

Quick answers to common problems

Mule ESB Cookbook

Over 40 recipes to effectively build your enterprise solutions from the ground up using Mule ESB



Dr. Zakir Laliwala Azaz Desai Abdul Samad Uchit Vyas



Mule ESB Cookbook

Over 40 recipes to effectively build your enterprise solutions from the ground up using Mule ESB

Dr. Zakir Laliwala Abdul Samad Azaz Desai Uchit Vyas



BIRMINGHAM - MUMBAI

Mule ESB Cookbook

Copyright © 2013 Packt Publishing

All rights reserved. No part of this book may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, without the prior written permission of the publisher, except in the case of brief quotations embedded in critical articles or reviews.

Every effort has been made in the preparation of this book to ensure the accuracy of the information presented. However, the information contained in this book is sold without warranty, either express or implied. Neither the authors, nor Packt Publishing, and its dealers and distributors will be held liable for any damages caused or alleged to be caused directly or indirectly by this book.

Packt Publishing has endeavored to provide trademark information about all of the companies and products mentioned in this book by the appropriate use of capitals. However, Packt Publishing cannot guarantee the accuracy of this information.

First published: August 2013

Production Reference: 1160813

Published by Packt Publishing Ltd. Livery Place 35 Livery Street Birmingham B3 2PB, UK.

ISBN 978-1-78216-440-1

www.packtpub.com

Cover Image by Rakesh Shejwal (shejwal.rakesh@gmail.com)

Credits

Authors

Dr. Zakir Laliwala Abdul Samad Azaz Desai Uchit Vyas

Reviewers

Lieven Heuninck Maurizio Turatti

Acquisition Editor Kartikey Pandey

Lead Technical Editor Dayan Hyames

Technical Editors Jalasha D'costa Amit Ramadas

Copy Editors

Gladson Monteiro Sayanee Mukherjee Aditya Nair Adithi Shetty Laxmi Subramanian

Project Coordinator Kranti Berde

Proofreader Linda Morris

Indexers Hemangini Bari Tejal Soni

Graphics Sheetal Aute

Production Coordinator Arvindkumar Gupta

Cover Work Arvindkumar Gupta

About the Authors

Dr. Zakir Laliwala is an entrepreneur, open source specialist, and hands-on CTO of Attune Infocom. Attune Infocom provides enterprise-level open source solutions, and services for SOA, BPM, ESB, Portal, Cloud computing, and ECM. At Attune Infocom, he is responsible for solutions and services delivery and product development. He is exploring new enterprise-level open source solutions, and defining architecture, roadmap, and best practices. He has provided consulting and training on various open source technologies to corporates around the world on Mule ESB, Activiti BPM, JBoss' jBPM and Drools, Liferay Portal, Alfresco ECM, Jboss SOA, and Cloud computing.

Dr. Zakir pursued Ph.D. in Information and Communication Technology from Dhirubhai Ambani Institute of Information and Communication Technology. He was the Adjunct Faculty at Dhirubhai Ambani Institute of Information and Communication Technology (DA-IICT) and CEPT University, where he now teaches students pursuing Master's.

He has published many research papers in IEEE and ACM international conferences on web services, SOA, Grid computing, and Semantic Web. He also serves as a reviewer at various international conferences and journals. He has contributed chapters on open source technologies and writes books on open source technologies.

Abdul Samad has more than seven years' hands-on experience in leading and implementing Java, J2EE, Portal, and ECM open source solutions. He has successfully migrated IBM WebSphere portal to Liferay Portal for a client based in the U.K. He has delivered successful training with his experience and expertise on Liferay Portal and jBPM to Sambaash, AT&T, Cognizant, Urja Technologies, and Protea Technologies. He was part of an enterprise-level, open source portal application implementation for media and publication houses, portal customization projects, and led a team of developers to achieve the client's requirements on time.

He has expertise in implementing J2EE technologies (JSP, Servlet, MVC Frameworks, BPM, ESB, and Portlet frameworks) to develop enterprise web applications. He has worked with various frameworks such as Mule ESB, jBPM, Liferay, Alfresco, and Oracle WebLogic portal on his journey.

Azaz Desai has more than three years' experience in Mule ESB, JBPM, and Liferay technologies. He is an Oracle Certified Java Programmer (OCJP). He is responsible for implementing, deploying, integrating, and optimizing services and business processes using ESB and BPM tools. He is a lead writer of *Mule ESB Cookbook, Packt Publishing*, as well playing a vital role of trainer on ESB to global clients at Attune Infocom.

He is very enthusiastic and active in understanding client-specific requirements on web service integration. He has done various integration of web services, such as Mule ESB with Liferay, Alfresco, jBPM, and Drools. He was a part of a key project on Mule ESB integration as a messaging system. He has worked on various web service standards and frameworks, such as CXF, AXIS, SOAP, and REST.

Uchit Vyas a B.Tech. graduate in Computer Science with a research interest in ESB and Cloud, is a certified Cloud Architect (AWS), Cisco (CCNA), VMware (VSP), and Red Hat Linux (RHCE) professional. He has an energetic strength to work on multiple platforms at a time and the ability to integrate open source technologies. He works as a Sr. Consultant and looks after AWS – Cloud, Mule ESB, Alfresco, Liferay and deploying Portal, and ECM system. He was previously working with TCS as an Assistant System Engineer.

With over three years' hands-on experience on open source technologies, he manages to guide the team and deliver projects and training sessions meeting client expectations. He has provided more than 13 training sessions on Cloud computing, Alfresco, and Liferay in a couple of months. In the last few years, he has moved over 80 percent of Attune Infocom business processes to the Cloud by implementing agile SDLC methodology on Amazon, Rackspace, and private Clouds such as Eucalyptus and OpenStack. His skills are not limited to designing and managing Cloud environment/infrastructure, server architecture. He is also active in Shell scripting, autodeployment, supporting hundreds of Linux and Windows physical and virtual servers hosting databases, and applications with continuous delivery using Jenkins/CruiseControl with Puppet/Chef scripting.

About the Reviewers

Lieven Heuninck holds an MSc. degree. He has an up-to-date knowledge on technology and is very capable in designing technical solutions using state of the art technological components. He knows how to combine his technical skills with a good practical knowledge of the various functional processes present in today's organizations. He is passionate about Enterprise Architecture, Service-Oriented Architecture, and all the tools that can bring these concepts into reality. He is so passionate about it that he co-started a company called Apogado (http://www.apogado.com) to provide dedicated services in his areas of expertise. Apogado currently conducts various missions for large organizations in private and public sectors. When Lieven is not working, he enjoys sailing.

Maurizio Turatti is a software integration architect with more than 15 years' professional experience in many SOA and enterprise integration projects in Europe and the Middle East. He is now working as an Open Source Team Leader. He has worked formerly at SeeBeyond, Sun Microsystems, and Alfresco. He loves to research and experiment with leading edge open source technologies. You can contact him through his CamelCase blog: http://blog.maurizioturatti.com/.

He is also the author of the book Instant Apache Maven Starter, Packt Publishing.

www.PacktPub.com

Support files, eBooks, discount offers and more

You might want to visit www.PacktPub.com for support files and downloads related to your book.

Did you know that Packt offers eBook versions of every book published, with PDF and ePub files available? You can upgrade to the eBook version at www.PacktPub.com and as a print book customer, you are entitled to a discount on the eBook copy. Get in touch with us at service@packtpub.com for more details.

At www.PacktPub.com, you can also read a collection of free technical articles, sign up for a range of free newsletters and receive exclusive discounts and offers on Packt books and eBooks.



http://PacktLib.PacktPub.com

Do you need instant solutions to your IT questions? PacktLib is Packt's online digital book library. Here, you can access, read and search across Packt's entire library of books.

Why Subscribe?

- ► Fully searchable across every book published by Packt
- Copy and paste, print and bookmark content
- On demand and accessible via web browser

Free Access for Packt account holders

If you have an account with Packt at www.PacktPub.com, you can use this to access PacktLib today and view nine entirely free books. Simply use your login credentials for immediate access.

Table of Contents

Preface	1
Chapter 1: Getting Started with Mule ESB	7
Introduction	7
Understanding Mule concepts and terminologies	8
Setting up the Mule IDE	13
Installing Mule Studio	20
Configuring Mule components	23
Deploying your first Hello World application on the Mule server	31
Chapter 2: Working with Components and Patterns	43
Introduction	43
Configuring the component	44
Using the Echo component to display the message payload	48
Using a Flow Reference component to synchronously execute another flow	57
Publishing a RESTful web service using the REST component	72
Publishing a SOAP-based web service using the SOAP component	84
Chapter 3: Using Message Property, Processors, and Sources	99
Introduction	99
Understanding components	100
Understanding message sources	112
Using message processors to control the message flow	114
Understanding message property scopes	122
Chapter 4: Endpoints	<u>133</u>
Introduction	133
Configuring the Generic Endpoint	133
Configuring the HTTP Endpoint	135
Configuring the IMAP Endpoint to retrieve e-mails	145
Using the JDBC Endpoint to connect to the database	147

Table of Contents	
Implementing the File Transport channel using the File Endpoint Sending messages asynchronously using the AJAX Endpoint Using the Servlet Endpoint to listen to events or messages from servlet requests	164 181 197
Chanter 5: Transformers	201
Introduction	201
Configuring the JSON-to-Object transformer	202
Configuring the Object-to-XML transformer	214
Configuring the Message and Variable transformers	223
Creating the custom transformer	226
Understanding the DataMapper transformer	235
Chapter 6: Configuring Filters	249
Introduction	249
Configuring the Logic filters – And/Or/Not	249
Performing filtering according to the exception type	258
Filtering messages by evaluating expressions	260
Handling incoming events or messages using the Message filter	261
Configuring the Wildcard filter	264
Creating a Custom filter	273
Chapter 7: Handling Exceptions and Testing	281
Introduction	281
Understanding Messaging Exception strategies	282
Configuring the Choice Exception Strategy	284
Configuring the Reference Exception Strategy	286
Configuring the Rollback Exception Strategy	288
Testing with JUnit in Mule ESB	289
Chapter 8: Introducing Web Services	311
Introduction	311
Proxying web services	312
Creating JAX-WS services	313
Creating web services using the REST component	322
Calling external web services using the SOAP component	329
Chapter 9: Understanding Flows, Routers, and Services	339
Introduction	339
Configuring the All Router/Flow Control	339
Configuring the Choice Router/Flow Control	350
Configuring the Splitter Flow Control	361

—**ii**—

	———— Table of Contents
Chapter 10: Configuring Cloud Connectors	371
Introduction	371
Configuring the Twitter Cloud Connector	371
Configuring the DropBoxIntegration folder	384
Index	405



Preface

Mule ESB is a lightweight Java-based enterprise service bus (ESB) and integration platform that allows developers to connect applications together quickly and easily, enabling them to efficiently exchange data. You can therefore use Mule ESB to allow different applications to communicate with each other via a transit system to carry data between applications within your enterprise or across the Internet. It is also useful if you use more than one type of communication protocol while integrating three or more applications/services.

Mule ESB Cookbook takes readers through the practical approach of using Mule ESB 3.3. This book solves numerous issues faced by developers working on Mule ESB in real time and provides use cases on how to integrate Mule with other technologies. It also focuses on development and delivery using Mule ESB through integrating, migrating, and upgrading advanced technological tools.

This book gives the reader a strong overview of the Mule framework using practical and easy-to-follow examples. It has three sections: problems, approaches, and solutions. The key aim of this book is to show you how to allow different applications to communicate with each other by creating a transit system to carry data between applications within your enterprise or across the Internet. Mule ESB enables easy integration of existing systems, regardless of the different technologies that the applications use, including JMS, web services, JDBC, HTTP, and more.

Mule ESB Cookbook will teach you everything to communicate between applications that are built on different platforms, as well as how to migrate them into your application across multiple platforms or on the Cloud.

Preface -

What this book covers

This book contains recipes related to deployment, scripting, and the API discussing core concepts of standard components, performance tuning, and Cloud integration through practical task-oriented recipes. This book will provide you practical knowledge of the Mule ESB architecture and its configuration. Core concepts and components required to understand how Mule ESB works are also explained.

Chapter 1, *Getting Started with Mule ESB*, discusses Mule core concepts and terminology. It also provides an environment setup for Mule ESB and Mule Studio. By the end of this chapter, you will be familiar with Mule IDE integration with Eclipse, and how to create a Hello World project and flow in Mule Studio. At the end of this chapter, you will learn how to configure Mule elements and deploy applications on the Mule server.

Chapter 2, Working with Components and Patterns, describes what a component is and its types, such as Echo, Logger, REST, SOAP, HTTP, and Java. You will also know how to configure a component, how to use it in a workflow, and what patterns are in Mule ESB.

Chapter 3, Using Message Property, Processors, and Sources, helps you understand what message sources, processors, and properties are. By the end of this chapter, you will be able to use processors in a workflow, and use message processors to control the message flow, and message property scopes.

Chapter 4, Endpoints, explains what an Endpoint is. Endpoints send and receive data and are responsible for connecting to external resources and delivering messages. The two types of Endpoints available in Mule Studio are: Inbound Endpoint and Outbound Endpoint. Inbound Endpoint is used for receive messages and Outbound Endpoint is used for sending messages.

Chapter 5, Transformers, explains what a transformer is. By the end of this chapter, you will be able to configure the JSON-to-Object and Object-to-XML transformers and DataMapper.

Chapter 6, *Configuring Filters*, explains what a filter is and how to configure the Logic filter. By the end of this chapter, you will be able to create a custom filter and configure the Message filter.

Chapter 7, Handling Exceptions and Testing, explains what an exception is. By the end of this chapter, you will be able to configure the Catch Exception Strategies, Rollback Exception Strategies, and JUnit testing.

Chapter 8, Introducing Web Services, explains what a web service is. By the end of this chapter, you will be able to create a JAX-WS web service and integrate external web services.

Chapter 9, Understanding Flows, Routers, and Services, explains what a Router is, and how to configure the Router and the Splitter Flow Control.

Chapter 10, Configuring Cloud Connectors, explains what a cloud connector is and how to integrate Twitter and Dropbox connectors.



What you need for this book

You will need the following software to be installed before running the code examples:

- Mule ESB requires JDK 6 or a later version. JDK 6 can be downloaded from the following site: http://www.oracle.com/technetwork/java/javase/ downloads/jdk6downloads-1902814.html.
- Mule Studio is a powerful, user-friendly Eclipse-based tool. Mule Studio is an Eclipse-based tool that has three main components: a package tree, a palette, and a canvas. Mule ESB easily creates flows and edits and tests them in a few minutes. It is based on drag-and-drop elements and supports two-way editing. Mule Studio can be downloaded from the following site: http://www.mulesoft.org/all-mulestudio-downloads.
- You will also require PostgreSQL 9.2. PostgreSQL is a web hosting database that is used to store website information such as user information. PostgreSQL is a powerful, open source object-relational database system. It runs on all major operating systems, including Linux, Unix, and Windows. It is fully ACID compliant, and has full support for foreign keys, joins, views, triggers, and stored procedures. PostgreSQL can be downloaded from the following site: http://www.piostgresql.org/download.
- You will need Selenium IDE. It is an integrated development environment for Selenium scripts. It is implemented as a Firefox extension and allows you to record, edit, and debug tests. Selenium IDE includes the entire Selenium Core, allowing you to easily and quickly record and play back tests in the actual environment that they will run on. Selenium IDE is not only a recording tool, but it is also a complete IDE. You can choose to use its recording capability, or you may edit your scripts by hand. Selenium IDE can be downloaded from the following site: http://docs. seleniumhq.org/download/.

Who this book is for

This book provides solutions for developers who are working on Mule ESB and for integrators or migrators who are integrating and migrating Mule with other technologies. It focuses on development and delivery using Mule ESB through integrating, migrating, and upgrading advanced technological tools.

Conventions

In this book, you will find a number of styles of text that distinguish between different kinds of information. Here are some examples of these styles, and an explanation of their meaning.

Code words in text are shown as follows: "Enter the project name, Logic Filter."



Preface -

A block of code is set as follows:

```
package com.org;
public class User {
  private String name;
  private String lname;
 public String getName() {
    return name;
  }
  public void setName(String name) {
    this.name = name;
  }
  public String getLname() {
    return lname;
  public void setLname(String lname) {
    this.lname = lname;
  }
}
```

Any command-line input or output is written as follows:

```
java -jar selenium-server-standalone-2.31.0.jar
```

New terms and important words are shown in bold. Words that you see on the screen, in menus or dialog boxes for example, appear in the text like this: "Click on Next and then on Finish."





Reader feedback

Feedback from our readers is always welcome. Let us know what you think about this book—what you liked or may have disliked. Reader feedback is important for us to develop titles that you really get the most out of.

To send us general feedback, simply send an e-mail to feedback@packtpub.com, and mention the book title via the subject of your message.

If there is a topic that you have expertise in and you are interested in either writing or contributing to a book, see our author guide on www.packtpub.com/authors.

Customer support

Now that you are the proud owner of a Packt book, we have a number of things to help you to get the most from your purchase.

Downloading the example code

You can download the example code files for all Packt books you have purchased from your account at http://www.packtpub.com. If you purchased this book elsewhere, you can visit http://www.packtpub.com. If you purchased this book elsewhere, you can visit http://www.packtpub.com. If you purchased this book elsewhere, you can visit http://www.packtpub.com. If you purchased this book elsewhere, you can visit http://www.packtpub.com/support and register to have the files e-mailed directly to you.

Errata

Although we have taken every care to ensure the accuracy of our content, mistakes do happen. If you find a mistake in one of our books—maybe a mistake in the text or the code—we would be grateful if you would report this to us. By doing so, you can save other readers from frustration and help us improve subsequent versions of this book. If you find any errata, please report them by visiting http://www.packtpub.com/submit-errata, selecting your book, clicking on the **errata submission form** link, and entering the details of your errata. Once your errata are verified, your submission will be accepted and the errata will be uploaded on our website, or added to any list of existing errata, under the Errata section of that title. Any existing errata can be viewed by selecting your title from http://www.packtpub.com/support.

5

Preface -

Piracy

Piracy of copyright material on the Internet is an ongoing problem across all media. At Packt, we take the protection of our copyright and licenses very seriously. If you come across any illegal copies of our works, in any form, on the Internet, please provide us with the location address or website name immediately so that we can pursue a remedy.

Please contact us at copyright@packtpub.com with a link to the suspected pirated material.

We appreciate your help in protecting our authors, and our ability to bring you valuable content.

Questions

You can contact us at questions@packtpub.com if you are having a problem with any aspect of the book, and we will do our best to address it.

- 6 -

1 Getting Started with Mule ESB

In this chapter, we will cover the following topics:

- Understanding Mule concepts and terminologies
- Setting up the Mule IDE
- Installing Mule Studio
- ► Configuring Mule components
- Deploying your first Hello World application on the Mule server

Introduction

Mule ESB is a lightweight Java programming language. Through ESB, you can integrate or communicate with multiple applications. Mule ESB enables easy integration of existing systems, regardless of the different technologies that the applications use, including JMS, web services, JDBC, and HTTP.

Getting Started with Mule ESB -

Understanding Mule concepts and terminologies

Enterprise Service Bus (**ESB**) is an application that gives access to other applications and services. Its main task is to be the messaging and integration backbone of an enterprise.

An ESB is a distributed middleware system to integrate different applications. All these applications communicate through the ESB. It consists of a set of service containers that integrate various types of applications. The containers are interconnected with a reliable messaging bus.

Getting ready

An ESB is used for integration using a service-oriented approach. Its main features are as follows:

- Polling JMS
- Message transformation and routing services
- Tomcat hot deployment
- ► Web service security

We often use the abbreviation, **VETRO**, to summarize the ESB functionality:

- ▶ V validate the schema validation
- ► E enrich
- ► **T** transform
- R route (either itinerary or content based)
- **0** operate (perform operations; they run at the backend)

8



Before introducing any ESB, developers and integrators must connect different applications in a point-to-point fashion.

How to do it...

After the introduction of an ESB, you just need to connect each application to the ESB so that every application can communicate with each other through the ESB. You can easily connect multiple applications through the ESB, as shown in the following diagram:





Getting Started with Mule ESB -

Need for the ESB

You can integrate different applications using ESB. Each application can communicate through ESB:

- To integrate more than two or three services and/or applications
- To integrate more applications, services, or technologies in the future
- To use different communication protocols
- To publish services for composition and consumption
- For message transformation and routing

What is Mule ESB?

Mule ESB is a lightweight Java-based enterprise service bus and integration platform that allows developers and integrators to connect applications together quickly and easily, enabling them to exchange data. There are two editions of Mule ESB: Community and Enterprise. Mule ESB Enterprise is the enterprise-class version of Mule ESB, with additional features and capabilities that are ideal for clustering and performance tuning, DataMapper, and the SAP connector. Mule ESB Community and Enterprise editions are built on a common code base, so it is easy to upgrade from Mule ESB Community to Mule ESB Enterprise.

Mule ESB enables easy integration of existing systems, regardless of the different technologies that the applications use, including JMS, web services, JDBC, and HTTP. The key advantage of an ESB is that it allows different applications to communicate with each other by acting as a transit system for carrying data between applications within your enterprise or across the Internet. Mule ESB includes powerful capabilities that include the following:

- Service creation and hosting: It exposes and hosts reusable services using Mule ESB as a lightweight service container
- Service mediation: It shields services from message formats and protocols, separate business logic from messaging, and enables location-independent service calls
- Message routing: It routes, filters, aggregates, and re-sequences messages based on content and rules
- Data transformation: It exchanges data across varying formats and transport protocols



Mule ESB is lightweight but highly scalable, allowing you to start small and connect more applications over time. Mule provides a Java-based messaging framework. Mule manages all the interactions between applications and components transparently. Mule provides transformation, routing, filtering, Endpoint, and so on.

10

Chapter 1

How it works...

When you examine how a message flows through Mule ESB, you can see that there are three layers in the architecture, which are listed as follows:

- Application Layer
- Integration Layer
- Transport Layer

Likewise, there are three general types of tasks you can perform to configure and customize your Mule deployment. Refer to the following diagram:



The following list talks about Mule and its configuration:

- Service component development: This involves developing or re-using the existing POJOs, which is a class with attributes and it generates the get and set methods, Cloud connectors, or Spring Beans that contain the business logic and will consume, process, or enrich messages.
- Service orchestration: This involves configuring message processors, routers, transformers, and filters that provide the service mediation and orchestration capabilities required to allow composition of loosely coupled services using a Mule flow. New orchestration elements can be created also and dropped into your deployment.



Getting Started with Mule ESB -

 Integration: A key requirement of service mediation is decoupling services from the underlying protocols. Mule provides transport methods to allow dispatching and receiving messages on different protocol connectors. These connectors are configured in the Mule configuration file and can be referenced from the orchestration layer. Mule supports many existing transport methods and all the popular communication protocols, but you may also develop a custom transport method if you need to extend Mule to support a particular legacy or proprietary system.



- Spring beans: You can construct service components from Spring beans and define these Spring components through a configuration file. If you don't have this file, you will need to define it manually in the Mule configuration file.
- ► **Agents**: An agent is a service that is created in Mule Studio. When you start the server, an agent is created. When you stop the server, this agent will be destroyed.
- Connectors: The Connector is a software component.
- Global configuration: Global configuration is used to set the global properties and settings.
- Global Endpoints: Global Endpoints can be used in the Global Elements tab. We can
 use the global properties' element as many times in a flow as we want. For that, we
 must pass the global properties' reference name.
- Global message processor: A global message processor observes a message or modifies either a message or the message flow; examples include transformers and filters.



- **Transformers**: A transformer converts data from one format to another. You can define them globally and use them in multiple flows.
- Filters: Filters decide which Mule messages should be processed. Filters specify the conditions that must be met for a message to be routed to a service or continue progressing through a flow. There are several standard filters that come with Mule ESB, which you can use, or you can create your own filters.
- **Models**: It is a logical grouping of services, which are created in Mule Studio. You can start and stop all the services inside a particular model.
- Services: You can define one or more services that wrap your components (business logic) and configure Routers, Endpoints, transformers, and filters specifically for that service. Services are connected using Endpoints.
- Endpoints: Services are connected using Endpoints. It is an object on which the services will receive (inbound) and send (outbound) messages.
- Flow: Flow is used for a message processor to define a message flow between a source and a target.

Setting up the Mule IDE

The developers who were using Mule ESB over other technologies such as Liferay Portal, Alfresco ECM, or Activiti BPM can use Mule IDE in Eclipse without configuring the standalone Mule Studio in the existing environment. In recent times, MuleSoft (http://www.mulesoft. org/) only provides Mule Studio from Version 3.3 onwards, but not Mule IDE. If you are using the older version of Mule ESB, you can get Mule IDE separately from http://dist. muleforge.org/mule-ide/releases/.

Getting ready

To set Mule IDE, we need Java to be installed on the machine and its execution path should be set in an environment variable. We will now see how to set up Java on our machine.

- Firstly, download JDK 1.6 or a higher version from the following URL: http://www.oracle.com/technetwork/java/javase/downloads/ jdk6downloads-1902814.html.
- 2. In your Windows system, go to Start | Control Panel | System | Advanced.

13

Getting Started with Mule ESB _____

3. Click on Environment Variables under System Variables, find Path, and click on it.

System Proper	ties				? 🛛
System Re	store	Autor	natic U	pdates	Remote
General	Comput	ter Name		Hardware	Advanced
You must be la	ogged on as	an Administr	ator to	make most	of these changes.
Visual effects	, processor :	scheduling, i	memor	y usage, and	l virtual memory
					Settings
User Profiles					
Desktop sett	ngs related t	o your logor	1		
					Settings
Startup and F	ecovery				
System startu	p, system fai	ilure, and de	buggir	ng informatio	n
					Settings
	Env	vironment Va	ariables	En	ror Reporting
		0	К	Cano	el Apply



4. In the **Edit** window, modify the path by adding the location of the class to its value. If you do not have the item **Path**, you may select the option of adding a new variable and adding Path as the name and the location of the class as its value.

System Properties 🔹 💽 🕅	System Properties
System Restore Automatic Updates Remote	System Restore Automatic Updates Remote
Environment Variables	Environment Variables
Edit System Variable	Edit System Variable
Variable name: JAVA_HOME	Variable name: Path
Variable value: C:\Program Files\Java\jdk1.6.0_20	Variable value: DAVA_HOME%\bin;%MULE_HOME%\bin;
OK Cancel	OK Cancel
Variable Value	Variable Value
EICDDrive F:\	NUMBER_OF_P 2 OS Windows NT
JAVA_HOME C:\Program Files\Java\jdk1.6.0_20 MAVEN_HOME C:\apache-maven-3.0.4 Mule C:\Pub_inc(MuleStudio ♥	Path C:\WINDOWS\system32;C:\WINDOWS\ PATHEXT .COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS; PROCESSOR_A x86
New Edit Delete	New Edit Delete
OK Cancel	OK Cancel

5. Close the window, reopen the command prompt window, and run your Java code.

How to do it...

If you go with Eclipse, you have to download Mule IDE Standalone 3.3.

 Download Mule ESB 3.3 Community edition from the following URL: http://www.mulesoft.org/extensions/mule-ide. Unzip the downloaded file and set MULE HOME as the environment variable.



 Download the latest version of Eclipse from http://www.eclipse.org/ downloads/.

S	system Properties 2	<
	System Restore Automatic Updates Remote	
ſ	Environment Variables	
	Edit System Variable	
	Variable name: MULE_HOME Variable value: e-standalone-3.2.0\mule-standalone-3.2.0	
	OK Cancel	
	System variables	
	variable value FP_NO_HOST_C NO JAVA_HOME C:\Program Files\Java\jdk1.6.0 MULE_HOME C:\mule-standalone-3.2.0\mule-standal MuleStudio C:\MuleStudio NUMBER_OF_P 2	
	New Edit Delete	
	OK Cancel]

After installing Eclipse, you now have to integrate Mule IDE in the Eclipse. If you are using Eclipse Version 3.4 (Galileo), perform the following steps to install Mule IDE. If you are not using Version 3.4 (Galileo), the URL for downloading will be different.

1. Open Eclipse IDE.

– 16

2. Go to Help | Install New Software....



- 3. Write the URL in the **Work with:** textbox: http://dist.muleforge.org/muleide/updates/3.4/ and press *Enter*.
- 4. Select the **Mule IDE** checkbox.
- 5. Click on the **Next** button.
- 6. Read and accept the license agreement terms.

17—

Getting Started with Mule ESB _____

7. Click on the **Finish** button.

This will take some time. When it prompts for a restart, shut it down and restart Eclipse.

🔘 Java EE - Eclipse 🛛 🔘 Install			🖬 🛃
File Edit Navigate Search	ware		Java EE
	nac you wish to instail.		Task List 🗖 🗖
Work with: http://www.international.com	://dist.muleforge.org/mule-ide/updates/3.4/	Add	t available.
	Find more software by wo	rking with the ' <u>Available Software Sites'</u> preferences.	
type filter text			
Name	Version		
	DE		
Details			
		=	
Show only the	atest versions of available software Hide items that	are already installed	
Group items by	category What is <u>already i</u>	installed?	
Contact an upu	ite sites during instan to ninu required sortware		
?	< Back	Next > Finish Cancel	\$° ▽ □ □
	Description A Resource f	Path Locat Type	
i ∏° 0 items selected			



Mule configuration

After installing Mule IDE, you will now have to configure Mule in Eclipse. Perform the following steps:

- 1. Open Eclipse IDE.
- 2. Go to Window | Preferences.



19 –

Getting Started with Mule ESB -

Image: Construction of the initial of the initialy	🖉 Java EE - Eclipse 🧖 Proformer		
Path: C:(mule-standalone-3,3.0/mule-standalone-3,3.0 Version: 3.3.0 Restore Defaults Apply OK Cancel	Java EE - Eclipse Project Evolver 2 Project Evolver 2	Mule Image: Comparison of the second select the default distribution to use with Mule projects. Add Mule installation directories and select the default distribution to use with Mule projects. Mule Distributions: Image: Comparison of the second select the default distribution to use with Mule standalone-3.3.0 Add Image: Comparison of the second select the default distribution to use with Mule standalone-3.3.0 Image: Comparison of the second select the default distribution to use with Mule standalone-3.3.0 Image: Comparison of the second select the default distribution to use with Mule standalone-3.3.0 Image: Comparison of the second select the default distribution to use with Mule standalone-3.3.0 Image: Comparison of the second select the default distribution to use with Mule standalone-3.3.0 Image: Comparison of the second select the default distribution to use with the second select the second select the default distribution to use with Mule select the second select the select the second select the second select the second sel	t available.
	U Rens Description	Peth: C:(mule-standalone-3.3.0)(mule-standalone-3.3.0) Version: 3.3.0 Restore Defaults Apply OK Cancel	<u>وہ</u> م

3. Select **Mule**, add the distribution folder mule as standalone 3.3; click on the **Apply** button and then on the **OK** button. This way you can configure Mule with Eclipse.

Installing Mule Studio

Mule Studio is a powerful, user-friendly Eclipse-based tool. Mule Studio has three main components: a package tree, a palette, and a canvas. Mule ESB easily creates flows as well as edits and tests them in a few minutes. Mule Studio is currently in public beta. It is based on drag-and-drop elements and supports two-way editing.

Getting ready

To install Mule Studio, download Mule Studio from http://www.mulesoft.org/
download-mule-esb-community-edition.



Chapter 1

How to do it...

Unzip the Mule Studio folder. Set the environment variable for Mule Studio. While starting with Mule Studio, the config.xml file will be created automatically by Mule Studio.

The three main components of Mule Studio are as follows:

- A package tree
- A palette
- A canvas

A package tree

A **package tree** contains the entire structure of your project. In the following screenshot, you can see the package explorer tree. In this package explorer tree, under src/main/java, you can store the custom Java class. You can create a graphical flow from src/main/resources.



In the app folder you can store the mule-deploy.properties file. The folders src, main, and app contain the flow of XML files. The folders src, main, and test contain flow-related test files. The Mule-project.xml file contains the project's metadata. You can edit the name, description, and server runtime version used for a specific project. JRE System Library contains the Java runtime libraries. Mule Runtime contains the Mule runtime libraries.

21

Getting Started with Mule ESB -

A palette

The second component is **palette**. The palette is the source for accessing Endpoints, components, transformers, and Cloud connectors. You can drag them from the palette and drop them onto the canvas in order to create flows. The palette typically displays buttons indicating the different types of Mule elements. You can view the content of each button by clicking on them. If you do not want to expand elements, click on the button again to hide the content.



- 22

A canvas

The third component is canvas; canvas is a graphical editor. In canvas you can create flows. The canvas provides a space that facilitates the arrangement of Studio components into Mule flows. In the canvas area you can configure each and every component, and you can add or remove components on the canvas.



Configuring Mule components

A simple POJO component will be invoked by Mule when a message is received. You can create your own custom component.

Getting ready

There are three types of components in Mule Studio:

- Simple component
- Java component
- Other components


How to do it...

Service components contain the business logic. Drag-and-drop a **Component** from the palette onto the canvas and configure the component. Double-click on the component.



How it works...

The following are the palette components present in Mule.

Palette components

There are different palette components available in Mule Studio, where each palette component has different uses. We will see that in detail here:

- Endpoints
- ► Components
- ► Transformers
- ▶ Filters
- Flow control



Chapter 1

- Routers
- Scopes
- Cloud connectors

Endpoints

Generally, Endpoints send and receive data, and are responsible for connecting to external resources and delivering messages. Endpoints can be Inbound or Outbound. An Inbound Endpoint receives messages via its associated transport. Each transport implements its own Inbound Endpoint element. An Outbound Endpoint sends messages via its associated transport. Each transport is associated transport. Each transport implements its own Outbound Endpoint element.



- FTP Endpoint: This Endpoint reads files from the FTP server. This Endpoint carries all the information for an FTP connection. The host and port values are required. The FTP Endpoint implements a file transport channel so that your Mule application can exchange files with an external FTP server. You can configure FTP as an Inbound Endpoint (which receives files) or Outbound Endpoint (which writes files to the FTP server). This is only used in the Enterprise edition.
- File Endpoint: This Endpoint reads a file from the filesystem. The File Endpoint implements a transport channel so that your Mule application can exchange files with a filesystem. You can implement the File Endpoint as an Inbound Endpoint (a message source), or as an Outbound Endpoint. This Endpoint implements only a one-way exchange pattern. The File Endpoint is used for transferring the file from one directory to another.



- Generic Endpoint: This Endpoint is used as a dynamic way to configure an Endpoint using Mule expressions and specifying paths. The Generic Endpoint allows for a wide array of configuration options by defining a particular transport to be used as the Endpoint.
- HTTP Endpoint: This Endpoint is used to process HTTP requests or responses. Mule uses HTTP Endpoints to send and receive requests over the HTTP transport protocol, or HTTPS over the SSL protocol. Configured as either Inbound (also known as message sources) or Outbound, HTTP Endpoints use one of the two patterns: request-response and/or one-way.
- ► **JMS Endpoint**: This Endpoint is used to send or receive messages from a JMS queue. The JMS Endpoint's two-way exchange patterns use: request-response and/or one-way.
- VM Endpoint: This Endpoint is used for an in-memory queue that allows you to integrate different flows or services in the same Mule configuration. The VM Endpoint's two-way exchange patterns use request-response and/or one-way.

Components

The Studio building blocks are known as components and fall into three categories: general, script, and web service.

General components execute whenever a message is received. The logic embedded into general components cannot be modified. Components such as Logger, Flow Reference, and Echo fall into this category.

Script components do not contain prepackaged logic; instead they allow the developer to specify the logic (in the form of a custom script or a Java class) to add into the component. Script components also allow you to:

- Configure interceptors
- Add Spring beans
- Change the value or reference of a specific property within the associated class

The Java component allows you to reference a Java class. The other script components support the Groovy, JavaScript, Python, and Ruby scripting engines.

Web service components, as the name implies, enable Mule to use SOAP and RESTful protocols to communicate with external web services. The SOAP and RESTful components use CXF and Jersey services to convert messages from Java to XML. Web service components also allow the developer to select or define the logic to be invoked by the component. If using the RESTful component, you only need to select a Java class and add a script to the component. On the other hand, SOAP configuration requires you to define attributes and select the operation method used to publish a SOAP web service.



As an example of how the SOAP component can be used, a SOAP message could be sent to a web-service-enabled website, such as a used car price database, with the parameters needed for a search. The site would then return an XML-formatted document with the resulting data; for example, prices, models, and features. The data returned is then integrated directly into a third-party website or application.



The following components are present in Mule:

- Echo: This component is used to echo a message payload to the console.
- Logger: This component is used to perform logging using an expression that determines what should be logged depending on the logging level. Use Logger to log messages, such as error messages or exceptions.
- REST: This component is used to make a REST service available via Jersey. REST is the formalized architecture of HTTP based on the concepts of resources, links, and a uniform interface. It uses the HTTP protocol. We can create a web service using the REST component.
- SOAP: This component is used to make a web service available via CXF. You can create a CXF web service in Studio by configuring a SOAP component in your Mule flow to perform any of the following CXF web service operations:
 - Publish a simple service
 - Publish a JAX-WS service
 - Proxy a published service
 - Consume a service using a simple client
 - Consume a service using JAX-WS client
 - Proxy to a service

Using Mule's SOAP component, you can also enable WS-security, specify data bindings, and add interceptors to your CXF web service.



Transformers

Transformers convert message payloads to formats expected by their destinations. Mule ESB provides many standard transformers, which you configure using predefined elements and attributes in your Mule XML configuration file. You can also configure custom transformers using the <custom-transformer> element. You can configure a transformer locally or globally.



The following transformers are present in Mule:

- Custom Transformer: Transformers in Mule are used to convert messages from one format to another or to manipulate the message information such as headers and attachments. Mule ESB also provides several standard transformers, including XML transformers. You can create your own custom Java class using the extended AbstractTransformer interface. Two ways to create a Custom Transformer are:
 - Use a transformer annotation on a method
 - Create a custom Java class
- Object-to-Xml transformer: This transformer is used to convert a Java object to an XML representation using XStream. You configure this transformer using the <object-to-xml-transformer> element.
- Script Transformer: This transformer is used to transform the payload using a script. This defines script components to be used as transformers. The Script transformer lets you select the particular scripting engine from a pull-down list. The predefined script transformers, namely, Groovy, JavaScript, Python, and Ruby, have the scripting engines already set.
- Transformer Ref: This transformer is used to reference a global transformer.



- > XSLT Transformer: This transformer is used to transform XML using XSLT.
- Xml-to-Object Transformer: This transformer is used to convert XML to Java bean graphs using XStream. You configure this transformer using the <xml-to-objecttransformer> element.

Filters

Filters specify conditions that must be met for a message to be routed to a service or continued progressing through a flow. There are several standard filters that come with Mule ESB, which you can use, or you can create your own filters. You can create a global filter and then reference it from your services and flows. You can define a filter locally or globally.



The following filters are present in Mule:

- Custom Filter: This filter is used as a user-implemented filter. The standard filters handle most filtering requirements, but you can create your own custom filter. To create a custom filter, you have to implement the Filter interface.
- **Exception Filter**: This is a filter that matches an exception type.
- **Expression Filter**: This filter evaluates a range of expressions providing different types of evaluators such as XPath, JXPath, and OGNL and also a custom evaluator.
- Message Property Filter: This filter allows you to add logic to your routers based on the value of one or more properties of a message. This filter can be very powerful because the message properties are exposed, allowing you to reference any transport-specific or user-defined properties.



- Filter Reference: This filter is used to reference to a globally defined filter.
- **Regular Expression Filter**: This filter is used on a filter that applies a regular expression pattern to the message payload.
- Wildcard Filter: This is a filter that matches string messages against wildcards.
- Payload Filter: This is a filter that checks the class type of the payload object inside a message.

Routers

Flow Controls/Routers route messages to various destinations in a Mule flow. Some Flow Controls incorporate logic to analyze and possibly transform messages before routing takes place.

Icon	Name	Description
-	All	Router that sends message to all routes.
~	Choice	Router that routes messages based on expressions.

The following Routers are present in Mule:

- All: This Router can be used to send the same message to multiple targets. It sends messages to all routes.
- **Choice**: The Router sends a message to the first message processor that matches. It routes messages based on expressions.

Cloud Connectors

A Cloud Connector easily integrates your Mule application with third-party web APIs. A Cloud Connector is built using the Cloud connect toolset, which resides in Mule Studio by default.





The following Cloud Connectors are present in Mule:

- Salesforce: This connector provides an easy way to integrate with the Salesforce API. This allows users to create flows which can query, create, and update information in Salesforce.
- **Twitter**: This connector provides an easy way to integrate with the Twitter API using Mule flows.

Deploying your first Hello World application on the Mule server

By creating a simple Hello World application, you will know how to create a flow and deploy the flow using the Mule server.

Getting ready

Using the steps for application deployment, given in this recipe, you will learn the execution of the flow, how that flow execution will occur, and what will be the output of the application code.

How to do it...

In this example you will see how to create and deploy the first "hello world" using Mule Studio.

1. Open Mule Studio and enter the name for the workspace name, as shown in the following screenshot:

¥ Workspace Launcher	
Select a workspace	
Mule Studio stores your projects in a folder called a workspace. Choose a workspace folder to use for this session.	
Workspace: E:\MuleStudio	Browse
Use this as the default and do not ask again	
	OK Cancel

31

 Create a project in Mule Studio. You will see the Mule Studio Welcome Page window. If you click on New Mule Project Based on a template, you will see an existing example. By clicking on Go to Mule Studio, it will start redirecting away from the Mule Studio Welcome Page window. You can even create a project from a menu bar by going to File | New | Mule Project.



🔀 Mule - Mule Studio	🎽 New Mule Projec	st 📃 🗖 🔀	
File Edit Source Refactor Navigate Search F : C ▼ , R R	Mule Project	ject.	
Type Hiera	Project Documentatio Name: Description:	n: Helloworld	
	Server Settings: Server Runtime: Templates: Create project ba	Mule Server 3.3.0 CE	
- -	Criefo Crder Discounter Order Fulfillment Legacy Moderniz Foreach Example Scripting Example Stock Quotes Flight Reservation	Esche example included in Mule distribution. Processes XML orders, applies tiered discounts, and adds Tw Process XML orders, spills into items and route for fufilment. Demo Mobile App This is a simple Mule Studio project that illustrates a Legacy An example showing the use of the Foreach element Helio Example Stock Quotes Example Stock Quotes Example Flight Reservation Example	Туре
E Outline 12 P	0	< Back Next > Finish Cancel	

3. Enter the project name called HelloWorld, click on **Next**, and enter the .mflow name as the filename. Then, click on the **Finish** button.

33

4. Go to src/main/java, right-click on it, and go to New | Class. Create a class called Greeting under the package com.org; here we have created the sayHi method and its return type is set to String.

```
public String sayHi(String str)
{
    return "Hello "+str;
}
```



Downloading the example code

You can download the example code files for all Packt books you have purchased from your account at http://www.packtpub.com. If you purchased this book elsewhere, you can visit http://www.packtpub. com/support and register to have the files e-mailed directly to you.

5. Go to the Greeting.Mflow file. In the following screenshot, the central part is called the canvas where we can put graphical elements and on the right-hand side you can see the group of elements; this area is called a palette. You have to drag-and-drop the **HTTP** Endpoint from the palette onto the canvas area.



35 —

6. Double-click on the HTTP Endpoint to configure it. You will see the hostname. If you want to change the hostname, you can change it. In this example, we use localhost. If you want to change the port number, you can change that as well. By default, port number 8081 will be taken by the Mule server.

Mule - helloworld/flows/Hell	Y Endpoint Properties		
File Edit Navigate Search Project	HTTP (Inbound Endpoint) The HTTP Endpoint allows Mule application to connect to web resources through the HTTP transport protocol. This endpoint can also implement security through HTTP5.		
Package E ≥ Type Hiera Package E ≥ Type Hiera Package E ≥ Pack	General Advanced References HTTP Settings Documentation Display Display Name: HTTP Exchange Patterns One-way @ request-response Basic Settings Enabling HTTPS will require configuring a HTTPS Connector Host: localhost Port: 8081 Path:		Fiter: Select Endpoints Fite Database FIP Fite Generic HITP MAP MS Jatty PO3
E outine 23	OK OK Message Flow Gobal Elements	Cancel	Quartz All Quartz Components Component
i = •			i e 🔝 @ 🕒 M

- 36 -

7. Drag-and-drop the **Java** component from the palette onto the canvas area. In the **Java** component, we will store the custom Java class. Configure the **Java** component. In this example, we configure the Greeting class that we created before.





8. Double-click on the **Java** component and configure it. Click on the Java icon and configure the Greeting class.



How it works...

In this section, you will see how to deploy the application and how it will run on the browser.

 If you haven't saved your application code, do save it. To deploy the application code in the Mule server, go to **Run As** | **Mule Application**; the Mule server will deploy your application.





2. If your application code is successfully deployed, you will see the following output screen:

🗑 Mule - helloworld/flows/HelloWorld.mflow - Mule Studio
File Edit Navigate Search Project Run Window Help
🖺 🕎 Mule
🖻 Console 🕴 📕 📓 🕞 🖉 🛃 🖆 🖓 👘 🖉
HelloWorld [Mule Application] C:/Program Files]ava/pre6/bin/javaw.exe (Nov 20, 2012 3:22:30 PM)
connected=true
supportedProtocols=[http]
serviceOverrides= <none></none>
INFO 2012-11-20 15:22:34,375 [main] org.mule.lifecycle.AbstractLifecycleManager: Starting connector: connector.http.mule.de
INFO 2012-11-20 15:22:34,375 [main] org.mule.lifecycle.AbstractLifecycleManager: Starting model: _muleSystemModel
INFO 2012-11-20 15:22:34,390 [main] org.mule.construct.FlowConstructLifecycleManager: Starting flow: HelloWorldFlow1
INFO 2012-11-20 15:22:34,390 [main] org.mule.processor.SedaStageLifecycleManager: Starting service: HelloWorldFlow1.stage1
INFO 2012-11-20 15:22:34,390 [main] org.mule.component.ComponentLifecycleManager: Starting component: commponent.27970580
INFO 2012-11-20 15:22:34,390 [main] org.mule.transport.http.HttpConnector: Registering listener: HelloWorldFlow1 on endpoin
INFO 2012-11-20 15:22:34,406 [main] org.mule.transport.service.DefaultTransportServiceDescriptor: Loading default response
INFO 2012-11-20 15:22:34,406 [main] org.mule.lifecycle.AbstractLifecycleHanager: Initialising: 'null'. Object is: HttpMessa
INFO 2012-11-20 15:22:34,421 [main] org.mule.lifecycle.AbstractLifecycleManager: Starting: 'null'. Object is: HttpMessageRe
INFO 2012-11-20 15:22:34,421 [main] org.mule.module.management.agent.WrapperManagerAgent: This JVM hasn't been launched by
INFO 2012-11-20 15:22:34,453 [main] org.mule.module.management.agent.JmxAgent: Attempting to register service with name: Mu
INFO 2012-11-20 15:22:34,453 [main] org.mule.module.management.agent.Jmxxgent: kegistered indpoint Service with name: Rule.
INFO 2012-11-20 15:22:34,453 [main] org.mule.module.management.agent.JmxAgent: kegistered connector service with name mule.
INFO 2012-11-20 15:22:34,466 [main] org.mule.module.laucher.application.beraultHuleApplication: Reload Interval: 3000
INFO 2012-11-20 15:22:34,466 [main] org.mule.beraultenuecontext:
t Ampliantion, bilanald t
· Application: nellowella ·
* *
t Agente Dunning:
A JBV from t
INFO 2012-11-20 15:22:34,468 [main] org.mule.module.launcher.DeploymentService:
+ Started app 'helloworld' +
+++++++++++ ⁺⁺ +++++++++++++++++++++++
s s

-40-

3. Copy the URL http://localhost:8081 and paste it on your browser.



4. Paste the URL on your browser, and you will see the following output:



41

2 Working with Components and Patterns

In this chapter, we will cover:

- Configuring the component
- ▶ Using the Echo component to display the message payload
- ▶ Using a Flow Reference component to synchronously execute another flow
- Publishing a RESTful web service using the REST component
- Publishing a SOAP-based web service using the SOAP component

Introduction

Mule has the ability of routing, filtering, transforming, and processing with components. Each of those abilities are assigned a good number of fine-grained processors. The configuration file of a Mule application that combines those elements can end up being large. The different types of configuration patterns provided by Mule are simple service pattern, bridge, validator, HTTP proxy, and WS proxy.

Working with Components and Patterns -

Configuring the component

In this recipe, you will see how to configure a component in Mule Studio. We will have a look at how to use different components throughout this chapter.

Getting ready

Mule uses HTTP Endpoints to send and receive requests over the HTTP protocol. Configured as either **Inbound** (also known as **message sources**) or **Outbound**, HTTP Endpoints use one of these two message exchange patterns: **request-response** or **one-way**.



The arrows in the preceding screenshot indicate the request-response type of message exchange.



The arrow in this screenshot indicates the one-way type of message exchange.



How to do it...

Double-click on the **HTTP** Endpoint to configure it. You will see a screen similar to the following screenshot on your window. You have to enter the **Host** and **Port** values. By default, the port number is **8081**; you can change the values of the **Host** and **Port** fields. However, these two fields are mandatory.

Mule - flowref/flows/FlowRef	¥ Endpoint Properties	🖬 🖬 📉
Mule - flowref/flows/FlowRef File Edt Navigate Search Project Pr		Filter: Select Select Filter: Database FIP File
Image: Second secon	Host localhost Port: 8888 Path:	File Generic HTTP MAP Sopes Composite Source File Flow Composite Source Flow Composite Source Composite Source Flow Composite Source Flow Composite Source Composite Source Composi
	OK Cancel Message Flow Global Elements Configuration XML	Components Compon
i □ *		i a 🖹 @ 📴 M 🚍

45 —

Working with Components and Patterns _____

The Java component

Double-click on the Java component to configure it. You can import the class you had created.

- 46 -

Custom filters

Filters specify conditions that must be met for a message to be routed to a service. If the condition is met, the message will go to another component. You can also create your own filter. To create a filter, implement the Filter interface, which has a single method. You can import a custom filter class using the extended Filter interface.

Mule - flowref/flows/FlowRef	Y Pattern Properties		🖬 🖻 🔀
File Edit Navigate Search Project	Custom The Custom Filter is a user-implemented filter.		
😫 Package E 🛛 🍃 Type Hiera	General Spring Documentation		- 0
E	Display		Filter:
General Activity General Activity General Activity	Display Name: Custom		Select
🕀 🛋 JRE System Library [JavaSE	Generic		Carl Endpoints
Mule Runtime [Mule Server	Class:		Copes Copes
🥮 src/test/java			Components
- 🥮 src/main/resources			a Transformers
src/test/resources			🔁 Filters 🛛 🗠
FlowRef.mflow			And And
- 🗁 mappings			Custom
mule-project.xml			Exception
🗉 😂 helloworld			Expression
🗄 😂 usdconverter			Filter Reference
			Idempotent Message
			Message
			Message Property
			Not 🛛
			Y Or
			Payload
Securities St			Regex
		_	Schema Validation
	OK Cancel		Wildcard
			Control
			arror Handling
			Cloud Connectors
	Message Flow Global Elements Configuration XML		
i ∎*			j a 🦹 @ 😣 M 🚍

Working with Components and Patterns -

How it works...

Components generally execute whenever a message is received; the logic embedded into components cannot be modified. Components such as **Logger**, **Echo**, **Java**, **Flow Ref**, and **Expression** all come under this category.

In the process of scripting a component, you have to develop your own business logic by writing a script, or you can import a script file written in scripting languages such as Ruby, Python, and Groovy. The Java component allows you to reference a Java class.

Using the Echo component to display the message payload

The **Echo** component is used to display the message payload. The Echo component is used for displaying message payloads, which receives the end user HTTP request and returns the payload message to the HTTP response, which is then sent to the end user. In this recipe, you will see how to use and configure the Echo component in Mule Studio.

Getting ready

In this example, we'll use the following components: HTTP, Logger, and Echo.

1. Open Mule Studio and enter the workspace name as shown in the following screenshot:

9 8	Vorkspace Launcher
	Select a workspace
B al	Mule Studio stores your projects in a folder called a workspace. Choose a workspace folder to use for this session.
]	Workspace: E:\MuleCookBook Srowse
(KR)	Use this as the default and do not ask again
8	OK Cancel
744] 	t WuleSoft ©2011-2012 MuleSoft, Inc. All rights reserved.



2. To create a new project, go to **File** | **New** | **Mule Project**. Enter the project name, Echo, and click on **Next** and then on **Finish**. Your new project is created. You can now start the implementation.



49 —

Working with Components and Patterns -

3. To create a flow, go to the Echo.mflow file, drag the **HTTP** Endpoint onto the canvas, and configure it by double-clicking on it.



How to do it...

In this section, we will see how we can use the Logger and Echo components in a flow.

1. Double-click on the **HTTP** Endpoint to configure it. You can change the hostname and port number. Here, we have used port number 8585.

Mule - echo/flows/Echo.mflov	Y Endpoint Properties	
File Edit Navigate Search Project : □ • □ • □ • □ • • • • • • • • • • • • •	HTTP (Inbound Endpoint) The HTTP Endpoint allows Mule application to connect to web resources through the HTTP transport protocol. This endpoint can also implement security through HTTP5.	
Package E Package E Packag	General Advanced References HTTP Settings Document/ation Display Display Name: HTTP Exchange Patterns one-way orequest-response Basic Settings Enabling HTTPS will require configuring a HTTPS Connector Host: localhost Port: Ø585 Path: 0	Filter:
E Outline 🛛	О Сок Сок	Cancel Script
	Message Flow Global Elements Configuration XML	Cloud Connectors
1 D*		i e 🖹 @ 😡 M 🖳
🐉 start 🔰 🜍 Echo Example	🔮 Chapter 2 - Mic 🔮 659_The Auth 🦉 untitled - Paint 🛛 🍟 Mule - echo/flo	👜 440105_01_1s 🕜 🖉 🥵 12:20 PM

Working with Components and Patterns -

 To display messages on the console, drag the Logger component onto the canvas and configure it. The Logger component uses an expression to determine what information in the message should be displayed on the console. Mule Expression Language (MEL) is the primary language used for formulating such expressions throughout the Mule ESB.



3. Double-click on the **Logger** component to configure it. You can see different types of levels. The Logger component level is used for displaying error messages or exceptions. We have selected the **INFO** level here. In the message box, you should use the expression # [message:payload]. This expression is used for displaying messages on the console.

Mule - echo/flows/Echo.mflov	Y Pattern Properties		🖬 🖬 🛛
File Edit Navigate Search Project	Logger The Logger Component performs logging using an expression that determines what should be logged. By default the current messages is logged using the DEBUG level.		
Package E	General Documentation Display Display Name: Logger Generic Message:		Filter: Select Select Select Components Com
E Outline 🛛			
	ОК СС	ancel	Filters
			Flow Control
			Cloud Connectors
	Message Flow Global Elements Configuration XML		Code Competers
i □ *			i e 🔝 @ 🔯 M 🚍

53 —

Working with Components and Patterns -

4. Drag the **Echo** component onto the canvas; there is no need to configure it. Messages sent to an **Echo** component simply return the message payload as the response to an end user.





How it works...

In this section, you will see how to deploy the application and how to run the application on the browser.

 Now we are ready for the deployment. If you haven't saved your application code, do save it. After saving your project, right-click on the Echo.mflow file and go to Run As | Mule Application.



55 —

Working with Components and Patterns _____

2. If your application code is successfully deployed, you will see the following message on the console: Started app 'helloworld'.

🗸 Mula - ballowedd@nweAlalloWorld m@nw - Mula Studio 📰 🗐 🔊
[1] [1] • [1] [2] [2] [2] · [
E Mule
🖸 Console 🖄 💼 🛱 🎆 🛃 💭 🛃 💭 🛃 🖓 👘 🖻 🖓 👘
HelloWorld [Mule Application] C:\Program Files\Java\tre6\bin\javaw.exe (Nov 20, 2012 3:22:30 PM)
connected=true
supportedProtocols=[http]
serviceOverrides= <none></none>
INFO 2012-11-20 15:22:34,375 [main] org.mule.lifecycle.AbstractLifecycleManager: Starting connector: connector.http.mule.de
INFO 2012-11-20 15:22:34,375 [main] org.mule.lifecycle.AbstractLifecycleManager: Starting model: _muleSystemModel
INFO 2012-11-20 15:22:34,390 [main] org.mule.construct.FlowConstructLifecycleManager: Starting flow: HelloWorldFlow1
INFO 2012-11-20 15:22:34,390 [main] org.mule.processor.SedaStageLifecycleManager: Starting service: HelloWorldFlow1.stage1
INFO 2012-11-20 15:22:34,390 [main] org.mule.component.ComponentLifecycleManager: Starting component: commponent.27970580
INFO 2012-11-20 15:22:34,390 [main] org.mule.transport.http.HttpConnector: Registering listener: HelloWorldFlow1 on endpoin
INFO 2012-11-20 15:22:34,406 [main] org.mule.transport.service.DefaultTransportServiceDescriptor: Loading default response
INFO 2012-11-20 15:22:34,406 [main] org.mule.lifecycle.AbstractLifecycleManager: Initialising: 'null'. Object is: HttpMessa
INFO 2012-11-20 15:22:34,421 [main] org.mule.Hifedycle.AbstractLifedycleAnaager: starting: 'null'. Object 1s: httphessageke
INFO 2012-11-20 15:22:34,421 [main] org.mule.module.management.agent.WrapperManagerAgent: This JVM hasn't been launched by
INFO 2012-11-20 15:22:34,453 [main] org.mule.module.management.agent.imxagent: Attempting to register service with name: Hu
INFO 2012-11-20 15:22:34,453 [main] org.mule.module.management.agent.JmxAgent: kegistered indpoint service with name: mule.
INFO 2012-11-20 15:22:34,453 [main] org.mule.module.management.agent.JmxAgent: kegistered connector service with name mule.
INFO 2012-11-20 15:22:34,460 [main] org.mule.module.launcher.appication.berauthulexppication: keidad interval: 3000
IN 0 2012-11-20 13:22:34,460 [main] ofg.mule.belationtext:
t implication: helloworld t
- Application. Hellowolld -
* *
* Agents Dunning.
* JW årent *
INFO 2012-11-20 15:22:34.468 [main] org.mule.module.launcher.DeploymentService:
+ Started app 'helloworld' +
+++++++++++++++++++++++++++++++++++++++
· · · · · · · · · · · · · · · · · · ·
🗗 🕈 🗶 🐵 😡 🖬 😜

3. Copy the URL http://localhost:8585 and paste it in your browser.

¥ Mule - echo/flows/Echo.mflow - Mule Studio 🔲 🗗	×
File Edit Navigate Search Project Run Window Help	
1 + 🖫 励 △ 歩 • Q • Q • I 🖶 ♂ • I 🎒 🖉 • I 💷 2 · □ I = V ↔ · · · · I ♡ ♡ 🕱	
E Mule	
😑 Console 🛛 🗧 🕷 🖓 🔛 🖅 🖓 🖻 🖆 🖓 🖆 🖻 - 🗂 - 🍟	8
Echo [Mule Application] C:\Program Files\Java\jre6\bin\javaw.exe (Nov 26, 2012 12:37:11 PM)	
	^
]15 [main] org.mule.lifecycle.AbstractLifecycleManager: Starting connector: connector.http.mule.default	
]15 [main] org.mule.lifecycle.AbstractLifecycleManager: Starting model: _muleSystemModel	
)15 [main] org.mule.construct.FlowConstructLifecycleManager: Starting flow: EchoFlow1	
331 [main] org.mule.processor.SedaStageLifecycleManager: Starting service: EchoFlow1.stage1	
331 [main] org.mule.component.ComponentLifecycleManager: Starting component: commponent.25001087	
]31 [main] org.mule.transport.http.HttpConnector: Registering listener: EchoFlow1 on endpointUri: http://localhost:8585	
)46 [main] org.mule.transport.service.DefaultTransportServiceDescriptor: Loading default response transformer: org.mule.trans	
)46 [main] org.mule.lifecycle.AbstractLifecycleManager: Initialising: 'null'. Object is: HttpMessageReceiver	
]62 [main] org.mule.lifecycle.AbstractLifecycleManager: Starting: 'null'. Object is: HttpMessageReceiver	
162 [main] org.mule.module.launcher.application.DefaultMuleApplication: Reload interval: 3000	

- 56 -

4. To see the output on the console, paste the URL in your browser and type in /EchoExample. When a user types http://localhost:8585/EchoExample in the browser, Mule returns a message in the browser that reads /EchoExample, as shown in the following screenshot:

Mule - echo/flows/Echo.mflow - Mule Studio
e Edit Navigate Search Project Run Window Help
11 + 🗄 🔂 🖆 🕻 🛠 + 🚺 + 🍕 - 🕴 🞯 + 🦢 🖋 + 🏾 🗉 🖢 🖉 - 🤤 🖓 - 🗯
ŝ 💟 Mule
⊇ Console 🛛 🗧 🗰 🚵 🕞 🖉 🛃 🖃 🛣 🖬 🖓 🐨 🖆 🖓 🖓 🖓
ho (Mule Application) C: (Program Flex) avalytes (bin) avaw exe (Nov 26, 2012 12:37:11 PM)
NFO 2012-11-26 12:37:15,031 [main] org.mule.transport.http.HttpConnector: Registering listener: EchoFlow1 on endpointUri: ≙
NFO 2012-11-26 12:37:15,046 [main] org.mule.transport.service.DefaultTransportServiceDescriptor: Loading default response
NFO 2012-11-26 12:37:15,046 [main] org.mule.lifecycle.AbstractLifecycleManager: Initialising: 'null'. Object is: HttpMessa
NFO 2012-11-26 12:37:15,062 [main] org.mule.lifecycle.AbstractLifecycleManager: Starting: 'null'. Object is: HttpMessageRe
NFO 2012-11-26 12:37:15,062 [main] org.mule.module.launcher.application.DefaultMuleApplication: Reload interval: 3000
NFO 2012-11-26 12:37:15,062 [main] org.mule.module.management.agent.WrapperManagerAgent: This JVM hasn't been launched by
NFO 2012-11-26 12:37:15,093 [main] org.mule.module.management.agent.Jmx&gent: &ttempting to register service with name: Mu
NFO 2012-11-26 12:37:15,093 [main] org.mule.module.management.agent.JmxAgent: Registered Endpoint Service with name: Mule.
NFO 2012-11-26 12:37:15,093 [main] org.mule.module.management.agent.Jmx&gent: Registered Connector Service with name Mule.
NFO 2012-11-26 12:37:15,093 [main] org.mule.DefaultMuleContext:

Application: echo *
OS encoding: Cp1252, Mule encoding: UTF-8 *
*
Agents Running: *
JMX Ågent *

Using the command prompt

To run a Mule application, enter the following command on the command prompt:

```
mule [-config <your-config.xml>]
```

Here, <your-config.xml> is the Mule configuration file you want to use. If you don't specify the configuration file, Mule looks for mule-config.xml, which is a generic name that does not exist in the default configuration file. If you have only one configuration file, you can name it mule-config.xml so that you can run Mule with it just by typing in mule. To stop Mule, press *Ctrl* + *C*.

Using a Flow Reference component to synchronously execute another flow

Flow Reference is used to synchronously execute another flow that is external to the current flow. If a message reaches the Flow Reference component, Mule invokes the external flow referenced by it. Once the referenced flow completes, the control passes back to the initiating flow only after the external process is completed.

57

Working with Components and Patterns -

Getting ready

To demonstrate this example, we'll use the following four components: HTTP, Logger, Java, and Flow Ref.

1. Open Mule Studio and enter the workspace name as shown in the following screenshot:

My Makwork Storage	distinui to	Nobile Park	V Workepace Launcher	
Nacycle din		endaraba	Select a workspace Mule Studio stores your projects in a folder called a workspace. Choose a workspace folder to use for this session.	
internat Explorer	TaamWawar 7	D Nulecooks	Workspace: 21/MuleCool:Book Browse Use this as the default and do not ask again	
	anaritarik-2.0		OK Cancel	
Reader 5.0	Nove Toxic		MuleSoft* @2011-2012 MuleSoft, Inc. All rights reserved.	

2. To create a new project, go to **File** | **New** | **Mule Project**. Enter the project name, FlowRef, and click on **Next** and then on **Finish**. Your new project has been created now; so we are ready to start the implementation.

🎽 Mule - Mule Studio					∎₽
File Edit Source Refactor Navigate	Search Project I	Run Window Help			
New	Alt+Shift+N	🗟 🔤 Mule Project		- * + + -	
Open File		/ Java Project			
Close	Ctrl+W	📸 Project			_
Close All	Ctrl+Shift+W	📌 Mapping Flow			
🔛 Save	Ctrl+S	Mule Flow			
Save As		🖶 Package			
i Save All	Ctrl+Shift+S	🎯 Class			
Revert		🞯 Interface			
Move		🞯 Enum			
Rename	F2	Annotation			
🗞 Refresh	F5	Source Folder			
Convert Line Delimiters To	•	Set			
i Print	Ctrl+P	E Folder			
Switch Workspace	,	Untitled Text File			
Restart		Example			
🚵 Import			a. 1. 1.		
Export		Uther	CENHIN	ļ	
🔯 Export diagram to					
Properties	Alt+Enter				
1 FlowRef.mflow [flowref/flows] 2 package.html [activity/src/main/] 3 Activity.mflow [activity/flows] 4 StringToNumber.java [usdconverter/]				
Exit					
All outline is not available.		_			
				a 👔 🖉	🙆 1/1 🛛

- 58 -

How to do it...

In this section, you will see how to configure the Java component and the Flow Ref component. Here you are creating a class, and the output will be displayed on the browser through this class.

1. To create a class, go to src/main/java, right-click on it, and go to New | Class.
Create a class named Greeting under the package com.org; here, we create the
muleCookBook method and its return type is set to String:

```
public String muleCookBook(String str)
{
    return "HelloMule"+str;
}
```

You can see the creation of this method in the following screenshot:




2. To create a class, right-click on the package. Create one more class, World, under the same package. Here, we create the method cookbook and its return type is set to String:

```
public String cookBook(String str)
{
    return "CookBook";
}
```

You can see the creation of this method in the following screenshot:



3. To create a flow, go to the FlowRef.Mflow file. Drag the **HTTP** Endpoint onto the canvas and configure it.



4. Double-click on the **HTTP** Endpoint to configure it. You can change the hostname and port number. We have used the port number 8989 here. Click on the **OK** button. By default, the **request-response** method is selected, as shown in the following screenshot:

Mule - flowref/flows/FlowRef	Y Endpoint Properties		
File Edit Navigate Search Project	HTTP (Inbound Endpoint) The HTTP Endpoint allows Mule application to connect to web resources through the HTTP transport protocol. This endpoint can also implement security through HTTP5.		
Mule Package E Type Hiera Package E Type Hiera Package E	General Advanced References HTTP Settings Documentation Display Display Name: HTTP Exchange Patterns One-way Orequest-response Basic Settings Enable HTTPS will require configuring a HTTPS Connector Host: Iocahost Port: 9999 Path: OK	Cancel	Filter: Select Selec
	Message Flow Global Elements Configuration XML		Cloud Connectors
1 D *			: a 🖹 @ 🔃 M 🖬

- 62 -

5. To import a class, drag the **Java** component and configure it.



6. Double-click on the **Java** component to configure it. Just click on the **Browse** button and a new window, **Class name browser**, will open. Here you can import the World. java class, which was created before, and click on the **OK** button.







7. To reference another flow name, drag the **Flow Reference** component onto the canvas.

8. Before configuring the **Flow Reference** component, you have to drag the **Java** component onto another flow (you can see this in the following screenshot). Configure that **Java** component. If you create another flow, just drag the component onto the canvas outside the first flow; this will create another flow.



9. Double-click on the **Java** component to configure it; change the display name so you can identify the class name. Import the Greeting.java class that was created before and click on the **OK** button.



Million (Immediation (Clause Contraction)	for the second se		
File File Nevinte Courts Design	Y Pattern Properties		
rile Luit Navigate Search Project	Flow Reference		
	The Flow Reference Component allows a Mule Flow to be referenced such that the message processing will		
🖹 📔 Mule	continue in the referenced flow before returning.		
🚦 Package E 🕺 🍃 Type Hiera	General Documentation		- 8
			Filter:
🕀 🔂 activity	Display Name: Flow Reference		
🔹 😂 echo			Select
Flowref			🚰 Endpoints 🛛 🗠
	How Name: 0[HowketHow2		💓 Ajax
			🗐 Database
			😑 FTP
			🤚 File
			Generic
			(HTTP
			IMAP
			Scoper (
			Acvnc
			Composite Source
			Flow
			Q Foreach
			Message Enricher
			() Poll
			Sub flow
E Outline 🛛			Components
		ancel	a Transformers
			Carliers
			Control
			Cror Handling
			Cloud Connectors
	Message Flow Global Elements Configuration XML		
			i e 🖹 @ 📴 🖬 🚍

10. Now double-click on the **Flow Ref** component to configure it. Assign another Flow Reference name. Now we are ready to deploy our application.

How it works...

In this section, you will see how to deploy the application and how to run the application in the browser.

 Now we are ready for the deployment. If you haven't saved your application code, do save it. After saving your project, right-click on the Echo.mflow file and go to Run As | Mule Application.



2. If your application code is successfully deployed, you will see the message Started app 'helloworld' on the console.

Y Mule - flowref/flows/FlowRef.mflow - Mule Studio					
File Edit Navigate Search Project Run Window Help					
: □ + □ □ □ : 歩 • ● • • • = = = + ● ● / • : = = : 2 - 2 - ↓ ← + → - : : ♡ ♡ ×					
🖺 🔛 Mule					
🕒 Console 🛛 🛛 📕 🖓 🔛 😸 🕮 🛃 🗁 🗂 - 🗂 - 🌱					
FlowRef [Mule Application] C:\Program Files\Java\jre6\bin\javaw.exe (Nov 27, 2012 11:50:22 AM)					
}					
INFO 2012-11-27 11:50:28 578 [main] org mule lifecycle AbstractLifecycleManager: Starting connector: connector bttn mule de					
INFO 2012-11-27 11:50:25,593 [main] org.mule.lifecycle.AbstractLifecycleManager: Starting model: muleSystemModel					
INFO 2012-11-27 11:50:28.593 [main] org.mule.construct.FlowConstructLifecvcleManager: Starting flow: FlowRefFlow1					
INFO 2012-11-27 11:50:28,593 [main] org.mule.processor.SedaStageLifecvcleManager: Starting service: FlowRefFlow1.stage1					
INFO 2012-11-27 11:50:28,609 [main] org.mule.component.ComponentLifecycleManager: Starting component: commponent.8835546					
INFO 2012-11-27 11:50:28,671 [main] org.mule.transport.http.HttpConnector: Registering listener: FlowRefFlow1 on endpointUr					
INFO 2012-11-27 11:50:28,781 [main] org.mule.transport.service.DefaultTransportServiceDescriptor: Loading default response					
INFO 2012-11-27 11:50:28,781 [main] org.mule.lifecycle.AbstractLifecycleManager: Initialising: 'null'. Object is: HttpMessa					
INFO 2012-11-27 11:50:28,859 [main] org.mule.lifecycle.AbstractLifecycleManager: Starting: 'null'. Object is: HttpMessageRe					
INFO 2012-11-27 11:50:28,859 [main] org.mule.construct.FlowConstructLifecycleManager: Starting flow: FlowRefFlow2					
INFO 2012-11-27 11:50:28,859 [main] org.mule.processor.SedaStageLifecycleManager: Starting service: FlowRefFlow2.stage1					
INFO 2012-11-27 11:50:28,859 [main] org.mule.component.ComponentLifecycleManager: Starting component: commponent.22665343					
INFO 2012-11-27 11:50:28,859 [main] org.mule.module.management.agent.WrapperManagerAgent: This JVM hasn't been launched by					
INFO 2012-11-27 11:50:28,906 [main] org.mule.module.management.agent.JmxAgent: Attempting to register service with name: Mu					
INFO 2012-11-27 11:50:28,906 [main] org.mule.module.management.agent.JmxAgent: Registered Endpoint Service with name: Mule.					
INFO 2012-11-27 11:50:28,921 [main] org.mule.module.management.agent.JmxAgent: Registered Connector Service with name Mule.					
INFO 2012-11-27 11:50:28,921 [main] org.mule.module.launcher.application.DefaultMuleApplication: Reload interval: 3000					
INFO 2012-11-27 11:50:28,921 [main] org.mule.DefaultMuleContext:					

* Application: flowref *					
* OS encoding: Cp1252, Mule encoding: UTF-8 *					
* *					
* Agents Running: *					
* JMX Agent *					
INFO 2012-11-27 11:50:28,921 [main] org.mule.module.launcher.DeploymentService:					
+ Started app 'flowref' +					

i 🗈 🔪 🖉 🖉 🖉 🐘 🖬					

- 70 -

3. Copy the URL http://localhost:8989 and paste it in your browser.



4. To see the output, paste the URL onto your browser. You will see the output as shown in the following screenshot. Here, Hello Mule is called from the Greeting class through the **Flow Reference** component, and CookBook is called from the World class:



Publishing a RESTful web service using the REST component

REST stands for **Representational State Transfer**. REST exposes a much simpler interface than SOAP. REST components are bound with HTTP. So, if you are designing an application to be used exclusively on the Web, REST is a very good option. RESTful applications simply rely on the built-in HTTP security. A REST design is good for database-driven applications and also when a client wants quick integration.

Getting ready

In this example, we'll use three components: HTTP, Logger, and REST.

1. Open Mule Studio and enter the workspace name as shown in following screenshot:

Y Workspace Launcher	
Select a workspace Mule Studio stores your projects in a folder called a workspace. Choose a workspace folder to use for this session.	
Workspace: Et/MuleCookBook	Browse
Use this as the default and do not ask again	OK Cancel
t WuleSoft* ©2011-2012 MuleSoft, Inc. All rights reserved.	



2. To create a new project, go to **File** | **New** | **Mule Project**. Enter the project name, REST, and click on **Next** and then on **Finish**. Your new project has been created; now you can start the implementation.

🎽 Mule - Mule Studio			
File Edit Source Refactor Navi	igate Search Project	Run Window Help	
New	Alt+Shift+N	🕩 😋 Mule Project	
Open File		Java Project	
Close	Ctrl+W	Project	
Close All	Ctrl+Shift+W	/ 🔐 Mapping Flow	
Save	Ctrl+S	Mule Flow	
Save As		Package	
Revert	Ctrl+Shift+S	Class	
		G Enum	
Move Rename	F2	Annotation	
Refresh	F5	💕 Source Folder	
Convert Line Delimiters To		🕨 🍄 Java Working Set	
Print	Ctrl+P	Folder	
		Intitled Text File	
Restart			
Ma Import		Example	
A Export		😭 Other	Ctrl+N
In Export diagram to		_	
Export diagram to		_	
Properties	Alt+Enter	_	
1 FlowRef.mflow [flowref/flows]			
2 package.html [activity/src/main 3 Activity mflow [activity/flows]	n/]		
4 StringToNumber.java [usdconv	/erter/]		
Exit		-	
All outline is not available.		_	
1 =0			
: •			

How to do it...

Here we will create a RESTful web service using the annotation. We will create a method named getwelcomeMsg().

 To create a class, go to src/main/java and right-click on it. Create a class named HelloWorldResource to print a message. Enter the package name and click on Next and then on Finish. Here we have used the JAX-WS annotation. For details on the JAX-WS annotation, you can refer to this URL: http://publib.boulder. ibm.com/infocenter/radhelp/v7r0m0/index.jsp?topic=/com.ibm. ws.jaxws.emitter.doc/topics/rwsandoc002.html.

```
package com.org;
import javax.ws.rs.GET;
import javax.ws.rs.Path;
@Path("/myrest")
```



```
public class HelloWorldResource {
    @GET
    public String getWelcomeMsg () throws Exception {
        return "Hi MuleCookBook!!!!";
    }
}
The @Get annotation indicates that the annotated method responds
to an HTTP GET request.
```

The @Path annotation is used to map a given URL.

You can see the creation of this method in the following screenshot:



- 74

2. To create a flow, go to the Rest.mflow file. First of all, you have to drag the **HTTP** Endpoint from the palette and drop it on the canvas area.



3. Double-click on the **HTTP** Endpoint to configure it. You will see the hostname and port number. You can change the **Host** and **Port** field values. These two fields are mandatory. By default, port number **8081** will be taken by the Mule server. We have used port number 4343 here.

Mule - rest/flows/Rest.mflow	Y Endpoint Properties	_ 2 🛛
File Edit Navigate Search Project : □ · · · · · · · · · · · · ·	HTTP (Inbound Endpoint) The HTTP Endpoint allows Mule application to connect to web resources through the HTTP transport protocol. This endpoint can also implement security through HTTPS.	
Package E Packa	General Advanced References HTTP Settings Documentation Display Name: Display Name: Exchange Patterns One-way Image: Enable HTTPS Image: Enable HTTPS Image: Enable HTTPS Image: Path:	Filter: Select Secopes Components Components Components Components Flow Reference Groovy HTTP Response Builder Java Javascript Logger Ogger Opython SEST
	OK Cancel Message Flow Global Elements	Kuby SOAP Script Script Transformers Filters Flow Control General Anding Fror Handing Could Connectors
: •	1 6	• 🖹 @ 🕒 M 🖏 🕴 🖷 🗐

- 76 -

77 –

4. To create a RESTful web service, drag the **REST** component from the palette and drop it on the canvas area. This **REST** component is used to make a REST service available via **Jersey**. Jersey is an open source, production-quality, JAX-RS (JSR 311) reference implementation for building RESTful web services.

REST is the formalized architecture of HTTP and is based on concepts of resources, links, and a uniform interface. It uses the HTTP protocol. We can create a web service using the **REST** component.



5. Double-click on the **REST** component to configure it. Here you can add a **Java** component that was created before.



- 78

6. To display messages on the console, drag-and-drop the **Logger** component from the palette onto the canvas area and configure it.



7. To configure the **Logger** component, double-click on it. You will see the **Message:** textbox; just enter the payload expression, # [payload].

Mule - rest/flows/Rest.mflow	V Pattern Properties		🖬 🖬 🔀
File Edit Navigate Search Project	Logger The Logger Component performs logging using an expression that determines what should be logged. By default the current messages is logged using the DEBUG level.		
Package E Type Hiera Ty	General Documentation Display Display Name: Caneric Message: [#[payload]] Level: INFO Category: Category: Image: Configuration XML Configuration XML	ancel	Filter: Select Select Select Select Components Components Components Components Components Components Select Selec
: •			: " 🕰 🥗 🖄 🖬 🖵 🗝

- 80 -

How it works...

To deploy your application go through the following steps:

1. If you haven't saved your application code, do save it. After saving your project, right-click on the Echo.mflow file and go to **Run As** | **Mule Application**.

🖌 Mule - rest/flows/Rest.mflow - Mule Studio 🗾 🖸 🔀						
File Edit Source Refactor Navigate Search Project Run Window Help						
i 📬 • 🔛 🕼 🗅 i 🏇 • 💽 • 🏊 • i 🏶 🎯 • i 🥭 🔗	i 🗈 • 🗒 🖄 i \$\$ • O • O₂ • i # @ • i @ \$\$ /? • i = i Ø · \$2 • \$0 • \$ • \$ 0 \$ \$ \$					
🖹 🔛 Mule						
😫 Package E 🛛 💲 Type Hiera 🗖 🗖 🚺 HelloWorldResource	e.java 🤎 *Rest 🕱					
□ 🔄 🏹		Filter:				
B B activity		Select				
i ⊕ 🔁 helloworld		Endpoints				
🖻 🖻 pic		Scopes				
e 😂 rest		Components 🗠				
I I I I I I I I I I I I I I I I I I I		(0+0) Echo				
Grand Strein Open F3		Expression				
😑 🌐 com. Open With 🕨 🕨	·	Flow Reference				
B → B →		Groovy				
		HTTP Response				
🖉 src/test/ 🗎 Copy Qualified Name	Logger REST	Builder				
😑 🗁 flows 📫 Paste Ctrl+V		🔬 Java				
Rest X Delete Delete		Javascript				
Build Path		Logger				
mule-prc Refactor Alt+Shift+T •		📀 Python				
🗷 😂 script	RE31	S REST				
Export		Ruby				
A Refrech F5		SOAP				
Assign Working Sets		Script				
E Outline S						
Run As	1 Dup op Server Alt+Shift+X D	a Transformers				
Debug As	2 Mule Application	Charles Filters				
Profile As		Control				
Team •	Run Configurations	🚰 Error Handling				
Compare With		Cloud Connectors				
Replace With	Elements Configuration XML					
Rest.n 2 CloudHub		i e 🖹 @ 📴 🖬 🗉 🦐				

81 -

2. If your application code is successfully deployed, you will see the message <code>Started app 'helloworld'</code> on the console.

Y Mule - rest/flows/Rest.mflow - Mule Studio	X
File Edit Navigate Search Project Run Window Help	
: 11 - □ @ △ : 参 - ② - Q : 巻 ଡ - : 参 - / : ◎ : 2 - ? - ♡ - : ◇ ◇ × : ◇ ◇ ×	
🖹 🔛 Mule	
🕒 Console 🛛 🔰 🛤 🔜 🖉 🖉 🖬 🗠 🔂 🖓 👘	
Rest [Mule Application] C:\Program Files\Java\ire6\bin\javaw.exe (Dec 4, 2012 11:31:43 AM)	8
connected=true	Ju
supportedProtocols=[http]	
serviceOverrides= <none></none>	
TNFO 2012-12-04 11:31:51 453 [main] org mule liferurle abstractLiferurleManager. Starting connector, connector bttp mule	
INFO 2012-12-04 11:31:51.468 [main] org.mule.lifegurle.kbstractLifegurleManager: Starting model: muleSystemMidel	
INFO 2012-12-04 11:31:51,468 [main] org.mule.construct.FlowConstructLifecvcleManager: Starting flow: testFlow1	
INFO 2012-12-04 11:31:51,468 [main] org.mule.processor.SedaStageLifecvcleManager: Starting service: testFlow1.stage1	
INFO 2012-12-04 11:31:51,468 [main] org.mule.component.ComponentLifecycleManager: Starting component: commponent.265173(
INFO 2012-12-04 11:31:51,468 [main] org.mule.transport.http.HttpConnector: Registering listener: testFlow1 on endpointUp	
INFO 2012-12-04 11:31:51,562 [main] org.mule.transport.service.DefaultTransportServiceDescriptor: Loading default respon	
INFO 2012-12-04 11:31:51,562 [main] org.mule.lifecycle.AbstractLifecycleManager: Initialising: 'null'. Object is: HttpMe	
INFO 2012-12-04 11:31:51,593 [main] org.mule.lifecycle.AbstractLifecycleManager: Starting: 'null'. Object is: HttpMessag	
INFO 2012-12-04 11:31:51,593 [main] org.mule.module.launcher.application.DefaultMuleApplication: Reload interval: 3000	
INFO 2012-12-04 11:31:51,609 [main] org.mule.module.management.agent.WrapperManagerAgent: This JVM hasn't been launched	
INFO 2012-12-04 11:31:51,656 [main] org.mule.module.management.agent.JmxAgent: Attempting to register service with name:	
INFO 2012-12-04 11:31:51,656 [main] org.mule.module.management.agent.JmxAgent: Registered Endpoint Service with name: Mu	
INFO 2012-12-04 11:31:51,671 [main] org.mule.module.management.agent.JmxAgent: Registered Connector Service with name Mu	
INFO 2012-12-04 11:31:51,671 [main] org.mule.DefaultMuleContext:	

* Application: rest *	
* OS encoding: Cp1252, Mule encoding: UTF-8 *	
* *	
* Agents Running: *	
* JMX Agent *	
INFO 2012-12-04 11:31:51,671 [main] org.mule.module.launcher.DeploymentService:	
+ startea app 'rest' +	
i 🗗	-6

- 82 -

3. Copy the URL http://localhost:4343/myrest and paste it in your browser.



4. To see the output, paste the URL on your browser and type /myrest; this is required because we have used the @Path annotation in the custom Java class.



83 -

Publishing a SOAP-based web service using the SOAP component

The Mule **SOAP** component is used for publishing, consuming, and proxying of SOAP web services within a Mule flow. Using the SOAP component, you can also enable Web Service Security. Apache CXF is an open source services framework. CXF helps you build services using frontend programming APIs such as JAX-WS and JAX-RS.

You can create a CXF web service by configuring a SOAP component in your Mule flow to perform any of the following CXF web service operations:

- Publish a simple service
- Publish a JAX-WS service
- Proxy a published service
- Consume a service using a simple client
- Consume a service using the JAX-WS client
- Proxy to a service

Getting ready

In this example, we will see how to create a SOAP-based web service using the SOAP component. To create a SOAP web service, we'll use three components: HTTP, Java, and SOAP.

1. Open Mule Studio and enter the workspace name as shown in the following screenshot:

Navibr		
SER.	V Workspace Launcher	
	Select a workspace Mule Studio stores your projects in a folder called a workspace. Choose a workspace folder to use for this session.	ANNINANA
כ	Workspace: E://MuleCookBook Browse	
aka	Use this as the default and do not ask again	
1 1/44	OK Cancel	
	WuleSoft* @2011-2012 MuleSoft, Inc. All rights reserved.	



2. To create a new project, go to **File** | **New** | **Mule Project**. Enter the project name, SOAP, and click on **Next** and then on **Finish**. Your new project is created and you are now ready to start the implementation.



3. To create a class, go to src/main/java, right-click on it, and go to New | Interface.
Create an interface named HelloService under the package com.org. Here, we
create the hiMule method and set its return type to String.

```
package com.org;
import javax.jws.WebService;
@WebService
public interface HelloService {
   public String hiMule(String str) throws Exception;
}
```

You can see the creation of this class in the following screenshot:



How to do it...

In this section, you will refer to the JAX-WS annotation and create the method hiMule(). We will use this method to generate the output using a browser.

1. Create a class called HelloServiceImpl under the same package directory (com. org) and implement it with the interface. Here, we have used the @WebService annotation to create a SOAP-based web service and override the hiMule method. You can refer to this URL for more information on the JAX-WS annotation: http:// publib.boulder.ibm.com/infocenter/radhelp/v7r0m0/index. jsp?topic=/com.ibm.ws.jaxws.emitter.doc/topics/rwsandoc002.html. package com.org; import javax.jws.WebService; @WebService(EndpointInterface="com.org.HelloService", serviceName="HelloService") public class HelloServiceImpl implements HelloService {

- 86

@Override

```
public String hiMule(String str) throws Exception {
    // TODO Auto-generated method stub
    return "Hello "+str;
  }
}
```

Through the @WebService annotation, we call the HelloService interface and also provide a service name.



Y Mule - soap/flows/Soap.mflow - Mule Studio File Edit Navigate Search Project Run Window Help i 📸 • 🔛 🕲 🗁 i 🕸 • 🕖 • 💁 • i 🏦 🞯 • i 🎒 🖋 • i 🖿 ii 🦢 🖉 • ii 🖉 · 🏹 · 🏷 🔶 • - - - - ii 🞸 📎 🗶 🖹 🔛 Mule 😫 Package E 🕺 👔 Type Hiera) 🖓 🗊 🚺 🖅 HelioServiceImpl.java 🚺 HelioService.java 🔛 *Soap 🖄 - -8 🕒 🔄 🎽 Filter: Ju detivity
 detivity
 detivity
 detivity
 detivity
 detivity ^ l> Select

 ⊕ ⊕ flowref

 ⊕ ⊖ pic

 ⊕ ⊖ pic

 ⊕ ⊖ rest

 ⊕ ⊖ script

 ⊖ ⊖ script

 ⊕ ⊕ script

 🚰 Endpoints 皹 Ajax 😑 Database 😑 FTP 😑 File flow: SoapFlow1 📝 Generic src/main/java HTTP НТТР 🕮 src/test/java HTTP B src/main/resources a Scopes src/test/resources Async Flows 🛃 Composite Source 🗁 mappings Flow src mappings src main bettest Q Foreach Message Enricher 1 Mule-project.xml O Poll usdconverter ~ > Sub flow V - 8 🗄 Outline 🛛 Components

Message Flow Global Elements Configuration XML

Carl Transformers

Flow Control

Cloud Connectors

j e 🔝 @ 🕓 M 🗉 🔫

2. To create a flow, go to the Soap.mflow file. Drag the **HTTP** Endpoint from the palette and drop it onto the canvas. You have to configure the **HTTP** Endpoint.

88

-

....

₽

3. Double-click on the **HTTP** Endpoint to configure it. You will see the **Host** and **Port** fields. These two fields are mandatory. In this example, we used localhost as the hostname. By default, port number **8081** will be taken by the Mule server. We have used port number 2121 here.

Mule - soap/flows/Soap.mflov	🗑 Endpoint Properties	_ _ N
File Edit Navigate Search Project	HTTP (Inbound Endpoint) The HTTP Endpoint allows Mule application to connect to web resources through the HTTP transport protocol. This endpoint can also implement security through HTTPS.	
Package E ⊠ type Hiera ⊕ ⊕ activity ⊕ ⊕ echo ⊕ ⊕ helloworld ⊕ ⊕ pic ⊕ ⊕ rest	General Advanced References HTTP Settings Documentation Display Display Name: HTTP Exchange Patterns One-way @ request-response Basic Settings	Filter: Ju Select Endpoints co Majax Database
	Enable HTTPS VI require configuring a HTTPS Connector Host: iceahost Port: 2121 Path:	FTP File Generic HTTP Commonstraints HTTP Commonstraints File Account
Construction Construction		Composite Source Flow Flow Foreach Message Enricher Pol Pol
	OK Cancel	Sub How Components Com
	Message Flow Global Elements Configuration XML	 Cloud Connectors

89 -

4. To create a SOAP-based web service, drag the **SOAP** component from the palette, drop it onto the canvas, and configure it. Here, we create web services using the **SOAP** component.



5. Double-click on the **SOAP** component to configure it. We select the **JAX-WS** services for the operation and then we will import the HelloService interface that was created before.

Mule_soap/flows/Soap.mflow	₩ Pattern Properties	
File Edit Navigate Search Project : ひ・ : 喩	SDAP The SOAP Component will publish a SOAP web service via JAX-WS Annotations, WSDL, or CXF Simple Service using Apache CXF.	
Package E Package E	General Interceptors Advanced Documentation Display Display Name: SOAP General: Config Reference: Image: Config Reference: Image: Config Reference: Operation: a) JAX-WS service Image: Config Reference: Image: Config Reference: Inbound Attributes Proxy Service Image: Config Reference: Image: Config Reference: Binding ID: DAX-WS Service Image: Config Reference: Image: Config Reference: Image: Config Reference: Port: Proxy Service Image: Config Reference: Image: Config Reference: Image: Config Reference: Image: Config Reference: Service: Common: Common: Generate from WSDL Image: Config Reference: Image: Config Reference: Service: Common: Image: Config Reference: Image: Config Reference: Image: Config Reference: Image: Config Reference: Validation Enabled Image: Config Reference: Image: Config Reference: Image: Config Reference: Image: Config Reference: Image: Config Reference: Image: Config Reference: <td< th=""><th>Filter: Select Selec</th></td<>	Filter: Select Selec
: •		j a 🖹 @ 📴 M 🗉 🖷

- Y Mule soap/flows/Soap.mflow Mule Studio File Edit Navigate Search Project Run Window Help i 📸 • 🔢 🗟 🖆 i 🕸 • 🚺 • 💁 • i 🏦 🞯 • i 🎒 🖋 • i 🖿 i 🤰 🖉 • i 🖛 i 🖢 - 🤤 - 🖓 • 🗰 🖹 🔛 Mule 😫 Package E 🕺 🦹 Type Hiera) 🖓 🗖 🚺 *HelloServiceImpl.java 🚺 HelloService.java 🛛 😭 Soap 🖄 - -8 🗏 🔄 🎽 Filter: Ju detivity
 detivity
 detivity
 detivity
 detivity
 detivity ^ Select

 B
 → P holworld

 B
 → P holworld

 B
 → P pic

 B
 → rest

 B
 → Script

 C
 → Scap

 B
 → RE System Library [JavaSE-1.6]

 B
 → Mule Furthine [Mule Server 3.3.0 C

 Carl Endpoints Copes Components (0+0) Echo Expression flow: SoapFlow1 📝 Flow Reference src/main/java ☆ Groovy SOAP HTTP Response Builder 🕮 src/test/java HTTP SOAP B src/main/resources 🛓 Java I src/test/resources Flows Javascript E Logger - 🗁 mappings SOAP Python src mappings src main bettest S REST -💎 Ruby Mule-project.xml SOAP sdconverter > Script - - B E Outline 🔀 a Transformers 🔁 Filters **∞**→∞→∞ Flow Control Carl Error Handling Cloud Connectors Message Flow Global Elements Configuration XML ∎≎ j e 🔝 @ 🕓 M 🗉 🔫
- 6. To import a class, drag the **Java** component from the palette, drop it on the canvas area, and configure it. If you want to change its name, you can do so.

7. Double-click on the **Java** component to configure it. Here, we import the HelloServiceImpl class that was created before.



How it works...

To deploy your application, right-click on your .mflow file and deploy your Mule application by performing the following steps:

1. If you haven't saved your application code, do save it. After saving your project, right-click on the Echo.mflow file and go to **Run As | Mule Application**.



2. If your application code is successfully deployed, you will see the message Started app 'helloworld' on the console.

New management	-) (=)	
¥ Mule - soap/Nows/Soap.mNow - Mule Studio	JĽ	
File Edit Navigate Search Project Run Window Help		
🎦・🗒 🗟 🏇・ 🖉・ 隆 🞯・ 🥭 🔗・ 🚥 🎦 - 🎦 - 🍄 - 中 - ク 🏷 🗶		
🖹 🔛 Mule		
🕒 Console 🕴 🐂 📓 🕞 🖉 🛃 🗁 🖬 🖬 🖉	. 9	••••
Soap [Mule Application] C:\Program Files\Java\jre6\bin\javaw.exe (Dec 4, 2012 6:15:16 PM)		8
connected=true	^	Ju
supportedProtocols=[http]		\square
serviceOverrides= <none></none>		
INFO 2012-12-04 18:15:21,218 [main] org.mule.hitecycle.abstractLitecycleManager: starting connector: connector.http.mu.	.ε	
INFO 2012-12-04 10:15:21,224 [main] org.mule.interview.asstactinecyclemanage: starting model: _mulesystemmodel		
INFO 2012-12-04 10.15.21,224 [main] org.mule.consci.uconsci.uconsci.uconterverina.ager. starting files. Sospirowi		
INFO 2012-12-04 10:15:27,204 [main] org.mult.processor.setascageLiferur(lawnager, Starting component, component 28211	sr	
THEO 2012-12-04 18:15:21 234 [main] organic: component of the Mith Connector: Peristering listener: Scamponent on andmoint	D.	
INFO 2012-12-04 18:15:21.250 [main] org.mule.transport.service.DefaultTransportServiceDescriptor: Loading default respo	11	
INFO 2012-12-04 18:15:21.250 [main] org.mule.lifecycle.hstractlifecycleManager: Initialising: 'null'. Object is: Httm	le le	
INFO 2012-12-04 18:15:21.265 [main] org.mule.lifecvcle.AbstractLifecvcleManager: Starting: 'null'. Object is: HttpMesse	ac	
INFO 2012-12-04 18:15:21.265 [main] org.mule.module.launcher.application.DefaultMuleApplication: Reload interval: 3000		
INFO 2012-12-04 18:15:21.265 [main] org.mule.module.management.agent.WrapperManagerAgent: This JVM hasn't been launched	1	
INFO 2012-12-04 18:15:21,296 [main] org.mule.module.management.agent.JmxAgent: Attempting to register service with name	4:	
INFO 2012-12-04 18:15:21,296 [main] org.mule.module.management.agent.JmxAgent: Registered Endpoint Service with name: N	£1	
INFO 2012-12-04 18:15:21,296 [main] org.mule.module.management.agent.JmxAgent: Registered Connector Service with name M	li I	
INFO 2012-12-04 18:15:21,296 [main] org.mule.DefaultMuleContext:		

* Application: soap *		
* OS encoding: Cp1252, Mule encoding: UTF-8 *		
* *		
* Agents Running: *		
* JMX Agent *		

INFO 2012-12-04 18:15:21,296 [main] org.mule.module.launcher.DeploymentService:		
+++++++++++++++++++++++++++++++++++++++		
+ Started app 'soap' +		

	-	
i 🛛 🖌 🖉 🖉 🖗 😡 🖬		-6
Working with Components and Patterns _____

3. Copy the URL ${\tt http://localhost:2121/}$ and paste it on your browser.

Mule - soap/flows/Soap.mflow - Mule Studio		
File Edit Navigate Search Project Run Window Help		
i 🗂 • 🗒 🖄 🖆 i 🏇 • 🕖 • 💁 • i 😫 🎯 • i 🥭 🖋 • i 💷 i 🖢 - 🖓 - 😓 - i 🖉 🗢 - i 🖉 🖏 👷		
🖺 🔛 Mule		
🖸 console 🕴 💼 🗰 🖓 🕼 🖬 💭 🛃 💭 🛃	•	
Soap [Mule Application] C:\Program Files\Java\jre6\bin\javaw.exe (Dec 4, 2012 6:15:16 PM)		8
	~	Ju
		\square
e.AbstractLifecycleManager: Starting connector: connector.http.mule.default		
e.AbstractLifecycleManager: Starting model: _muleSystemModel		
:t.FlowConstructLifecycleManager: Starting flow: SoapFlow1		
r.SedaStageLifecycleManager: Starting service: SoapFlow1.stage1		
t.ComponentLifecycleManager: Starting component: commponent.28921506		
t.http.HttpConnector: Registering listener: SoapFlow1 on endpointUri: http://localhost:2121		
t.service.DefaultTransportServiceDescriptor: Loading default response transformer: org.mule.transport.http.transform	ers.	
e.AbstractLifecycleManager: Initialising: 'null'. Object is: HttpMessageReceiver		
.e.AbstractLifecycleManager: Starting: 'null'. Object is: HttpMessageReceiver		
.auncher.application.DefaultMuleApplication: Reload interval: 3000		
wanagement.agent.WrapperManagerAgent: This JVM hasn't been launched by the wrapper, the agent will not run.		
wanagement.agent.JmxAgent: Attempting to register service with name: Mule.soap:type=Endpoint,service="SoapFlow1",cons	iecto:	
anagement.agent.JmxAgent: Registered Endpoint Service with name: Mule.soap:type=Endpoint,service="SoapFlow1",connect	or=c	
anagement.agent.JmxAgent: Registered Connector Service with name Mule.soap:type=Connector,name="connector.http.mule	defa	
[uleContext:		

*		
.autoret.vepioymentsetvice.		
+		
	>	1
	5 I/I 🔄	1 =6

- 96 -

4. To see the output, paste the URL on your browser and type in /hi?wsdl; here, wsdl stands for **Web Services Description Language**.

🗋 localhost:2121/hi?wsd 🛛 🗙 📃		D	23
← → C	\$	6	Ξ
This XML file does not appear to have any style information associated with it. The document tree is shown below.			^
<pre>w<wsdl:definitions "="" helloserviceservice"="" http:="" org.com="" targetnamespace="http://org.com/" xmlns:ns1="http://schemas.xmlsoap.org/soap/http" xmlns:soap="http://schemas.xmlsoap.org/wsdl/so
xmlns:tns=" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns:xsd="http://www.w3.org/2001/XMLSc
name="> */wsdl/tymes></wsdl:definitions></pre>	ap/" hema"		
<pre>v<xs:schema attributeformdefault="unqus
elementFormDefault=" targetnamespace="http://org.com/" unqualified"="" xmlns:tns="http://org.com/" xmlns:xs="http://www.w3.org/2001/XMLSchema"> <xs:element name="hillul" type="ths:hillul"></xs:element></xs:schema></pre>	lifie	ed"	
<xs:element name="hiMuleResponse" type="tns:hiMuleResponse"></xs:element> <td></td> <td></td> <td></td>			
<pre></pre>			Ξ
 ▼ <xs:complextype name="hiMuleResponse"></xs:complextype>			
<pre>vxs:sequence></pre>			
 <xs:element name="Exception" type="tns:Exception"></xs:element>			
<pre>w <xs:complexippe name="ixception"></xs:complexippe></pre>			
<wsdl:part element="tns:hiMuleResponse" name="parameters"></wsdl:part> 			
<pre>w<usdl:message name="hiMule"> <wsdl:part element="ths:hiMule" name="parameters"></wsdl:part> <td></td><td></td><td></td></usdl:message></pre>			
<pre><!-- word::message name="Exception"--> <!--/word::part--> <!--/word::part element="tim::Exception" name="Exception"--><!--/word::part--> }</pre>			
 ▼ <wsdl:porttype name="HelloService"></wsdl:porttype>			
<pre>v<wsdl:operation name="hiMule"></wsdl:operation></pre>			*

In this chapter, we will cover the following topics:

- Understanding components
- Understanding message sources
- ▶ Using message processors to control the message flow
- Understanding message property scopes

Introduction

A message source is the Endpoint where the Mule inbound elements receive messages. Message sources can be Inbound Endpoints, polls, or the custom message receiver. All of these Endpoints receive messages and depend on their corresponding message processors for further execution. Mule has transformers, filters, components, Routers, and other message-processing elements to be used and nested freely as required. They all implement a common message processor interface and can be used interchangeably.

Understanding components

The **Script** component is used for executing different types of scripts such as Ruby, Java, JavaScript, Python, and Groovy. We execute a script that receives a response from the client before the payload is processed. The Script component also provides the option of integrating custom script into a flow.

Getting ready

When using a Script component, the developer must select a script engine that is compatible with the language used to create the custom script.

The **Java** component is used to create custom Java code that is executed when the component receives a message. The **Java** component (whose icon is shown in the following screenshot) can be used to enhance the functionality. To configure the Java component, import a custom Java class; additionally, you have to configure the **Spring** and **Singleton** objects. The Singleton object's purpose is to control object creation and limiting the number to one, while allowing the flexibility to create more objects if the situation demands it.



The **Python** component (whose icon is shown in the following screenshot) can be used to configure the Python scripting language. You can write a custom script in Python for an application. You can also add a scripting file inside the src/main/resources folder. The Python script is executed at runtime only.



In the **Ruby** component (whose icon is shown in the following screenshot), you have to integrate the custom script language or you can add a script file inside the src/main/ resources folder.



In the **Groovy** component (whose icon is shown in the following screenshot), you have to integrate the custom script language or you can add a script file inside the src/main/resources folder.



In the **Javascript** component (whose icon is shown in the following screenshot), you have to integrate a custom script language in that component or you can add a script file inside the src/main/resources folder. It also allows the developer to configure interceptors and alter the values or references of particular properties in a script. All the scripting components are configured in a similar way.



How to do it...

We will now use the **Groovy** component to demonstrate how to change the filename of an image.

1. Open Mule Studio and enter the name for the workspace. We have to use the **Groovy** component, the **Logger** component, and the **HTTP** Endpoint.

Workspace Launcher
lect a workspace
Iule Studio stores your projects in a folder called a workspace. hoose a workspace folder to use for this session.
orkspace: E:\MuleCookBook Browse
Use this as the default and do not ask again
OK Cancel



2. To create a new project, go to **File** | **New** | **Mule Project**. Enter the project name as Script and click on **Next** and then on **Finish**. Your new project has been created. You can now start implementing it.

Edit Source Refactor Navigate Search Project Run Window Help New Alt+Shift+N Mule Project Open File Mapping Filew Mapping Filew Close Ctrl+W Project Project Save All Ctrl+Shift+S Mapping Flow Save All Ctrl+Shift+S Prolocu Save All Ctrl+Shift+S Prolocu More Prolocu Project Refresh P2 Source Folder Convert Line Delimiters To Sava Warking Set Pinto Ctrl+P
New Alk+Shift+N Row Open File
Open File
Close Ctrl+W I* Project Close All Ctrl+Shift+W
Close All Ctrl+Shift+W
Save Ctrl+S Image Mule Flow Save All Image Package Save All Image Package Save All Image Package Revert Image Package Move Image Package Rename F2 Pander Show Show Show Show Show Show Show Show
Save As Save As Save As Save All Ctrl+Shft+S Ctrl+Shft
Save All Ctrl+Shift+S G Class Revert G Interface Move Rename F2 G Annotation Refresh F5 G Source Folder Convert Line Delimiters To Ctrl+P Ctrl+
Revert If Interface Move If Interface Move If Interface Rename F2 Refresh F5 Convert Line Delimiters To If Israe Vorter Pintu CritteP
Move Rename F2 Refresh F5 Convert Line Delimiters To Pintu Ctri+P Convert Line Delimiters Convert Li
Rename F2 @ Annotation Refresh F5 @ Source Folder Convert Line Delimiters To M Surve Working Set Pinto Chri+P
Refresh F5 Hall Source Folder Convert Line Delimiters To Image: Source Folder Print Carl+P
Convert Line Delimiters To
Print Ctrl+P
File
Switch Workspace
Restart Example
Import
Export Ctrl+N
Export diagram to
Properties Alt+Enter
1 FlowRef.mflow [flowref/flows]
2 package.html [activity/src/main/] 3.0ctivity_melow_factivity/flowe1
4 StrinoToNumber, iava [usdconverter]]
Ena
EXIL Ducking IS THOL BY BIRDING.
¢



3. Go to the file Script.mflow. Drag the HTTP Endpoint from the palette and drop it on the canvas. You will now have to configure the HTTP Endpoint.



Double-click on the HTTP Endpoint to configure it. You will see the Host: and Port: fields. You can change the hostname if you want to. In this example, we have used localhost. If you want to change the port number, you can do that as well. By default, the Mule server takes up the port number 8081. Here we have used port number
 2121. We have used only the one-way exchange pattern, so we send requests from the HTTP component and responses will come from the Script component.

Y Mule - script/flows/Script.mf	Y Endpoint Properties	- 7 🛛
File Edit Navigate Search Project	HTTP (Inbound Endpoint) The HTTP Endpoint allows Mule application to connect to web resources through the HTTP transport protocol. This endpoint can also implement security through HTTPS.	
Package E Package E	General Advanced References HTTP Settings Documentation Display Display Name: HTTP Exchange Patterns one-way request-response Back Settings Enable HTTPS	Fiter: Ju Select Compoints co Compoints co Co Compoints co Co Co Co Co Co Co Co Co Co Co Co Co Co
E Outline 🛛	? Ск. Салсе!	Components
		Flow Control
. D+	Message Flow Global Elements Configuration XML	i # 🖹 @ 😡 VI 📮 🗝

5. Drag the **Logger** component from the palette and drop it on the canvas. Configure the **Logger** component.



6. Double-click on the **Logger** component to configure it. You will see the **Message**: textbox. Enter the payload expression in it. After configuring this, click on the **OK** button and you will see a screen similar to the following screenshot:

Y Mule - script/flows/Script.mf	Y Pattern Properties	
File Edit Navigate Search Project	Logger The Logger Component performs logging using an expression that determines what should be logged. By default the current messages is logged using the DEBUG level.	
📮 Package E 🕺 🍃 Type Hiera	General Documentation	
	Display	Filter:
	Display Name: Logger	Select
🗄 😂 flowref	Generic	Endpoints 🗠
im im im im im im im im im	#[payload]	🧭 Ajax
😟 😂 rest	message:	📑 Database
🖶 🔄 JRE System Library [JavaSt		E FTP
Mule Runtime [Mule Server	Category:	File
- # src/test/java		HTTP
src/main/resources		
E E flows		🗖 JMS
Script.mflow) Jetty
E 🗁 src		🖂 РОРЗ
mule-project.xml		🕓 Quartz
🗉 🧭 suap		E RMI
		SFTP SFTP
		SMTP
		Copes
🗄 Outline 🛛		Components
	(?) OK Cancel	Transformers
∞ → 5 3		Control
·		Error Handling
	Message Flow Global Flements Configuration XMI	Cloud Connectors
: =0	ressage new stabilities contigui adon And	
8 UT		: 🗗 🔣 @ 😣 🖊 🔛 🤜



- Y Mule script/flows/Script.mflow Mule Studio File Edit Navigate Search Project Run Window Help : 🛅 • 🖫 🗟 🗄 🕸 • 🕖 • 🍇 • 🗄 📽 🎯 • 🗄 🅭 🖋 • 🗄 🚥 🗄 🧏 • 🖓 • 🤃 🔶 🐥 • 🖓 • 🐇 🔛 🔛 Mule ቹ Package E 🕺 🍃 Type Hiera 🗖 🗖 🎽 Script 🖄 - -8 ■ 🔄 🏹 Filter: echo
 echo
 flowref
 pic
 rest
 script Ju Select endpoints 皹 Ajax 📑 Database 😑 FTP Script
 Script
 Mole Runtime [Mule Server 3.3.0 CE]
 Scriman/java
 scriman/java
 scriptiva
 scriptiva
 scriptivesources
 Scriptivesources
 Scriptivesources
 Scriptivesources 😑 File Generic HTTP flow: ScriptFlow1 📝 🖂 IMAP Flows 📼 JMS Logger Groovy ≻ ≻ 🔘 Jetty HTTP Logger Groovy 🖂 POP3 🕒 Quartz <u>.</u> RMI 😐 🔛 usdconverter 🔁 SFTP SMTP Copes 2 - -E Outline 🛛 Components a Transformers Filters Flow Control **∞** → 63 → 63 Crror Handling Cloud Connectors Message Flow Global Elements Configuration XML ∎≎ j e 🔝 @ 🕓 M 🗉 🔫
- 7. Drag the **Groovy** component from the palette and drop it on the canvas. Configure the **Groovy** component.

8. Double-click on the **Groovy** component to configure it. Here we have written a simple Groovy script. With that script, we can change the image filename that is located in the location C:/MyPicture. You can also add a script file inside the src/main/ resources folder.

```
printlnfile.getName() + " -> " + f.getName() }
```



How it works...

In this section, you will learn how you can deploy your application using Mule Studio. After deploying this application, you will see how it works.

 If you haven't saved your application code, do save it. To deploy the application code in the Mule server, go to **Run As** | **Mule Application**; the Mule server will deploy your application.

₩ Mule - script/flows/Script.mflow - Mule Studio	
File Edit Source Refactor Navigate Search Project Run Window Help	
i 📬 • 🗐 🕼 🖻 i 🎄 • 🔘 • 🍕 • i 🔐 🎯 • i 🍅 🔗 • i 🔤 i 🖉 • 🖏 •	(-) - (-
	- 6)
	8
	Filter: Ju
activity	Select
i ⊕	Endpoints 🗠
🕀 🔛 pic	🥦 Aja×
B B C rest	S Database
Script New	E FTP
🗑 🛋 Mule Ru Open F3	(E) File
-😕 src/maii Open With	Generic
Show In Alt+Shift+W	() НТТР
src/heet Copy Ctrl+C	IMAP
🖶 🕞 🕞 Copy Qualified Name	
Logger	croovy I Petty
🗁 mappin 💢 Delete Delete Logger	Groovy
Build Path	
Befactor Alt+Shift+T	Quarcz
😟 😂 usdconvert 🛶 Import	RMI
PA Export	E SFTP
	SMTP
^{or} Refresh F5	Copes
Assign working Sets	Components
Validate	Car Transformers
Run As 1 Run on Server Alt+Shift+	X, R
Profile As	Flow Control
Team Run Configurations	Error Handling
Compare With	Cloud Connectors
Replace With al Elements Configuration XML	
Mule Management Console	
CloudHub	: · · · · · · · · · · · · · · · · · · ·

2. If your application code is successfully deployed, you will see the following screenshot as the output on your console:

Y Mule - script/flows/Script.mflow - Mule Studio	×
File Edit Navigate Search Project Run Window Help	
: 11 + 22 弦 合 : * 参 + ② + Q + 1 # G* : # / 参 / / = = : 2 - 2 - 2 - 2 - 2 - 2 - 2 - 2 - 2 - 2	
😰 💟 Mule	
Script [Mule Application] C:\Program Files\Java\jre6\bin\javaw.exe (Dec 5, 2012 12:23:42 PM)	-
connected true	JU
active activ	
Setviceoverrides- <none></none>	
INFO 2012-12-05 12:23:45,656 [main] org.mule.lifecvcle.AbstractLifecvcleManager: Starting connector: connector.http.mule	
INFO 2012-12-05 12:23:45,656 [main] org.mule.lifecycle.AbstractLifecycleManager: Starting model: muleSystemModel	
INFO 2012-12-05 12:23:45,656 [main] org.mule.construct.FlowConstructLifecycleManager: Starting flow: ScriptFlow1	
INFO 2012-12-05 12:23:45,656 [main] org.mule.processor.SedaStageLifecycleManager: Starting service: ScriptFlow1.stage1	
INFO 2012-12-05 12:23:45,656 [main] org.mule.component.ComponentLifecycleManager: Starting component: commponent.288460	
INFO 2012-12-05 12:23:45,656 [main] org.mule.transport.http.HttpConnector: Registering listener: ScriptFlow1 on endpoint	
INFO 2012-12-05 12:23:45,671 [main] org.mule.transport.service.DefaultTransportServiceDescriptor: Loading default respon	
INFO 2012-12-05 12:23:45,671 [main] org.mule.lifecycle.AbstractLifecycleManager: Initialising: 'null'. Object is: HttpMe	
INFO 2012-12-05 12:23:45,687 [main] org.mule.lifecycle.AbstractLifecycleManager: Starting: 'null'. Object is: HttpMessa	
INFO 2012-12-05 12:23:45,687 [main] org.mule.module.management.agent.WrapperManager&gent: This JVM hasn't been launched	
INFO 2012-12-05 12:23:45,703 [main] org.mule.module.management.agent.JmxAgent: Attempting to register service with name:	
INFO 2012-12-05 12:23:45,703 [main] org.mule.module.management.agent.JmxAgent: Registered Endpoint Service with name: Mu	
INFO 2012-12-05 12:23:45,718 [main] org.mule.module.management.agent.JmxAgent: Registered Connector Service with name M	
INFO 2012-12-05 12:23:45,718 [main] org.mule.module.launcher.application.DefaultMuleApplication: Reload interval: 3000	
INFO 2012-12-05 12:23:45,718 [main] org.mule.DefaultMuleContext:	
* Application: script	
• US encoding: Cp1252, Mule encoding: UIF-8	
· · · · · · · · · · · · · · · · · · ·	
* Agents kunning:	
INFO 2012-12-05 12:23:45.718 [main] org.mule.module.launcher.DeploymentService:	
+ Started app 'script' +	
+++++++++++++++++++++++++++++++++++++++	
i D*	-6

-110-

3. Copy the URL http://localhost:2121/ and paste it on your browser.



4. Here you can see the output on the console, and you can see that we have changed the filename using the **Script** component.

🖌 Mule - script/flows/Script.mflow - Mule Studio	J X
File Edit Navigate Search Project Run Window Help	
E 11 ▼ 🖩 🗟 🖹 🅸 ▼ 🚺 ▼ 💁 📽 🥙 ▼ 🗄 🥙 🖋 👘 🗐 ½ ▼ 🖓 − 🤅 🔤 🖓 🖓 ↓ 📖 🗄 ½ ▼ 🖓 → 🖓 ⇔ ↔ → 👘 🖉 ♥ 💥	
🟥 🔛 Mule	
🖻 Console 🛛 🔰 🖷 🦓 🕅 🚛 🚛 🥮 🛃 🖅 📬 🖓 🖓	p
Script [Mule Application] C:\Program Files\Java\jre6\bin\javaw.exe (Dec 5, 2012 12:33:24 PM)	
INFO 2012-12-05 12:33:27,750 [main] org.mule.processor.SedaStageLifecycleManager: Starting service: ScriptFlow1.stage1 🔮	Ju
INFO 2012-12-05 12:33:27,765 [main] org.mule.component.ComponentLifecycleManager: Starting component: commponent.2900450	
INFO 2012-12-05 12:33:27,765 [main] org.mule.transport.http.HttpConnector: Registering listener: ScriptFlow1 on endpoint	
INFO 2012-12-05 12:33:27,765 [main] org.mule.transport.service.DefaultTransportServiceDescriptor: Loading default responses to the service of	
INFO 2012-12-05 12:33:27,781 [main] org.mule.lifecycle.AbstractLifecycleManager: Initialising: 'null'. Object is: HttpMe	
INFO 2012-12-05 12:33:27,796 [main] org.mule.lifecycle.AbstractLifecycleManager: Starting: 'null'. Object is: HttpMessag	
INFO 2012-12-05 12:33:27,796 [main] org.mule.module.management.agent.WrapperManagerAgent: This JVM hasn't been launched	
INFO 2012-12-05 12:33:27,828 [main] org.mule.module.management.agent.JmxAgent: Attempting to register service with name:	
INFO 2012-12-05 12:33:27,828 [main] org.mule.module.management.agent.JmxAgent: Registered Endpoint Service with name: M	
INFO 2012-12-05 12:33:27,828 [main] org.mule.module.management.agent.JmxAgent: Registered Connector Service with name M	
INFO 2012-12-05 12:33:27,828 [main] org.mule.module.launcher.application.DefaultMuleApplication: Reload interval: 3000	
INFO 2012-12-05 12:33:27,828 [main] org.mule.DefaultMuleContext:	
* Application: Script	
- Us encoding: Cp1252, Mule encoding: Ulr-6	
· · · ·	
* Agents Kunning: "	
INFO 2012-12-05 12:33:27 828 [main] or mule module launcher DeploymentService.	
+ Started app 'script' +	
INFO 2012-12-05 12:33:33,437 [[script].ScriptFlow1.stage1.02] org.mule.api.processor.LoggerMessageProcessor: /	
INFO 2012-12-05 12:33:33,437 [[script].ScriptFlow1.stage1.03] org.mule.api.processor.LoggerMessageProcessor: /favicon.ic	
1jpg -> 1.infocom.jpg	
2jpg -> 2.infocom.jpg	
Thumbs.db -> Thumbs.db	4
1jpg -> 1.infocom.jpg	
2jpg -> 2.infocom.jpg	
Thumbs.db -> Thumbs.db	
	4
D°	

Understanding message sources

A message source generally receives or generates new messages to be processed by Mule. Once a message has been received from a message source, it is processed by Mule using one or more message processors.

Getting ready

We can use message processors in the flow in two ways: the one-way exchange pattern and the request-response pattern. Setting the exchange pattern of a message source to one-way enables asynchronous processing of a flow, while setting the exchange pattern of a message source to request-response enables synchronous processing of a flow.



How to do it...

As mentioned in the introduction to this chapter, a message source is the Endpoint where Mule inbound elements receive messages. Message sources can be Inbound Endpoints, polls, or the custom message receiver. All of these Endpoints receive messages and depend on their corresponding message processors for further execution. Mule supports the following types of message sources:

- Inbound Endpoints
- Polls
- Custom message sources

Inbound Endpoints

Inbound Endpoints receive new messages from channels. The following is an example of a code snippet that configures an Inbound Endpoint for a flow:

```
<flow name="MessageSources">
<http:inbound-endpoint address="http://localhost:8080/endpoint"
exchange-pattern="one-way"/>
<jms:outbound-endpoint queue="messages"/>
</flow>
```

This flow indicates how to asynchronously bridge an HTTP request to the JMS.

Polls

Instead of using an Inbound Endpoint, you can poll any message processor and use the result as the source of your flow. Regularity can be configured with milliseconds as the unit or else the default of one second can be used. To arrange polling, use the <poll> section. The following is an example of a code snippet that configures a poll for a flow:

```
<flow name="PollExample">
<poll frequency="500">
<http:outbound-endpoint host="localhost"port="4343"/>
</poll>
<processor ref=""/>
<processor ref=""/>
</flow>
```

113

Custom message sources

Custom message sources are used to restore any Inbound Endpoint in a flow. You organize the custom message source using the <custom-source> element. You can further organize the routine message source using Spring bean properties. The following is an example of a code snippet that configures a custom message source for a flow:

```
<flow name="MyCustomMessage">
<custom-source class="com.org.cookbook.Message">
<spring:property name="threads" value="100"/>
</custom-source>
<vm:outbound-endpoint path="output" exchange-pattern="one-way"/>
</flow>
```

How it works...

You can use message processors in a flow. A message source receives or generates new messages to be processed by Mule.

Using message processors to control the message flow

A message processor is the basic building block of all elements in Mule. These blocks can be glued together to create Mule flows. The message processor is a necessary building block for any project in Mule. You will often need to perform some business logic as part of your flow.

Getting ready

Let's see an example of message processors. In this example, we will see how we can send and receive messages within a flow.

1. We use two components: **STDIO** and **Java**. Open Mule Studio and enter the name of the workspace as shown in the following screenshot:

¥ Workspace Launcher	
Select a workspace	
Mule Studio stores your projects in a folder called a workspace. Choose a workspace folder to use for this session.	
Workspace: E:\/MuleCookBook	Browse
Use this as the default and do not ask again	
	OK Cancel



2. To create a new project, go to **File** | **New** | **Mule Project**. Enter the project name as StudioConnector and click on **Next** and then on **Finish**. Your new project has been created now. You are now ready to start the implementation.

🍟 Mule - Mule Studio				
File Edit Source Refactor Navig	ate Search Project	Run Window Help		
New	Alt+Shift+N	🕨 😋 Mule Project		
Open File		🏄 Java Project		
Close	Ctrl+W	Project		c
Close All	Ctrl+Shift+W	📌 Mapping Flow		
🕌 Save	Ctrl+S	Mule Flow		
Save As		🖶 Package		
in Save All	Ctrl+Shift+S	🞯 Class		
Revert		🗊 Interface		
Move		🕞 Enum		
Rename	F2	@ Annotation		
🔊 Refresh	F5	Bource Folder		
Convert Line Delimiters To		Folder		
🚔 Print	Ctrl+P	File		
Switch Workspace		Untitled Text File		
Restart			_	
Import		Example	_	
Z Export		📸 Other Ctrl+	N	
📧 Export diagram to		-		
Properties	Alt+Enter	-		
1 FlowRef.mflow [flowref/flows]				
2 package.html [activity/src/main/	(]			
3 Activity.mflow [activity/flows]				
4 StringToNumber.java [usdconve	erter/]	_		
Exit				
Arroddine is not available.		_		
: 0				- 🖉 🖷 🕅 🗤

How to do it...

Mule supports components implemented in Java using scripting languages. Message processors are used within flows to control how messages are sent and received within that flow. In this section, we will learn how to use the STDIO connector and how the message processer works.

1. To create a class, go to the folder src/main/java and right-click on it. Go to New | Class. Create a class named MuleCookBook under the package com.org.cookbook. Here, we have created the Welcome method and its return type is set to String.

```
public String Welcome(String name)
{
    return "Hello "+name;
}
```





The STDIO connector allows reading and writing of streaming data to Java's System. out and System.in objects for debugging:

2. Go to the StudioConnector.mflow file and click on the **Configuration XML** tab. Here you create the custom stdio connector tag. First, insert the namespace and the schemaLocation attribute for the STDIO connector; schemaLocation is used in the configuration file.

```
xmlns:stdio=http://www.mulesoft.org/schema/mule/stdio
xsi:schemaLocation="
    http://www.mulesoft.org/schema/mule/stdiohttp://www.mulesoft.org/schema/mule/stdiohttp://www.mulesoft.org/schema/mule/stdiohttp://www.mulesoft.org/schema/mule/stdiohttp://www.mulesoft.org/schema/mule/stdiohttp://www.mulesoft.org/schema/mule/stdiohttp://www.mulesoft.org/schema/mule/stdiohttp://www.mulesoft.org/schema/mule/stdiohttp://www.mulesoft.org/schema/mule/stdiohttp://www.mulesoft.org/schema/mule/stdiohttp://www.mulesoft.org/schema/mule/stdiohttp://www.mulesoft.org/schema/mule/stdiohttp://www.mulesoft.org/schema/mule/stdiohttp://www.mulesoft.org/schema/mule/stdiohttp://www.mulesoft.org/schema/mule/stdiohttp://www.mulesoft.org/schema/mule/stdiohttp://www.mulesoft.org/schema/mule/stdiohttp://www.mulesoft.org/schema/mule/stdiohttp://www.mulesoft.org/schema/mule/stdiohttp://www.mulesoft.org/schema/mule/stdiohttp://www.mulesoft.org/schema/mule/stdiohttp://www.mulesoft.org/schema/mule/stdiohttp://www.mulesoft.org/schema/mule/stdiohttp://www.mulesoft.org/schema/mulesoft.org/schema/mule/stdiohttp://www.mulesoft.org/schema/mulesoft.org/schema/mule/stdiohttp://www.mulesoft.org/schema/mule/stdiohttp://www.mulesoft.org/schema/mule/stdiohttp://www.mulesoft.org/schema/mule/stdiohttp://www.mulesoft.org/schema/mulesoft.org/schema/mule/stdiohttp://www.mulesoft.org/schema/mulesoft.org/schema/mulesoft.org/schema/mulesoft.org/schema/mulesoft.org/schema/mulesoft.org/schema/mulesoft.org/schema/mulesoft.org/schema/mulesoft.org/schema/mulesoft.org/schema/mulesoft.org/schema/mulesoft.org/schema/mulesoft.org/schema/mulesoft.org/schema/mulesoft.org/schema/mulesoft.org/schema/mulesoft.org/schema/mulesoft.org/schema/mulesoft.org/schema/mulesoft.org/schema/mulesoft.org/schema/mulesoft.org/schema/mulesoft.org/schema/mulesoft.org/schema/mulesoft.org/schema/mulesoft.org/schema/mulesoft.org/schema/mulesoft.org/schema/mulesoft.org/schema/mulesoft.org/schema/mulesoft.org/schema/mulesoft.org/schema/mulesoft.org/schema/mulesoft.org/schema/mulesoft.org/schema/mu
```

http://www.mulesoft.org/schema/mule/stdiohttp://www.mulesoft. org/schema/mule/stdio/current/mule-stdio.xsd

You have to create the <stdio:connector> tag; inside the configuration file, you have to use three parameters, name, messageDelayTime, and promptMessage. This connector is configured globally, which means you can use this connector in the flow.

```
<stdio:connector name="stdio" messageDelayTime="1000" promptMessage="Enter Name :" doc:name="STDIO"/>
```



3. Click on the **Global Elements** tab; you can see that the STDIO connector has been created. This will be used in the flow.



4. Here, we have used two STDIO connectors <stdio:inbound-endpoint> and <stdio:outbound-endpoint>, one is for input and the other is for output. We used the connector-ref parameter. Assign the name of the global STDIO connector. Between the two STDIO connectors, we used the Java component tag to import the Java class:

```
<flow name="stdio_component" doc:name="stdio_component">
<stdio:inbound-endpoint system="IN" connector-ref="stdio"
doc:name="STDIO"/>
<component class="com.org.cookbook.MuleCookBook" doc:name="Java"/>
<stdio:outbound-endpoint system="OUT" connector-ref="stdio"
doc:name="STDIO"/>
</flow>
```

-118

The following screenshot shows the entire config.xml file. In this XML file, we can configure the global STDIO connector.



The full configuration file will look like the following code:

<?xml version="1.0" encoding="UTF-8"?>

```
<mulexmlns="http://www.mulesoft.org/schema/mule/core"
xmlns:stdio="http://www.mulesoft.org/schema/mule/stdio"
xmlns:doc="http://www.mulesoft.org/schema/mule/documentation"
xmlns:spring="http://www.springframework.org/schema/beans"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
version="CE-3.3.0"
xsi:schemaLocation="
http://www.mulesoft.org/schema/mule/stdio http://www.mulesoft.org/
schema/mule/stdio/current/mule-stdio.xsd
http://www.springframework.org/schema/beans http://www.
springframework.org/schema/beans http://www.</pre>
```

```
http://www.mulesoft.org/schema/mule/core http://www.mulesoft.
org/schema/mule/core/current/mule.xsd ">
<stdio:connector name="stdio" messageDelayTime="1000"
promptMessage="Enter Name :" doc:name="STDIO"/>
<flow name="stdio_component" doc:name="stdio_component">
<stdio:inbound-endpoint doc:name="STDIO"/>
<stdio:inbound-endpoint system="IN" connector-ref="stdio"
doc:name="STDIO"/>
<component class="com.org.cookbook.MuleCookBook" doc:name="Java"/>
<stdio:outbound-endpoint system="OUT" connector-ref="stdio"
doc:name="STDIO"/>
</flow>
</mule>
```

Your graphical flow will look like the following screenshot:



-120

How it works...

To deploy the application, right-click on the .mflow file and deploy your Mule application. After deploying the application, you will see how to run that application on the console.

1. To deploy the application code in the Mule server, go to **Run As** | **Mule Application**; the Mule server will deploy the application.



2. Enter MuleCookBook on the console; you will see the output on the console. So, in this way, we can send and receive messages with the flow through the STDIO connector.

Mule - studioconnector/flows/StudioConnector.mflow - Mule Studio	ЪХ
File Edit Navigate Search Project Run Window Help	
┆╒╬╺╘╗╘╬╸Ѻ╸Q₂╸┆╋╺╔╸╡╔╝╱╴┆┉┊╢╴╢╺┶┍╸┽┆╝╚╲╨	
E Mule	
	F)
StudioConnector [Mule Application] C:\Program Files\Java\re6\bin\javaw.exe (Dec 18, 2012 12:47:41 PM)	- 8
	- Ju
INFO 2012-12-18 12:47:44,031 [main] org.mule.lifecycle.AbstractLifecycleManager: Starting connector: stdio	
INFO 2012-12-18 12:47:44,046 [main] org.mule.lifecycle.AbstractLifecycleManager: Starting model: _muleSystemModel	
INFO 2012-12-18 12:47:44,046 [main] org.mule.construct.FlowConstructLifecycleManager: Starting flow: stdio_component	
INFO 2012-12-18 12:47:44,046 [main] org.mule.processor.SedaStageLifecycleManager: Starting service: stdio_component.stag	
INFO 2012-12-18 12:47:44,046 [main] org.mule.component.ComponentLifecycleManager: Starting component: commponent.274498"	
INFO 2012-12-18 12:47:44,078 [main] org.mule.transport.stdio.PromptStdioConnector: Registering listener: stdio_component	
INFO 2012-12-18 12:47:44,109 [main] org.mule.lifecycle.AbstractLifecycleManager: Initialising: 'null'. Object is: Stdiof	
INFO 2012-12-18 12:47:44,109 [main] org.mule.lifecycle.AbstractLifecycleManager: Starting: 'null'. Object is: StdioMesse	
INFO 2012-12-18 12:47:44,125 [main] org.mule.module.management.agent.WrapperManagerAgent: This JVM hasn't been launched	
INFO 2012-12-18 12:47:44,156 [main] org.mule.module.management.agent.JmxAgent: Attempting to register service with name:	
INFO 2012-12-18 12:47:44,171 [main] org.mule.module.management.agent.JmxAgent: Registered Endpoint Service with name: M	
INFO 2012-12-18 12:47:44,171 [main] org.mule.module.management.agent.imxAgent: Registered Connector Service with name M	
INFO 2012-12-18 12:47:44,171 [main] org.mule.module.launcher.appication.beraulthuleAppication: Reload interval: 3000	
INFO 2012-12-10 12:47:44,171 [main] org.mule.beraliteleontext:	
* hunlightion: studiogonmentor *	
A Generative Schulzes meeding: HTE-8	
* *	
* Agents Running: *	
* JN Agent *	

INFO 2012-12-18 12:47:44,171 [main] org.mule.module.launcher.DeploymentService:	
+ Started app 'studioconnector' +	
++++++	
Enter Name :Hule CookBook	2
INFO 2012-12-18 12:48:02,703 [[studioconnector].stdio.dispatcher.01] org.mule.lifecycle.MstractLifecycleManager: Initic	
INFO 2012-12-18 12:48:02,703 [[studioconnector].stdio.dispatcher.01] org.mule.lifecycle.AbstractLifecycleManager: Start	
Hello Mule CookBook	
Enter Name :	~

Understanding message property scopes

The Mule Studio message property transformer has been deprecated and replaced by four new transformers grouped under the title Message and Variable transformers. There are four types of scopes in the message properties, as follows:

 Inbound: Inbound properties are those that are received in an Inbound Endpoint or as the response of a call to an Outbound Endpoint. For instance, when an HTTP message is received, each HTTP header will be placed as a Mule message inbound property.

123

- Outbound: Outbound properties are the ones that will be part of the outgoing messages. For instance, if a message with an outbound property content type is sent through HTTP, the content-type property will be placed as an HTTP header in the outbound message.
- Invocation: Invocation is used mostly internally by Mule for the duration of this service's call, not typically utilized, nor meant for the end user.
- Session: Session values are passed from invocation to invocation.
- Application: This scope is used when you create two different applications.

Use invocation properties if you need to set a property that you want to then use in the same flow. A typical example of where you will use an invocation property is when you wish to make a variable available for use during a flow. Typically, the invocation property is created for a value that is to be re-used in multiple places in a flow.

Getting ready

In this section, we will see how to create a custom component. We have to use one HTTP Endpoint and one Java component.

🗑 Workspace Launcher
Select a workspace
Mule Studio stores your projects in a folder called a workspace. Choose a workspace folder to use for this session.
Workspace: E:\MuleStudio Browse
Use this as the default and do not ask again
OK Concel

1. Open Mule Studio. Enter a name for the workspace.

2. To create a new project, go to **File** | **New** | **Mule Project**. Enter the project name as HelloWorld and click on **Next** and then on **Finish**. Your new project has been created now. We are ready to start the implementation.



How to do it...

In this section, you will see how to create a custom component using business logic.

 To create a class, go to src/main/java and right-click on it. Go to New | Class. Create a class under the package com.org and name it Greeting; here, we have created the sayHi() method and its return type is set to String.

```
public String sayHi(String str)
{
    return "Hello "+str;
}
```

-124

You can see the creation of this method in the following screenshot:



2. To create a flow, go to the Greeting.mflow file. In the following screenshot, the central part is called a **canvas**, which has been explained in previous chapters, where we insert graphical elements. On the right-hand side, you can see a group of elements, which is called **palette**. First of all, you have to drag the **HTTP** Endpoint from the palette and drop it on the canvas area.





3. Double-click on the **HTTP** Endpoint to configure it. You will see the hostname. Here, the **Host:** and **Port:** fields are mandatory. You have to configure both the attributes.

🛛 Mule - helloworld/flows/Hell	¥ Endpoint Properties		
File Edit Navigate Search Project	HTTP (Inbound Endpoint)		
i 📬 • 🔚 🗟 🗄 🔅 🍫 • 💽	The HTTP Endpoint allows Mule application to connect to web resources through the HTTP transport protocol. This applications are implement and with through HTTPS		
🖹 🔛 Mule	mis endpoint can also implement security through mines.		
洋 Package E 🛛 🍃 Type Hiera	General Advanced References HTTP Settings Documentation		
	Display		Filter:
🖻 😂 helloworld	Display Name: HTTP		Select
Intersection Contraction (Javas) Intersection Contraction (Javas) Intersection Contraction (Javas) Intersection Contraction (Javas)	Exchange Patterns		C Endneinte
🗉 进 src/main/java	🔘 one-way 💿 request-response		
com.org	7 Basic Settings		Database
- 🗁 src/test/java	Enable HTTPS 0		E
- C src/main/resources	Enabling HTTPS will require configuring a HTTPS Connector	_	File
□ Flows	Host: localhost	_	Generic
HelloWorld.mflow	Port: 8081		HTTP
mappings frc	Path:		IMAP
mule-project.xml			📥 JMS
) Jetty
			DOP3
			🕒 Quartz
			RMI
			🔁 SFTP
			SMTP-
			Scopes
🗄 Outline 🛛			Components
	? ОК Са	ncel	Elterr
Name of Street S			Flow Control
100			Error Handling
2			Cloud Connectors
	Message Flow Global Elements Configuration XML		
i □ *			j a 🔝 @ 🔂 VI

4. Drag the **Java** component from the palette and drop it on the canvas area. We will store the custom Java class in the **Java** component. Configure the **Java** component. In this example, we have configured the Greeting class that we had created earlier.







5. Double-click on the **Java** component to configure it and then configure the Greeting class.

How it works...

To deploy the application, right-click on the .mflow file and deploy the Mule application.

1. To deploy the application code in the Mule server, go to **Run As** | **Mule Application**; the Mule server will deploy your application.



2. If your application code is successfully deployed, you will see the output screen as shown in the following screenshot:

V Mula ballaure Halla World or flour, Mula Studio
International and the second model with the second se
[1] [1] · [1] (1] (1) · (2)
🗈 💟 Mule
🕒 console 🕱 🛛 🗧 🗮 🎆 🕞 🖉 🛃 🖓 🖬 🖉 🖉
HelloWorld [Mule Application] C:\Program Files\Java\re6\bin\javaw.exe (Nov 20, 2012 3:22:30 PM)
connected=true
supportedProtocols=[http]
serviceOverrides= <none></none>
INFO 2012-11-20 15:22:34,375 [main] org.mule.lifecycle.AbstractLifecycleManager: Starting connector: connector.http.mule.de
INFO 2012-11-20 15:22:34,375 [main] org.mule.lifecycle.AbstractLifecycleManager: Starting model: _muleSystemModel
INFO 2012-11-20 15:22:34,390 [main] org.mule.construct.FlowConstructLifecycleManager: Starting flow: HelloWorldFlow1
INFO 2012-11-20 15:22:34,390 [main] org.mule.processor.SedaStageLifecycleManager: Starting service: HelloWorldFlow1.stage1
INFO 2012-11-20 15:22:34,390 [main] org.mule.component.ComponentLifecycleManager: Starting component: commponent.27970580
INFO 2012-11-20 15:22:34,390 [main] org.mule.transport.http.HttpConnector: Registering listener: HelloWorldFlow1 on endpoin
INFO 2012-11-20 15:22:34,406 [main] org.mule.transport.service.DefaultTransportServiceDescriptor: Loading default response
INFO 2012-11-20 15:22:34,406 [main] org.mule.lifecycle.AbstractLifecycleManager: Initialising: 'null'. Object is: HttpMessa
INFO 2012-11-20 15:22:34,421 [main] org.mule.lifecycle.AbstractLifecycleManager: Starting: 'null'. Object is: HttpMessageRe
INFO 2012-11-20 15:22:34,421 [main] org.mule.module.management.agent.WrapperManagerAgent: This JVM hasn't been launched by
INFO 2012-11-20 15:22:34,453 [main] org.mule.module.management.agent.JmxAgent: Attempting to register service with name: Mu
INFO 2012-11-20 15:22:34,453 [main] org.mule.module.management.agent.Jmxlgent: Registered Endpoint Service with name: Mule.
INFO 2012-11-20 15:22:34,453 [main] org.mule.module.management.agent.JmxAgent: Registered Connector Service with name Mule.
INFO 2012-11-20 15:22:34,468 [main] org.mule.module.launcher.application.DefaultMuleApplication: Reload interval: 3000
INFO 2012-11-20 15:22:34,468 [main] org.mule.DefaultMuleContext:

* Application: helloworld *
* OS encoding: Cp1252, Mule encoding: UTF-8 *
* *
* Agents Running: *
* JHX Agent *
INFO 2012-11-20 15:22:34,455 [main] org.mule.module.launcher.DeploymentService:
+ statica app inclineering +
*
Using Message Property, Processors, and Sources -

3. Copy the URL http://localhost:8081 and paste it on your browser.

🖌 Mule - helloworld/flows/HelloWorld.mflow - Mule Studio 💦 🗐 🔀
File Edit Navigate Search Project Run Window Help
╡▆┓╸▐ ▝▓ ▪ Ø ▪ Q₂ ▪ ╡ #° Ø ▪ ╡ Ø - ⋞ ⊨ ╡ Ø - ∜ ← ♀ → ↓ ↓ ♡ ♡ X
E Mule
🕒 Console 🕴 📑 🙀 📓 🖉 🖉 ピ 🖓 ピ 🖓 😁 😁
HelloWorld [Mule Application] C:\Program Files\Java\jre6\bin\javaw.exe (Nov 20, 2012 2:46:17 PM)
in] org.mule.lifecycle.AbstractLifecycleManager: Starting connector: connector.http.mule.default
in] org.mule.lifecycle.AbstractLifecycleManager: Starting model: _muleSystemModel
in] org.mule.construct.FlowConstructLifecycleManager: Starting flow: HelloWorldFlow1
in] org.mule.processor.SedaStageLifecycleManager: Starting service: HelloWorldFlow1.stage1
in] org.mule.component.ComponentLifecycleManager: Starting component: commponent.3946484
in] org.mule.transport.http.HttpConnector: Registering listener: HelloWorldFlow1 on endpointUri: http://localhost:8081
in] org.mule.transport.service.DefaultTransportServiceDescriptor: Loading default response transformer: org.mule.transport.
in] org.mule.lifecycle.AbstractLifecycleManager: Initialising: 'null'. Object is: HttpMessageReceiver
in] org.mule.lifecycle.AbstractLifecycleManager: Starting: 'null'. Object is: HttpMessageReceiver
in] org.mule.module.management.agent.WrapperManagerAgent: This JVM hasn't been launched by the wrapper, the agent will not
in] org.mule.module.management.agent.JmxAgent: Attempting to register service with name: Mule.helloworld:type=Endpoint,serv
in] org.mule.module.management.agent.JmxAgent: Registered Endpoint Service with name: Mule.helloworld:type=Endpoint,service
in] org.mule.module.management.agent.JmxAgent: Registered Connector Service with name Mule.helloworld:type=Connector,name="
in] org.mule.module.launcher.application.DefaultMuleApplication: Reload interval: 3000
in] org.mule.DefaultMuleContext:

*
ing: UTF-8 *
*
*
*

in] org.mule.module.launcher.DeploymentService:

4. After pasting the URL in your browser, type in /Mule (it is the string type in your browser) and you will see the output. In this example, the message hits the component as it passes through the Inbound Endpoint. Mule properties are handled by Mule and move between scopes either implicitly or explicitly.

C localhost:8081/Mule ×	
← → C Socahost:8081/Mule	☆ 📴 🔧
Hello /Mule	
🛐 start 🔰 🦻 localhost:6061 🎓 Downloads 🔯 Deploying App 🦞 Mule - hellowor 🔯 mulefirstchapte 🦉 untitled - Paint 🤻) 🔽 🗞 💕 🛛 2:49 PM

4 Endpoints

In this chapter, we will cover the different types of endpoints. You will learn the following:

- ► Configuring the Generic Endpoint
- Configuring the HTTP Endpoint
- Configuring the IMAP Endpoint to retrieve e-mails
- Using the JDBC Endpoint to connect to the database
- Implementing the File Transport channel using the File Endpoint
- ▶ Sending messages asynchronously using the AJAX Endpoint
- ▶ Using the Servlet Endpoint to listen to events or messages from servlet requests

Introduction

An Endpoint is used for sending and receiving messages through a service. Endpoints can be Inbound or Outbound. An Inbound Endpoint receives messages via its associated transport. Each transport implements its own Inbound Endpoint element. An Outbound Endpoint sends messages via its associated transport. Each transport implements its own Outbound Endpoint element. Different types of Endpoints are available in Mule, such as HTTP, JMS, IMAP, SMTP, and AJAX. We will see how to configure the Endpoint.

Configuring the Generic Endpoint

The Generic Endpoint is a string representation of the information you use to configure your Endpoint. This Endpoint is configured by the path specified in the **Address:** field.

Getting ready

The Generic Endpoint employs two types of exchange pattern, one-way and request-response. The Generic Endpoint uses the one-way pattern by default. The VM Endpoint can be created from a Generic Endpoint by specifying the VM transport in this field, for example, vm://foo. Similarly, an FTP Endpoint can be created with the Generic Endpoint component by specifying the FTP transport string in the component's **Address:** field under **Transport Settings**; for example: ftp://user:password@host/directory?connector-ref=myFtpConnector&binary=true.

How to do it...

After dragging the **Generic** Endpoint from the palette and dropping it onto the canvas, double-click on the Endpoint icon. This invokes the **Endpoint Properties** window for configuration of this Endpoint. The **Address:** field in this window is where you set up the path for the **Generic** Endpoint, as shown in the following screenshot:

🛛 Mule - helloworld/flows/Hello	Y Endpoint Properties	🗙 🗖 🔁 🔀
File Edit Navigate Search Project	Generic (Dutbound Endpoint)	
i 📬 • 🔛 🕼 🗁 i 🏇 • 💽 •	Normal Action Control Con	
🖹 🕎 Mule	· · · · ·	
洋 Package E 🛛 🍃 Type Hiera	General Advanced References Documentation	
E	✓ Display	Filter:
🗉 😂 database	Display Name: Generic	JU
Gropboxdemo	Exchange Patterns	L2 Select
Ellowref	one-way Orequest-response	Endpoints 🗠
😑 🔛 helloworld		🦉 Ajax
🗄 🛋 JRE System Library [JavaSE	Transport Settings	Database
Hule Rundme [Mule Server	Address:	FTP
🥮 src/test/java		File
- C src/main/resources		Generic
src(test)resources		() HTTP
HelloWorld.mflow		IMAP
C mappings		Scopes 🗠
mule-project.xml		Async
🗉 😂 http-security_filter		😞 Composite Source
🖲 🔛 jdbc		Flow
🖮 🗁 jms 🖮 🔛 mule-activiti		Q Foreach
🗉 🔛 pic		Message Enricher
🗈 😂 rest		C Poll
		Sub flow
E Outline 🛛		Components
		Carl Transformers
		Filters
₩→₩→₩		Control
		Carl Error Handling
		Cloud Connectors
	Message How Global Liements Configuration XML	
÷ 📭		e 🛃 @ 😣 M 🗝 🛛 e 🖻
🛃 start 🔊 Mule	🔏 Tomcat 🔍 Mule 🏷 Mule 🖏 659 🖏 Chap 🏷 Gene 🏷 Mule 🕝	Micro

134

How it works...

The Endpoint will be defined based on Mule expressions or a specific address. For instance, if the address includes HTTP at the beginning, you are configuring an HTTP Endpoint. If the address includes File, you are configuring a File Endpoint.

Configuring the HTTP Endpoint

Mule uses HTTP Endpoints to send and receive requests over the HTTP transport protocol. Configured as either Inbound (also known as message sources) or Outbound, HTTP Endpoints use one of the two message exchange patterns: request-response or one-way. We will see an example of how it works.



The arrows indicate the request-response type of exchange pattern.



The arrow indicates the one-way type of exchange pattern.

Getting ready

We have to use two components: the HTTP Endpoint and the Java component.

1. Open Mule Studio and enter the workspace name as shown in the following screenshot:

¥ Workspace Launcher	
Select a workspace	
Mule Studio stores your projects in a folder called a workspace. Choose a workspace folder to use for this session.	
Workspace: E:\MuleCookBook	Browse
Use this as the default and do not ask again	
	OK Cancel



2. For creating a new project, go to **File** | **New** | **Mule Project**. Enter the project name called Echo and click on **Next** and then on **Finish**. Your new project is created now, so you can start the implementation.

🖌 Mule - Mule Studio						
File Edit Source Refactor Navig	gate Search Project I	Run Window Help				
New	Alt+Shift+N	🗠 😋 Mule Project		\$~ () -		
Open File		🏄 Java Project				
Close	Ctrl+W	📑 Project				
Close All	Ctrl+Shift+W	Annoing Flow				
U Save	Chrl+S	Mule Flow				
Save As		Package				
Save All	Ctrl+Shift+S	G Class				
Revert		🗊 Interface				
Move		🞯 Enum				
Rename	F2	Annotation				
🔊 Refresh	F5	😂 Source Folder				
Convert Line Delimiters To	,	🖌 b Java Working Set				
A Print	Ctrl+P	😂 Folder				
		File				
Switch Workspace	,	Untitled Text File				
Restart		Example				
🚵 Import		■ • Other	Christen			
🛃 Export			carri			
🔯 Export diagram to						
Properties	Alt+Enter	_				
1 FlowRef.mflow [flowref/flows]						
2 package.html [activity/src/main	/]					
3 Activity.mflow [activity/flows]						
4 StringToNumber.java [usdconv	erter/]					
Exit						
All outline is not available.		_				
□*						j a 🚼 @ 😡 M 🚍
🐉 start 🔰 💿 Echo Exam	ple - M 🔯 Chapt	er 2 - Micros 🔯 6	59_The Author a	W untitled - Paint	Mule - Mule Studio	🗌 🔿 🔥 🔁 🥩 🗞 💽 10:59 AM 🔒

-136-

How to do it...

In this section you will see the Hello World example, which is created in Mule Studio. In this example two components are used: one is the HTTP Endpoint and the other is the Java component.

 To create a class, go to src/main/java, right-click on it, and go to New | Class. Create a class called Greeting under the package com.org; here, we have created the sayHi method and its return type is set to String.

```
public String sayHi(String str)
                                               {
                                                         return "Hello"+str;
                                                }
Y Mule - helloworld/src/main/java/com/org/Greeting.java - Mule Studio
                                                                                                                                                                                                                                                                                                                                                                                           File Edit Source Refactor Navigate Search Project Run Window Help
   : 🗂 • 🔛 🕼 👜 :: 🏇 • 🔕 • 🤮 📽 • : : 🎒 🔗 • : : 🖗 🕖 • : : 🍁 🖉 • : : 🌾 🕖
 🔛 🔛 Mule
 📱 Packa 🕴 🁔 Type 📄 🖓 *HelloWorld 🕖 Greeting.java 🖄
                                                                                                                                                                                                                                                                                                                                                                                             - -
                                              E 🔄 ▽ 📕 1 package com.org;
                                                                                                                                                                                                                                                                                                                                                                                                              æ
                                                                                                                                                                                                                                                                                                                                                                                                              Ju
                                                                                     2
3
4
public class Greeting (
   🗄 🔛 database
                                                                           ^

    in the second se
                                                                                          50
                                                                                                                public String sayHi(String str)
   1 flowref
                                                                                          6
    🖹 🔛 helloworld
                                                                                                                {

    JRE System Library [Java
    Java
    Mule Runtime [Mule Serve
                                                                                          8
                                                                                                                             return "Hello "+str;
                                                                                          9
           😑 🇀 src/main/java
                                                                                         10
                 i territariy)are
                                                                                                               }
                                                                                         11 }
12
                 Greeting.java
                # src/main/resources
# src/test/resources
          Flows
             🗁 mappings
    😟 😥 jdbc
    🗄 🛃 jms
   🗄 📂 mule-activiti
   i inir
∢
                                                                           ×
                                                                >
 🗄 Outline 🛛
                          Greeting
                       sayHi(String) : String
    •
                                                                                                  8 Writable
                                                                                                                                                                                                                                                                                                                               e 🖹 @ 😥 M 🖏 🕴 e 🚍
                                                                                                                                                  Smart Insert
                                                                                                                                                                                  1:1
                                                                                                                                                                                                                                                                                                                              🗐 44... 🔇 🥵 🕭 🔽 🗞 5:48 PM
   🏄 start 🔰 👩 az.
                                                                    💰 To
                                                                                                    ₩м..
                                                                                                                           🗀 М.
                                                                                                                                                          @] 65
                                                                                                                                                                                     🗐 Ch.
                                                                                                                                                                                                                                            🗀 Mule
                                                                                                                                                                                                                                                                     🗐 Ch..
                                                                                                                                                                                                                                                                                                🔁 Do
```



2. Go to the HelloWorld.mflow file. Firstly, you have to drag the HTTP Endpoint onto the canvas; to configure it, double-click on the HTTP Endpoint.



3. You will see a similar screen on your system. You have to enter the **Host** and **Port** field values. By default, the port number is **8081**. You can change the hostname and port number, but note that these two fields are mandatory.

Image: Sector Protect HTTP (Inbound Endpoint) The endpoint as do implement security through the HTP transport protocol. Image: Sector Protect Image: Sector Protect Image: Sector Protect Image: S	⊻ Mule - helloworld/flows/Hello	Y Endpoint Properties	3 🔹 🗗 🗙
Image:	File Edit Navigate Search Project	HTTP (Inbound Endpoint)	
It is endpoint can also implement security through HTPS. It stata It is the endpoint can also implement security through HTPS. It stata It is the endpoint can also implement security through HTPS. It stata It is the endpoint can also implement security through HTPS. It is the endpoint can also implement security through HTPS. It is the endpoint can also implement security through HTPS. It is the endpoint can also implement security through HTPS. It is the endpoint can also implement security through HTPS. It is the endpoint can also implement security through HTPS. It is the endpoint can also implement security through HTPS. It is the endpoint can also implement security through HTPS. It is the endpoint can also implement security through HTPS. It is the endpoint can also implement security through HTPS. It is the endpoint can also implement security through HTPS. It is the endpoint can also implement security through HTPS. It is the endpoint can also implement security through HTPS. It is the endpoint can also implement security through HTPS. It is the endpoint can also implement security through HTPS. It is the endpoint can also implement security through HTPS. It is the endpoint can also implement security through HTPS Connector It is the endpoint can also implement security through HTPS Connector <td>i 📬 - 📓 🗟 i 🏇 - 💽 -</td> <td>The HTTP Endpoint allows Mule application to connect to web resources through the HTTP transport protocol.</td> <td></td>	i 📬 - 📓 🗟 i 🏇 - 💽 -	The HTTP Endpoint allows Mule application to connect to web resources through the HTTP transport protocol.	
Packa Type General Advanced References HTTP Settings Documentation Press General Advanced References Press General Advanced References General Advanced References Components General Advanced References Components Co	📫 🔛 Mule	This endpoint can also implement security through HTTP5.	
Image: Section of the sec	😫 Packa 🕺 🍃 Type 📄 🗖	General Advanced References HTTP Settings Documentation	
Image: Second	E 🕏 🏹	Display	Filter:
Big dropboxdeno Exchange Patterns Image: Platterns Image: Platterns Image: Platterns	🗉 🗁 database 🧖	Display Name: HTTP	D. Caluat
Image: Strain of the server Image: Strain of th	Gropboxdemo	Exchange Patterns	12 Delect
Image: Sources Bask: Settings Port: 2121 Path:	B B flowref		Endpoints
Bit Setting Bit Setting </td <td>🖨 😂 helloworld</td> <td></td> <td>Scopes</td>	🖨 😂 helloworld		Scopes
Bit Mule Ruthine [Mule Servi Bit Structure [Mule Servi Bit Structure [Mule Servic Bit Structure	🕀 🛋 JRE System Library [Java	Basic Settings	Components 🗠
Image: Signal state of the	H - Mule Runtime [Mule Serve	Enable HTTPS	[000] Echo
Post: incahoot Post: 2121	- 🗁 src/test/java	Enabling HTTPS will require configuring a HTTPS Connector	Connues Expression
Pott 2121 Pott 2121 Path: Path: Path: Path: Path: Starscript Path: Starscript Path: Path: Path: Starscript Path: Path: Path: Path: <td>- 🕮 src/main/resources</td> <td>Host: localhost</td> <td>Flow Reference</td>	- 🕮 src/main/resources	Host: localhost	Flow Reference
Path: Image: Transformers Image: Transformers Image: Transformers	- C src/test/resources	Port: 0 2121	Groovy
Image project.xml Image project.xml	HelloWorld.mflow	Path:	HTTP Response Builder
Image: section of the section of th	- C mappings		🔬 Java
Image: Security of the security	🕀 🗁 src		Javascript
Image: Second	mule-project.xml		Logger
Image: set in the set in	i idbc		Python
Image: Second	🗈 🔁 jms		REST
Image: Solution of the state of the sta	🗷 🗁 mule-activiti		Ruby
Image: Control in the second secon	E St		SOAP
BE Outline X OK Cancel Image: Control Image: Control Image: Control Im	🕀 🚔 scrint 🗳		Son
Control Concel Conce			Denje
Image: Concest of the second secon	🗄 Outline 🛛 📃		J
		(?) OK Cancel	Transformers
Flow Control	manana		Filters
Message Flow Global Elements Configuration XML			Flow Control
Message Flow Global Elements Configuration XML			Error Handling
		Maccasa Elevi Clobal Elements Configuration VM	Cloud Connectors
	: =¢		
	5 U.		

4. To import a class, drag the **Java** component and configure it. Here we have imported the Greeting.java class that was created before.





5. To configure the **Java** component, double-click and configure it as we have seen earlier. Just click on the Browse button (beside the **Class Name:** field), and a new window will open. Here, you can import the Greeting.java class that was created before and click on the **OK** button.

🛛 Mule - helloworld/flows/Hello	🗑 Pattern Prop	erties						D	<	- 2	×
File Edit Navigate Search Project	Java The Java Compon received.	ent is a simple P	'OJO (Plain Old	d Java Object) (hat will be inv	oked by Mule	when a messag	e is	1		
Mule Packa X Type database dropboxdemo deba dropboxdemo dropbo	General Advanc Display Display Display Name: Generic Class Name: Object:	ed Documenta Java com.org.Greeti	ing				OK	Cancel	Filter: Filter: Select Sopes Components Filter: Sopes Components Filter: Sopes Components Filter: Sopes Components Filter: Sopes Components Filter: Sopes Components Filter: Sopes Sopes Components Filter: Sopes Sope	co	8 Ju
//		t test i l							Cloud Connect	ors	
: =¢	Message Flow G	obal Elements	Configuration	XML	_ ;			1	- • • • •		-
: 0	_	_	1	0	2	1	1 -	1	e 🔝 @ 😡 M 1	-6 : - E	5
🛃 start 🔰 🌀 az 🧉	То 🤡 М	🧀 M	· 🗐 65	🗐 Ch	🜍 нт	🗀 Mule	🔮 Ch	🗁 Do	🖄 44 🛛 🔇 🕵 🧥	🔁 🗞 6:18 PM	

6. To deploy the application code in the Mule Server, go to **Run As** | **Mule Application**, and the Mule Server will deploy your application.





7. If your application code is successfully deployed, you will see the message Started app 'helloworld' on the console.

🖌 Mule - helloworld/flows/HelloWorld.mflow - Mule Studio	
File Edit Navigate Search Project Run Window Help	
I 🗂 • 🔛 🐚 🗁 I 🏇 • O • O • O • I 🕮 O • I 🙋 🖋 • I == I ½ → ½ → ♡ → ♡ → ↓ ♡ ♡ 🕱	
🗄 🔛 Mule	
🖻 Console 🖄 🔰 📓 💭 🚱 🖻 🖻 😁 🗂 🖓	8
Helloworld [Mule Application] C:\Program Files\Java\ire6\bin\javaw.exe (Dec 27, 2012 6:46:06 PM)	
connected=true	<u>∧</u> Ju
supportedProtocols=[http]	
serviceOverrides= <none></none>	
INFO 2012-12-27 10:40:10,570 [main] org.mule.lifeCycle.AsstractLifecycleAnanger: starting connector: connector.http.mule	e .
INFO 2012-12-27 18:46:10.576 [mail] ofg.mule.interviet.Abstractifictyclenamager: Statting moder: Hellowinderia	
INFO 2012-12-27 18:46:10.578 [main] org.mule.processor.SedaStareLifectcleManager: Statting service: HelloWorldFloW1.sta	ιc
INFO 2012-12-27 18:46:10.578 [main] org.mule.component.ComponentLifecvcleManager: Starting component: commponent.11357	5C
INFO 2012-12-27 18:46:10,593 [main] org.mule.transport.http.HttpConnector: Registering listener: HelloWorldFlow1 on end	13
INFO 2012-12-27 18:46:10,593 [main] org.mule.transport.service.DefaultTransportServiceDescriptor: Loading default respo	1
INFO 2012-12-27 18:46:10,593 [main] org.mule.lifecycle.AbstractLifecycleManager: Initialising: 'null'. Object is: HttpM	Ie
INFO 2012-12-27 18:46:10,609 [main] org.mule.lifecycle.AbstractLifecycleManager: Starting: 'null'. Object is: HttpMesse	ις
INFO 2012-12-27 18:46:10,625 [main] org.mule.module.management.agent.WrapperManagerAgent: This JVM hasn't been launched	ι
INFO 2012-12-27 18:46:10,640 [main] org.mule.module.management.agent.JmxAgent: Attempting to register service with name	::
INFO 2012-12-27 18:46:10,640 [main] org.mule.module.management.agent.JmxAgent: Registered Endpoint Service with name: M	it 👘
INFO 2012-12-27 18:46:10,640 [main] org.mule.module.management.agent.JmxAgent: Registered Connector Service with name N	a
INFO 2012-12-27 18:46:10,656 [main] org.mule.module.launcher.application.DefaultMuleApplication: Reload interval: 3000	
INFO 2012-12-27 18:46:10,656 [main] org.mule.peraultaulecontext:	
t Ampligation: balloworld t	
* OS encodinor: Chil250 Mule encodinor: HTF-8 *	
* *	
* Agents Running: *	
* JNX Agent *	
*******	в
INFO 2012-12-27 18:46:10,656 [main] org.mule.module.launcher.DeploymentService:	
+ Started app 'helloworld' +	
***************************************	_
	<u> </u>
	* 🖳
🕂 start 🔰 🔍 U 🧉 To 🔍 M 🗇 M 🗇 65 🗇 Ch 🗇 HT 🎓 Mule 🔍 Ch 🎓 Do 🗐 44 🔇 🕵 🔊 😪	6:46 PM

8. Copy http://localhost:2121 and paste it in your browser to see the output.



How it works...

By putting the URL in the browser, you can see the following output. The word, Hello, is called from the Greeting class through the HTTP request-response exchange pattern. When a request is received, the Java component simply returns whatever was sent as part of the request.





Configuring the IMAP Endpoint to retrieve e-mails

IMAP stands for **Internet Message Access Protocol**. The IMAP/POP3 connector allows you to receive e-mail messages from a mail server using IMAP/POP3. The IMAP Endpoint is configured as Inbound with a one-way exchange pattern. POP3 is similar to IMAP by functionality. **POP3** stands for **Post Office Protocol Version 3**. The POP3 Endpoint can be configured as a one-way exchange pattern.

Getting ready

In this section, you will learn how to configure the namespace and schema location in the Mule configuration XML file. When you are dealing with XML documents in Mule, you need to declare any namespaces used by the document. You can specify a namespace globally so that it can be used by XPath expressions across Mule. You can declare a namespace locally also in filters and routers using the <namespace>element.

The namespace for the IMAP XML namespace is:

xmlns:imap "http://www.mulesoft.org/schema/mule/imap"

The syntax for the IMAP XML namespace is:

xmlns:imaps "http://www.mulesoft.org/schema/mule/imaps"

The IMAP transport can be used for receiving messages from IMAP inboxes using the javax. mail API. When we use IMAP, we will have to use the following schema location:

http://www.mulesoft.org/schema/mule/imap/3.1/mule-imap.xsd

The IMAPS transport uses secure connections over SSL/TLS. When we use IMAPS, we will have to use the following schema location:

http://www.mulesoft.org/schema/mule/imaps/3.1/mule-imaps.xsd

The namespace for the POP3 XML namespace is:

xmlns:pop3 "http://www.mulesoft.org/schema/mule/pop3"

The syntax for the POP3 XML namespace is:

xmlns:pop3s http://www.mulesoft.org/schema/mule/pop3s

The POP3 transport can be used for receiving messages from POP3 inboxes using the following XML schema location:

http://www.mulesoft.org/schema/mule/pop3/3.1/mule-pop3.xsd



The POP3S transport connects to POP3 mailboxes using the javax.mail API using the following XML schema location:

http://www.mulesoft.org/schema/mule/pop3s/3.1/mule-pop3s.xsd

How to do it...

Drag the **IMAP** Endpoint from the palette and drop it on the canvas. Double-click on it and fill the **User:**, **Host:**, **Port:**, and **Password:** fields, which are mandatory. For IMAP, the default port number is **143**. For IMAPS, the default port number is **993**. If you are using IMAP, your hostname will be <code>imap.gmail.com</code>, and if you are going with POP, your hostname will be <code>pop.gmail.com</code>. The default port number is **110** for POP3S, and for POP3 the default port number is **995**.

In the following screenshot, you will see how to configure the IMAP Endpoint:

∑ Mule - imap/flows/IMAP.mflo	Y Endpoint Properties	3 🗖 🗖 🛛
File Edit Navigate Search Project	IMAP (Inbound Endpoint) The IMAP Endpoint allows Mule applications to receive messages from IMAP inboxes using the javax.mail API. This endpoint includes optional security with IMAPS for connections over SSL/TLS.	
Packa 2 Type database dropboxdemo dropbo	General Advanced References Security Documentation Display Name: IMAP IMAP Server Information Host: imap.gmail.com Port: 993 User: azazdesai@gmail.com Password: eeeeeeeee	Fiter: Select Fiter: Dutabase FTP File Generic HTTP IMAP Scopes Composite Source Flow Composite Source Flow Flow File Source Flow File Source Flow Flow File Foreach Message Enricher Poll
E Outline 🛛 🗖 🗖	OK Cancel	Components
	Message Flow Global Elements Configuration XML	Flow Control
 ∃ □ \$		e 🖹 @ 🕓 И 🖏 🕴 e 🗨
🛃 start 💿 Activiti + Mule	ESB – 🧕 Retrieving Email Atta 📓 Chapter 4 - Microsoft 📓 659_The Author and 🦞 Mule - in	hap/flows/IM



How it works...

Internet Message Access Protocol (IMAP) is used for retrieving e-mails on a mail server from multiple computers and devices. For this, you must have configured the **IMAP** Endpoint.

Using the JDBC Endpoint to connect to the database

The JDBC Endpoint is used to communicate with the database. It's used for retrieving, updating, deleting, and inserting database records. We will see an example on how it works. In this example, we will retrieve data from the database and store it in a file on our local hard drive.

Getting ready

In this section, we will see how to use and configure the JDBC component in Mule Studio. In this example you will use three components: the JDBC Endpoint, the Object-to-String transformer, and the File Outbound Endpoint.

1. We will use one JDBC Endpoint, a data source, and the File Outbound Endpoint. Retrieve the record from the MySQL database and store it on the local hard drive.

Y Workspace Launcher	
Select a workspace	
Mule Studio stores your projects in a folder called a workspace. Choose a workspace folder to use for this session.	
Workspace: E:\MuleCookBook	Browse
Use this as the default and do not ask again	
	OK Cancel



2. To create a new project, go to **File** | **New** | **Mule Project**. Enter the project name called Database and click on **Next** and then on **Finish**. Your new project is created. Now you can start the implementation.

🎽 Mule - Mule Studio						- 7 ×
File Edit Source Refactor Naviga	ate Search Project I	Run Window Help				
New	Alt+Shift+N	• 🚰 Mule Project	-			
Open File		🏄 Java Project				
Close	Ctrl+W	📸 Project				- 6
Close All	Ctrl+Shift+W	Mapping Flow				
Save .	Ctrl+S	Mule Flow				
📓 Save As		🖶 Package				
🕼 Save All	Ctrl+Shift+S	🞯 Class				
Revert		🞯 Interface				
Move		ઉ Enum				
Rename	F2	@ Annotation				
🔊 Refresh	F5	Source Folder				
Convert Line Delimiters To	,	🔸 🍄 Java Working Set				
Print	Ctrl+P	Folder				
Switch Workenace		File				
Restart						
N		Example				
Import		T Other	Ctrl+N			
Export						
Export diagram to						
Properties	Alt+Enter					
1 FlowRef.mflow [flowref/flows]						
2 package.html [activity/src/main/]					
3 Activity.mflow [activity/flows]						
4 StringToNumber.java [usdconve	rter/]	_				
Exit						
All outline is not available.		_				
: =0						
: 0*						= 🔣 @ 😣 🖬 📮
🛃 start 💿 Echo Examp	ole - M 🛛 🗐 Chapt	er 2 - Micros 🛛 🗐 659	9_The Author a	🦉 untitled - Paint	Mule - Mule Studio	🗌 🔿 🔼 🚬 🧶 🗞 🔝 10:59 AM

-148-

3. Download the postgresql-9.2-1002.jdbc4.jar file from the source code on the Packt Publishing website. First, you have to add the postgresql JAR file in your class path. Right-click on your project, select **Properties**, go to the Java build path, and click on the **Add External JARs.** button.





4. Select the JAR file, postgresql-9.2-1002.jdbc4, and click on the **OK** button.

How to do it...

In this section we will configure the PostgreSQL database in Mule Studio and learn how to use this JDBC Endpoint in a flow.

1. To create a flow, click on Database.mflow and go to Global Elements | Create | Data Sources | PostgreSQL Data Sources.



2. You have to configure **PostgreSQL Data Sources**. In the URL textbox, enter jdbc:postgresql://localhost:5432/Test as the value. Test is our database, which is created in PostgreSQL. In the end, enter the PostgreSQL user credentials.

⊻ Mule - object-to-xml/flows/0	🖌 Global Elei	ment Properties				X			٥X
File Edit Navigate Search Project	PostgreSQL I Configure Post	Data Source greSQL Data Source							
Package E S Type Her	General Adva General Adva General Name: Connection - URL: User: Password:	nced Documentation PostgreSQL_Data_Source ydbc:postgresgi://localhost:54: postgres	32/Test		OK Cance		*2	Create Edit Delete	
i □ *			6		Ð	8 @	😥 M	e 🗄 i e	P 📮
🐉 start 🛛 🧿 Running Mu	. 📄 Downloa	ds 🔄 440105_04	Y Mule - obje	谢 untitled - Paint	🔄 chapter 4 [b	ellomule	 4: 	37 PM

-152-

🕙 Mule - database/flows/Databa	Y Choose Global Type					_ ð 🗙
File Edit Navigate Search Project : 😭 - 📄 😭 🗁 :: 🏇 - ⊘ - El 🎦 Mule	Choose Global Type Choose the type of global element to	create.				
Packa 2 Type C database database cho cho cho cho cho cho cho cho	Filter: Global Endpoints Global Endpoints Global Endpoints Global Configurations Component configurations Global Configurations Global Endpoints Global Endpoints				Create Edt Delete	
E Outline 🛛 🖓 🖓			0	Cancel		
	Message Flow Global Elements Conl	figuration XML				
: •		-		: 8	🖹 @ 🔯 И 🖏	i a 📮
🛃 start 🚺 🧿 azaz.desai@-	attun 🧊 IMAP Endpoint Ref	Chapter 4 - Micros	659_The Author a	Y Mule - database/fl		3:44 PM

3. Go to **Connectors** | **Database**.

4. To configure the Database connector, select the **Database Specific:** name, **PostgreSQL_Data_Sources**, that was created before and click on **OK**.



5. To create a flow, click on the **Message Flow** tab, drag the **JDBC** Endpoint from the palette, and drop it on the canvas.



6. To configure the JDBC Endpoint, double-click on it. Click on the **Queries** tab and click on the plus icon, as shown in the following screenshot. Now enter the query key, login, and enter the query in the query box; for example, select * from mule. Here, mule is our table name, which is created in PostgreSQL.





7. Click on the **General** tab, select the query key, login, and click on **OK**.

😸 Mule - database/flows/Databa	Y Endpoint Properties	. 3 🗙
File Edit Navigate Search Project	Database (Inbound Endpoint)	
i 📬 • 🔛 💼 🗁 i 🏇 • 🔘 •	The JDBC Endpoint allows Mule applications to read and write to databases which support a JDBC driver.	
😭 🕎 Mule		
洋 Packa 🕺 🍃 Type 🛛 🗖	General Advanced References Queries Documentation	
□ 🛱 🏹	Display	Filter:
🖃 😂 database 🛛 🔺	Display Name: JDBC	Ju
🗷 🛋 JRE System Library [Java		Select
	Query	🚰 Endpoints 🛛 🗠
src/main/java	Query Key: 👸 login 🛛 💆 🖉	🔎 Ajax
- # src/main/resources	Transaction	Database
- 🕮 src/test/resources	Type: No Transaction	ETP
🗄 🛋 Referenced Libraries		
Flows	Action:	File
Database.mflow	Timeout:	Generic
	Interact With External 🟮	() HTTP
mule-project.xml		IMAP
🕀 🔛 dropboxdemo		Scopes 👳
🗈 🔁 echo		Async
		Composite Source
hellowond		
🗄 🔂 imap		Flow
🕀 🔛 jdbc		Q Foreach
🕀 🔁 jms		Message Enricher
🖶 🗁 mule-activiti		O Poll
		Sub flow
🗄 Outline 🛛 🗖 🗖		Components
		Carl Transformers
		🚰 Filters
and a set		Control
		a Error Handling
		Cloud Connectors
	Message Flow Global Elements Configuration XML	
i □ *	😢 i i	e 🔝 @ 😡 M 🗝 🕴 e 📃
🛃 start 🚺 💿 Subram	🥱 IMAP E 💿 Chapte 💿 659_Th 🍟 Mule 🌾 MySQL 📄 data 📑 Dil(test	🔇 🔥 🔍 🖉 🗞 🔟 4:08 PM



8. Drag the **Object to String** transformer from the palette and drop it on the canvas. No need to configure this transformer because it converts the data object to string.

9. Drag the File Outbound Endpoint from the palette and drop it on the canvas.



10. To configure the **File Outbound** Endpoint, double-click on it. Select a system path and output pattern, which is myfile#[function:datestamp].txt. Now, enter the filename and the expression. Through the expression you can display the current date and time, and here the file format is .txt. Now click on the **OK** button.

∑ Mule - database/flows/Databa	🖌 Endpoint Properties 💦 👔	3 🗖 🖉
File Edit Navigate Search Project	File (Outbound Endpoint)	
📫 🗄 🗟 🏇 • 💽 •	The File Endpoint allows Mule applications to read or write files on the local file system.	
🖹 🔛 Mule		
洋 Packa 🕴 🍃 Type 📄 🗖	General Advanced References Documentation	
E 😫 🗆	C Display	Filter:
🖲 🚰 jdbc 🖉	Display Name: File	JU
🗉 🔛 jms	- Path Information	L∂ Select
	Dath: Difestidata	Endpoints 👳
🗊 🔁 rest	Paul Diposipolo	🔎 Ajax
🗉 😂 script	Output Pattern: myfile#[function:datestamp].txt	🕃 Database
🗈 🔛 soap		E FTP
studioconnector		😑 File
😑 🔂 testdb		Generic
🗉 🛋 JRE System Library [Java		() HTTP
H Mule Runtime [Mule Serve		IMAP
- # src/test/java		a JMS
- 🥮 src/main/resources		Tetty
B src/test/resources		
Referenced Libraries Flows		
testdb.mflow		Quartz
- Company mappings		RMI
🕀 🌮 src		Redis (Streaming)
		SFTP
		Scopes
🗄 Outline 🛛 🗖 🗖		Components
		arransformers
		Pilters
ei = ei		Control
		a Error Handling
		Cloud Connectors
	Message Flow Global Elements Configuration XML	
8 ∎*		е 🔝 @ 💁 И 🤜 🕴 е 📃
🛃 start 🛛 👩 aza 🐧) IMA 🔄 Cha 🔯 659 🍟 Mule 🌾 MyS 🗁 data 📑 D:\t 🗁 Dow	🗐 440 🔇 🍋 🛃 🗞 6:31 PM

-160-

11. Now we are ready for the deployment. If you haven't saved your application code, first save it. After saving your project, right-click on the Database.mflow file and go to **Run As** | **Mule Application**.



12. If your application code is successfully deployed, you will see the message Started app 'database' on the console.



How it works...

Once your deployment is done successfully, you can see the log on the console. It will show that the files are stored in the particular drive. When you see the log on the console, you will see that the files are transferred to the destination path.

1. You can see that the database records are stored in a particular file. Here, the files are stored into the D:/.

– Chapter 4

Mule - database/flows/Database.mflow - Mule Studio		
File Edit Navigate Search Project Run Window Help		
: 🗂 • 🔛 🐘 🌰 : 🏘 • 🕖 • 🗛 • 三 🕮 🞯 • 三 🙋 🚀 • 三 💷 三 短 - 石 = 〇 - 〇 - 三 〇 〇 〇 🕱		
🖺 🔛 Mule		
🖸 Console 🖄 📃 🗮 🙀 🛃 💭 🛃 💭 🛃 💭 🛃 🖓 🖬		
<terminated> Database [Mule Application] C:\Program Files\Java\;re6\bin\javaw.exe (Jan 2, 2013 6:38:25 PM)</terminated>		8
	^	Ju
tractLifecycleManager: Initialising: 'connector.file.mule.default.disnatcher.2645268', Object is: FileMessageDisnatcher		
utractLifecycleManager: Starting: 'connector.file.mule.default.disnatcher.2645268', Object is: FileMessageDisnatcher		
uracLifecycleManager: Initialising: 'connector.file.mule.default.disnatcher.2915013'. Object is: FileMessageDisnatcher		
tractLifecvcleManager: Starting: 'connector.file.mule.default.disnatcher.2915013', Object is: FileMessageDisnatcher		
tractLiferveleManager. Initializing, Competer file mule default disnatcher 4750048; Chiert is: FileMessagelisnatcher		
iractliferurilamanger: Starting, Jonnartor file mule default dienercher 4750481 (biert is: FileMassembienercher		
Prachiferral Hanager, Stateng, Connector file mult default disarcher 51483801 Chief is FileMessarahienstcher		
irrarlifarurilamangari Starting, Jonnartor fila mula default dienarcher 51483801 (hiertie: FileMacanabiensteher		
irracLiferveleManager. Juiting, connector file mule default dignatcher 18/59800. Object is FileMerigerspacehienatche	r	
stractliferveleManager: Starting, Connector file mule default disnatcher 182598001 (Disct is FileMessageDisnatcher	-	
irachliferydlamager. Juitiliging Connector file mule default dienetrios 23692851 Object is: FileWessageDienetrie		
brachlifernellenanger. Hittalising, Commetter filmmet default dispatcher 2005000, Gujett is, Filenessager bjøtte		
a FileConvertanteget, Schleget for Directive FileOcol 112 (1920) - Color		
a FileConvector. Writing file to: Ditest/data/wrfile02-0113 18-38-30 187 txt		
a FileConnector, Writing file to, Divestidate/writie02-01-13_10-30-30.107.000		
a FileConnector, Writing file to, Diversidated writie02.01.13 18.38.30 187 tvt		
interesting of the second s	. 🛄	
irracLiferveleManager: Starting, Connector file mult default dignatcher 86686101, Object is, FileMasageDignatcher		
a FileConnector. Writing file to: D:/test/data/wrfileO2_01_13 18_38_30 187 tvt		
e FileConnector: Writing file to: D:\test\deta\muffle02_01_13_18_38_30_187_tvt		
e FileConnector: Writing file to, DitestAdra wrfileD2-01-13 18-38-30 187 txt		
tractifectuleManager: Initializing: 'connector.file.mult.default.disnatcher.30170403', Chiert is: FileMessageDisnatche	r	
uractLifecterleManager: Starting: 'connector file mule default disnatcher 30170403' Object is: FileMessageDisnatcher	~ ~	
Concontractal and a second sec	>	
i D ^M 📕 🔂 👘 🔛 👘 🔂 👘 🔂	8	
🔰 Start 🕥 azaz 🧿 IMAP 📓 Chapt 📓 659_T 🏆 Mule 🌮 MySQ 🖆 data 🕋 Downl 📓 4401 🔇 👀 🗞	6:44	PM

2. The file in Notepad++ looks like the following screenshot. This data is retrieved from the database in the mule table, which was created in the PostgreSQL database.

D:\test\data\myfile02-01-13_18-38-30.187.txt - Notepad++	
File Edit Search View Encoding Language Settings Macro Run TextFX Plugins Window ?	Х
myfie02-01-13_18-38-30.187.txt	
1 {OPEN_PRICED=54.32, HIGH_PRICE=54.32, pk=9, CLOSE_PRICE=21.33, LOW_PRICE=32.45, SYMBOL=KAS}	
v Normal text file length : 91 lines : 1 Ln : 1 Col : 1 Sel : 0 Dos\Windows ANSI	INS
🔧 start 🔰 😒 aza 💿 IMA 😨 Cha 🔯 659 🖤 Mule 🎓 MyS 🎓 data 🎓 Dow 🔯 440 😭 Dijt 🔇 🔊 🖓	6:48 PM



Implementing the File Transport channel using the File Endpoint

The File Endpoint is a transport channel. We can transfer files from one directory to another through the File Endpoint. The File Inbound Endpoint is used for setting the source path, and the File Outbound Endpoint is used to set the destination path. You can define the File Endpoint globally as well. If you declare the File Endpoint globally, you can set the reference tab to assign the name of the File Endpoint.

Getting ready

In this section, you will see how to use the File Endpoint and how to transfer files from one location to another using it.

 Use the File Endpoint, a Choice Router, and the Echo component for transferring files from one location to another. Open Mule Studio and enter the workspace name as shown in the following screenshot:

Y Workspace Launcher	
Select a workspace	
Mule Studio stores your projects in a folder called a workspace. Choose a workspace folder to use for this session.	
Workspace: El\MuleCookBook	Browse
Use this as the default and do not ask again	
	OK Cancel

2. To create a new project, go to File | New | Mule Project. Enter the project name, File Transfer, and click on Next and then on Finish. Your new project is created. Now you can start the implementation.

🖌 Mule - Mule Studio						
File Edit Source Refactor Navi	gate Search Project I	Run Window Help				
New	Alt+Shift+N	🗠 😋 Mule Project	-	*		
Open File		/ Java Project				
Close	Ctrl+W	📑 Project				- 8
Close All	Ctrl+Shift+W	Apping Flow				
🗑 Save	Ctrl+S	Mule Flow				
📓 Save As		瞪 Package				
院 Save All	Ctrl+Shift+S	🞯 Class				
Revert		🗊 Interface				
Move		🞯 Enum				
Rename	F2	Annotation				
🔊 Refresh	F5	Source Folder				
Convert Line Delimiters To)	 Interview of the set of the set				
👜 Print	Ctrl+P	Folder				
Switch Workspace	,	File Intitled Text File				
Restart		Oncoded Texe The				
N- Import		Example				
A Export		📑 Other	Ctrl+N			
		-				
Export diagram to		_				
Properties	Alt+Enter					
1 FlowRef.mflow [flowref/flows]						
2 package.html [activity/src/main	v]					
3 Activity.mflow [activity/flows]						
4 StringToNumber.java [usdconv	erter/]	_				
Exit						
An oddine is not available.						
: =¢						
: U'				1		: = 🖸 @ 🖄 M 🗳
😤 start 👘 🕤 Echo Exan	nple - M 🛛 🗐 Chapt	er 2 - Micros 🛛 🗐 65	9 The Author a.		Mule - Mule Studio	🗌 🔿 🔥 🚬 🧶 🗞 💽 10:59 AM

Endpoints

How to do it...

In this section you will see how to create a flow in Mule Studio and how it works using the File Endpoint. In this example, you will use four components: the File Inbound Endpoint, the Choice Router, the Echo component, and the File Outbound Endpoint.

1. To create a flow, go to the file Transfer.mflow file. Firstly, you have to drag the **File** Endpoint from the palette and drop it on the canvas.



2. To configure the File Endpoint, double-click on it and set the source path shown in the following screenshot. In the D:\, create a folder called Order data. In this folder, create two folders: an Input folder and an Output folder. In the Input folder, place three XML files and these three XML files will be transferred to a specific folder (the Output folder).


-V Y Endpoint Properties File (Inbound Endpoint) 📑 • 🔛 🗟 🖻 🗄 🏘 • 🔘 The File Endpoint allows Mule applications to read or write files on the local file system. 🔛 🔛 Mule - -洋 Packa 🛛 🍃 Type General Advanced References Documentation 8 🖻 🕏 -Display -Filter: Ju 🗉 🔛 database Display Name: File Select Path Information endpoints 😑 📂 file_transfer Path: D:\OrdersData\Input ... 皹 Ajax Ini_danser
 I Move to Pattern: 📑 Database - 🥮 src/main/java - 🥮 src/test/java ... Move to Directory: 😑 FTP 📁 File src/main/resources Polling Information B src/test/resources Generic Polling Frequency: 1000 File Transfer.mflow HTTP File Age: 500 🖂 IMAP 🗁 mappings File Name Filter 르 JMS + × File Name Regex Filter:) Jetty 🖂 POP3 + Helloworld 🕚 Quartz 🗉 🔂 imap 🚥 RMI 😟 🔂 jdbc 窿 Redis (Streaming) 👝 SFTP > Copes 2 🗄 Outline 🖾 Components a Transformers ? OK Cancel Filters Flow Control . Crror Handling Cloud Connectors Message Flow Global Elements Configuration XML • 8 i e 🔝 @ 😥 M 🗝 📋 e 🖃

🔄 Chapter 4 - Micr

🗐 659_The Auth

Y Mule - file_trans.

🍟 Mule - file_trans... 🛛 🔇 🦻

Here, you can configure the File Inbound Endpoint. Once you configure it, click on the ${\bf OK}$ button.

-168

🛃 start

💿 azaz.desai@att

3. For identification or display purposes, we drag the **Echo** component from the palette and drop it on the canvas. Change the **Echo** component name to **File Transfer to Specific Folder**.



4. To transfer a file to a specific folder, drag the **Choice** Router or Flow Control. In the **Choice** Router, you can define the condition. If the condition matches with the expression, the flow will be executed.





- Y Mule file_transfer/flows/File Transfer.mflow Mule Studio File Edit Navigate Search Project Run Window Help : 📫 • 🔛 🗟 🗄 🕸 • 🕖 • 🍇 • : | 🖶 🎯 • : | 🅭 🛷 • :| 🚥 :| ½ - २] - ७- (- ० - : | 🗸 🗠 🗙 🔛 🔛 Mule 😫 Packa 🛛 🍃 Type 🗖 🗖 😭 *File Transfer 🔀 - -8 🖻 🙀 ≧ Filter: Ju 🗉 🔛 database ^ 😞 Select endpoints 🚰 😑 📂 file_transfer 皹 Ajax 🖶 🛋 JRE System Library [Java flow: File_TransferFlow1 📝 🗐 Database 🗷 🛋 Mule Runtime [Mule Serve # src/main/java 😑 FTP 😑 File src/main/resources B src/test/resources Generic File Transfer.mflow HTTP 🖂 IMAP 🗁 mappings flowref 르 JMS {[[1 1]]} Echo FILE) Jetty File File Transfer to Spacific Folder Choice + Helloworld 🖂 РОРЗ Default 🕚 Quartz 🗉 🔂 imap 🚥 RMI FILE 😟 🔂 jdbc 窿 Redis (Streaming) File 👝 SFTP ~ > Copes 2 -🗄 Outline 🖾 Components a Transformers Filters ----Flow Control Carl Error Handling -- 10 Cloud Connectors < Message Flow Global Elements Configuration XML а 🖹 @ 📴 И 🔫 80 ə 📃 Mule - file_t... 🛃 start 💧 💿 azaz.desai 📓 659_The Au. 000 Y Mule - file_t.. 🗀 Inpu
- 5. To set the destination target, drag three **File** Endpoints from the palette and drop them on the canvas, as shown in the following screenshot:

6. To configure the Choice Router or Flow Control, double-click on it. Here, you can see the When partition; this is our condition area. If the condition is true, it will execute; for example, in the condition /shiporder/shipto/city='Ahmedabad', shiporder is a tag which is used in the XML file shown in the following screenshot:



7. This is an XML file, which is used in the Input folder.



Endpoints _____

8. Double-click on the first **File** Endpoint to configure it. In the same way, you can configure the other two File Outbound Endpoints, one for the BANG folder and the other for the Other folder.

⊻Mule - file_transfer/flows/Fil	🖌 Endpoint Properties 🛛 🛛 🗙	- B 🗙
File Edit Navigate Search Project	File (Outbound Endpoint)	
i 📫 • 🔛 🕼 🖻 i 🏇 • 💽 •		
😫 📔 Mule	This element most have an accidutes for one or sets.[parit] or [address] or [rei]	
洋 Packa 🕴 🍃 Type 📄 🗖	General Advanced References Documentation	
	CDisplay	Filter:
🗄 😂 database 🛛 🔺	Display Name: File	
😟 😥 dropboxdemo		Select
🗉 😂 echo	Path Information	Endpoints 📀
😑 😂 file_transfer	Path:	(III) Diay
JRE System Library [Java	Output Pattern:	Ajax
Mule Runtime [Mule Serve		Database
src/main/java		FTP FTP
src/cest/java	Browse For Folder	File
😕 src/test/resources		Generic
🖃 🧁 flows		нттр
File Transfer.mflow		
- mappings	🗉 🦳 NewSpring	IMAP
	😑 🦳 OrdersData	I IMS
flowref	🗀 Input	Jetty
🗄 😂 helloworld	🗏 🚞 Output	POP3
🖲 😂 http-security_filter		O Quartz
🕀 😥 imap	Definer and Definer	E RMI
⊞ 🎦 jdbc		Redic (Streaming)
i≡re⊃ jms		(Screaning)
	Folder: AHD	SHIP .
< >>		Copes Scopes
🗄 Outline 🛛 🗖 🗖	Make New Folder OK Cancel	_ 🧁 Components
		arransformers
	UK Cancel	🚰 Filters
		Control
		arror Handling
		> 🔁 Cloud Connectors
	Message Flow Global Elements Configuration XML	
E □ \$	l i i	F 🛃 @ 😥 M 🖏 🕴 F 🗳
Home . Tim	Tile Endogin V Mula - Ela L 🕅 Chanter 4 . 🕅 659. The Au	Mula - fila k 🖉 💷 🔍 2-22 DM



9. In the **Choice** Router, you need to add an attribute evaluator="xpath" after the expression attribute to read the XML tag, as shown in the following screenshot:



175



10. To deploy the application code in the Mule server, go to **Run As** | **Mule Application**, and the Mule Server will deploy your application.

How it works...

Once your application is successfully deployed, you can see the log on the console. All files are transferred to that particular directory.

If your application code is successfully deployed, you will see the message ${\tt Started}$ app 'file transfer' on the console.

🖌 Mule - file_transfer/flows/File Transfer.mflow - Mule Studio 🗧 💽							
File Edit Navigate Search Project Run Window Help							
i 白・田 岡 ি · · · · · · · · · · · · · · · · · ·							
🖹 🔛 Mule							
🖸 Console 🖄 📲 🕷 🔐 🥵 🖉 🖳 🗂 🖓 👘							
ile Transfer [Mule Application] C:\Program Files]Java)res[bin]javaw.exe (Jan 3, 2013 4:01:43 PM)							
createMultipleTransactedReceivers=true							
connected=true							
supportedProtocols=[file]							
serviceOverrides= <none></none>							
INFO 2013-01-03 16:01:47,406 [main] org.mule.lifecycle.AbstractLifecycleManager: Starting connector: connector.file.mule							
INFO 2013-01-03 16:01:47,421 [main] org.mule.lifecycle.AbstractLifecycleManager: Starting model: _muleSystemModel							
INFO 2013-01-03 16:01:47,421 [main] org.mule.construct.FlowConstructLifecycleManager: Starting flow: File_TransferFlow1							
INFO 2013-01-03 16:01:47,421 [main] org.mule.processor.SedaStageLifecycleManager: Starting service: File_TransferFlow1.:							
INFO 2013-01-03 16:01:47,421 [main] org.mule.component.ComponentLifecycleManager: Starting component: commponent.2777203							
INFO 2013-01-03 16:01:47,437 [main] org.mule.transport.file.FileConnector: Registering listener: File_TransferFlow1 on (
INFO 2013-01-03 16:01:47,453 [main] org.mule.lifecycle.AbstractLifecycleManager: Initialising: 'null'. Object is: FileMe							
INFO 2013-01-03 16:01:47,453 [main] org.mule.lifecycle.AbstractLifecycleManager: Starting: 'null'. Object is: FileMessa							
INFO 2013-01-03 16:01:47,453 [main] org.mule.module.launcher.application.DefaultMuleApplication: Reload interval: 3000							
INFO 2013-01-03 16:01:47,453 [main] org.mule.module.management.agent.WrapperManager&gent: This JVM hasn't been launched							
NFO 2013-01-03 16:01:47,515 [main] org.mule.module.management.agent.JmxAgent: Attempting to register service with name:							
NFO 2013-01-03 16:01:47,515 [main] org.mule.module.management.agent.JmxAgent: Registered Endpoint Service with name: M							
INFO 2013-01-03 16:01:47,515 [main] org.mule.module.management.agent.JmxAgent: Registered Connector Service with name Mu							
INFO 2013-01-03 16:01:47,531 [main] org.mule.DefaultMuleContext:							

* Application: file_transfer *							
* OS encoding: Cp1252, Mule encoding: UTF-8 *							
*							
* Agents Running: *							
* JMX Agent *							

INFO 2013-01-03 16:01:47,531 [main] org.mule.module.launcher.DeploymentService:							
+ Started app 'file transfer' +							
+++++++++++++++++++++++++++++++++++++++							
INFO 2013-01-03 16:01:48,453 [[file_transfer].connector.file.mule.default.receiver.01] org.mule.transport.file.FileMessev							
🙀 Start 🤍 Home - Iim 🦁 File Endpoin 🍟 Mule - File_t 🖉 Chapter 4 🖉 659_The Au 📁 Input 🍡 🖓 Mule - File_t 🔇 🖉 🗞 4:01 P							

You can see that each file is transferred to a specific folder in the following screenshot:



You can see that the first XML file is transferred to the first destination folder:



The second XML file is transferred to the second destination folder:



And, the third XML file is transferred to the third destination folder:



Sending messages asynchronously using the AJAX Endpoint

AJAX stands for **Asynchronous Java and XML**. The AJAX connector allows Mule actions to be sent and received asynchronously to and from the web browser. We will see how to use AJAX Endpoint in this recipe.

Getting ready

In this section, you will see a Google spell check example and how it works. Here, we use the HTTP Endpoint, the AJAX Endpoint, the Echo component, an Object to XML transformer, and the XSLT transformer. In this example you will see how to configure the AJAX component.

1. Open Mule Studio and enter the workspace name as shown in the following screenshot:

¥ Workspace Launcher	
Select a workspace	
Mule Studio stores your projects in a folder called a workspace. Choose a workspace folder to use for this session.	
Workspace: E:\MuleCookBook	Browse
Use this as the default and do not ask again	
	OK Cancel

2. To create a new project, go to **File** | **New** | **Mule Project**. Enter the project name called AJAX and click on **Next** and then on **Finish**. Your new project is created. Now you can start the implementation.

🍟 Mule - Mule Studio						_ = = X
File Edit Source Refactor Navigate S	Search Project I	Run Window Help				
New	Alt+Shift+N 🕨	🔄 Mule Project				
Open File		/ Java Project				
Close	Ctrl+W	Project				- 8
Close All	Ctrl+Shift+W	📌 Mapping Flow				
🔛 Save	Ctrl+S	Mule Flow				
📓 Save As		🖶 Package				
Nave All	Ctrl+Shift+S	🞯 Class				
Revert		🗊 Interface				
Move		🕼 Enum				
Rename	F2	@ Annotation				
🗞 Refresh	F5	Source Folder				
Convert Line Delimiters To	,	Solder				
i Print	Ctrl+P	File				
Switch Workspace	,	Untitled Text File				
Restart						
Magnet		Example				
Z Export		📑 Other	Ctrl+N			
🔯 Export diagram to		-				
Properties	Alt+Enter	-				
1 FlowRef.mflow [flowref/flows] 2 package.html [activity/src/main/] 3 Activity.mflow [activity/flows] 4 StringToNumber.java [usdconverter/	.]					
Exit						
All oddine is not available.						
i 0*						i a 🖹 @ 📴 M 🗉
🐉 start 💿 Echo Example - M	📓 Chapt	er 2 - Micros 🛛 👜	659_The Author	a 🦉 untitled - Paint	Y Mule - Mule Studio	🔇 🐴 💐 🧶 🗞 🔝 10:59 AM

-182

How to do it...

In the following section, you'll see how the spell check example works, and you will also see how to configure each component in flow.

1. Create a folder inside src/main/app called docroot as shown in the following screenshot. Inside that folder, create an XML file called SpellCheck.html. In this file you have to create an HTML page, which will be displayed on the browser. We call the JavaScript function, callEcho, through the onkeyup function.

₩ Mule - ajax/src/main/app/dooc	croot/SpellChecker.html - Mule Studio	_ 7 🗙					
File Edit Source Refactor Navigate Search Project Run Window Help							
i 🗈 • 🔛 🗟 🖆 i 🕸 • 🕥 •	🌯 • E 🔀 🞯 • E 🖄 🛷 • E 📖 E 🖢 - 🖗 • 😓 • 🗢 -						
😭 🔛 Mule							
🕌 Packa 🛛 🔓 Type 📃 🗖	SpellChecker.html 🕅						
□ 🕏 🏹	1 html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/htm</th <th>JU</th>	JU					
i 🗁 ajax 🔼	3⊜ <head></head>						
🗄 📑 JRE System Library [JavaSE	<pre>4 cmeta http-equiv="Content=Type" content="text/html: charset=TS0-8859-1"></pre>						
Mule Runtime [Mule Server :	5 <title>Aiax Smell Checker</title>						
	6 <acrint arc="mule-resource/is/mule.is" twne="text/iavascript"></acrint>						
🗁 src/test/java	7						
src/main/resources	8 <script type="text/iavascript"></th><th></th></tr><tr><th>src/test/resources</th><th>9 function callEcho()</th><th></th></tr><tr><th>Hows</th><th>10 (</th><th></th></tr><tr><th>wonm.xALA</th><th><pre>11 var data = new Object();</pre></th><th></th></tr><tr><th>mappings</th><th><pre>12 data.phrase = document.getElementById('txt1').value;</pre></th><th></th></tr><tr><th>Si Consta</th><th>13 mule.rpc("/services/echo", data, callEchoResponse);</th><th></th></tr><tr><th>i i i i i i i i i i i i i i i i i i i</th><th>14 }</th><th></th></tr><tr><th></th><th>15</th><th></th></tr><tr><th></th><th>16</th><th></th></tr><tr><th></th><th>17 function callEchoResponse(message)</th><th></th></tr><tr><th></th><th>18 (</th><th></th></tr><tr><th>mule-app:prope</th><th colspan=6>19 document.getElementById("response").innerHTML = "Response: inbsp;" + message.d</th></tr><tr><th>indie-deploy.prc</th><th colspan=6>• }</th></tr><tr><th>wile-project yml</th><th colspan=6></script>						
databace	22						
database database	23						
echo V	24 [©] <body></body>						
<	<pre>25 <input id="txt1" onkeyup="callEcho()" type="text"/></pre>						
🗄 Outline 🕱 🛛 🗖 🖓 🗖 🗖	26						
	ar 28. div.id="reepone=">./div.						
B btol	29						
	30						
	31 <th></th>						
		-					
SpellChecker.html - ajax/sr	rc/main/app/doocroot	; e 📮					
🛃 start 💿 azaz.desai@att	t 👩 AJAX Transport 🦙 Mule - ajex/flo 📓 Chapter 4 - Mic 🖆 MuleCookBook 🛛 🍟 Mule - ajax/src/ 🕼 🕏	4:49 PM					

This is a SpellCheck.html file; we have to put this file inside the src/main/app/ docroot location:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;
charset=ISO-8859-1">
<title>Ajax Spell Checker</title>
<script type="text/javascript" src="mule-resource/js/mule.js"></
script></script></script></script></script></script></script></script></script></script></script></script>
```



```
<script type="text/javascript">
  function callEcho()
  {
    var data = new Object();
    data.phrase = document.getElementById('txt1').value;
    mule.rpc("/services/echo", data, callEchoResponse);
  }
function callEchoResponse(message)
{
    document.getElementById("response").innerHTML =
"<b>Response:&nbsp;</b>" + message.data + "\n";
}
</script>
</head>
<body>
<input type="text" id="txt1" onkeyup="callEcho()" />
   <div id="response"></div>
   </body>
</html>
```

2. Create a transform.xsl file inside src/main/app; this is how it will look:



The transform.xsl file provides instructions for extracting data from incoming messages and translating that data into a form that applications can digest.

```
<?xml version="1.0" encoding="utf-8" ?>
<xsl:stylesheet version="2.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
<spellrequest textalreadyclipped="0" ignoredups="0"
ignoredigits="1" ignoreallcaps="1">
<text>
<xsl:value-of select="map/entry/string[2]" />
</text>
</spellrequest>
</xsl:template>
</xsl:template>
```

3. To create a flow, go to the AJAX.mflow file. Then, go to the **Global Elements** tab; click on **Create** and go to **Connectors** | **Ajax**, and then click on **OK**.





4. To configure the AJAX connector, enter the server URL http://l27.0.0.1:8090/ Ajax in the Server URL: field. In Resource Base, enter the .html file path, which is located at src/main/pp/docroot.

🗑 Mule - ajax/flows/AJAX.mflov	Y Global Element Properties				. 2 🛛
File Edit Navigate Search Project	Ajax				
🗄 📬 📲 🗟 🖻 🖗 • 🕥 •	Connector configuration for Ajax end	points.			
📫 🔛 Mule					
Mule Packa S Type ajax Packa S	General Tmings Security Docum Servert Name: Ajax Server URL: Ibtp://it Resource Base: Ibtable Reply To Settings Log Level: Icon Disable Reply To Settings Log Level: Icon Disable Reply To Solo Commented	entation 27.0.0.1:8090/Ajax //app/docroot	OK Canc		Create Edit Delete
	Message Flow Global Elements Con	figuration XML			
: •		<u>e</u>		8	@ 🖳 M 🗝 📄 🖷 🖳
🛃 start 🚺 👩 azaz desai@a	attunein 🧖 XSLT Transformer Re	Chanter 4 - Microsoft	Mule - aiex/flows/Aie V M	le - aiav/flow	c/A1 🔹 🖓 🛸 12:08 PM



- Whule ajax/flows/AJAX.mflow Mule Studio File Edit Navigate Search Project Run Window Help : 📫 • 🔛 🗟 🗄 🕸 • 🕖 • 🍇 • : | 🖶 🎯 • : | 🅭 🛷 • :| 🚥 :| ½ - २] - ७- (- ० - : | 🗸 🗠 🗙 📑 🔛 Mule 😫 Packa 🕺 🏂 Type 🛛 🗖 🗈 SpellChecker.html 🛛 😭 *AJAX 🖄 🗶 transform.xsl - -8 🖻 🔄 ∑ Filter: Ju ajax ^ 😞 Select endpoints 🚰 # src/main/java
 # src/test/java 🔎 Ajax # src/main/resources
 # src/test/resources 😑 Database 😑 FTP 🗉 🧁 flows 😑 File 🗁 mappings 🗏 🧁 src Generic 😑 🗁 main HTTP 😑 🗁 app flow: AJAXFlow1 📝 app
 doocroot
 AJAX.xml
 mule-app.pro
 mule-deploy.i
 transform.xsl 🖂 IMAP a Scopes AJAX Async Ajax 🐻 Composite Source 🕞 test Flow test mule-project.xml database for pboxdemo for pboxdemo for pboxdemo for pboxdemo for pboxdemo Q Foreach Message Enricher Poll File transfer > Sub flow ~ - -E Outline 🛛 Components a Transformers <mark>è</mark> Filters Flow Control 100 a Error Handling Cloud Connectors Message Flow Global Elements Configuration XML ∎\$ 8 i e 🖹 @ 🔯 M 🗝 🗄 e 🖃 🛃 start 💿 azaz.desai@attunei 🍟 Mule - ajax/flows/AJ... 🛛 🔿 🥵 🗐 Chapter
- 5. Go to the message flow, drag the **AJAX** Endpoint from the palette, and drop it on the canvas.

6. Double-click on the **Ajax** Endpoint to configure it. You have to enter a channel path, /services/echo, the same path which you have assigned in the SpellCheck. html file. Go to the **Reference** tab, select the connector reference name and click on the **OK** button.

Y Mule - ajax/flows/AJAX.mflo	Y Endpoint Properties				. 3 X
File Edit Navigate Search Project	Ajax (Inbound Endpoint)				
i 📬 • 🖩 🗟 🖆 i 🏇 • 🔘	The Ajax Endpoint allows Mule events to be	e sent and received asynchronou	usly to and from the web browser.		
🖹 🔛 Mule					
😫 Packa 🕺 🍃 Type 📄 🗖	General Advanced References Docum	entation			
E 😫 🗌	Global References			5	Filter:
📮 😂 ajax 🔮	Connector Reference: Ajax		v 🕹 🖉		
Intersection State (Java)	Endpoint Reference:		÷ 0		13 Select
mule Rundine [mule serve	- Transformers References: Request -				Endpoints 📀
🕮 src/test/java	Thansioniers Kererences, Kegaese				🐖 Ajax
- 🥮 src/main/resources	Global Transformers:		Y		🕃 Database
src/test/resources	Transformers to be applied:				E FTP
- 🗁 mappings					🥦 File
🖻 🧁 src					Generic
🖃 🧀 main					() HTTP
😐 🦢 doocroot					IMAP
					Geopes 🛛 🗠
mule-app.pro					Async
transform.xsl					😞 Composite Source
- 🧀 test					Flow
mule-project.xml					Q Foreach
arter diabase arter diabase arter diabase					••• Message Enricher
🗉 🔛 echo					(Poll
file transfer					Sub flow
🗄 Outline 🛛 🗸 🖓 🗖 E					Components
	0		OK Cancel		and Transformers
	U		Cancer		htters
Ref.					Control
Para a					Caror Handling
					Cloud Connectors
	Message Flow Global Elements Configura	tion XML			
1 D*		<u></u>		- 8	
🛃 start 💿 azaz.desai@	attunein 🤇 SLT Transformer Re	Chapter 4 - Microsoft	Mule - ajex/src/main/ Y Mule	- aiaxi	flows/AJ 🔍 😼 🗞 12:55 PM

7. Drag the **Object to XML** transformer from the palette and drop it on the canvas. It is used to convert a JavaScript object to XML.



8. Drag the **XSLT** transformer from the palette and drop it on the canvas.



- _ 7 🗙 🖌 Pattern Properties XSLT 📑 • 🔛 🗟 🖻 🗄 🏘 • 🔘 The XSLT Transofrmer will transform XML using XSLT. 🔛 🔛 Mule 😫 Packa 🕺 🍃 Type 📄 - -General Advanced Documentation 8 🖻 🕏 -Display -Filter: Ju ajax Display Name: XSLT Select Mule Runtime [Mule Serve XSLT Carl Endpoints XSLT Content: - 🥮 src/main/java - 🌁 src/test/java a Scopes # src/main/resources Components 进 src/test/resources Transformers 🗉 🧁 flows Property 🗁 mappings 🗏 🧁 src 🤌 Python 😑 🗁 main 💎 Ruby 😑 🗁 app app
 doocroot
 AJAX.xml
 mule-app.pro
 mule-deploy.q
 transform.xsl ••= Script Serializable to Byte Array Session Variable src/main/app/transform.xsl XSL File: Set Payload 🕞 test test mule-project.xml database for pboxdemo for pboxdemo for pboxdemo for pboxdemo for pboxdemo String to Byte Array Output Encoding: ~ String to Email Transformer Reference 🗄 🟳 file transfer venue Variable > ML to Object ~ -🗄 Outline 🖾 SLT XSLT ? OK Cancel Filters Flow Control **≈**→ = →= a Error Handling Cloud Connectors Message Flow Global Elements Configuration XML 8 e 🔝 @ 🔯 🖊 🔫 📋 e 🚍 Mule - ajex/flow Image: Second sec 🛃 start 🔰 😒 Blogs posted by 🗀 Mule Training 👜 Chapter 4 - Micr 🍟 Mule - ajax/flo
- 9. Double-click on the **XSLT** transformer to configure it. In **XSL File:**, enter the path of the .xsl file, which is located in src/main/app and click on the **OK** button.

10. Drag the **HTTP** Endpoint from the palette and drop it on the canvas.



11. Double-click on the **HTTP** Endpoint to configure it, and in the **Host:** field enter the spell check API link. Select the **POST** method.

∑ Mule - ajax/flows/AJAX.mflov	¥ Endpoint Properties	🔽 🗖 🔽
File Edit Navigate Search Project	HTTP (Outbound Endpoint) The HTTP Endpoint allows Mule application to connect to web resources through the HTTP transport protocol. This endpoint can also implement security through HTTPS.	
Packa 23 Type Packa 23 Packa 24 Packa 24 Packa 25 P	General Advanced References HTTP Settings Documentation Display Display Name: HTTP Exchange Patterns One-way @request-response Basic Settings Enable HTTPS @ Enable HTTPS will require configuring a HTTPS Connector Host: www.google.com/tbproxy/spel?lang=en Port: 80 Path: Image: Configure Con	Filter: Select Ju Select Codpoints Cod Parabase FTP Coduction FTP FTP Coduction FTP FTP FTP FTP FTP FTP FTP FTP
i □ ◆	🔂 🗄	i e 🔝 @ 😥 M 🗝 i e 🗉
Blogs posted b	y 👩 Ajax Spell Chec 📓 Chapter 4 - Mic 🍟 Mule - ajex/flo 🍟 Mule - ajax/flow 🍃	Mule Training 🔹 🔊 🔥 4:01 PM



12. For display purposes, drag the **Echo** component from the palette and drop it on the canvas.

13. Now we are ready for the deployment. If you haven't saved your application code, first save it. After saving your project, right-click on the AJAX.mflow file and go to Run As | Mule Application.



14. If your application code is successfully deployed, you will see the message Started app 'ajax ' on the console.



How it works...

Paste this URL in your browser: http://127.0.0.1:8090/Ajax/SpellChecker.html, and type any word into the textbox. You can see the related search results through the HTTP response by the Google spell check API.



Using the Servlet Endpoint to listen to events or messages from servlet requests

Identify the path to the servlet through which the event or message is received. The Path property can be set from the General tab. We will see each tab that is used in the Servlet Endpoint. The Servlet Endpoint contains four tabs:

- ► General
- Advanced
- References
- Documentation



Getting ready

The **General** tab contains two fields: one is **Display Name:** and the other is **Path:**. The display name is used for identity purposes, that is, to identify the channel over which your Servlet Endpoint is going to communicate with the client web page. Here you have to assign the path, and use the same path for the browser.

Mule - mail_notification/flows	/Mail_Notification.mflow - Mule Studio	🔳 🗗 🚺
File Edit Navigate Search Project	Run Window Help	
i 📬 • 🔛 🕼 🗁 i 🕸 • 🕥 •	🖌 Endpoint Properties 🛛 🔀	
🗈 🕎 Mule	Servlet (Inhound Endpoint)	
😫 Packa 🕴 🔭 🗖	Attribute 'path' is required	- 8
The aiex	General Advanced Defension Demonstration	Filter: serv
🖶 🔁 allrouter	Direlau	🔓 Select
🗈 📴 catchexcaption	Dispay	Suggestions ∞
🗷 🗁 cxf	Uispiay Name: Serviec	Servlet
i international	Generic	DEST
iller	Path: 0	COAD.
🗉 🔛 hellomule		SUCAP
😑 😂 mail_notification		FTP
Image: A state of the state		HTTP
Hule Ruhume [Hule Server 3 Hule strong and the server 3		Jetty
🕮 src/test/java		👝 SFTP
- 🥮 src/main/resources		Endpoints 🗠
- C src/test/resources		Servlet
Mail Notification of Iow		=
🗉 🗁 src		
mule-project.×ml		
H 🔁 nametransfer		
SPIIP Spellcheck		
🗉 🔛 testtransformer		
🗄 😂 transformer		
twiter		
Witchintegration		
	OK Cancel	_
< >	Message Flow Global Elements Configuration XML	
: •	loading http://wwwcurrent.xsd 💷 👘 🧝 🦹 @	🔍 VI 🗄 # 🗄 🗍 # 🗳
Start SPMN 2.0 / Act	i 🧑 Servlet Endpoint 📓 Chapter 4 - Micr 🍟 untitled - Paint 🛛 🍟 Mule - servlet/fi 🍟 Mul	e - mail_notifi 🔇 🥩 4:03 PM

How to do it...

In this section, you will see how to configure the Servlet Endpoint and how to use it.

In the **Advanced** tab, enter the address for the Endpoint; for example, <code>localhost:8080</code>. Response timeout identifies how long the Endpoint must wait for a response. The default is **10000** ms.



How it works...

The **References** tab is used when you are creating a global element. If you use a global element to configure the **Server** Endpoint, you have to configure the **References** tab.

Mule - mail_notification/flows	/Mail_Notification.mflow - Mule Studio	💶 🗗 🗙
File Edit Navigate Search Project	Run Window Help	
i 📬 • 🔛 🕼 🗁 i 🎄 • 🕥 •	🖌 Endpoint Properties 🛛 🔀	
🗈 🔛 Mule	Servlet (Inhound Endpoint)	
😫 Packa 😫 隆 Type 📃 🗖	Attribute 'path' is required	
aiev	Convert Advanced References Doministration	Filter: serv
🖶 🔁 allrouter		la Select
atchexcaption		🔶 Suggestions 👳
E CXf		Servlet
E C filetransfer	Endpoint Reference:	REST
🗈 😂 filter	Transformers References: Request	SOAP
	Global Transformers:	ETP
mail_notification	Transformers to be applied:	
A Mule Runtime [Mule Server 3		HIP
🗈 🕮 src/main/java		Jetty
- 🕮 src/test/java		SFTP
src/main/resources		Endpoints 🗠
signestifiesturces signesturces		💽 Servlet
Mail_Notification.mflow	Transformers References: Response	
- Company mappings	Child Tura Suman	
🗄 🧁 src		
Inde-project.xiii Image project.xiii	Transformers to be applied:	
😟 🔁 SMTP		
🖶 🗁 spellcheck		
🗉 😥 spring		
testtransformer		
🗈 🔛 twiiter		
🗄 🔛 twiiterintegration		
I webservice-cxf	(?) OK Cancel	a Austable y
		CS AYGIIGDIC X
	Updature of the second se	es are available for your software.
	Message How Global Liements Configuration XML Set u	Reminder options
E □ *	😥 i	serie e e pre 🚽 🕽
👭 start 💦 🕤 BPMN 2.0 / Act	i 🧿 Servlet Endpoin 👜 Chapter 4 - Mic 🦉 untitled - Paint 🛛 🍟 Mule - servlet/fl 🍟 Mule -	mail_noti 🔇 🖉 🗞 4:17 PM

5 Transformers

In this chapter, we will cover the different types of transformers. You will learn about the following topics:

- ► Configuring the JSON-to-Object transformer
- ► Configuring the Object-to-XML transformer
- ► Configuring the Message and Variable transformers
- Creating the custom transformer
- Understanding the DataMapper transformer

Introduction

A **transformer** is used for converting a message from one format to another. You can create a custom transformer as well, which we will see later. There are different types of transformers, such as the **Java** transformer, the **DataMapper** transformer, the **XSLT** transformer, and the **Append String** transformer. Standard transformers are easy to use compared to custom transformers. For example, we use the Append String transformer to append a string text to the original message payload with a single line of code, as follows:

```
<append-string-transformer name="Greeting" message=", How are you?"/>
```

If the original string message is Hello, it will convert it to Hello, How are you?.

Transformers -

Configuring the JSON-to-Object transformer

The JSON-to-Object transformer is used to transform a JSON string to object data and vice versa. Using this transformer, we convert JSON data into an object type.

Getting ready

In this example, you will see how to store data in the database using the JSON-to-Object transformer. We need four components for this: the File Endpoint, the JSON-to-Object transformer, the Collection Splitter, and the Database Endpoint. You have to install PostgreSQL on your system.

How to do it...

In this section, you will see how to use the JSON-to-Object transformer in Mule Studio.

1. Open Mule Studio and enter a name for the workspace as shown in the following screenshot:

Y Workspace Launcher	X
Select a workspace	
Mule Studio stores your projects in a folder called a workspace. Choose a workspace folder to use for this session.	
Workspace: E:\MuleCookBook	Browse
Use this as the default and do not ask again	OK Cancel



2. To create a new project, go to **File** | **New** | **Mule Project**. Enter the project name as JSON and click on **Next** and then on **Finish**. Your new project is created and now you have to start the implementation.


3. Go to the Json.mflow file, navigate to the **Global Elements** tab, and click on the **Create** button. Go to **Data Sources** | **PostgreSQL Data Sources**. In the **URL:** textbox, enter the database name that was created in PostgreSQL. Here, we entered Test. Enter the PostgreSQL username and password and click on the **OK** button.

∑ Mule - json/flows/Json.mflow	🖌 Global Element Properties	- - X
File Edit Navigate Search Project	PostgreSQL Data Source Configure PostgreSQL Data Source	
Mule Package E Type Hiera Package E Type Hiera Package E	General Advanced Documentation General Name: PostgresQL_Data_Source Connection URL: jdbc:postgresgl://localhost:5432/Test User: postgres Password: ••••	Create Edit Delete
		 i a 🖹 a 🗟 🖬 🗖

-204-

4. Click on the Create button again and go to Connectors | Database. Go to the Queries tab and enter the query key InsertRecord and the query INSERT INTO tblemployee(empid,empname,empage,empdesigna tion) VALUES(#[message.payload.empid],#[message.payload.empname],#[message.payload.empage],#[message.payload.empdesignation]). After that, go to the General tab, change the database name, and select the data-specific name in that box. Click on the OK button.

Y Mule - json/flows/Json.mflow	🖌 Global Element Properties		- 2 🛛
File Edit Navigate Search Project C • • • • • • • • • • • • • • • • • • •	Database Connector configuration for JDBC endpoints. General Advanced Properties Reconnection Queries Documentation JDBC Queries Defines a set of queries. Each query has a key and a value (SQL statement). Queries are later	r referenced by key.	
Allrouter Allrouter Allrouter	🕂 🖉 🛤		Create
choice_router	Query Key Query	Query Reference	Edit
Corn-expression Corn-expressi	INSERT INTO tblemployee(empid, empname, empage, empdesignation) VALUES InsertRecord (#[message.payload.empid], #[message.payload.empidesignation])		Delete
	0	OK Cancel	
	Messane Flow Gobal Flements Configuration XM		
÷ =+			: a 🔊 @ 🕼 🕫 🖻

5. Go to **Message Flow**. Drag the **File** Endpoint onto the canvas and then double-click on it. Here, you will have to enter the path of the JSON file.

🞽 Mule - json/flows/Json.mflow	🖌 Endpoint Properties 🛛 🛛 🔀	🗖 🗖 🗖 🔀
File Edit Navigate Search Project : 🗂 • 🔛 🔞 🗁 : 券 • 🕥 • :: 🎦 Mule	File (Inbound Endpoint) The File Endpoint allows Mule applications to read or write files on the local file system.	
Package E Package E Type Hiera Package E Package E	General Advanced References Documentation Display Display Name: File Path Information Path: C:Documents and Settings\azadesa\Desktop\Input Move to Pattern: Move to Directory: Polling Information Polling Information Polling Information Polling Information Polling Information Polling Frequency: File Age: File Name Filter File Name Regex Filter: @ OK	Fiter: json Select Suggestions (*) 25 DataMapper Transformers (*) 10 Diject to JSON
	Message Flow Global Elements Configuration XML	i = 🖹 @ 📐 M 📮

-206-

- Wule json/flows/Json.mflow Mule Studio File Edit Navigate Search Project Run Window Help _ 7 🛛 i 📬 • 🖫 🗟 🗄 i 🏇 • 🗘 • 隆 🐨 • i 🤔 🖋 • i 🚥 i 🖉 - i 🖓 - v - i 🕫 🛠 🗴 🖹 🔛 Mule ቹ Package E 🕺 🍃 Type Hiera 🗖 🗖 🎽 *Json 🔀 - -■ 🕏 🌣 Filter: json ajex
 ajex
 alrouter
 catchexcaption
 choice_router ^ l> Select 🚖 Suggestions... B debite_router
 B debite_router 2/1 DataMapper a Transformers JSON to Object ••• Object to JSON flow: JsonFlow1 📝 FILE ••• > File JSON to Object setestinesources
 Sete src
 mule-project.xml > < E Outline 🔀 **•** Message Flow Global Elements Configuration XML ∎⇔ 📄 mule/flow/#text 8 i e 🔝 @ 🔯 🖊 📮
- 6. To convert the JSON format to object format, drag the JSON to Object transformer.

7. Drag the **Collection Splitter** onto the canvas, which will transform the JSON data into Java.Util.List, and then divide it into several Java.Util.Maps items. Finally, we will write it to the database.





8. To configure the database, drag the **Database** Endpoint onto the canvas. We can now configure the **Database** Endpoint.



9. For configuring the **Database** Endpoint, click on the **References** tab and select the connector reference name. Select JSON-file1 and then click on the **General** tab and select the query key that was created in the database connector.

😸 Mule - json/flows/Json.mflow	Y Endpoint Properties	
Mule - json/flows/Json.mflow File Edit Navigate Search Project Project Project Package E 23 Type Hera Package E 23 Type Hera C S alex C C C C C C C C C C C C C			Filter: json Select Suggestions 00 Jog DataMapper Transformers 00 Transformers 00 Transformers 00 Transformers 00 Transformers 00
bellomule jon jon JPE System Library (JavaSE Mule Runtime (Mule Server Strc/main/java Strc/main/resources Strc/test/java Strc/te	Transformers References: Response Global Transformers: Transformers to be applied:		
	OK Cancel	1	
: •	Message Flow Global Elements Configuration XML		: a 🖹 a 🗟 🕫 🖻

__210_

10. To import a JAR file for the PostgreSQL database, right-click on your project and click on **Properties**. Here, you can import the jdbc connector JAR file to PostgreSQL.

Mule - json/flows/Json	nerties for ison			
Prop Prop <th>ter text source diders rabuild Path ra Code Style radec Location rgect References ryDebug Settings ryer ski Tags idation</th> <th>Java Build Path Source Projects Libraries Order and Export JARs and class folders on the build path:</th> <th>Add JARs Add External JARs Add External JARs Add Uariable Add Class Folder Add Class Folder Add External Class Folder Edit Remove Migrate JAR File</th> <th>Filter: json Select Suggestions Co Solutional Transformers Co Transformers Co Cobject to JSON</th>	ter text source diders rabuild Path ra Code Style radec Location rgect References ryDebug Settings ryer ski Tags idation	Java Build Path Source Projects Libraries Order and Export JARs and class folders on the build path:	Add JARs Add External JARs Add External JARs Add Uariable Add Class Folder Add Class Folder Add External Class Folder Edit Remove Migrate JAR File	Filter: json Select Suggestions Co Solutional Transformers Co Transformers Co Cobject to JSON
	Message Flow Glo	bal Elements Configuration XML		
i 🔤 🔶 json		&		е 🖹 @ 🕒 И 旦

How it works...

In this section, you will learn how to deploy your application using Mule Studio. After deploying the application, here's how to run this example:

1. To deploy the application code in the Mule server, go to **Run As** | **Mule Application**. The Mule server will deploy your application.



_212

2. In the console output window, you can see that four rows are updated in the database, as seen in the following screenshot:

🎽 Mule - json/flows/Json.mflow - Mule Studio 📃 🗗 🔀
File Edit Navigate Search Project Run Window Help
: 🗂 • 🖫 🗟 🖻 : 恭 • 🕖 • 💁 • : 🕮 🞯 • : : 😕 🖋 • : : 🚥 : : ½ - : : : : : : : : : : : : : : : :
🗈 💟 Mule
🖻 Console 🛛 🔰 🖷 🖓 📓 🖉 🖉 🛃 🖓 👘 🖉 🖓 👘 👘
Json [Mule Application] C. Program Files Java)re6 (bin)tavaw.exe (Mar 7, 2013 2:43:44 PM)
=.launcher.application.DefaultMuleApplication: Reload interval: 3000
≥.management.agent.WrapperManagerAgent: This JVM hasn't been launched by the wrapper, the agent will not run.
=.management.agent.JmxAgent: Attempting to register service with name: Mule.json:type=Endpoint,service="JsonFlow1",connector
E.management.agent.JmxAgent: Registered Endpoint Service with name: Mule.json:type=Endpoint,service="JsonFlow1",connector=co
:.management.agent.JmxAgent: Registered Connector Service with name Mule.json:type=Connector,name="connector.file.mule.defau
E.management.agent.JmxAgent: Registered Connector Service with name Mule.json:type=Connector,name="Json.file1"
ltMuleContext:

*
*
*
*
*
3. Jauncher, JeploymentService:

*
TTTTTTTTT - www.default wareiner Oll and wule trenewart file FileWearene Desciner, Leek abteined on file, C.\Desuwarts and Sattings\ar
- mute default receiver.olj org.mule transportine riters sagekeeleer book obtained on file: Cidocuments and securings as
spatcher.olj org.mule intervite.abstractifervitemanager. interatisnig. oson-inter.dispatcher.istatioo. object is. idoomeenses matcher oli org.mule lifervile Matractifervitemanager. Statismi, Idoo-file dispatcher 24.21251. Object is. IdooMeessesse
spatcher.01 org.mule.inferguie.abstracturieggiennanger. Statchny. Usun-file.inferdenet.station. Goject is. Suddenessager statcher.01 org.mule liferunie NetrartiferunieMenager. Thitaliens, Jonn-file idenstrier 315640141 Ohiersie.JohoMes
spatcher 04) org mule liferurle BetractliferurleManarar. Starting. Jaco-Filitararder 3154014. Object 16. JobeMessare
spaceher.021 org.mule_lifecycle_bstractlifecycleManager: Initialising, John file1.disatcher.19888721. Object is Johnsburgs
matcher.03] org.mule.lifecycle.bstractLifecycleManager: Initialising: 'Json-filel.dismatcher.23578365', Object is: JdbcMes
spatcher.02] org.mule.lifecvcle.AbstractLifecvcleManager: Starting: 'Json-file1.dispatcher.1989872'. Obiet is: JdbcMessageD
spatcher.03] org.mule.lifecycle.AbstractLifecycleManager: Starting: 'Json-file1.dispatcher.23578365'. Object is: JdbcMessage
spatcher.01] org.mule.transport.jdbc.sqlstrategy.SimpleUpdateSqlStatementStrategy: Executing SQL statement: 1 row(s) updated
spatcher.03] org.mule.transport.jdbc.sqlstrategy.SimpleUpdateSqlStatementStrategy: Executing SQL statement: 1 row(s) updated
spatcher.02] org.mule.transport.jdbc.sqlstrategy.SimpleUpdateSqlStatementStrategy: Executing SQL statement: 1 row(s) updated
spatcher.04] org.mule.transport.jdbc.sqlstrategy.SimpleUpdateSqlStatementStrategy: Executing SQL statement: 1 row(s) updated
×
: r 🕐 🖉 😥 🖉 😥 😥 😥 😥 😥 😥

3. Open the PostgreSQL database and have a look at the tblemployee table. You will see that all the data has been inserted into this table:

		1	· · · · ·	,
File Edit	: View Tools	Help		
: 🔳 🛔	9 🧑 🗎	🔒 🛡 🍸	🛛 💡 🚦 No limit	t 😽
	empid text	empname text	empage text	empdesignation text
1	01	AzazDesai	25	Liferay/Mule Developer
2	03	Mohib Mansu	25	Web Designer
3	02	Uchit Vyas	25	Cloud Admin
4	04	Tapan	27	Liferay/Alfeshco Developer

Configuring the Object-to-XML transformer

The **Object-to-XML** transformer is used to transform a Java object to XML data. In this example, you will see how to retrieve data from the database and how to store data in a particular XML file through the Object-to-XML transformer. Here, you will use the table from the previous example. We will use three components: the Database Endpoint, the Object-to-XML transformer, and the File Outbound Endpoint.

Getting ready

In this section, you will see how to configure the Database Endpoint in Mule Studio.

1. Open Mule Studio and enter a name for the workspace as shown in the following screenshot:

🖌 Workspa	ace Launcher	
Select a w	orkspace	
Mule Studio Choose a wo	stores your projects in a folder called a workspace. orkspace folder to use for this session.	
Workspace:	E:\MuleCookBook	Browse
Use this a	s the default and do not ask again	OK Cancel

2. To create a new project, go to File | New | Mule Project. Enter the project name as Object-to-xml and click on Next and then on Finish. Your new project is created. You can now start implementing it.



How to do it...

In this section, you will learn how to configure the Object-to-XML transformer and the File Outbound Endpoint.

 Navigate to the Object-to-xml.mflow file. To configure the database, go to the Global Elements tab, click on the Create button, and go to Data Sources | PostgreSQL Data Sources. In the URL: textbox, enter the name of the database that was created in PostgreSQL. Here we enter Test. Enter the username and password for PostgreSQL and click on the OK button.

Y Mule - object-to-xml/flows/0	Global Element Properties		🗙 🗖 🔁 🔀
File Edit Navigate Search Project : □3 •	ostgreSQL Data Source Configure PostgreSQL Data Source		
Mule Package E Type Hiera Con-expression C	Configure PostgreSQL Data Source General Advanced Documentation Generic Name: PostgreSQL_Data_Source Connection URL: jdbc:postgresgl://localhost:5432 User: postgres Password: ••••	i/Test	Create Edit Delete
€ →=→ =	Message Flow Global Elements Config	guration XML	»
÷ •			: e 💦 @ 🔍 M 🗉

2. Click on the **Create** button and go to **Connectors** | **Database**. Go to the **Queries** tab, enter the query key name Update List and the query SELECT * FROM tblemployee. After this, go to the **General** tab, change the database name, and select a data-specific name in the box. Click on the **OK** button.

⊻ Mule - object-to-xml/flows/0	Y Global Element Properties	X	🗖 🖬 🔁
File Edit Navigate Search Project	Database Connector configuration for JDBC endpoints.		
Package E Type Hera Connexpression Connexpression	General Advanced Properties Reconnection Queries Defines a set of queries. Each query has a key and a value (SQL statement). Queries are later referenced by key. Image: Comparison of the set of query is a comparison of query is a compa		Create Edt Delete
: =0	Global Elements Configuration XML		

3. Go to the **Message Flow** tab, drag the **Database** Endpoint onto the canvas, and configure it. Click on the **References** tab and select the connector-ref name. Here, select the **JDBC** connector. Click on the **General** tab and select the query key that was created in the **Database** connector.

⊻ Mule - object-to-xml/flows/0	Y Endpoint Properties		🔳 🗗 🗾
File Edit Navigate Search Project	Database (Inbound Endpoint)		
i 📫 • 🔛 💼 🖆 i 🏇 • 🔘 •	The JDBC Endpoint allows Mule applications to read and write to databases which support a JDBC driver.		
😭 🔛 Mule			
洋 Package E 🛛 🍃 Type Hiera	General Advanced References Queries Documentation		- 8
	GlobalReferences		Filter:
🗈 😂 filetransfer	Connector Reference: JDBC-Connection 🗸 🗸 🖉		
🗈 😂 filter	Endpoint Beference		Select
			Carl Endpoints 🗠
ametransfer	Traisforniers References: Request		🔎 Ajax
😑 😂 object-to-xml	Global Transformers:		🕞 Database
🗊 🛋 JRE System Library [JavaSt	Transformers to be applied:		FTP
Mule Runtime [Mule Server			(👝 File
- 🕮 src/test/java			Generic
- 🚈 src/main/resources			ПНТТР
Beferenced Libraries			IMAP
Green and an and a set of the set of th			ms Ims
Object-to-xml.mflow	Transformers References: Response		letty
- mappings	Global Transformers: 🔍 💌 💆		
mule-project.xml	Transformers to be applied:		
🗈 😂 quartz			
😐 😂 quartz_scheduler			RML
restbasedwebservice			Redis (Streaming)
I Snellcheck			SF1P
			Copes Scopes
🗄 Outline 🛛			Components
	(?) OK Cancel		a Transformers
		_	Pilters
			Flow Control
			Crror Handling
			Cloud Connectors
	Message Flow Global Elements Configuration XML		
•			e 🖹 @ 💽 M 📮

-218-

4. To import a JAR file for the PostgreSQL database, right-click on your project and go to **Properties**. Here, you have to import the PostgreSQL jdbc connector JAR file.



5. To convert an object to XML format, drag the **Object to XML** transformer onto the canvas. The transformer will create a map with the result of the query and send the map in a message. You can now use this map and convert it to XML.



6. To store data in a file, drag the **File** Endpoint onto the canvas and configure it. Double-click on the **File** Endpoint. Firstly, select the storage file path in the following output pattern: Database#[function:datestamp].xml (Database represents a filename and datestamp represents the current date and time).

Y Mule - object-to-xml/flows/0	🖌 Endpoint Prope	rties		🗖 🗖 🔁
File Edit Navigate Search Project	File (Outbound En The File Endpoint allow	dpoint) ws Mule applications to read or write files on the local file system.		
😫 Package E 🕺 🔋 Type Hiera	General Advanced	References Documentation		
allrouter	Display Display Name: File			Filter:
Choice_router	Path:	C:\Documents and Settings\azazdesai\Desktop\Input		Carl Endpoints 🛛 🔅
filer	Output Pattern:			E Dacabase
i i i i i i i i i i i i i i i i i i i	l l			
Object-to-xml Object-t				
				Async
				Flow C Foreach
Object-to-xml.mflow propings				Message Enricher
E Outline X				Components
	?	ОК	Cancel	Carteria Transformers
				Control
	Message Flo	Global Elements Configuration XML		Cloud Connectors
: •				i a 🖹 @ 🔃 H 📮

How it works...

In this section, you will learn how to deploy the application in Mule Studio and how to run the application in the browser after deploying it.

1. To deploy the application code in the Mule server, go to **Run As** | **Mule Application**; the Mule server will deploy your application.



2. You can see in the console output that the file has been created in the mentioned path with the name and current date/time, as seen in the following screenshot:



Configuring the Message and Variable transformers

Message properties and variables are the most frequently used features in Mule. Message properties have mainly four scopes: **invocation**, **outbound**, **session**, and **application**. A Mule message consists of three parts: header, payload, and attachments.

223

Getting ready

You can configure the Message Properties transformer as shown in the following screenshot. A Mule message supports two types of properties: **inbound properties** and **outbound properties**.

🖌 Mule - variable-transformer/	Y Pattern Properties	- - X
File Edit Navigate Search Project	Massage Properties	
i 📬 • 🗐 🕼 🖻 i 🎄 • 🔘 •	The Manuary Dispertition Transformer and delate an annual second se	
🖹 📔 Mule	The message properties Transformer can ado, delete, or rename message properties.	
😫 Package E 🐰 🔭 Type Hiera	General Advanced Designmentation	- 8
n 🗇 skiest te suit	Display -	Filter:
Diject-co-xim	Display Name: Message Properties	Select
guartz scheduler	General	Cardeninka
🗉 😂 restbasedwebservice		
🗉 🔛 restwebservice		Scopes
🗈 😂 spellcheck	Scope: e session	Components
🗈 📅 spring	invocation	a Transformers 🛛 🗠
testtransformer	session	••= Object to JSON
transformer	💁 🖉 application	Object to
twitterintegration	Key Value Value Reference	JmsMessage
avariable-transformer	🕅 property-key azazdesai	Object to MIME
😟 🛋 JRE System Library [JavaSt		Object to String
🗉 🛋 Mule Runtime [Mule Server		Object to XMI
💷 🥭 src/main/java		Collect to And
🖃 🌐 com.org		Parse Template
myclass.java		Property
src/resc/java		Python
src/test/resources		Ruby
E 🗁 flows	Delete Message Properties	Script
Variable-Transformer.m		Serializable to Byte
<		Array
E Outline 🛛	кеу	Session Variable
		Set Payload
	OK Cancel	Pilters
		General Flow Control
		🚰 Error Handling
		Cloud Connectors
	Message Flow Global Elements Configuration XML	
: •	······	i a 🖹 @ 💁 🖊 🖻

How to do it...

In this section you will learn how to add session properties into the Mule config.xml file and how to use the session scope in the Mule configuration file.

How to add session properties

Drag the properties of the transformer onto the canvas. To configure, double-click on the **Message Properties** transformer.

```
<message-properties-transformer scope="session">
     <add-message-property key="property-key" value="property-value"/>
</message-properties-transformer>
```

You can also trigger a Mule session using MuleEventContext.

```
public class MyClass implements Callable
{
    public Object onCall (MuleEventContext eventContext) throws
Exception
    {
        eventContext.getSession().setProperty("property-key","some
value");
        return eventContext.getMessage();
    }
}
```

A Mule session is maintained while the message goes through Mule instances, so don't store things you don't need, to avoid wasting memory.

How it works...

In this section, you will learn what a **Variable** transformer is and how to use it in Mule Studio. You will see how to set variables in Mule Studio.



The Variable transformer

The Variable transformer allows activation of variables. To activate the variable, first drag the **Variable** transformer onto the canvas. Double-click on the **Variable** transformer. You can declare a variable as an expression or as a literal. If you declare the variable as an expression, Mule evaluates the variable against the content of the current message.

😸 Mule - variable-transformer/	Y Pattern Properties	
File Edit Navigate Search Project	Variable & Attribute 'variableName' is required	
Package E 🕺 🍃 Type Hiera	General Documentation	
Package E	General Documentation Display Name: Variable Settings Operation: ③ Set Variable Remove Variable Name: ④ #[] Value: ● #[] Output Ø OK Carcel	Fiter: Select Select Secopes Components Components Property Python Python Python Python Session Variable Session Variable Session Variable Session Variable Session Variable Session Variable Session Variable Session Variable String to Byte Array String to Byte Array String to Email Transformer Reference Reference String to Dyte Array String to Email Transformer Reference String to Dyte Array String to Email Transformer Reference String to Dyte Array String to Email
		Error Handling
	Message Elow Global Elements Configuration XMI	Cloud Connectors
: •	hessage now jacobal clements Conniguration Amb	

In the **Name:** textbox, specify the variable name as a string. In the **Value:** textbox, type a string that specifies the variable value, either in expression or literal, depending on your choice.

Creating the custom transformer

A **custom transformer** is a user-defined transformer class that implements the org. mule.api.transformer transformer. Your class can extend AbstractTransformer or AbstractMessageAwareTransformer, depending on your requirements.

226

Getting ready

In this section, you will learn what a custom transformer is and how to configure it.

1. Open Mule Studio and enter a name for the workspace as shown in the following screenshot:

Y Workspace Launcher	×
Select a workspace	
Mule Studio stores your projects in a folder called a workspace. Choose a workspace folder to use for this session.	
Workspace: E:\MuleCookBook	Browse
Use this as the default and do not ask again	
	OK Cancel

2. To create a new project, go to **File** | **New** | **Mule Project**. Enter the project name as NameTransfer and click on **Next** and then on **Finish**. Your new project is created. You can now start the implementation.

le Edit Source Refactor Navigate	Search Project	Run Window Help			
New	Alt+Shift+N	Mule Project		\$P (P = P) =	
Open File		/ Java Project			
Close	Ctrl+W	Project			
Close All	Ctrl+Shift+W	📝 Mapping Flow			
Save	Ctrl+S	Mule Flow			
🗂 Save As		🖶 Package			
🖹 Save All	Ctrl+Shift+S	🞯 Class			
Revert		🚯 Interface			
Move		🚯 Enum			
Rename	F2	Annotation			
👔 Refresh	F5	Source Folder			
Convert Line Delimiters To		 Inva Working Set Inva Working Set 			
Print	Ctrl+P	Folder			
Switch Workspace		Intitled Text File			
Restart					
N= Import		Example			
A Export		📑 Other	Ctrl+N		
Export diagram to		-			
Properties	Alt+Enter	-			
1 FlowRef.mflow [flowref/flows] 2 package.html [activity/src/main/] 3 Activity.mflow [activity/flows] 4 StringToNumber.java [usdconverter/.]	_			
Exit		-			
er oukling is the aralised					

How to do it...

In this section, you will use four components: the HTTP Endpoint, the custom transformer, the Java component, and the Append String transformer. Also, you will learn how to create a custom transformer and a custom Java component in Mule Studio.

1. Right-click on src/main/java and go to **New | Class**. Enter the package name as com.org and the class name as Greeting; click on the **Finish** button.

Mule - corn-expression/flows/Corn-Exp	Y New Java Class		
File Edit Source Refactor Navigate Search	Java Class		
i 📬 • 🔛 🗟 📄 i 🏇 • 🕥 • 💁 • i	Create a new Java class.		
😭 🕎 Mule		9	
📱 Package E 🕺 🐮 Type Hiera			
	Source folder: nametransfer/src/main/java	Browse	et l
	Package: com.org	Browse	Filter:
til alrouter	Enclosing type:	Browice	Select
🗄 🔁 catchexcaption			Endpoints (
🗉 😂 choice_router			
🕀 🗁 corn-expression	Name: Greetings		Ajax
🗄 📂 cxt	Modifiers: Opublic Odefault Oprivate Oprotected		😇 Database
file_transfer	🔄 abstract 🔄 final 🔄 static		😑 FTP
	Superclass: java.lang.Object	Browse	(==) File
🗄 🔁 hellomule	Interfaces:		Generic
😟 🔛 json		<u>Haa</u>	нттр
🛱 😂 nametransfer			
Image: Barrier Barr		Remove	
H Aule Runtime [Mule Server 3.3.0 C	Which method stubs would you like to create?	-	🚰 Scopes 🛛 🗠
src/main/java	public static void main(String[] args)		Async
- # src/main/resources	Constructors from superclass		Composite Source
- 🥭 src/test/resources			Elow
😑 🧁 flows	Do you want to add comments? (Configure templates and default value bere)		C Smarth
····₩ NameTransfer.mflow	Generate comments		C Poreach
- mappings			Message Enricher
mule-project ym			C Poll
<			Sub flow
🗄 Outline 🛛 🔍 🖓 🗖 🗖			Components
	Enis		arransformers
			Pilters
			Control
			Crror Handling
			Cloud Connectors
	Message Flow Global Elements Configuration XML		
src/main/java - nametransfer	8		a 🖹 🔍 😡 🖬 🚍

2. Here, you can create a simple sayHi method; the return type is set to String.

```
package com.org;
public class Greeting {
  public String sayHi(String str)
    {
      return "Hello "+str;
    }
}
```

3. In the same package, you had created another class called NameTransfer. This class is called custom transfer, and through this class you will remove the extra forward slash. Here you can extend AbstractTransformer.

```
package com.org;
import org.mule.api.transformer.TransformerException;
import org.mule.transformer.AbstractTransformer;
public class NameTransfer extends AbstractTransformer{
@Override
  protected Object doTransform(Object src, String enc)
      throws TransformerException {
    if(src instanceof String)
    {
      String name=((String)src);
      {
    if (name.charAt(0) = = '/')
       {
         name.substring(1);
       }
      }
    }
    return src;
  }
}
```

229

4. Go to the NameTransfer.mflow file. To configure the localhost URL, drag the **HTTP** Endpoint onto the canvas. Double-click on the **HTTP** Endpoint. Enter the port number and hostname and click on the **OK** button.

⊻ Mule - nametransfer/flows/N	Y Endpoint Properties	🗖 🗖 🔁
File Edit Navigate Search Project	HTTP (Inbound Endpoint) The HTTP Endpoint allows Mule application to connect to web resources through the HTTP transport protocol. This endpoint can also implement security through HTTPS.	
Mule Package E Autor Package E Autor Auto	This endpoint can also implement security through HTPS.	Image: Solution of the solution
: =0	Message Flow Global Elements Configuration XML	
÷ 0*	81	: 8 🕺 @ 😣 🖊 📮



5. To configure the custom class, drag the **Java** transformer onto the canvas. Doubleclick on the **Java** transformer. Here, you need to import the NameTransformer class that was created earlier.



6. Drag the **Java** component onto the canvas. To configure it, double-click on the **Java** component. Here, you import the Greeting class that was created earlier.



7. Drag the **Append String** transformer onto the canvas. To configure, double-click on it. In the **Message** textbox, enter the value ESB and click on the **OK** button.



How it works...

In this section, you will learn how to deploy the application in Mule Studio and how to run this application in the browser after deployment.

1. To deploy the application code in the Mule server, go to **Run As** | **Mule Application**; the Mule server will deploy your application.



2. Copy the localhost URL http://localhost:5454. Open the browser, paste the URL, and type in Mule. You can see that the forward slash has been removed. The custom transformer removed the forward slash.

🗅 localhost:5454/Mule 🛛 🗙 🔽		x
← → C C localhost:5454/Mule	☆ 🧧	≡
Hello MuleESB		

Understanding the DataMapper transformer

The **DataMapper** transformer is more powerful and flexible as compared to the rest of the transformers that are provided with Studio. Like other transformers, DataMapper can transform data across formats and manipulate the payload values as well. DataMapper can map an input field to a different output field.

Getting ready

The DataMapper transformer works only in the Mule Studio Enterprise edition. DataMapper supports only six formats: CSV, XML, Java, Map, JSON, and Excel.

1. Open Mule Studio and enter a name for the workspace as shown in the following screenshot:

Workspace Launcher	X
Select a workspace	
Mule Studio stores your projects in a folder called a workspace. Choose a workspace folder to use for this session.	
Workspace: E:MuleCookBook	Browse
Use this as the default and do not ask again	



2. To create a new project, go to **File** | **New** | **Mule Project**. Enter the project name as Mapper and click on **Next** and then on **Finish**. Your new project is created. You will now have to start the implementation.



-236

How to do it...

In this section, you will learn what is DataMapper, how to configure the DataMapper in Mule Studio, and when to use DataMapper in Mule Studio.

In this example, you will use three components: the File Inbound Endpoint, DataMapper, and Database.

1. Go to the Mapper.mflow file and drag the **File** Inbound Endpoint onto the canvas. To configure, double-click on the **File** Endpoint. Here, you import the CSV file.

⊻ Mule - mapper/flows/Mapper	🗑 Endpoint Properti	es		- - X
File Edit Navigate Search Project	File (Inbound Endpoi The File Endpoint allows I	nt) Iule applications to read or write files on the local file system.		
Project Explorer 23 detamapper detamapper detamapper mapper m mapper m m m m m m m mapper m	General Advanced Re Display Name: File Path Information Path: Move to Pattern: Move to Directory: Poling Information Poling Frequency: File Age: File Name Regex Filte	ferences Documentation C:\Documents and Settings\azazdesa\Desktop\Input C:\Documents and Settings\azazdesa\Desktop\Input I000 C: C:C:C:C:C:C:C:C:C:C:C:C:C:C:C:C:	····	Filter: da Select Suggestions (C) C DataMapper Database Database Database Transformers (C) C DataMapper
1 D*		R 1	i # 🖹 @	0 🖬 🗄 🖶 🗎 🖉 🖉

This is how the sample-output.csv file looks:

C :		17	• (° •)	Ŧ				sam	ole-output -	Microsoft E	xcel					-	•	x
	Угн	lome	Insert	Page La	yout Fo	rmulas (Data R	eview	View							0 -		x
ľ	*		Calibri	* 11	A A	= = =	≫~~		General	*	S		J 🕻	Insert •	Σ·Α	A		
Pa	ste 🧹	,	BIU	• 🖽 • 🗸	<u>> A</u> -			•4• •	\$ • %	, •.0 .00 .00 ⇒.0	Conditional Formatting *	Format as Table * S	Cell tyles -	Format *	Sort &	Find & Select *		
Clip	board			Font	1	Alig	nment	5	Numb	er 👒		Styles		Cells	Editin	g		
	4	41	•	• (•	<i>f</i> ∞ id													×
	A		В	С	D	E	F	G	н	1	J	К	L	M	N	0		
1	id		name	age	designatio	on												
2		1	azaz	25	Softeware	e Engineer												
3																		
4																		
5																		

2. To configure mapping, drag the **DataMapper** transformer onto the canvas and double-click on it.



-238

3. Here, you configure the **DataMapper** transformer and enter the name CSV 2 Map in the **Name** textbox. The filename .grf automatically appears in the **File name** textbox. The .grf extension file is used for the attributes that you map. Click on the **Next** button.

Y Mule - mapper-book/flows/Mapper-Book.mf	Y New Data Mapping Flow.	
File Edit Navigate Search Project Run Window : □ • □: □: □: □: □: □: □: □: □: □: □: □: □:	Data Mapping Flow Create a data mapping flow.	1
Mule Project Explorer 23 Project Explorer 23 datamapper datamape	General Information Name: CSV 2 MAP File name: csv_2_map.grf Description:	Filer: data
1 D*	8 N N N N N N N N N N N N N N N N N N N	: P 👗 @ 🖳 😫 : P 🔚 : P 🚺
Transformers -

4. The DataMapper feature requires three files to hold the configuration information: the .grf file, the input type, and the output type. For the input type, you have to select the CSV type from the drop-down list. Click on the ellipsis (...) button to the right and choose the .csv file. In the output pane, use the drop-down list in the Type: field and select Maps/List of Maps. Note that you cannot specify a sample file for the Maps format. Click on the Edit Fields button to the right of the Name: field in the Output pane.

Mule - mapper-book/flows/Mapper-Book.mf	🖌 New Data Mapping Flow.	
File Edit Navigate Search Project Run Window	Select Input and Output Type	
<u>□ </u>		
📫 🔛 Mule		
Project Explorer 🛛 🖓 M	Territ	- 8
datamapper dena dena dena mapper mapper-book map	Type: SV Type: SV CSV example: C:\Documents and Settings\azazdesai\Desktop\Input\sar Name sample_output Output Type: Maas / Liet of Maas	Filter: data
Image: Image: State of the s	Type: Traps / List of Praps	
C Flows Box Mapper-Book.mflow C mappings B C src	Name outputMetadata Edit Fields Generate default	iranstormers 💿
M Data Mapping X	The Key/Values needs to be specified by hand	
	Image: Control of the sector over and any per component operations in the sector over and any per component operations in the sector over any per component operation in the sector over any per cover any per component operation in the sector	w file.
i •	😢 🗄	i a 🖹 @ 🔕 🗎 i a 📰 i a 🚺

5. In the **Edit Map Fields** window, click on the plus 🔂 button. Here, you enter the same column name that was entered in the database. Click on **OK** to finish the **DataMapper** wizard.



Transformers -

6. After you click on the **Finish** button, you will see a screen similar to the one shown in the following screenshot. The area on the left-hand side of the screen is for the sample.csv file attribute and the right-hand side of the screen area is for the DataMapper attribute.



7. Click on **id: string** and drag it on top of **empid: string**. The solid black line indicates mapping.

11 Data Mapping 🛛	al 🖓 🕸 🗖 🗖 🗖
Input 🗣 🐨 type filter text	Current Element Mapping sample_output_to_s * sample_output_to_s * *
57 Mappings O Preview	
i D°	😥 i e 🔛 🤅 e 🔛 i e 🔛 i e 🕨

8. Drag all the attributes to be mapped onto the right-hand side. The final output looks like the following screenshot:

III Data Mapping 🛛		al - 🗆 🖓 🖗
Input 🌵 🖉 Eype filter text	Current Element Mapping sample_output_to_s 🗸 🕂 🥢 🗙	Output 🖉 🍪 type filter text
 a, sample_output d: string anme: string age: string designation: string Input arguments Lookup Tables 		kry sample_output @ enpid: string @ enponene : string @ enponene : string @ enpodesignation : string
14 Mappings O Preview		 Orag an Input attribute to an Output attribute to assign/concatenate. Orag an Input element list to an Output one to map their attributes.
[] →	 Mule - mapper-book/flows/Map	oper-Book.mflow - Mule Studio 🖉 🔝 🥥 😡 🗉 📜 🖉 🛃

9. Click on **Preview**, and then click on **Run Mapping** on the right-hand side of your screen.

Data Mapping 23		
Input sample data: C:/Documents and Settings/azazdesai/Deskt	pp/Input/sample-output.csv	🚺 Run Mapping Input Arguments
Text Graphic View		Run Graph csv. 2 map.grf
		>
Mappings O Preview		
i □•	6	i e 🖹 e 🔃 i e 📰 i e 🕅



Transformers -

You will see the result being displayed on the screen. One really attractive feature is the possibility to test your mappings without the need to launch your Mule application.

🕼 Data Mapping 🛛	4 🗢 🗖 🗖
Input sample data: C:/Documents and Settings/azazdesai/Desktop/Input/sample-output.csv	💽 Run Mapping Input Arguments
Text Graphic Wew [[
11 Mappings O Preview	

10. Click on the **Global Elements** tab. Here, you have to configure the database that we have already seen in the previous example.

Y Mule - mapper-book/flows/Mapper-Boo	ok.mflow - Mule Studio				- 7
File Edit Navigate Search Project Run Win	ndow Help				
i 📬 • 🔛 🕼 🗁 i 🏇 • 🕥 • 💁 • i	瞪 🞯 • 🗄 🥭 🛷 • 🗄 🖑	🍤 🗶 🗄 🗰 🗄 🖢 - 🖓	🌾 ⇔ - ⇔ -		
E Mule					
Project Explorer 🛛 🗖 🗖	Mapper Mapper-Boo	k 🛿			
□ 🕏 🏹	🗑 Global Mule Confi	guration Elements			
🗈 🗁 datamapper	-	.			
🗈 🔛 demo	Туре	Name	Description		Create
🗄 🗁 Idap	DataMapper	CSV_2_MAP			
mapper	PostgreSQL Data Source	PostgreSQL_Data_Source			Edit
The System Library []avaSE-1 6]	Database	esv to map			Delete
Mule Runtime [Mule Server 3.3.1 EE]					
🗉 🕮 src/main/java					
🗷 🇀 src/test/java					
Image:					
Image: Strate					
postgresql-9.2-1002.jdbc4.jar - C:\Doi					
Hows					
Mapper-Book.mriow					
The process of the pr					
mule-project.xml					
Webservices					
<	Message Flow Global Elements Co	onfiguration XML			
		a :	1		
: U				" 🔟 🔍 🖳 📄	

-244

Your final flow should look like the following screenshot:



Transformers -

How it works...

In this section, you will learn how to deploy the application in Mule Studio and how it works.

1. To deploy the application code in the Mule server, go to **Run As** | **Mule Application**; the Mule server will deploy your application.

Mule - mapper-book/flows/Mapper-Book.mflow - Mule Studio	- 7
File Edit Navigate Search Project Run Window Help	
😰 🕎 Mule	
Project Explorer X PD Mapper	
Project Explorer S Mapper **Mapper-Book S	Filter: data
Debug As 2 Mule Application	
Profile As Run Configurations	
Compare With	
Replace With Configuration XML	
Mule Management Console	9 # BE # M

2. Open the PostgreSQL database and check the table entries. You will observe that the CSV data is transferred to the external database.

Properties Statistics Dependencies Dependents Edit Data - Testdb (localhost: 5432) - Test - tblemp File Edit View Tools Help Properties Statistics Dependencies Properties Statistics Dependencies Dependencies Properties Statistics Dependencies Dependencies Displace Displace Displace Displace Properties Statistics Dependencies Displace Properties Displace <th></th> <th></th> <th></th> <th></th> <th></th> <th></th>						
Properties Statistics Dependencies Dependents Edit Data - Testdb (localhost:5432) - Test - tblemp Image: Comparison of text in the state of text in text	SQL		- ېټې 🖉			
Edit Data - Testdb (localhost:5432) - Test - tblemp File Edit View Tools Help Image:	× Pr	operties Sta	tistics Deper	ndencies De	pendents	
File Edit View Tools Help Image: Second Secon	🔲 Edit Data - Testdb (localhost:5432) - Test - tblemp					
Image: Second	File Edit View Tools Help					
empid empname empage empdesignation text	: 🔳 🛛	🦻 🧑 🗎	6 T T	🛛 💡 🕴 No limi	t 🔽	
		empid text	empname text	empage text	empdesignation text	
I azaz 25 java	1	1	azaz	25	java	

In this chapter, we will cover the different types of filters. You will also learn the following:

- ► Configuring the Logic filters And/Or/Not
- Performing filtering according to the exception type
- Filtering messages by evaluating expressions
- Handling incoming events or messages using the Message filter
- Configuring the Wildcard filter
- Creating a Custom filter

Introduction

Filters specify conditions that must be met for a message to be routed to a service. Several standard filters come with Mule where you can create your own filters. You will learn about some filters in this chapter.

Configuring the Logic filters – And/Or/Not

Use the And filter to join two or more filters. The And filter accepts a message and returns true only if all of its enclosed filters return true. The Or filter accepts the message if the message matches the criteria of any of its filters. The Wildcard filter applies a wildcard pattern to the message payload. This filter applies a string to the payload, so you might also want to apply a Payload Type filter to the message using an And filter to make sure the payload is a string.

Getting ready

In this section, you will use three components: the HTTP Endpoint, the Or filter, and the Java component. You will also learn how to configure the Or filter. Perform the following steps to create a new project in Mule Studio:

1. Open Mule Studio and enter the workspace name as shown in following screenshot:

Y Workspace Launcher	×
Select a workspace	
Mule Studio stores your projects in a folder called a workspace. Choose a workspace folder to use for this session.	
Workspace: E:\MuleCookBook	Browse
Use this as the default and do not ask again	
	OK Cancel

2. To create a new project, Go to File | New | Mule Project. Enter the project name Logic Filter, click on Next and then on Finish. Your new project is created now, so you can start with the implementation.



How to do it...

In this section, you will see how to use a Logic filter and how to configure the Java component.

1. Go to the LogicFilter.mflow file and drag the **HTTP** Endpoint onto the canvas. Double-click and configure it. Change the port number, and click on the **OK** button.

∑ Mule - logicfilter/flows/Logic	¥ Endpoint Properties	X	🔳 🗗 🗾
File Edit Navigate Search Project	HTTP (Inbound Endpoint) The HTTP Endpoint allows Mule application to connect to web resources through the HTTP transport protocol. This endpoint can also implement security through HTTPS.		
Package E Arype Hiera Corn-expression Corn-expres	General Advanced References HTTP Settings Documentation Display Display Name: HTTP Exchange Patterns Image: Connection of the set of the		Filter: Select Select Components Components Filters Filters Filters Filters Exception Exception Filter Reference Idempotent Message Message Message Property Not Cor Filter Reference Idempotent Message
	OK Cancel		Schema Validation
			Control
	Message Flow Global Elements Configuration XML		Cloud Connectors
i □ *		. 8	🖁 е 🔒 И 📋 е 📮

2. To create a class, go to src/main/java, right-click on it, and go to New | Class.
Create a class called Greeting under the package com.org; here, we have created
the sayHi method and its return type is set to String.





3. To configure a Logic filter, go to the LogicFilter.mflow file and drag the **Or** filter onto the canvas. Double-click and configure it. Here, you have to select the **Add Wildcard** filter twice.



4. In the **Add Wildcard** filter, you have added a pattern /Hello*. Add another pattern in the second **Add Wildcard** filter, /Hi*, and click on the **Finish** button. This pattern means that the string starts with Hi or Hello; if not, the output will not be displayed on the console.

Y Mule - logicfilter/flows/Logic Pattern Properties	
File Edit Navigate Search Project	
The Or Filter returns true if any of the enclosed filters returns true.	
Package E 🔅 👔 Type Hiera General Documentation	
A C A State of the	Filter:
alrouter The Wildcard Eilter matches string messages against wildcards.	Select
	Endpoints
Corn-ex General	Scopes
the tran	Components
Filetrans Pattern: /Hello*	Filters 🗢
titer	And
	Custom
바 날 json 금 1월 logicfilter	Exception
🗑 🛋 JRE	Expression
Hi 🖹 Mule	Filter Reference
	Idempotent Message
- → sril	Message
- 🥮 sicil	Message Property
- 🥭 src/l	
	Y Payload
	Regex
	Schema Validation
	Wildcard
Finish Cancel	Ger Flow Control
	Crror Handling
Massan Flav, Clabel Flavante Canfin antice VM	Cloud Connectors
i no interse i	

- ¥ Pattern Properties - - X Java 📬 • 🔛 🗟 🗄 🏇 • 🔘 The Java Component is a simple POJO (Plain Old Java Object) that will be invoked by Mule when a message is received. 🔛 🔛 Mule 洋 Package E 🛛 🏦 Type Hiera - -General Advanced Documentation Display Filter: Display Name: Java Select Generic andpoints ... Class Name: com.org.Greeting a Scopes 🕂 🗙 🕯 Object: Components (0+0) Echo Expression Flow Reference Firstwebservice 索 Groovy ison HTTP Response Builder 🛓 🛋 JRE System Library [Java 🔬 Java 🗉 🛋 Mule Runtime [Mule Server Hole Kulture (nue ber Broc/main/java Com.org D Greeting.java Src/test/java Javascript 📑 Logger 🤌 Python src/main/resources S REST 进 src/test/resources 💎 Ruby Flows SOAP < Script 🗄 Outline 🖾 a Transformers ? OK Cancel <mark>è</mark> Filters $\overline{\underline{w}} \to \overline{\underline{w}} \to \overline{\underline{w}}$ Flow Control a Error Handling Cloud Connectors Message Flow Global Elements Configuration XML ∎≎ 8 e 🔝 @ 🕒 M 🕴 e 🗉
- 5. To configure a custom class, you have to drag the **Java** component onto the canvas and import the custom Java class that you created before.

How it works...

In this section, you will see how to deploy the application using Mule Studio, and after deploying the application, you will see how it will run.

 Now you are ready for the deployment. If you haven't saved your application code, please do it. After saving your project, right-click on the LogicFilter.mflow file and go to Run As | Mule Application.



2. If your application code is successfully deployed, you will see the message Started app 'LogicFilter' on the console.

🖌 Mule - logicfilter/flows/LogicFilter.mflow - Mule Studio
File Edit Navigate Search Project Run Window Help
🖹 🔛 Mule
🖸 Console 🛛 🛛 🖷 💥 📓 🖉 🖉 🛃 🖓 👘 🛃 🖓
LogicFilter [Mule Application] C:\Program Files\Java\jre6\bin\javaw.exe (Mar 28, 2013 5:02:49 PM)
connected strue
supportedProtocols=[http]
serviceOverrides= <none></none>
INFO 2013-03-28 17:03:02,875 [main] org.mule.lifecycle.AbstractLifecycleManager: Starting connector: connector.http.mule.de
INFO 2013-03-28 17:03:02,875 [main] org.mule.lifecycle.AbstractLifecycleManager: Starting model: _muleSystemModel
INFO 2013-03-28 17:03:02,875 [main] org.mule.construct.FlowConstructLifecycleManager: Starting flow: LogicFilterFlow1
INFO 2013-03-28 17:03:02,890 [main] org.mule.processor.SedaStageLifecycleManager: Starting service: LogicFilterFlow1.stage1
INFO 2013-03-28 17:03:02,890 [main] org.mule.component.ComponentLifecycleManager: Starting component: commponent.30684172
INFO 2013-03-28 17:03:02,921 [main] org.mule.transport.http.HttpConnector: Registering listener: LogicFilterFlow1 on endpoi
INFO 2013-03-28 17:03:02,968 [main] org.mule.transport.service.DefaultTransportServiceDescriptor: Loading default response
INFO 2013-03-28 17:03:02,968 [main] org.mule.lifecycle.AbstractLifecycleManager: Initialising: 'null'. Object is: HttpMessa
INFO 2013-03-28 17:03:03,015 [main] org.mule.lifecycle.AbstractLifecycleManager: Starting: 'null'. Object is: HttpMessageRe
INFO 2013-03-28 17:03:03,031 [main] org.mule.module.management.agent.WrapperManagerAgent: This JVM hasn't been launched by
INFO 2013-03-28 17:03:03,078 [main] org.mule.module.management.agent.JmxAgent: Attempting to register service with name: Mu
INFO 2013-03-28 17:03:03,078 [main] org.mule.module.management.agent.JmxAgent: Registered Endpoint Service with name: Rule.
INFO 2013-03-28 17:03:03.078 [main] org.mule.module.management.agent.JmxAgent: Registered Connector Service with name Mule.
INFO 2013-03-28 17:03:03,0.78 [main] org.mule.module.launcher.application.beraulthuleApplication: keload interval: 3000
INFO 2013-03-68 1/:05:05,059 [main] org.mult.DerauthuleContext:
t Amplication: Logicfilter t
approveding togrammer in the encoding IITE-8 t
* *
* Agents Running:
* JMX Agent *

INFO 2013-03-28 17:03:03,093 [main] org.mule.module.launcher.DeploymentService:
+++++++++++++++++++++++++++++++++++++++
+ Started app 'logicfilter' +
· · · · · · · · · · · · · · · · · · ·
i 🕐 🐁 👘 👘 👘 🔛 👘 🖉

3. To see the output in your browser, copy the URL http://localhost:7676/ HiAzaz and paste it in your browser. If the string doesn't match the filter, the output will not be displayed on the console.



Performing filtering according to the exception type

An exception is a problem that arises during the execution of a program. An exception can occur for many different reasons, including the following: user errors, programmer errors, and physical resources that have failed in some manner.

Getting ready

You can supply the class indicating the exception type to the **Expected Type:** property. For example, for a null pointer exception type, you might set **Expected Type:** to java.lang. NullPointerException.

How to do it...

From the palette, drag the Exception filter onto the canvas. The Exception filter handles errors.



How it works...

Double-click on the **Exception** filter to configure it. Here, you can define the exception depending upon the requirement.

∑ Mule - exceptionfilter/flows/I	Y Pattern Properties	- 7 🛛
File Edit Navigate Search Project	Exception	
╡┇╸╢╚╘┊ॐ╸Ѻ╸ ╸━━	The Exception Filter matches to the type of an exception.	
📮 Package E 🛛 🧏 Type Hiera	General Documentation	
E	Display	Filter:
	Display Name: Exception	Select
🗉 📴 catchexcaption	Generic	Care Endpoints
Choice_router Corp-expression	Expected Type: java.lang.NullPointerException	Copes
∎ 😫 cxf		Components
exceptionfilter		Carl Transformers
A Mule Runtime [Mule Server		🚔 Filters 🛛 🗠
😑 🥮 src/main/java		And
🖃 🌐 com.org 🗄 🕕 🚺 Greeting.java		Custom
🕮 src/test/java		Exception
# src/main/resources		Expression
E 🗁 flows		Filter Reference
ExceptionFilter.mflow		Message
🖅 mappings		Message Property
mule-project.xml		Not
file_transfer filetransfer		V Or
😟 🔛 filter		Payload
E Outline M		Regex
		Schema Validation
	OK Cancel	Wildcard
		Control
		Crror Handling
		Cloud Connectors
	Message How Global Llements Configuration XML	

Filtering messages by evaluating expressions

Mule ESB provides several default expression evaluators allowing you to embed expression logic in a variety of expression languages; alternatively, you can create your own evaluators to support additional languages. This filter lets you evaluate a range of expressions. It supports expression types such as header, payload, regex, and wildcard. Set the evaluator to specify the expression evaluator type to be used. The **RegEx** filter applies a regular expression pattern, such as a pattern that includes wildcards or other character substitution symbols in the message payload. The filter applies the toString() method to the payload to convert the payload into a string.

Getting ready

In Mule, there are different types of evaluator expressions, such as the header payload type, the exception type, wildcard, regex, bean, and groovy. Each expression depends on the evaluator type. For example, if the expression type is XPath, bean, or OGNL, the expression should be a Boolean. Expressions allow you to extract information from the current message or determine how to handle the message. Expressions are very useful with routers.

How to do it...

In this section, you will see how to create a custom evaluator and how to use it in a Mule configuration file.

- To create a custom evaluator, the first step is to implement the ExpressionEvaluator interface. Note that this interface implements NamedObject, which allows the evaluator to be named. This is the name you use for the evaluator attribute when using this evaluator in the configuration. After that, create another class named MessageHeaderExpressionEvaluator that implements with the interface, which we have created before.
- 2. After creating your custom expression evaluator, you must register it with Mule. If you are using an XML configuration, you can just configure your expression evaluator as a bean, and Mule will discover it.

260

How it works...

In this section, you will see different types of expressions.

XPath expressions

XPath expressions use the standard XPath query language based on JAXP—the Java API for XML processing. Refer to Using Filters (http://www.mulesoft.org/documentation/ display/current/Using+Filters) for more information about XPath expressions.

```
<expression-filter evaluator="xpath" expression="(msg/header/
resultcode)='success'"/>
```

JXPath expressions

JXPath is an XPath interpreter that can apply XPath expressions to graphs of objects of all kinds, including JavaBeans, Maps, Servlet contexts, DOM objects, and mixtures of these objects. For more information about JXPath, refer to Using Filters (http://www.mulesoft.org/documentation/display/current/Using+Filters).

```
<expression-filter evaluator="jxpath" expression="(msg/header/
resultcode)='success'"/>
```

OGNL expressions

OGNL is a simple, but a very powerful, expression language for plain Java objects. Similar to JXPath, OGNL works on object graphs. Filters using OGNL expressions enable simple and efficient content routing for payloads. Refer to Using Filters (http://www.mulesoft.org/documentation/display/current/Using+Filters) for more information.

```
<expression-filter evaluator="ognl" expression="[MULE:0].equals(44)"/>
```

Handling incoming events or messages using the Message filter

The Message filter is used for deciding whether to handle incoming events or messages. You can use the **On Unaccepted** property to optionally specify the name of the message processor that should handle any unaccepted events.

261

Getting ready

Drag-and-drop the **Message** filter from the palette on the canvas and configure it. Double-click on the **Message** filter. Here you can see the **Throw on Unaccepted** checkbox. Select this checkbox to throw an exception if a message or event is not handled. The default—when not checked—is to not throw an exception. You can use the **On Unaccepted:** property to optionally specify the name of the message processor.

	Y Pattern Properties	
File Edit Navigate Search Project	Message	
	🛞 Nested element 'Nested Filter' is required	
😫 Package E 🛛 🏌 Type Hiera	General Documentation	
	Display	Filter:
💷 🚰 ajex	Display Name: Message	Select
	Generic	C
🛓 🔁 choice_router	On Unaccepted:	Contraction Contraction Contraction
Corn-expression		Scopes
E CXF	Throw On Unaccepted	Components
E Generation inter	Nested Filter	Transformers
🗷 🔛 filetransfer	Nested Filter:	📥 Filters 🗠
🗈 😂 filter		And
		Custom
ison		Exception
🗈 🔛 logicfilter		Expression
mametransfer		Filter Reference
guartz		Idempotent Message
🗷 🔛 quartz_scheduler		Message
restbasedwebservice		Message Property
spellcheck		Vot
🗄 🔂 spring		Y Or
testtransformer		Payload
		Regex
DE Outline 23		Schema Validation
	(?) OK Cancel	Wildcard
		Plow Control
		lerror Handling
		Cloud Connectors
	Message Flow Global Elements Configuration XML	
E □ ◆	en 1	: e 🖹 @ 🔍 M i e 🖻

How to do it...

The Message filter is used to route messages using logic based on the values of message properties. You can define one or more message properties. Using this filter, you can access the message metadata, including transport-specific and user-defined properties. These message properties represent all the meta information about the message that is available from the underlying transport. For example, if you receive a message via an HTTP Endpoint, you can use this filter to check HTTP header values. The pattern is expressed as a key/ value pair, where key is the name of the property. By default, the **Case Sensitive** checkbox is selected representing that the comparison is case sensitive. Deselect this box if you want the association to be case insensitive.



How it works...

This filter can be very powerful because the message properties are exposed allowing you to reference any transport-specific or user-defined properties. For example, you can match one or more HTTP headers for an HTTP event, match properties in JMS and e-mail messages, and much more. You can define **Scope** in the message filter properties. There can be different types of scopes available in the message properties. You will see different types of scopes in the following screenshot. The expression syntax has been improved to support scopes. The **Scope** part is optional and is case insensitive. The default scope is **outbound**. The general syntax pattern is <evaluator>:<scope>:<expression>; for example, header:OUTBOUND:CookBook.

Mule - exceptionfilter/flows/	Y Pattern Properties	_ 2 🛛
Mule - exceptionfilter/flows/ File Edit Navigate Search Project Package E Package E Type Hiera Package E Type Hiera Package E Type Hiera Package E Context Package E Context Pack	Pattern Properties Message Property Attribute 'pattern is required General Documentation Display Name: Message Property General Display Name: Message Property General Case Sensitive 0 Scope:outbound	Filter:
Control of a data in the final set version of the set vers version of the set version of the set version of the set versio	nvocadon session application	And Custom Custom Exception Filter Reference Filter Reference Message Message Not Solution Payload Pa
	OK Cancel	Schema Validation Wildcard
	Message Flow Global Elements Configuration XML	Cloud Connectors
E mule/flow/message-p	roperty-filter	e 🖹 @ 🗟 M 🛛 e 🗳

You will see the following different types of scopes being used in the **Message Property** window:

- inbound: This specifies the properties/headers that come from a client's request.
- invocation: This is used mostly internally by Mule for the duration of this flow's call.
- **outbound**: This specifies the values deemed to be sent out from this flow. They become either request properties or response properties for the next flow in the case of a synchronous invocation.



- session: This specifies the values that are passed from invocation to invocation.
- application: This scope is used when you create two different applications.

Configuring the Wildcard filter

The Wildcard filter applies a wildcard pattern to the message payload. This filter applies toString() to the payload, so you might also want to apply a payload type filter to the message using an And filter to make sure the payload is a string.

Getting ready

After dragging the Wildcard filter from the palette and dropping it on the canvas, double-click on the filter icon. A **Pattern Property** window shows up. There is only one attribute to configure for this filter and it is the pattern. You will see the example of using the Wildcard filter in this section.

How to do it...

In this section, you will use three components: the HTTP Endpoint, the Wildcard filter, and the Custom transformer.

1. Open Mule Studio and enter the workspace name as shown in the following screenshot:

Y Workspace Launcher	
Select a workspace	
Mule Studio stores your projects in a folder called a workspace. Choose a workspace folder to use for this session.	
Workspace: E:\MuleCookBook	Browse
Use this as the default and do not ask again	OK Cancel

2. To create a new project, go to **File** | **New** | **Mule Project**. Enter the project name Wildcard Filter, click on **Next** and then on **Finish**. Your new project is created now, so you can start the implementation.

🎽 Mule - Mule Studio		
File Edit Source Refactor Navigate Search Project	Run Window Help	
New Alt+Shift+N	🕨 🛜 Mule Project	- *\$ () -
Open File	挫 Java Project	
Close Ctrl+W	Project	
Close All Ctrl+Shift+W	📌 Manning Flow	
Save Ctrl+S	Mule Flow	
Save As	📅 Package	
iii Save All Ctrl+Shift+S	🞯 Class	
Revert	🞯 Interface	
Move	🕼 Enum	
Rename F2	Annotation	
& Refresh F5	Source Folder	
Convert Line Delimiters To	Java Working Set	
Print Ctrl+P	Folder	
Switch Workspace	Intitled Text File	
Restart		
	Example	
A Export	📬 Other Ctrl+N	
Export diagram to		
	-	
Properties Alt+Enter	_	
1 FlowRef.mflow [flowref/flows]		
2 package.html [activity/src/main/] 2 Activity.mflaw.[activity/flaws]		
4 StringToNumber.java [usdconverter]]		
Evit		
Art Oddine is hot available.		
i □ *		i a 🔝 @ 😣 M 📃

3. Go to src/main/java, right-click on it, enter the package name com.org and the class name User, and click on the **Finish** button. In this class, you have to use two private String variables, and then you can generate the get or set method.



You create a class called User and generate the get and set methods:

```
package com.org;
public class User {
    private String name;
    private String lname;
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public String getLname() {
        return lname;
    }
}
```

```
}
public void setLname(String lname) {
   this.lname = lname;
}
```

4. Go to src/main/java, right-click on it and go to **New** | **Class**, enter the package name com.org and the class name HTTPReqToUser, and click on the **Finish** button. Here, you create a custom transformer class and extend it with AbstractMessageTransformer. This way you override the transformMessage method.



You create a class and extend it with the AbstractMessageTransformer interface:

```
import org.mule.api.MuleMessage;
import org.mule.api.transformer.TransformerException;
import org.mule.transformer.AbstractMessageTransformer;
```

public class HTTPReqToUser extends AbstractMessageTransformer

```
{
    @Override
    public Object transformMessage(MuleMessage message, String
    outputEncoding)
        throws TransformerException {
        User user = new User();
        String name = message.getInboundProperty("name");
        String lastName = message.getInboundProperty("lname");
        user.setLname(lastName);
        user.setName(name);
        return user;
    }
}
```

5. Go to src/main/java, right-click on it and go to **New | Class**, enter the package name com.org and the class name HTTPReqToName, and click on the **Finish** button.



-268

```
Here, you will create a class called HTTPReqToName and extend it with the
AbstractMessageTransformer interface.
package com.org;
import org.mule.api.MuleMessage;
import org.mule.api.transformer.TransformerException;
import org.mule.transformer.AbstractMessageTransformer;
public class HTTPReqToName extends AbstractMessageTransformer
{
  @Override
  public Object transformMessage(MuleMessage message, String
outputEncoding)
      throws TransformerException {
    String name = message.getInboundProperty("name");
        String lastName = message.getInboundProperty("lname");
        return name+" "+lastName;
  }
}
```

6. Go to the Filter.mflow file and drag the **HTTP** Endpoint onto the canvas. Double-click and configure it. Enter the port number and the hostname, and click on the **OK** button.

Y Mule - filter/flows/Filter.mfl	Y Endpoint Properties	🔳 🖻 🗾
File Edit Navigate Search Project	HTTP (Inbound Endpoint) The HTTP Endpoint allows Mule application to connect to web resources through the HTTP transport protocol. This endpoint can also implement security through HTTPS.	
Package E	General Advanced References HTTP Settings Documentation Display Mane: HTTP Exchange Patterns Orne-way Orequest-response Basic Settings Enable HTTPS II require configuring a HTTPS Connector Host: Iscahost Port: 9999 Path:	Filter: Select Endpoints Components Co
	CK Cancel Message Flow Global Elements Configuration VM	Filters
		e 🖹 @ 🕵 M 🕴 e 🛢



7. Drag the **Wildcard** filter onto the canvas. Double-click and configure it. In the **Wildcard** filter, you have to add a pattern, /Mule*; this means that a pattern should start with the text Mule.





8. Drag the Java transformer onto the canvas, and double-click on it to configure it.

Y Mule - filter/flows/Filter.mflc	Y Pattern Properties	X 🗖 🛛 🗙
File Edit Navigate Search Project	Java the Java Transformer delegates to a Java class.	
📮 Package E 🕺 🍃 Type Hiera	General Advanced Documentation	- 8
Package E Package E	General Advanced Documentation Display Display Name: Java Transformer Settings Transformer Class: com.org.HTTPReqToName Property	Filter: Select Filter: Select Components Compone
n tweise vice		HTTP Response to Object
E E Ionicfilter		HTTP Response to String
🗄 Outline 🛛		JSON to Object
	OK Cancel] Java
		Flow Control
		Error Handling
	Macroso Eleve Clobal Elemente Configuration VM	Cloud Connectors
	message now global clements coninguration Amb	
		e 🚼 @ 😥 🖊 🛛 e 🚍

How it works...

In this section, you will see how to deploy the application on Mule Studio, and also how it works after deployment.

1. To deploy the application code in the Mule server, go to **Run As** | **Mule Application**, and the Mule server will deploy your application.



2. Then, open the browser and paste the URL http://localhost:9999/ Mule?name=azaz&lname=desai to see the output.





Creating a Custom filter

Use a Custom filter to reference a user-implemented filter. Note that the reference is to a class implementing the Filter interface.

Getting ready

The Filter class is required if this is a global filter. After dragging the **Custom** filter from the palette and dropping it onto the canvas, double-click on the filter icon. A **Pattern Property** window shows up. There is only one attribute to configure for this filter and it is the Filter class.

How to do it...

In this section, you will use three components: the HTTP Endpoint, the Echo component, and the Custom filter.

1. Open Mule Studio and enter the workspace name as shown in the following screenshot:

🖌 Workspace Launcher	<u> </u>
Select a workspace	
Mule Studio stores your projects in a folder called a workspace. Choose a workspace folder to use for this session.	
Workspace: E:\MuleCookBook	Browse
Use this as the default and do not ask again	
	OK Cancel



2. To create a new project, go to **File** | **New** | **Mule Project**. Enter the project name Custom Filter, and click on **Next** and then on **Finish**. Your new project is created now, so you have to start the implementation.



3. Go to src/main/java, right-click on it and go to New | Class, enter the package
 name com.mule.filter and the class name CustomMuleFilter, and click on
 the Finish button. In this class, we implement the Filter interface, and in this filter
 interface class, we override the accept method.


Configuring Filters —

4. Go to the CustomFilter.mflow file and drag the **HTTP** Endpoint onto the canvas. Double-click on it to configure it. Enter the port number and the hostname, and click on the **OK** button. By doing this, you enter the pathname called customFilter.

Y Mule - customfilter/flows/Cus	Y Endpoint Properties	
Mule - customfilter/flows/Cus File Edit Navigate Search Project	Endpoint Properties If TP (Inbound Endpoint) The HTP Endpoint allows hule application to connect to web resources through the HTP transport protocol. This endpoint can also implement security through HTTPS. General Advanced References HTTP Settings Documentation Display Display Name: HTTP Exchange Patterns one-way @ request-response Basic Settings Enable HTTPS will require configuring a HTTPS Connector Hots: bcahost Port: 4343 Path: customFilter	Fiter: Custo
A Jrk: system Lubrary Lavase A Jrk: system Lubrary Lavase Consume Linker Consume Linker Consume Linker Consummed Linker Consummed Linker Consummed Linker Societies Linker Societies Linker Consummed Linker Societies Linker Societies Societi	Host: localhost Port: 4343 Path: customFilter	
	OK Cancel Message Flow Global Elements Configuration XML]
i □ ◆	😪 🕴	je 🛃 @ 🔯 M 🗍 e 📮



- _ 7 🗙 🖌 Pattern Properties Echo 📑 • 🔛 🗟 🖻 🗄 🏘 • 🔘 The Echo Component logs the message and returns the payload as the result. 🔛 🔛 Mule 洋 Package E 🛛 🏦 Type Hiera - -General Documentation Display Filter: custo Display Name: Configuer the Custom Filter Select

 Image: Catchexcaption

 <t 눹 Filters Custom a Flow Control Custom Aggregator src/main/resources src/test/resources
 flows
 CustomFilter.mflow 🗁 mappings mappings
 mappings
 mappings
 male-project.xml
 male-project.xml
 made-project.xml
 mappings
 ma E Outline 🛛 ? OK Cancel 60 4-----Message Flow Global Elements Configuration XML ∎⇔ e 🖹 @ 😣 M 🔢 e 🗳
- 5. To display the log on the console, drag the **Echo** component, drop it on the canvas, and configure it.

277—

Configuring Filters _____

6. Drag the **Custom** filter from the palette and drop it on the canvas. Double-click and configure it. Import the class that was created before, as shown in the following screenshot:

Y Mule - customfilter/flows/Cus	Y Pattern Properties		. 3 🗙
Mule - customfilter/flows/Cus File Edt Navigate Search Project P	Y Pattern Properties Custom The Custom Filter is a user-implemented filter. General Spring Documentation Display Display Name: Custom Generic Class: Com.mule.filter.CustomMuleFilter		Filter: Custo Filter: Custo Filters Co Filters Custom Filters Custom Filters Custom Filters Custom Aggregator
Forvs Forvs Forvs CustomFilter.mflow Forvs For		Cancel	
: •	message riow Global Elements Configuration XML		
: U			* 🔊 🧠 📫 👘 🚍



How it works...

In this section, you will see how to deploy the application.

1. To deploy the application code in the Mule server, go to **Run As** | **Mule Application**, and the Mule server will deploy your application.

🖌 Mule - customfilter/flows/CustomFilter.mflow - Mule Studio 💿 💽 🔀										
File Edit Source Refactor Navigate Search Project Run Window Help										
: *** • 目前 6: : ** • 0 • 0 • : ** 6* • : ** 2 • : ** : 5 • 5 • ** 4 • : * : * : * : * : * : * : * : *										
📲 Package E 🛛 🏌 🏌 Ty	pe Hiera	mFilter 🛛 🚺 '	*CustomMuleFilter.java							
	🖻 🔄 🎽			Filter: custo						
🗉 🚰 ajex	~			N. Colort						
🗉 🚰 allrouter				19 pelect						
the catchexcaption				🚰 Filters 🛛 🗠						
				Custom						
a customfilter				Elow Coptrol						
🗷 🛋 JRE System Librar	New	•								
🗷 🛋 Mule Runtime [Mu				Custom Aggregator						
😑 🌁 src/main/java	Open	F3								
🖃 🌐 com.mule.filte	Upen with	Alter Children H.								
🔛 🕗 🚺 CustomMu		AIC+SNIFC+W F								
src/resc/java	📄 Сору	Ctrl+C	Echo							
→ Sic/Hair/resources	🗎 Copy Qualified Name		Configuer the Custom Filter Custom							
😑 🗁 flows	💼 Paste	Ctrl+V								
	💢 Delete	Delete								
- 🧀 mappings	Build Path	•								
src	Refactor	Alt+Shift+T								
mide-project.xm										
🗄 🔂 exceptionfilter	import									
🗉 🔂 file_transfer	Kara Export									
🗄 🗁 filetransfer	🔗 Refresh	F5								
	Assign Working Sets									
🗄 Outline 🛛	Validate									
	Run As	•	1 Run on Server Alt+Shift+X, R							
	Debug As	Þ	V 2 Mule Application							
	Profile As	•								
	Team	•	Run Configurations							
	Compare With	•								
	Replace With	•	Configuration XML							
	👏 Mule Management Console	+								
CustomFilter.mf	🔀 CloudHub	•	e e e e e e e e e e e e e e e e e e e	🔣 @ 😫 M 🛛 🗧 📮 🖉						

2. Open the browser and paste the URL http://localhost:4343/customFilterthe URL that starts with http://localhost:4343/customFilter?name=azaz& lname=desai.





THandling Exceptions and Testing

In this chapter, we will cover different ways of handling exceptions and testing. You will learn how to do the following:

- Understanding Messaging Exception strategies
- ► Configuring the Choice Exception Strategy
- ► Configuring the Reference Exception Strategy
- Configuring the Rollback Exception Strategy
- ▶ Testing with JUnit in Mule ESB

Introduction

Mule errors are represented by **exceptions**; so when your Endpoint, Router, component, or any other processor fails, it throws an exception. When an exception is thrown, you need a way to handle it. In Mule there are different types of exception strategies. You will have a look at them in this chapter. There are two places where you could configure exception handling strategies: on the service and on the connector. The src/test/directory in every Mule ESB Maven project incorporates both unit and functional tests. Handling Exceptions and Testing

Understanding Messaging Exception strategies

Mule calls for the Messaging Exception Strategy whenever an exception is thrown in a flow. All exceptions are handled through the Messaging Exception Strategy. There are five types of Messaging Exception Strategies:

- Default Exception Strategy
- Catch Exception Strategy
- Rollback Exception Strategy
- Reference Exception Strategy
- Choice Exception Strategy

The Catch Exception Strategy

You can define a Catch Exception Strategy to customize the way Mule handles messages with errors. A Catch Exception Strategy catches all the exceptions thrown within flow and processes them. From the **Error Handling** palette group, drag **Catch Exception Strategy** and drop it onto the canvas.

Getting ready

In this section, you will see how to use the Catch Exception Strategy in Mule Studio.





Double-click on Catch Exception Strategy and configure it.



How to do it...

In this section, you will see how to configure the catch exception properties in Mule:

- 1. In the Catch Exception Strategy properties panel that appears on the screen, enter a name for your Catch Exception Strategy in the **Name:** field under the **General** tab.
- 2. Check the **Enable Notifications** checkbox to instruct Mule to send an exception notification to a registered listener, say, the Mule management console, whenever the Catch Exception Strategy accepts and handles an exception.
- 3. To enter an exception, enter information in the **Execute When:** field to indicate the kind of exception the catch exception handles when it resides within a Choice Exception Strategy.

Use case

HTTP Inbound Endpoint receives a message through the custom Java component and processes it through the flow. If the execution is successful, return a Java component message. If the execution fails then log the current message and return a Java component error response. You can define only one exception at a time.



Handling Exceptions and Testing

Configuring the Choice Exception Strategy

A Choice Exception Strategy catches all the exceptions thrown within its parent flow, examines the message content and exception type, and routes messages to the appropriate exception strategy for processing.

Getting ready

You can define a Choice Exception Strategy to customize the way Mule handles a message with an error based on the message's content the moment it throws an exception. Usually, you can define more than one exception strategy within a Choice Exception Strategy.

How to do it...

 From the Error Handling palette group, drag Choice Exception Strategy and drop it onto the canvas. In the Choice_Exception_Strategy box that appears, double-click on the configuration icon in the header.



-284

 From the Error Handling palette group, drop another Catch Exception Strategy or Rollback Exception Strategy icon on the Choice_Exception_Strategy box. In Choice Exception Strategy you can define one or more exception strategies, but in Catch Exception Strategy you can define only one.



Use case

Use a Choice Exception Strategy to enable Mule to make decisions about how to handle each error that occurs in a flow. A Choice Exception Strategy can evaluate the exception type of each message that throws an exception in this flow and route them to one of three exception strategies:

- A Catch Exception Strategy to process and discard all the "already processed" exceptions
- A second Catch Exception Strategy to process all the "validation exceptions" and send them to an invalid orders queue
- A Rollback Exception Strategy to roll back the order transaction in order to retry processing in the parent flow



Handling Exceptions and Testing

Configuring the Reference Exception Strategy

Use a Reference Exception Strategy to teach a flow to employ the error handling performance defined by a global Rollback Exception Strategy. In other words, you must ask your flow to refer to a global exception strategy for instructions on how to handle errors.

Getting ready

Reference Exception Strategies are created as global elements. You can create one or more strategies or re-use the one(s) in the flow throughout your Mule application. Create global exception strategies such as Catch and Rollback; choose the one which your exception strategies refers to. In the following screenshot you will see how to configure the global element:



-286

How to do it...

Perform the following steps:

1. From the **Error Handling** palette group, drag-and-drop the **Reference Exception Strategy** icon onto the canvas.





Handling Exceptions and Testing -



2. To configure the **Reference Exception Strategy**, double-click on **Components** and select a reference name that was created as a global element.

Configuring the Rollback Exception Strategy

You can define a Rollback Exception Strategy. This makes sure that a message that throws an exception in a flow is rolled back for reprocessing. Use a Rollback Exception Strategy when you cannot correct an error when it occurs in a flow.

-288

Getting ready

The Rollback Exception Strategy is used to loop the executions infinitely until the conditional exceptional strategy becomes true. Let's say for example, we have a server with a count value = 0. Now if 50 requests come simultaneously and you want the server to handle not more than 50 instances, you will set a rollback exception that sets the count variable to 0 each time it reaches 50. This process is looped infinitely.

How to do it...

Usually, you can use this rollback exception in a transaction. If the transaction fails or if a message throws an exception while being processed, the Rollback Exception Strategy rolls the transactions back to where they exist in the flow. If the Inbound Endpoint is transactional, Mule delivers the message to the Inbound Endpoint of the parent flow again to reattempt processing.

How it works...

A Rollback Exception Strategy gives a message a few attempts to move through the flow successfully before the transaction is declared as "failed" and the message is consumed.

For example, suppose you have a flow that involves a bank transaction to deposit funds into an account. You configure a Rollback Exception Strategy to handle the errors that occur in this flow; when an error occurs during processing, a flow external bank account database is temporarily unavailable and the message throws an exception. The Rollback Exception Strategy catches the exception, and rolls the message back to the beginning of the flow to reattempt processing. During the second attempt at processing, the database is online again and the message successfully reaches the end of the flow.

Mule attempts to deliver the message again when your flow uses one of the following two types of transports: transactional or reliable.

Testing with JUnit in Mule ESB

JUnit is a framework for implementing unit testing in Java. An open source Java testing framework is used to write and run repeatable automated tests. In JUnit 4.0, we do not need to extend from the JUnit framework to TestCase. Also, there is no need to use the prefix Test with the test method. You can run the test using JUnit4TestAdapter and the @NAME syntax, which already has been introduced. In the Dropbox integration example, you will learn about JUnit 4 and Selenium testing.

289

Handling Exceptions and Testing —

Getting ready

Create a new project on Mule to test with JUnit using the following steps:

1. Open Mule Studio and enter the workspace name as shown in the following screenshot:

¥ Workspace Launcher	
Select a workspace	
Mule Studio stores your projects in a folder called a workspace. Choose a workspace folder to use for this session.	
Workspace: Et/MuleCookBook	Browse
Use this as the default and do not ask again	
	OK Cancel

2. To create a new project, go to **File** | **New** | **Mule Project**. Enter the name of the project, DropBox_Integration, and click on **Next** and then on **Finish**. Your new project is created and you can now start the implementation.

Edit Source Refactor Nevigate Search Project Run Window Help New AddShifetdi Open File () Mule Project Close Crif+SW Close Al Crif+SW/t+W () Save A. () Project Save A.I. () Project Save A.I. () Project Save A.I. () Project () Project	🖌 Mule - Mule Studio					
New At+Shift+1 Image: Constraint of the state of	File Edit Source Refactor Nav	vigate Search Project	Run Window Help			
Open File I Save Project Close CrH+W Close All CrH+Shift+W If Save As If Mapping Flow Print If Mapping Flow Save As If Mapping Flow Print Print Convert Line Delmitters To If Mapping Flow Save Asta If Mapping Flow Export Other Export Other Export Other Propetts At+Enter 1 FlowRef.mflow [flows] Satabay.mflow [addocrwetter]] Exd If Satabay.mflow [addocrwetter]] Exd If Satabay.mflow [addocrwetter]] Exd If Satabay.mflow [addocrwetter]] Exd <td< th=""><th>New</th><th>Alt+Shift+N</th><th>🕨 🚰 Mule Project</th><th></th><th></th><th></th></td<>	New	Alt+Shift+N	🕨 🚰 Mule Project			
Close Cri+W Close Al Cri+S Mapping Flow Save A. Save A. Save A. Cri+Shift-S Protage Protage </th <th>Open File</th> <th></th> <th>🏄 Java Project</th> <th></th> <th></th> <th></th>	Open File		🏄 Java Project			
Close All Ch+Sht+W Mapping Flow Save Ch+Sht+W Male Flow Save A A Save A A Save A A Ch+Sht+S Class Cherric More Reman F2 Arindation Save A Morkspace Print Ch+P Switch Workspace Restart Import Export Export Export Source Folder Chere Switch Workspace Restart Import Export Export StringToNamber.java Jackby.mifw [dativity/src/main/] Satchity.mifwe [attivity/src/main/]	Close	Ctrl+W	🏫 Project			
Save Ctrl+S Save Ad Package Save Ad Procentian F2 Save Anotation Save Ad Save Ad Print Ctrl+P Switch Workspace Restat Save Ad	Close All	Ctrl+Shift+W	Apping Flow			
Save As Save As Save All CH+Shit+S Revet Move Rename F2 Refresh F5 Convert Line Delimiters T0 Print Ch+P Switch Workspace Restart Source Folder File Switch Workspace Restart Source Folder Cother Ch+P Switch Workspace Restart Source folder Cother Ch+P Switch Workspace Restart Source folder Cother Ch+P Switch Workspace Restart Source folder Cother Ch+P Switch Workspace Restart Source folder Cother Ch+P Switch Workspace Restart Source folder Cother Ch+P Switch Workspace Restart Source folder Source folder Cother Ch+P Switch Workspace Restart Source folder Source folder Cother Ch+N Cher Ch+N Source folder Source folder	🖁 Save	Ctrl+5	W Mule Flow			
Save All Chi+Shit+3 Revert © Class Move © Enum Rename F2 Refresh F5 Convert Line Delimiters To © Source Folder Pint Chi+P Switch Workspace Pider Restart © Other Import Chi+P Switch Workspace Pider Restart © Other Chi-Witting Calvity/filows] Alk+Enter 1 FlowRef.mflow [flowref/flows] 2 package.html 4 StringToNumber.java [usdconverter/] Exit	Save As		🖶 Package			
Revert Move Rename F2 Sefresh F5 Convert Line Delimiters To Print Ctrl+P Switch Workspace Restart Import Export Suprost diagram to Properties Alt+Enter 1 FlowRef.mflow [flowref/flows] 2 package.html [activity/src/mai/] 3 Activitymflow [activity/srlows] 4 StringToNumber.java [usdconverter/] Exit	🗑 Save All	Ctrl+Shift+S	🞯 Class			
Move Rename Parame Parame Parame Parame Parame Print Curiert line Delimiters To Properties Alt+Enter 1 FlowRef.mflow [Bouverfflows] 2 package.html [activity/src/main/] 3 Activity.mflows [ustorwriter/] Exit	Revert		🗊 Interface			
Rename F2 Refeath F5 Convert Line Delimiters To Source Folder Print Ctrl+P Switch Workspace Pile Unbitted Text File Export Export Export Properties Alt+Enter 1 Flowef.mflow (flower/flows) 2 package.html [activity/src/mair/] 3 Activity.mflow [activity/flows] 4 StringToNumber.java [usdconverter/]	Move		🞯 Enum			
Refresh F5 Convert Line Delimiters To Print Chi+P Print Chi+P Switch Workspace Restart Properties Alt+Enter 1 FlowRef.mflow [flowref/flows] 2 package.html [activity/sr/main] 3 Activity.mflow [activity/flows] 4 StringToNumber.java [usdconverter/] Exit	Rename	F2	Annotation			
Convert Line Delimiters To	🔊 Refresh	F5	Source Folder			
	Convert Line Delimiters To		 Image: Set and a set an			
Switch Workspace Restart Ulubited Text File Example Export diagram to Properties Al+Enter I FlowRef.mflow [flowref/flows] 2 package.html [activity/src/man/] 3 Activity.mflow [activity/src/man/] Exit	👜 Print	Ctrl+P	Folder			
Sind Workplace Restart Import Export Dother Ctrl+N Properties Alt+Enter I FlowRef.mflow [flowref/flows] 2 package.html [activity/src/main/] 3 Activity.mflow [activity/flows] 4 StringToNumber.java [usdconverter/] Exit	Switch Workspace		File Intitled Text File			
	Restart		E Oncided Texe The			
Cri+N	- Termark		Example			
	Manager Street		📑 Other	Ctrl+N		
	Export		_		,	
Properties Alk+Enter 1 FlowRef.mflow [flowref/flows] 2 package.html [activity/src/main/] 3 Activity.mflow [activity/flows] 4 StringToNumber.java [usdconverter/] Exit Introduction for not arcacute. 0° If @ @ @ M	Export diagram to					
1 FlowRef.mflow [flowref/flows] 2 package.html [activity/src/main/] 3 Activity.mflow [activity/flows] 4 StringToNumber.java [usdconverter/] Exit Information for net avalable:	Properties	Alt+Enter				
2 package.html [activity/src/main/] 3 Activity.mflow [activity/filows] 4 StringToNumber.java [usdconverter/] Exit Tordume to not available. 0 °	1 FlowRef.mflow [flowref/flows]	1				
3 Activity.mflow [activity./flows] 4 StringToNumber.java [usdconverter/] Exit 10° 10° 10° 10° 10° 10° 10° 10°	2 package.html [activity/src/mai	in/]				
4 StringToNumber.java [usdconverter]] Exit Troduce of the available:	3 Activity.mflow [activity/flows]					
	4 StringToNumber.java [usdcon	verter/]				
	Exit					
i o 🖸 🕴 🖉 🖉 🖉 🖉	AIT OUTING IS NOT AVAILABLE.					
					i a 🖹 @	

290

How to do it...

In this section, you will see how to configure the Dropbox connector in Mule Studio.

1. Navigate to the dropbox_Integration.mflow file in the **Global Elements** tab and click on the **Create** button. Go to **Cloud Connector** | **Dropbox**. Here you need to enter the API key or the secret key. You also have to generate this key on the Dropbox site.

└── Mule - dropbox_integration/f	Y Global Elemen	Properties			. ð 🗙
File Edit Navigate Search Project	Dropbox				
	Global Dropbox conf	guration information.			
📮 Package E 🛛 🏦 Type Hiera	General Oauth D	ocumentation			
E	Generic				
ti i i i i i i i i i i i i i i i i i i	Name:	Dropbox			Contra
E 🔁 catchexcaption	General				Create
choice_router corp-expression	Content Server:			_	Edit
E Customfilter	Server:			_	Delete
🖻 😂 cxf	Port:	8081			
Gropbox_integration JRE System Library [JavaSE	App Key:	ppr2sh8h1is39ks			
🗉 🔿 Mule Runtime [Mule Server	App Secret:	h3v5hzqz9x6lula			
■ # src/main/java	🗌 Debug ⁰ 🙆				
E 🕖 TestCase.java					
🗈 🔠 com.org.drop					
src/main/resources					
🕮 src/test/resources					
Referenced Libraries Generation					
DropBox Integration.mf					
🗁 mappings					
arc					
E Outline 🛛					
	?		ОК	Cancel	
•	Message F	Iow Global Elements Configuration XML			
: □*			6	1	e 🔝 @ 😥 VI 🕴 e 🗉

291—

Handling Exceptions and Testing -

2. Click on the **Message Flow** tab and drag the **HTTP** Endpoint and the **Dropbox** Cloud Connector onto the canvas. Here you need to select the reference name, which was created earlier, and select the authorized operation.





3. Now drag the **HTTP** Endpoint on the canvas and configure it.

Y Mule - dropbox_integration/f	Y Endpoint Properties	×	🗖 🗗 🔁
File Edit Navigate Search Project	HTTP (Inbound Endpoint) The HTTP Endpoint allows Mule application to connect to web resources through the HTTP transport protocol. This endpoint can also implement security through HTTPS.		
Package E Type Hiera Package E Type Hiera Package E Type Hiera Package E Type Hiera Package E Package	General Advanced References HTTP Settings Documentation Display Name: HTTP Exchange Patterns One-way @ request-response Basic Settings Enabling HTTPS @ Enabling HTTPS 0 Enabling HTTPS 0 Port: 0081 Part: in O OK Cancel		Fiter: Select Select Select Sopes Fiters
	KTTP	>	Paypal
E mule/flow/http:static	resource-handler	÷ 6	

293—

Handling Exceptions and Testing -

- Y Mule - dropbox_integration/flows/DropBox Integration.mflow - Mule Studio File Edit Navigate Search Project Run Window Help i 💼 • 🖫 🗟 🗁 i 🕸 • 🚺 • 🏰 • i 🌐 🎯 • i 🙋 🖋 • i 🖿 i 🖢 i 🖉 - 🏹 • 😓 - i 🗸 🖓 🗙 🖹 🔛 Mule DropBox Integration 🛛 🕖 TestCase.java - 2 8 1 Filter: 10 lacktrice Select •+= Carl Endpoints 8 Copes 🚰 Create Folder Object to JSON Components a Transformers ●→■ 🚰 Filters Logger Control Delete Object to JSON Logger Choice Logger er Mar arror Handling Default Cloud Connectors box Boxnet Logger 💽 CMIS Logger 🧠 Cloudhub 💿 Dropbox 👼 Github HBase 💿 Intacct 🦹 Jira 🝈 Magento o Metanga 💽 Mongo DB Netsuite S Objectstore 👂 Paypal Message Flow Global Elements Configuration XML 8 e 🖹 @ 😣 M 🕴 e 🗳 e mule/flow/http:static-resource-handler
- 4. Drag the **Choice** Router and two **Dropbox** connectors onto the canvas. You are creating two operations now: one is for creating a folder and another is for deleting it.

-294

5. Double-click on the **Choice** Router to configure it. Here you can assign the following condition: if operation equals to "createF", it will create a folder otherwise it will be deleted.

Y Mule - dropbox_integration/	✓ Pattern Properties		🗖 🗖 🔀
File Edit Navigate Search Project	Choice		
i 📬 • 🔛 🕼 🗁 i 🏇 • 💽	Routes messages based on expressions.		
🖹 🔛 Mule	- · ·		
····· 😭 DropBox Integration 🕱	Choice Properties Documentation		- 8
8			Filter:
	When	Route Message to	
	<pre>#[payload.get(op) == creater] #[payload.get('op') == 'delete']</pre>	Delete	Select
••••	Default	Logger	a Endpoints
8			Scopes
8º			Components
			Transformers
			Eilters
er Map Logger			Control
ci nap zoggor			Crror Handling
			Cloud Connectors ©
			Source Source
			CMIS Church the
			Cit L
			HBase
			X Jira
			Magenco Maharan
	0	OK Cancel	Marana DR
	\odot	Calica	Mongo DB
K	The Committee Man		r dypdi
message How Global Elements	onriguration XML		
🗄 📑 💽 mule/flow/http:statio	-resource-handler	🗞 :	e 🔣 @ 🔩 🖊 🛛 e 💻

295

Handling Exceptions and Testing _____

6. To configure the Cloud Connector, double-click on the Create Folder connector. Here you can select operation create folder and write an expression #[payload['dropboxPath']]. In the same way, you can create or delete operations on your own.

🔀 Mule - dropbox_integration/f 🎽 Pattern Prop	erties	X	🗖 🖬 🔁
File Edit Navigate Search Project □ □ • □	nector. The Dropbox Connector will allow to use the Dropbox REST API. Almost every be done via the API can be done thru this connector.		
General Docume Display Display Name: General Config Referer Operation: General	rkation Create Folder ce: Dropbox 🕑 🖉 🖉 Create folder		
er Map	#[payload['dropboxPath']]		Filters Flow Control Connectors Could Connectors Could Connectors Could Connectors Could Connectors Could Note
	OK Cancel		 Metanga Mongo DB Netsuite Objectstore
Maccane Flow Clobal Flemente Configuration YM			Paypal
Hessage How Gobal Elements Connightation AML			

7. To create a third flow, drag the HTTP Endpoint onto the canvas and configure it. After this, you have to add static resources handler manually as follows: <http:static-resource-handler resourceBase="\${app.home}/docroot" defaultFile="home.html"></http:static-resource-handler.</p>

─ Mule - dropbox_integration/f	Y Endpoint Properties		X	🔳 🗗 🔀
File Edit Navigate Search Project	HTTP (Inbound Endpoint)			
: C ¹ ▼ 🔡 (2) 🔄 : 券 ▼ 🔾 ▼	The HTTP Endpoint allows Mule application to This endpoint can also implement security thr	o connect to web resources through the HTTP rough HTTPS.	transport protocol.	
🕎 🔛 DropBox Integration 🗶 💡	General Advanced References HTTP Se	ettings Documentation		- a
	Display			Filter:
2	Display Name: HTTP			Select
Ferry Authoriza EV	Exchange Patterns			
B	○ one-way ③ request-response			
	Basic Settings			Components
	Enable HTTPS ⁰			Transformers
	Enabling HTTPS will require configuring a l	HTTPS Connector		🔁 Filters
	Host: localhost			🔁 Flow Control
	Port: 8081			a Error Handling
	Path: home			🔁 Cloud Connectors 🛛 🗠
				bax Boxnet
-1				CMIS
				Cloudhub
flow: HTML 📝				Oropbox
				👼 Github
				H HBase
HTTP htt				Intacct
				Y Jira
			/	Magento
27D	0		OK Cancel	Metanga
нттр	\odot			Mongo DB
L			~	
<	anfin waking VAN		>	- aypar
		A :		
: U e mule/flow/http:static	resource-handler			
Start Start Scienced	Mule - drop 🖾 Chapter 7	C:\WINDO 😢 Home - Dro	🗀 MuleCookB 🗐 440	105 0 🛛 🔿 🔜 😪 4:02 PM

8. Create a docroot folder inside src/main/app. Create a Home.html file inside the docroot folder, where index.css is stored.

The home.html file looks like the following:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<html>
<head>
<link rel="stylesheet" type="text/css" href="index.css" />
<meta http-equiv="Content-Type" content="text/html;
charset=UTF-8">
```



Handling Exceptions and Testing

```
<title>Insert title here</title>
</head>
<body>
  <script type="text/javascript">
  function processElements ( elements, style) {
    for (var i = 0; i < elements.length; i++)
      elements[i].style.display=style;
    }
  }
  function updateOptions(value) {
  var hideElements = document.getElementsByClassName('hidden');
  var showElements = document.getElementsByClassName(value);
 processElements(hideElements, 'none');
  processElements(showElements, 'block')
  }
  </script>
  <!--onsubmit="this.action=document.getElementById('op').
options[document.getElementById('op').selectedIndex].value;"-->
  <form action=/in method="post">
    Welcome to AttuneInfocom!!! <br /> <br />
    Operation: <select id="op" name="op"
onchange="updateOptions(this.options[this.selectedIndex].value);">
          <option value="selectoption">--Select Option--</option>
          <option value="createF">Create Folder</option>
          <option value="delete">Delete</option>
        </select><br /><br />
    <div class="hidden upFile createF delete downFile
list getLink" id="dropboxPath">Path:<input type="text"</pre>
name="dropboxPath" /></div>
    <input type="submit" value="Submit"/>
  </form>
</body>
</html>
```

298

```
The index.css file has the following code snippet in it:
form {
background: -webkit-gradient(linear, bottom, left 175px,
from(#CCCCCC), to(#EEEEEE));
background: -moz-linear-gradient(bottom, #CCCCCC, #EEEEEE 175px);
margin:auto;
position:relative;
width:350px;
height:350px;
font-family: Tahoma, Geneva, sans-serif;
font-size: 14px;
font-style: italic;
line-height: 24px;
font-weight: bold;
color: #09C;
text-decoration: none;
-webkit-border-radius: 10px;
-moz-border-radius: 10px;
border-radius: 10px;
padding:10px;
border: 1px solid #999;
border: inset 1px solid #333;
-webkit-box-shadow: 0px 0px 8px rgba(0, 0, 0, 0.3);
-moz-box-shadow: 0px 0px 8px rgba(0, 0, 0, 0.3);
box-shadow: 0px 0px 8px rgba(0, 0, 0, 0.3);
}
textarea#feedback {
width:375px;
height:150px;
}
textarea.message {
display:block;
}
input.button {
width:100px;
position:absolute;
right:20px;
bottom:20px;
```

299—

Handling Exceptions and Testing -

```
background:#09C;
color:#fff;
font-family: Tahoma, Geneva, sans-serif;
height:30px;
-webkit-border-radius: 15px;
-moz-border-radius: 15px;
border-radius: 15px;
border: 1p solid #999;
}
input.button:hover {
background:#fff;
color:#09C;
}
textarea:focus, input:focus {
border: 1px solid #09C;
}
img,a {
display:none;
}
#obj {
display:none;
}
.hidden {
display:none;
}
```

9. Here you have worked with three different flows. First, you deploy localhost:8081/ auth, and authorize the Dropbox application. Then you can call localhost:8081/ home.

-300

10. To deploy the application code in the Mule server, go to **Run As** | **Mule Application**; the Mule server will deploy your application.



301—

Handling Exceptions and Testing -

11. Open the browser and enter localhost:8081/auth. Once you enter the URL in the browser, it will redirect you to the Dropbox site and open a screen that looks like the following screenshot. Now click on the **Allow** button.





12. Enter the URL http://localhost:8081/home. Here, you can create a folder using the selected operation shown in the following screenshot:

🕹 Insert title here - Mozilla Firefox					-	PX
Eile Edit View History Bookmarks Tools Help						
Home - Dropbox	× +					
Comparison (Comparison (Compar	Ĩ	⊂ C'	🐠 = Ask.com	\mathcal{P}	⋒	🥐 🖣
	Welcome to AttuneInfocom!!!					
	Operation: Create Folder					
	Path: Submit					

13. Once you click on the **Submit** button, you will see an output similar to the following screenshot:

🗲 🕙 localho	st:8081/in								⊂ C ⁱ	<i>©</i> -	Ask.com		P	Â	*	-
{"revision":	11929,	"rev":	"2e9901a00460",	"thumb_exists"	false,	"bytes": 0,	"modified":	"Mon,	22 Ap	2013	10:54:31	+0000",	"path":	″/Te	stDro	рВо

303

Handling Exceptions and Testing —

- 14. Open your Dropbox account. You will see that a folder is created with the name TestDropBox.
- 15. Similarly, you can run the deployment for JUnit 4. For that, you will have to download the Selenium IDE testing plugin on Firefox.
- 16. Once you download the testing plugin, a part of the IDE is completed. Restart Firefox. After restarting, go to **Tools** and select **Selenium IDE**.
- 17. Enter localhost:8081/home, select the operation, and click on the **Submit** button. Then stop recording in Selenium IDE.

🥹 Insert title here - Mozilla Firefox		- 8 🛛
Eile Edit View History Bookmarks Tools Help		
Home - Dropbox	e x +	
Comparison and the second seco	☆ マ C 🖉 - Ask.com 🔎 1	r 🥐 🔻
	Welcome to AttuneInfocom!!! Selenium IDE 1.10.0 Ele Edit Actions Options Help Base URL https://www.dropbox.com/ East Slow Exet Case Untitled package com.example.tests; import org.junit.After; import static org.junit.After; import static org.junit.Assett.*; public class Untitled (private Selenium.statt(); Selenium.statt(); Selenium.statt(); Failures: 0 Import Note Import Note Import Note import Note Selenium.statt(); Selenium.statt(); Import Note Selenium.statt();	

-304



18. Now you have to convert the JUnit test case into an appropriate format. Go to **Options** | **Format** | **Java** / **JUnit 4** / **Remote Control**.

305

Handling Exceptions and Testing -

19. You will see that the JUnit test case is created. Copy this test case and paste it in your application.



-306

You can create such a test case in your application as well:



307—

Handling Exceptions and Testing —

How it works...

In this section, you will see how to deploy the JUnit test case in Mule Studio.

1. First, you have to import three Selenium JAR files: serializer-2.7.1, seleniumjava-2.31.0, and selenium-server-standalone-2.31.0.

Mule - dropbox_integration	YProperties for dropbox_in	itegration	
File Edit Source Refactor Na Constraints of the second se	Resource Builders Java Build Path Java Build Path Java Code Style Java Code Style Java Code Style Java Compiler Java Editor Javadoc Location Project Facets Project References Deformed Michae	Java Build Path	
	RunjDebug Settings Task Tags ∰ Validation	Add Class Folder Add External Class Folder Edk Remove Migrate JAR File	"https://www.drop
mappings	?	OK Cancel	
Gettine ☆	31 pub. 32 32 33 } 34 } 35	<pre>ic voia tearpown() throws fxception { selenium.stop();</pre>	Y
dropbox_integration		ед III III III III III III III III III I	Р 🗈 🔍 И 🕴 🗖

2. Go to the command prompt. Run the selenium-server-standalone-2.31.0.jar file using the following command:

java -jar selenium-server-standalone-2.31.0.jar

On execution, you will see a screen similar to the following screenshot:

🛿 C:WINDOWS\system32\cmd.exe - java -jar selenium-server-standalone-2.31.0.jar	. 🗗 🗙
Microsoft Windows XP [Version 5.1.2600] (C) Copyright 1985-2001 Microsoft Corp.	_
C:\Documents and Settings\azazdesai>cd Desktop	
C:\Documents and Settings\azazdesai\Desktop>java -jar selenium-server-standalone-2.31.0.jar pp 22, 2013 4:36:38 PH org.openga.grid.selenium.GridLauncher main 15:46:37.25 1PN0 - Javai Sun Miere 15:46:37.25 1PN0 - 05: Windows XP 5.1 X86 15:46:37.25 1PN0 - 05: Windows XP 5.1 X86 15:46:37.108 1PN0 - 02: Ji.0. Built from revision 1bd294d 15:46:37.108 1PN0 - V2.31.8. with Core v2.31.0. Built from revision 1bd294d 15:46:37.108 1PN0 - V2.31.8. with Core v2.31.9. Built from revision 1bd294d 15:46:37.108 1PN0 - Version Jetty/5.1.× 15:46:37.116 1PN0 - Started HttpContext[/selenium-server/driver,/selenium-server/driver] 15:46:37.116 1PN0 - Started HttpContext[/selenium-server/driver] 15:46:37.116 1PN0 - Started HttpContext[/selenium-server] 15:46:37.131 1PN0 - Started HttpContext[/selenium-server] 15:46:37.163 1PN0 - Started org.openga.jetty.jetty.ServerP128e28a	

3. Go to your application and right-click on the TestCase class. Go to Run As | JUnit.





Handling Exceptions and Testing -

- Y Mule dropbox_integration/src/main/java/com/example/tests/TestCase.java Mule Studio File Edit Source Refactor Navigate Search Project Run Window Help 🔯 • 🔛 🗟 🗄 🎄 • 🔕 • 🗳 📽 🞯 • 🗄 🤔 🔗 • 🗄 🖓 • 🗄 🖓 • 🖄 💷 🗄 🖉 • 🏷 🔶 • 🔶 🔛 🔛 Mule 📲 🚮 💊 🔒 🔳 🗒 • 🍸 🗆 🗖 😫 Package E 🕺 🍃 Type Hiera 🦳 🗖 🤎 DropBox Integration 🕼 TestCase.java 🕺 😑 🗖 JUnit 🕅 1 package com.example.tests; 🗏 🔄 🗸 Finished after 34.562 seconds 2
 3* import com.thoughtworks.selenium.*; Runs: 1/1 Errors: 0 E Failures: 0 🗄 🔁 customfilter 🗄 🔛 cxf aropbox_integration 10 public class TestCase { B - A JRE System Library [JavaSE-1.6] 🗉 🔚 com.example.tests.TestCase [Runner: JUnit 4] (18.422 s) private Selenium selenium; 🗉 🛋 Mule Runtime [Mule Server 3.3.0 C 130 0Before 14 public void setUp() throws Except: 🗄 🎣 TestCase.java 15 selenium = new DefaultSeleniur 🗉 🔠 com.org.drop 16 selenium.start(); / 🕮 src/test/iava) src/main/resources 18 src/test/resources 190 0 Test 🗉 🔜 Referenced Libraries 20 public void testCase() throws Exce i i lows selenium.open("http://localhos Y DropBox Integration.mflow assertEquals("Insert title her 🗁 mappings selenium.select("id=op", "labe 😑 🐎 src selenium.type("name=dropboxPat 😑 😥 main selenium.click("css=input[type 😑 🧁 app E Failure Trace selenium.waitForPageToLoad("30 😑 🗁 docroot 27 assertEquals("", selenium.get] - index.css 28 } 29 M DronBox Integration.xr 🔳 🗶 🔆 📴 🛃 🗐 🚝 🛃 🗖 - 🗂 -📮 Console 🛛 DropBox Integration [Mule Application] C:\Program Files\Javaljret(bin]javaw.exe (Apr 22, 2013 4:13:33 PM) nccp://iocarnosc:ouoi/aucn-nccpnessagereceiver(cnis-isopise, receiverrey-nccp://iocarnosc:ouoi/aucn, enopoinc-nccp://iocarno http://localhost:8081/0b07c574-5390-4994-92f4-87f72736c666=HttpMessageReceiver(this=1c3e3fb, receiverKey=http://localhost:80 http://localhost:8081/in=HttpMessageReceiver(this=1aca5e2, receiverKey=http://localhost:8081/in, endpoint=http://localhost:8 http://localhost:8081/home=HttpMessageReceiver(this=1be9a50, receiverKey=http://localhost:8081/home, endpoint=http://localho INFO 2013-04-22 16:53:14,116 [[dropbox_integration].connector.http.mule.default.receiver.06] org.mule.api.processor.LoggerB < > • 82 e 🚼 @ 😣 M 🗄 e 🗐
- 4. If your JUnit test case runs successfully, you can see in your Dropbox account. A folder is created automatically in Dropbox.

-310

8 Introducing Web Services

In this chapter, you will learn about web services, integration of web services, and much more. The following topics will be covered in this chapter:

- Proxying web services
- Creating JAX-WS services
- Creating web services using the REST component
- ► Calling external web services using the SOAP component

Introduction

A web service is an application that is written to meet Internet and Extensible Markup Language (XML) technology standards. It performs a specific task and is made available to other users through a network. In this section, we will learn about the two main types of web services in use: SOAP-based and REST-based. The key characteristics of web services are flexibility, interoperability, and transportability. Today, they are used in a variety of ways, including various web APIs, integration frameworks, and architecture models such as service-oriented architecture. Web services allows different applications to talk to each other and share data and services among themselves. Other applications can also use the services of web services. For example, a VB or .NET application can talk to Java web services and vice versa. So, web services are used to make the application platform and technology independent.
Proxying web services

A proxying web service is a very common application used for different purposes, such as decoupling clients and producers. CXF proxies support working with the SOAP body or the entire SOAP envelope. By default, only the SOAP body is sent as payload, but the payload mode can be set only via the payload attribute to the envelope needed. You can define two types of proxying web services: server-side and client-side.

Getting ready

Mule can perform as a web service proxy. A proxy can perform several useful functions:

- ► Security enforcement
- ► WS-Policy enforcement
- Routing to the proper backend service, whether a remote service or a local service
- Protocol bridging, such as HTTP to JMS
- Message transformations, such as converting from old versions of the message format to new versions
- Validation

Mule provides several utilities that help you do these.

In Mule Studio, you can create different types of web services: WS-Security, WS-Proxy, and protocol binding.

- WS-Security web service: WS-Security provides means to secure your services above and beyond transport level protocols such as HTTPS through a number of standards such as XML-Encryption, and headers defined in the WS-Security standard.
- Protocol binding: This allows you to forward requests from one Endpoint to another. This is generally the best option for proxying web services.
- ► WS-Proxy web service: This allows you to service WSDL files locally while proxying remote web services.

How to do it...

First we will see proxying web services.



Web service proxying

In web service proxying, you have to use CXF proxying for the following:

- ▶ To implement WS-Policy assertions
- ▶ To easily service a WSDL associated with your service
- ▶ To work directly with the SOAP body; for example, adding XML directly to it

A CXF web service standard supports the use of WS-Security and WS-Addressing.

Protocol binding

Protocol binding is used for forwarding a request from one Endpoint to another via service bridging. You can forward the data streams directly or process and transform the XML.

How it works...

The following code snippet is a simple configuration example that forwards a request from one HTTP Endpoint to another:

Creating JAX-WS services

In this recipe, you will learn how to create a JAX-WS service, The JAX-WS specification defines a series of APIs and annotations that help you build web services.

Getting ready

Before creating a JAX-WS service, perform the following steps:

1. Open Mule Studio and enter a name for the workspace as shown in the following screenshot:

Y Workspace Launcher	×
Select a workspace Mule Studio stores your projects in a folder called a workspace. Choose a workspace folder to use for this session.	
Workspace: E:(MuleCookBook	Browse
Use this as the default and do not ask again	OK Cancel

2. To create a new project, go to **File** | **New** | **Mule Project**. Enter the project name, soap-jax-ws, and click on **Next** and then on **Finish**. Your new project is created. We can start the implementation now.

e Edit Source Refactor Navi	gate Search Project	Rup Window Help		
New	Alt+Shift+M	Mule Project	L. M. A. A.	
Open File	Page 200 CTTV	🖄 Java Project		
Close	CELLAW	Project		
Close All	Ctrl+Shift+W	Manning Eleve	—	L
J Save	Chrl+S	Mule Flow		
Save As		📅 Package		
Save All	Ctrl+Shift+S	🞯 Class		
Revert		🗊 Interface		
Move		🕼 Enum		
Rename	F2	Annotation		
Refresh	F5	B Jource Folder		
Convert Line Delimiters To		Folder		
Print	Ctrl+P	File		
Switch Workspace		Untitled Text File		
Restart		Example	-	
inport			_	
Export				
Export diagram to				
Properties	Alt+Enter			
1 FlowRef.mflow [flowref/flows]		_		
2 package.html [activity/src/main	v[]			
3 Activity.mflow [activity/flows]	wrter/ 1			
4 Sching rondinbert Java Lasaconn	verter)]	-		
Exit in outline is not available.				

How to do it...

After you're done with the steps mentioned in the *Getting ready* section, perform the following steps to create a JAX-WS service:

1. You begin by writing the service interface. Firstly, go to src/main/java, right-click
 on it, and create an interface called HelloWorld. You can write an operation called
 sayHello to greet anyone who submits his or her name through a web browser.

```
import javax.jws.WebService;
public interface HelloWorld {
    @WebService
    String sayHi(String text);
}
```

The following screenshot shows this process:





```
2. To create a class, right-click on src/main/java and create a class called
    HelloWorldImpl; implement it with the interface HelloWorld.
    import javax.jws.WebService;

    @WebService(endpointInterface = "com.org.HelloWorld",
        serviceName = "HelloWorld")
    public class HelloWorldImpl implements HelloWorld
    {
        @Override
        public String sayHi(String text) {
            // TODO Auto-generated method stub
            return "Hello"+text;
        }
    }
}
```

The following screenshot shows the implementation of HelloWorldImpl with HelloWorld:



3. Go to the SOAP-JAX-WS.mflow file. Drag the **HTTP** Endpoint onto the canvas. Double-click on the **HTTP** Endpoint to configure it. Enter the port number and path.

∑ Mule - soap-jax-ws/flows/Soa	Y Endpoint Properties	< Image: Image: Image:
File Edit Navigate Search Project	HTTP (Inbound Endpoint)	
i 📬 • 🔡 🗟 📄 i 🏇 • 🔘 •	The HTTP Endpoint allows Mule application to connect to web resources through the HTTP transport protocol. This endpoint can also implement security through HTTP5.	
Package E	General Advanced Deferences HTTP Settings Desumativity	
	Display	Elter:
🗊 🗁 filter	Display Name: HTTP	Ju
🗉 🔛 flowref		Select
🗈 🗁 hellomule	Exchange Patterns	🚰 Endpoints 🛛 🗠
The security_riter	One-way 🕑 request-response	皹 Ajax
ison	Basic Settings	Database
🗈 🔛 jsp	Enable HTTPS 0	EFTP
🗄 😂 logicfilter	Enabling HTTPS will require configuring a HTTPS Connector	Ele Ele
handbranster	Host: localhost	
augurtz	Port: 4234	Generic
🕀 🔛 quartz_scheduler	Path: - hello	() HIIP
restbasedwebservice	. and the	
restwebservice		Scopes 🗠
script		Async
🗄 🛋 JRE System Library [JavaSt		😹 Composite Source
🗉 🛋 Mule Runtime [Mule Server		Flow
🖨 🚰 src/main/java		O Foreach
🖃 🍻 com.org		Message Enricher
HelloWorldInpl.iava		
🗁 🕮 src/test/java		Control Control Control
- C src/main/resources		Sub How
- C src/test/resources		Components
Soap-JAX-WS.mflow	(?) OK Cancel	Carl Transformers
a poop sint to since the second secon		Pilters
🗉 😥 src		Control
mule-project.xml		Crror Handling
spelicheck	▼	Cloud Connectors
	Message Flow Global Elements Configuration XML	
i □ *	😥 i 🕴 🚼	@ 😡 N 🛛 # 🗄 🗍 # 🗳

4. To configure the interface, drag the **SOAP** component onto the canvas. Double-click on the **SOAP** component and configure the service class, and choose the operation **JAX-WS service**.

⊻ Mule - soap-jax-ws/flows/Soa	Y Pattern Properties	I I I I I I I I I I I I I I I I I I I
File Edit Navigate Search Project	SOAP	
	The SOAP Component will publish a SOAP web service via JAX-WS Annotations, WSDL, or CXF Simple Service using Apache CXE.	
	General Interceptors Advanced Documentation	- Ether
🕒 🔁 filter	Display Name: SOAP	Ju
flowref		Select
Image: Security_filter	Config Reference:	Carl Endpoints
🕀 🔁 jms	Operation: 10X-WS cervice	Scopes
🗉 🤛 json 🗄 🔛 jsp		Components 🗠
🗉 😂 logicfilter	Binding ID:	Expression
🗉 📂 nametransfer	Port	Elow Reference
🗉 🔛 quartz	Namespace:	Groovy
guartz_scheduler je restbasedwebservice	Service:	HTTP Response
restwebservice	Service Class: com.org.HelloWorld Generate from WSDL	Java
🗉 🔂 script		JS Javascript
😟 🛋 JRE System Library [Java58	Validation Enabled	E Logger
IIII → Mule Runtime [Mule Server		💽 Python
🖻 🌐 com.org		S REST
🗄 🛃 HelloWorld.java		Ruby
🗁 src/test/java		SOAP
src/main/resources		Script
E E flows		Transformers
Soap-JAX-W5.mHow	OK Cancel	Filters
🗉 😥 src		Control
■ mule-project.xml ■ Belicheck		Crror Handling
🗄 🔁 spring	Message Flow Global Elements Configuration XM	Cloud Connectors

-318-

5. Drag the Java component onto the canvas. Your flow will look like this:



6. Double-click on the **Java** component to configure it. Here, you import the HelloWorldImpl class that was created before.

∑ Mule - soap-jax-ws/flows/Soa	Y Pattern Properties	- - X
File Edit Navigate Search Project	Java	
i ⊡• 🛛 🕲 🖄 • 💽 •	The Java Component is a simple POJO (Plain Old Java Object) that will be invoked by Mule when a message is	
🖹 🔛 Mule		
📮 Package E 🛛 🏌 🎦 Type Hier	General Advanced Documentation	
E	Display	Filter:
🗄 😂 filter	Display Name: Java	N Salast
	Generic	12 Select
Helioniae	Class Name: Com.org. HelloWorldImpl	Endpoints
🖶 🔁 jms		Copes Scopes
😟 🔛 json	Object:	Components 🗠
🗄 🗁 jsp		(m-m) Echo
logichilter		Committee Expression
object-to-xml		Flow Reference
🕀 🔁 quartz		Groovy
🗉 😂 quartz_scheduler		HTTP Decreases
🗈 🗁 restbasedwebservice		Builder
erript		🛓 Java
soap-iax-ws		Javascript
🗊 🛋 JRE System Library [JavaSt		D Logger
🗉 🛁 Mule Runtime [Mule Server		Duther
🖃 🚰 src/main/java		Pychon
E ge com.org		KEST
HelloWorldIngl.java		Ruby
🕮 src/test/java		SOAP
- 🥮 src/main/resources		Script
- C src/test/resources		
Soap-14X-WS mflow		Carl Transformers
- Coop stat in station		Geren Filters
🗉 😥 src		Row Control
mule-project.xml		Caror Handling
spelicheck	~	Cloud Connectors
< III	Message Flow Global Elements Configuration XML	
		0 🕒 U # 🗄 # 🗐 🖉

How it works...

In this section, you will see how to deploy your application using Mule Studio.

1. To deploy the application code in the Mule server, go to **Run As** | **Mule Application**; the Mule server will deploy your application.

🍟 Mule - soap-jax-ws/f	lows/Soap-JAX-WS.mflow	- Mule Studio					- 7 🔀
File Edit Source Refacto	r Navigate Search Project	Run Window He	lp.				
8 📬 • 🖪 🗟 🔒 🕴	\$ • Q • Q • # @	• 🙋 🔗 •	🗰 🗄 🖢 - 🔁	* 🏷 🔶 -	- i 🤣 😂 🗶		
🗈 🔛 Mule	New	•	L				
📲 Package E 🛛 比 T	Open	F3	HelloWorldImpl.java	Soap-JAX-W	vs 🖾		
	Open With	+				Filter:	
🕀 🕞 filter	Show In	Alt+Shift+W 🕨				T IICO T	Ju
🗈 😂 flowref	Conv	Chrl+C				🔓 Selec	:t
🗈 😂 hellomule	Copy Qualified Name	carre				Carl Endpoin	its
E 🗁 http-security_filter	Reste	Ctrl+V				Scopes	
ison	💢 Delete	Delete				Compor	nents 👳
🗄 😂 jsp	n.din.ik					(mail Ech	no
🗄 😂 logicfilter	Build Path Refactor	∆lt+Shift+T ▶				tunnin Evr	pression
🗄 🔛 nametransfer		MICTORIUCT 1	8				u Defense er
Dect-to-xmi	import						WREIEFEILE
🗄 🔁 quartz scheduler	🛃 Export					😪 Gro	JOVY
🗉 😂 restbasedwebservic	🔗 Refresh	F5	SOAP	Java		HT Bui	TP Response ilder
🗈 🗁 restwebservice	Assign Working Sets		SUAP	Java		d. Jay	va
🗄 🗁 script	Validate						vascrint
	Run As	۱.	Y 1 Mule Application				- ascript
🗷 🛋 Mule Runtime [M	Debug As	•					jger
🖃 🥵 src/main/java	Profile As	+	Run Configuration	5		🜔 Pyt	:hon
🖻 🌐 com.org	Team	•				S RE:	ST
Hellowo	Compare With					💎 Ru	by
- 🗁 src/test/java	Replace With					SO 50	AP
🥮 src/main/resour	Cloud Hub					📄 Scr	ipt
- 🕮 src/test/resourc	Ciodanab	•					
E E flows	Properties	Alt+Enter				🚰 Transfo	rmers
	Similow					🔁 Filters	
src						Flow Co	untrol
mule-project.xm	1 I I I I I I I I I I I I I I I I I I I					Error Ha	andling
🗉 📂 spellcheck						Cloud C	ionnectors
< spring	Message F	low Global Element	s Configuration XML				
Soap-JAX-WS	.mflow - soap-jax-ws/flows		<u> </u>			# 🔝 @ 🕓 M	e 🔚 🕴 e 📮

2. Open a browser and paste this URL: http://localhost:4234/hello?wsdl. This will display the WSDL generated by CXF.



Creating web services using the REST component

Using **REST**, which stands for **Representational State Transfer**, applications can transmit the information needed to function as web services using only the native functions of a given protocol. In the context of web services, this generally means that RESTful web services communicate via pure HTTP using XML or JSON to encapsulate the data and metadata.

Getting ready

In this section, you will use three components: the HTTP Endpoint, the Logger component, and the REST component. The REST component is used for creating a REST-based web service.

1. Open Mule Studio and enter a name for the workspace as shown in the following screenshot:

Workspace Launcher	×
Select a workspace	
Mule Studio stores your projects in a folder called a workspace. Choose a workspace folder to use for this session.	
Workspace: E:\MuleCookBook	Browse
Use this as the default and do not ask again	
	OK Cancel



2. To create a new project, go to **File** | **New** | **Mule Project**. Enter the project name, restbasedwebservice, and click on **Next** and then on **Finish**. Your new project is created. We can start the implementation now.

🖌 Mule - Mule Studio 🔄				
File Edit Source Refactor Navio	gate Search Project	Run Window Help		
New	Alt+Shift+N	🕨 🚰 Mule Project	- ♥= (⇒) -	
Open File		🏄 Java Project		
Close	Ctrl+W	📑 Project		- F
Close All	Ctrl+Shift+W	Apping Flow		
🖁 Save	Ctrl+5	Mule Flow		
Save As		📸 Package		
🔞 Save All	Ctrl+Shift+S	🞯 Class		
Revert		🗊 Interface		
Move		ઉ Enum		
Rename	F2	Annotation		
🔊 Refresh	F5	Source Folder		
Convert Line Delimiters To		Sava Working Set		
👜 Print	Ctrl+P	File		
Switch Workspace		Intitled Text File		
Restart				
and Import		Example		
A Export		📸 Other Ct	rl+N	
in Course discusses in		-		
Export diagram to		_		
Properties	Alt+Enter			
1 FlowRef.mflow [flowref/flows]				
2 package.html [activity/src/main	4]			
3 Activity.mflow [activity/flows]	autor 1			
4 Schingronumber, Java [usuconv	erter)]	-		
Exit Arr outling is not available.				
•				: a 💦 @ 🔃 M 🗉

How to do it...

In this section, you will see how to create a REST-based web service using Java annotations. Java annotations are one of the main development features that was introduced in JDK 5. Annotations are like meta tags that you can add to your code and apply to package declarations. Annotations carry data over a runtime.

1. Go to src/main/java, right-click on it, and create a class called HelloWebService.

```
import javax.ws.rs.GET;
import javax.ws.rs.Path;
@Path("/myrest")
public class HelloWebServices {
   @GET
```



```
public String getWelcomemsg() throws Exception
{
   return "Hello RestBased Webservice";
}
```

The following screenshot shows the creation of this class:



2. Drag the **HTTP** Endpoint onto the canvas. Double-click on the **HTTP** Endpoint to configure it. Enter the port number as 3232.

∑ Mule - restbasedwebservice/	Y Endpoint Properties	X	🔳 🗗 🔀
File Edit Navigate Search Project	HTTP (Inbound Endpoint)		
1 📑 • 🖩 🖻 🖻 1 🏇 • 🔘 •	The HTTP Endpoint allows Mule application to connect to web resources through the HTTP transport protocol.		
😰 🔛 Mule	This endpoint can also implement security through HTTPS.		
洋 Package E 🕺 🍃 Type Hier	General Advanced References HTTP Settings Documentation		
Θ	C Display		Filter:
🗉 😂 object-to-xml	Display Name: HTTP		
🖲 😂 quartz			la Select
🖶 🗁 quartz_scheduler	Exchange Patterns		Carl Endpoints 🛛 🗠
restbasedwebservice	O one-way () request-response		🙉 Aia×
JRE System Library [Javase Mule Puntime [Mule Server	Basic Settings		Databace
Gride Kandine Engle Berver	Enable HTTPS 0		
😑 🌐 com.org	Enabling HTTPS will require configuring a HTTPS Connector		E FTP
🗷 🚺 HelloWebServices.j.	Unative localization	1	🨑 File
- 🕮 src/test/java			Generic
src/main/resources	Port: 3232		(iiiiii) HTTP
src/test/resources	Path:		
Provis			
Restbasedwebservice.			Scopes 🗠
src			Async
😟 🗁 main			Composite Source
🦾 🗁 test			Flow
mule-project.xml			O Foreach
restwebservice			Massas Fasisher
			Message Enricher
spellcheck			CC Poll
spring			Sub flow
🕫 🔁 studioconnector			Components
🔹 😂 testtransformer			Carl Transformers
🛛 🖼 🗁 transformer	OK Cance		C Filters
🖬 🗁 twiiter			
twiterintegration			How Control
web-services			Crror Handling
webservice-cxf			Cloud Connectors
	Message Flow Global Elements Configuration XML		
i ∎*	😪 i i 🖉	2 @	🔒 И 🛛 е 🔚 🗍 е 🔲

3. Drag the **Logger** component onto the canvas. The **Logger** component is used for displaying logs on the console. Double-click on the **Logger** component to configure it.

∀Mule - restbasedwebservice/	Y Pattern Properties] 📃 🖬 🗙
File Edit Navigate Search Project	Longer	
i 📫 • 🔛 🗟 🗎 🏇 • 🔘 •	The Longer Component performs longing using an expression that determines what should be longed. By	
😭 🔛 Mule	default the current messages is logged using the DEBUG level.	
洋 Package E 🕺 🍃 Type Hier	General Documentation	
E	Display	Filter:
🗉 😂 object-to-xml	Display Name: Logger	J u
🗉 🔛 quartz		Select
🗉 📂 quartz_scheduler	Generic	and points 🗠
restbasedwebservice	#[payload]	👰 Ajax
JRE System Library [Javast Mule Runtime [Mule Server	Message:	Database
□ / / / / / / / / / / / / / / / / / / /		
😑 🌐 com.org	Level: TNFO	
🗈 🚺 HelloWebServices.j.		File
- 🥭 src/test/java	Category:	Generic
src/main/resources		(iii) НТТР
Sicilescresources		IMAP
RestBasedWebservice.		Scopes 🐡
🗁 mappings		Artime
🛱 🗁 src		Async
🗈 🗁 main		Composite Source
test		Flow
restwebservice		Q Foreach
🗄 🔁 script		••= Message Enricher
🗷 😥 soap-jax-ws		C Poll
spellcheck		WHI Sub flow
🗄 📅 spring		
studioconnector		Components
Transformer		a Transformers
🗄 🔁 twiiter		Gereillers
🗈 🔛 twiiterintegration		Control
usdconverter		Caror Handling
web-services	~	Cloud Connectors
Webservice-cxi	Message Flow Global Elements Configuration XML	
i □ ◆	😥 ::	@ 🚱 N # 🔠 # 🗳

4. Drag the **REST** component onto the canvas. Double-click on it and configure it.



How it works...

In this section, you will see how to deploy applications in Mule Studio.

1. To deploy the application code in the Mule server, go to **Run As** | **Mule Application**; the Mule server will deploy your application.

🖌 Mule - restbasedwebservio	ce/flows/RestBasedWebserv	ice.mflow - Mule Studio				
File Edit Source Refactor Navigate Search Project Run Window Help						
i 📬 • 🔲 🕞 🖻 i 🎄 • 🌔						
Package E 🛛 👔 Type Hie	r 🗌 🕗 HelloWebServic	es.java 🍟 RestBasedWebservice 🖄 🔪				
			Filter:	Tr		
🗉 😂 object-to-xml	~		N. S. L. L			
🗈 🔛 quartz			12 Select			
uartz_scheduler			Carl Endpoints			
resubasedwebservice	(aSE-1.6]		Copes Copes			
Mule Runtime [Mule Ser	ver 3.3.0		Components	400		
😑 🇀 src/main/java	New	•	(mett) Echo			
😑 🌐 com.org	0	52	Gaussie Expression			
HelloWebServic	Open With	F3 . 2/				
src/test/java	Show In	Alt+Shift+W		rence		
# src/test/resources		/	Groovy			
🖃 🧁 flows	Copy	Ctri+C er REST	HTTP Res	ponse		
RestBasedWebserv	Copy Qualified Name	er REST	lava			
mappings	Paste	Ctrl+V				
src	💢 Delete	Delete	38 Javascripo	-		
test	Build Path	•	Logger			
mule-project.xml	Refactor	Alt+Shift+T	Python			
🗉 😂 restwebservice	Pag Import		REST			
🗄 📴 script	Export		Ruby			
soap-jax-ws			SOAP			
	🖑 Refresh	F5	B Script			
🖶 😂 studioconnector	Assign Working Sets		- Series			
🗉 🔛 testtransformer	Validate					
🗈 😂 transformer	Run As	🕨 🎽 1 Mule Application				
🗄 📂 twiiter	Debug As	Bup Configurations	Charles and			
converter	Profile As		Flow Control			
web-services	Compare With		Error Handling			
🗄 🔁 webservice-cxf	Replace With		Cloud Connect	ors		
<	Mule Management Console	guration XML				
RestBasedWebservice		🕨 💼 🔛	j 8 🖹 @ 😡 M 🗍 8 🖁	e 🗄 e 🖳		

2. Open a browser and paste the URL http://localhost:3232/myrest in the browser.





Calling external web services using the SOAP component

In this recipe, you will see how to create a simple web service using Eclipse in Java; this web service will be called in Mule Studio using the SOAP component. You will see how to call external web services in Mule. You will then create a simple web service that will display the current date and time and the reverse string.

Getting ready

In this section, we will create a web service in Eclipse.

1. Open Eclipse and enter a name for the workspace. Create a class called RequestHandler. In this class, we create two methods: one to display the current time and date and the other to generate the reverse string.

```
import java.text.SimpleDateFormat;
import java.util.Date;
public class RequestHandler {
    public String reverseYourName(String name)
    {
       return new StringBuffer(name).reverse().toString();
    }
    public String getCurrentDate()
    {
       Date date= new Date();
       SimpleDateFormat dateFormat=new SimpleDateFormat("dd-MMMMM-
yyyy h:mm a");
       return dateFormat.format(date);
    }
}
```

329

The following screenshot shows the creation of this class:





2. The following screenshot shows the class you created as a web service. We will generate the WSDL file as well.



331

How to do it...

In this section, you will see how to run the WSDL file and see the output on the WSDL page.

1. Right-click on the RequestHandler.wsdl file and publish the WSDL file. Once the file is successfully published, you will see something similar to the following screenshot on your system. You can run the .wsdl file in two ways: right-click on the .wsdl file or right-click on the project and run the .wsdl file. You will see the output on the **Status** tab.





2. Open Mule Studio and enter the workspace name as shown in the following screenshot:

¥ Workspace Launcher	X
Select a workspace Mule Studio stores your projects in a folder called a workspace. Choose a workspace folder to use for this session.	
Workspace: E:\MuleCookBook	Browse
	OK Cancel

3. To create a new project, go to **File** | **New** | **Mule Project**. Enter the project name, external_webservice, and click on **Next** and then on **Finish**. Your new project is created. We can start the implementation now.

🎽 Mule Debug - Mule Studio			
File Edit Navigate Search Project F	Run Window He	lp	
New	Alt+Shift+N	🕨 🚰 Mule Project	
Open File		🏄 Java Project	
Close	Ctrl+W	Project	
Close All	Ctrl+Shift+W	🞽 Mule Flow	
🕌 Save	Ctrl+S	🖶 Package	
📓 Save As		🞯 Class	
i Save All	Ctrl+Shift+S	🞯 Interface	
Revert		() Enum	
Move		Annotation Sector Californ	
Rename	F2	Java Working Set	
Refresh	F5	Eolder	
Convert Line Demnicers 10		File	
i Print	Ctrl+P	📑 Untitled Text File	
Switch Workspace	1	Evample	
Restart		L Example	
🚵 Import		Cther	Ctrl+N
🛃 Export			
Export Studio Documentation			
Export diagram to			
Properties	Alt+Enter	-	
1 amazon sgs.mflow [amazon-s3/flows	;]	-	
2 twiiter.mflow [twiiter/flows]			
3 File Transfer.mflow [file_transfer/]	l		
4 Transformer.mflow [transformer/flow	/s]	_	
Exit			

4. Go to the External WebService.mflow file. Drag the **HTTP** Endpoint onto the canvas and double-click on it to configure it.

⊻ Mule - external_webservice/I	Y Endpoint Properties	📕 🖬 🖾
File Edit Navigate Search Project	HTTP (Inhound Endpoint)	
📬 • 🔛 🗟 🗎 🏇 • 💽 •	The HTTP Endpoint allows Mule application to connect to web resources through the HTTP transport protocol.	
😭 🕎 Mule	This endpoint can also implement security through HTTP5.	
📮 Package E 😫 🍃 Type Hier	General Advanced References HTTP Settings Documentation	
Ξ	Display	Filter:
🗄 😂 customfilter	Display Name: HTTP	JU
H Cxf	Exchange Patterns	13 Select
dropbox integration	O ope-way () request-response	Endpoints 🗢
😑 🔁 external_webservice		🥦 Ajax
🗷 🛋 JRE System Library [JavaSB	Basic Settings	🕞 Database
Hule Runtime [Mule Server]	Enable HTTPS ¹⁰	FTP
src/main/java	Enabling HTTPS will require configuring a HTTPS Connector	rile 🔁
- # src/main/resources	Host: localhost	Generic
- 🕮 src/test/resources	Port: 0 2121	П НТТР
🖹 🗁 flows	Path:	
External WebService.m		IMAP
		Scopes 👳
mule-project.xml		Async
🗉 📂 file_transfer		🙈 Composite Source
🗄 😂 filter		Flow
Flowref		Q Foreach
Helioniale Helioniale		•• Message Enricher
🗊 🔛 jms		C Poll
🗉 😂 json		Sub flow
🕀 🔛 jsp		JOD TOW
Importanter		Components
	(?) OK Cancel	Carl Transformers
🗉 🔛 quartz		Carl Filters
🗉 😂 quartz_scheduler		Control
🗉 📂 restbasedwebservice		Carl Error Handling
restwebservice	✓	Cloud Connectors
	Message Flow Global Elements Configuration XML	
. ∎ ♦	🚓 ::	0 💽 M 🛛 🖉 🗄 🖉

-334-

5. Drag the **SOAP** component onto the canvas. Double-click on the **SOAP** component to configure it. First click on the **Generate from WSDL** button; you will have to enter the WSDL URL, http://localhost:8080/WebService/services/RequestHandler?wsdl; this URL was created in the external web service in Eclipse. Enter the package name as com.org. This package name comes from the external web service. Click on the **OK** button.

⊻ Mule - external_webservice/I	🛛 Pattern Properties	🗙 🗖 🔁 🔀
File Edit Navigate Search Project	SOAP	<u> </u>
: 🖸 • 📓 🕼 🖆 : 🕸 • 🚺 •	The SOAR Component will publich a SOAR web cervice via 10X-WS Apportations. WSDL or CXF Simple Service	
🖹 🔛 Mule	Generate from WSDL	
洋 Package E 🙁 🍃 Type Hier	Generate from WSDL	
•	Import WSDL and generate JAX-WS classes.	Filter: so
🗉 🔁 customfilter		
🗷 📴 cxf	WSDL Source	13 Select
datamapper	Use WSDL loaded from a URL	🟫 Suggestions 🗠
Gropbox_Integration	WSDL URL: ttp://localhost:8080/WebService/services/RequestHandler?wsdl	S REST
Bre System Library [JavaSt	O Use WSDL from a file in the current project.	SOAP
🗈 🛋 Mule Runtime [Mule Server	WSDI File	
- 🥭 src/main/java	WOULFING:	Scopes 🗠
- 🥮 src/test/java	Source Generation	Composite Source
src/main/resources	Package Name: com.org	
src/test/resources		
External WebService m		
- C> mappings		
🗈 🗁 src	Generate from WSDL	
mule-project.×ml		
🗄 🔛 file_transfer		
⊞ 📂 Filter		
+ Helonide		
ims		
🗷 🗁 json		
🗄 😂 jsp		
🗈 🔛 logicfilter		
H 🗁 nametransfer		¬
Object-to-xmi	OK Cancel	
🕫 🔁 restbasedwebservice		
😐 😂 restwebservice		Compared to
🗄 🗁 script	Massan Elan Clabel Elanasta CarGametica VM	
	Message How Godal Liemencs Configuration XmL	
	🚓 i i 🖉 🔭	@ 💁 M 🛛 a 🔚 🗋 a 🚍

335

6. You will see that, under src/main/java, classes are created automatically, as shown in the following screenshot:



How it works...

In this section, you will see how to deploy the application.

1. To deploy the application code in the Mule server, go to **Run As** | **Mule Application**; the Mule server will deploy your application. At the same time, you should run the Tomcat server in Eclipse.

₩ Mule - external_web	New	•	- Mule Studio		
File Edit Source Refacto	Open Open With Show In	F3 Alt+Shift+W	9 100 : 월 - 원 - 박 수 • 수 · : (
Package E 🕺 🔭 T	Сору	Ctrl+C	xternal WebService 🛛		
1	Copy Qualified Name			Filter: so	
🗄 🗁 customfilter	Paste	Ctrl+V		Ju	
🗷 🗁 cxf 🛛 🗶	Delete	Delete		Select	
🗄 🔛 datamapper	Build Path	•		Suggestions ∞	
🗄 🚰 dropbox_integration	Refactor	Alt+Shift+T 🕨		REST	
external_webservici	Import			SOAD	
🗄 🛋 Mule Runtime [M	Export			JOAP	
🖨 🤔 src/main/java	Export			Scopes 🗠	
😑 🔑 com.org 🛷	Refresh	F5	iowi P	🐼 Composite Source	
🗄 🗾 GetCurr	Assign Working Sets				
GetCurr GetCurr	Validate				
E D package	Run As	E State	🕌 1 Run on Server 🛛 Alt+Shift+X, R		
E Deckage	Debug As	•	Y 2 Mule Application		
🕀 🚺 Request	Profile As	•	Due Carlie methods		
🕀 🚺 Request	Team		Run Conngurations		
🗄 🗾 Reverse	Compare With				
Reverse	Replace with				
- Chesci Java ()	Mule Management Console		SOAP		
🦉 src/test/resourd	Ciouunub				
🖻 🗁 flows	Properties	Alt+Enter			
External WebSer	vice.mflow				
mappings					
mule-project.xml					
🗉 🔂 file_transfer					
🗉 🔛 filter					
flowref					
hellomule					
ims	~			Components	
	Message Flo	Global Element	s Configuration XML		
External WebServio	😳 External WebService.mflow - external_webservice/flows 🏠 🗄 🕫 🔁				

2. Once you have successfully deployed the application, you will see the following output on your console:

¥ Mule - external_webservice/flows/External WebService.mflow - Mule Studio	
File Edit Navigate Search Project Run Window Help	
╡┇╸╝╺╠╝╸┇╬╸Q╸╉╸╡╋╺Q╸╡ ╚╶ ╎╘ ╔ ╶┊╔╶┊┚┈╬╺╬╸┽╴╗╗	
E Mue	
🖸 Console 🛛 🛛 📕 🕷 💭 🚱 🕐 🖾 • 😭 • 🖓 • 🖓	
External WebService [Mule Application] C:\Program Files\Java\jre6\bin\javaw.exe (Apr 25, 2013 6:23:29 PM)	-
createMultipleTransactedReceivers=true	JU
connected=true	
supportearrotocols=[http]	
Setviceoverrides-(hone)	
INFO 2013-04-25 18:23:33,609 [main] org.mule.lifecycle.AbstractLifecycleManager: Starting connector: connector.http.mule	
INFO 2013-04-25 18:23:33,625 [main] org.mule.lifecycle.AbstractLifecycleManager: Starting model: muleSystemModel	
INFO 2013-04-25 18:23:33,625 [main] org.mule.construct.FlowConstructLifecycleManager: Starting flow: External_WebService	
INFO 2013-04-25 18:23:33,625 [main] org.mule.processor.SedaStageLifecycleManager: Starting service: External_WebServiceF	
INFO 2013-04-25 18:23:33,640 [main] org.mule.transport.http.HttpConnector: Registering listener: External_WebServiceFlot	
INFO 2013-04-25 18:23:33,640 [main] org.mule.transport.service.DefaultTransportServiceDescriptor: Loading default respo	
INFO 2013-04-25 18:23:33,640 [main] org.mule.lifecycle.AbstractLifecycleManager: Initialising: 'null'. Object is: HttpMe	
INFO 2013-04-25 18:23:33,656 [main] org.mule.lifecycle.AbstractLifecycleManager: Starting: 'null'. Object is: HttpMessa	
INFO 2013-04-25 18:23:33,656 [main] org.mule.module.management.agent.WrapperManagerAgent: This JVM hasn't been launched	
INFO 2013-04-25 18:23:33,687 [main] org.mule.module.management.agent.JmxAgent: Attempting to register service with name:	
INFO 2013-04-25 18:23:33,687 [main] org.mule.module.management.agent.JmxAgent: Registered Endpoint Service with name: M	
INFO 2013-04-25 18:23:33,887 [main] org.mule.module.management.agent.JmxAgent: Registered connector Service with name M	
INFO 2013-04-23 10:23:33,607 [main] org.mule.module.launcher.appitation.betautchulexppitation: Reload Interval: 3000	
INFO 2013-04-23 10:23:33,000 [main] olg.mule.belauthuideonteeo:	
* innlightion: external vehervice *	
* OS encoding: Ch1252. Mule encoding: HTF-8 *	
* *	
* Agents Running: *	
* JMX Agent *	

INFO 2013-04-25 18:23:33,687 [main] org.mule.module.launcher.DeploymentService:	
+++++++++++++++++++++++++++++++++++++++	
+ Started app 'external_webservice' +	
+++++++++++++++++++++++++++++++++++++++	
×	
· · · · · · · · · · · · · · · · · · ·	

3. Open a browser and paste the URL http://localhost:2121/?wsdl in your browser bar.



-338

9 Understanding Flows, Routers, and Services

In this chapter you will learn different types of Routers/Flow Controls, and also the following:

- ► Configuring the All Router/Flow Control
- ► Configuring the Choice Router/Flow Control
- Configuring the Splitter Flow Control

Introduction

Flow Controls route messages to various destinations in a Mule flow. Some Flow Controls in business logic are implemented to study and possibly transform messages before routing takes place. Through Flow Controls, you will see how messages are sent and received within a Mule flow. In this chapter, you will see all types of Routers/Flow Controls. There are different types of Routers in Mule Studio: the **Choice Router**, the **All Router**, and the **Splitter**.

Configuring the All Router/Flow Control

The All Router/Flow Control is used for sending a message to multiple targets. It is also used to route the same message to more than one processor component.

Understanding Flows, Routers, and Services -

Getting ready

In this example, you will use four components: the File Inbound Endpoint, the File Outbound Endpoint, the All Router, and the Echo component.

1. Open Mule Studio and enter the workspace name as shown in the following screenshot:

Y Workspace Launcher	X
Select a workspace	
Mule Studio stores your projects in a folder called a workspace. Choose a workspace folder to use for this session.	
Workspace: El/MuleCookBook	Browse
Use this as the default and do not ask again	
	OK Cancel

2. To create a new project, go to File | New | Mule Project. Enter the project name, allrouter, and click on Next and then on Finish. Your new project is created, now you have to start the implementation.

Edit Source Refactor Navigate Search Project Run Window Help New Ab+Shift-HN Open File Mapping Flow Close All Ctrl+Shift-HV Close All Ctrl+Shift-HV Save As Project Save As Ctrl+Shift-HV Save As Ctrl+Shift-HV Save As Ctrl+Shift-HV Save As Ctrl+Shift-HV Save As Ctrl+Shift-HS Save As Ctrl+Shift-HV Save As Ctrl+Shift-HS Save As Project Save As Paramet Paramet F2 Refresh F5 Source Folder Source Folder Convert Line Delimiters To Source Folder Print Ctrl+P Stubt, Workspace Example Restart Example Import Ctrl+P Stubt, Mono (activity/isrc/main/] Att+Enter 1 FlowRef.mflow [flows] Att+Enter 1 FlowRef.mflow [activity/isrc/main/] Att+Enter 1 FlowRef.mflow [activity/isrc/main/]	🎽 Mule - Mule Studio			
New Alk+Shit+N Image: Constraint of the state of	File Edit Source Refactor Navigate	Search Project	Run Window Help	
Open File Image: Save Project Close Al Ctrl+W Close Al Ctrl+Shift+W Save Al Ctrl+Shift+S Save Al Ctrl+Shift+S Revert Image: Ctrl+Shift+S Project Image: Ctrl+Shift+S Rename F2 Package Image: Ctrl+Shift+S Rename F2 Project Image: Ctrl+Shift+S Project F5 Convert Line Delmiters To Source Folder Print Ctrl+P Switch Workspace Image: Ctrl+P Print Ctrl+P Switch Workspace Untitled Text File Restart Untitled Text File Export Other Properties Alt+Enter 1 FlowRef.mflow [flows] Satury.mflow [ctrl+Kyflows] 3 struty.mflow [ctrl+Kyflows] Satury.mflow [ctrl+Kyflows] 4 StringTo	New	Alt+Shift+N	🕨 🚰 Mule Project	
Close Ctrl+W Close All Ctrl+Shift+W Sove Al Ctrl+Shift+W Save As Package Save Al Ctrl+Shift+S Revert Ctrl+Shift+S Move Ctrl+Shift+S Revert Ctrl+Shift+S Move F2 Ramane F2 Switch Workspace Source Folder Restart Print Import Ctrl+P Studgram to Properties Properties Alt+Enter 1 FlowRef.mflow [flows] 4 stringToNumber.java [usdconverter/] Ext Ext StringToNumber.java [usdconverter/] Ext Ext Ture strates	Open File		/ Java Project	
Cbrie All Cbrie Shift+W Mapping Flow Save As Mule Flow Save As Package Save All Cbrie Shift+S Revert Interface Move Package Revert Interface Move F2 Refresh F2 Switch Workspace F5 Switch Workspace F1 Switch Workspace F2 Report Chriefflows] Import Chriefflows] Storperture filows/filows] Alx+Enter 1 FlowRef.milow [flowerf/flows] Alx+Enter 1 StorgeToNumber.java [usdconverter/] Example Exat Storgetontocover	Close	Ctrl+W	🏥 Project	
Save Ctrl+S While Flow Save AS Ctrl+Shit+S Class Save AI Ctrl+Shit+S Class Move Forma Forma Remarks F2 Source Folder Convert Line Delimitiers To Source Folder Switch Workspace F3 Source Folder Switch Workspace Example Restart Winbled Text File Export Other Properties Alt+Enter 1 flowfer.min/wo [flows] Source Folder 2 package.html [activity/src/main/] Source Folder Source Folder Source Folder * StringToNumber.java [usdconverter/] Example Ext Source Folder * StringToNumber.java [usdconverter/] Ext Ext Text * StringToNumber.java [usdconverter/] Ext	Close All	Ctrl+Shift+W	🔐 Mapping Flow	
Save A3 Save A3 Save A3 Save A3 Save A3 Save A3 Save A3 Save A3 Save A3 Save A3 Save A3 Save A3 Save A3 Save A3 Save A3 F2 Save A4 F3 Save A4 F4 Save A4 F4 Save A4 F4 Save A4 F5 Save A4	📙 Save	Ctrl+S	Mule Flow	
Save Al Ctrl+ShF+5 G Cass Revert Move Rename F2 Refresh F5 Convert Line Delimiters To Print Ctrl+P Switch Workspace Restat Export dagram to Properties Alt+Enter I flowRef.nflow [flowref/flows] S package kines/flows] Sativity.mflow [activity/flows] ShtimgToNumber.java [usdconverter/] Exi	📓 Save As		🖶 Package	
Revest Move Maname Paname Print Convert Line Delimiters To Print Print Chri+P Print Chri+P Switch Workspace Restart Import Export Export Export Export Properties Alt+Enter 1 flowRef.mflow [flows] 2 backage.hml (activet/yflows] 4 StingToNumber.java (usdconverter/] Exit	i Save All	Ctrl+Shift+S	Class	
Move F2 Refresh F5 Convert Line Delimiters To Source Folder Print Ctrl+P Print Ctrl+P Print Ctrl+P Swtch Workspace Image: Ctrl+P Restart Image: Ctrl+P Import Export Export Other Export Other Stargent to Properties Alx+Enter I Flowef.mflow [flows] 2 package.html [activity/src/mai/] S Activity.mflow [ctrl/tflows] 4 StringToNumber.java [usdconverter/] Exit	Revert		🕜 Interface	
Refresh F2 F4 Housson Parensh F5 Source Folder Convert Line Delmiters To Folder Print Ctrl+P Folder Switch Workspace File Example Import Example Example Import Other Ctrl+N Export. Other Ctrl+N Export. Other Ctrl+N Export Other Ctrl+N Export Other Ctrl+N Export Other Ctrl+N Export StringToNumber.java [usdconverter/] Example Example IstingToNumber.java [usdconverter/] Example Ext Total convertion or ansure Total convertion or ansure	Move		Enum Appetation	
Series For Convert Line Delimiters To Convert Line Delimiters To Print Properties Alt+Enter 1 flowRef.mflow [flows] 4 StringToNumber.java [usdconverter/] Ext	Rename	F2	Source Folder	
Print Ctrl+P Switch Workspace Pile Restart Inport Import Export Export Other Properties Alt+Enter 1 HowRef.mflow [flows] 2 package.html [activity/flows] 2 package.html [activity/flows] 4 StringToNumber.java [usdconverter/] Exit StringToNumber.java [usdconverter/]	Convert Line Delimiters To	F5	Java Working Set	
Print Ctrl+P Switch Workspace Prior Restart Import Import Export Export Other Export Other Properties Alt+Enter 1 FlowRef.mflow [flowref/flows] 2 package.html [activity/flows] 3 Activity.mflow [activity/flows] 4 StringToNumber.java [usdconverter/] Exit Exit			Folder	
Switch Workspace Restart Subtch Workspace R	Print	Ctrl+P	📑 File	
Restart Example Strampler Ctrl+N Sother Sother Sother Sother Sother Sother Sother Sother Sother Sother Sother Sother	Switch Workspace		Untitled Text File	
Import Export Ctrl+N Ct	Restart		Example	
Export Export degram to Properties Alt+Enter I flowRef.mflow [flows] 2 package.html [activity/src/man/] 3 Activity.mflows] 4 StringToNumber.java [usdconverter/] Ext	🚵 Import			CIRLEN
Export dagram to Properties Alt+Enter I FlowRef.mflow [flowes] 2 package.html [activity/src/main/] 3 Activity.mflow [activity/flowes] 4 StringToNumber.java [usdconverter/] Exit	🛃 Export		U outer	CUITIN
Properties Alt+Enter 1 FlowRef.mflow [flows] 2 package.html [activity/src/main/] 3 Activity.mflow [activity/flows] 4 StringToNumber.java [usdconverter/] Exit	😰 Export diagram to			
1 FlowRef.mflow [flowes] 2 package.html [activity/src/main/] 3 Activity.mflow [activity/flows] 4 StringToNumber.java [usdconverter/] Exit Toroutine or analysis	Properties	Alt+Enter		
2 package.html [activity/src/main(] 3 Activity.mflow [activity/fibws] 4 StringToNumber.java [usdconverter/] Exat	1 FlowRef.mflow [flowref/flows]			
4 StringToNumber (ava [usdonverter/] Exat	2 package.html [activity/src/main/]			
Exit	4 StringToNumber.iava [usdconverter	d]		
	Evit		-	
	All outline is not available.			
□◆				
	•			

How to do it...

In this section, you will see how to use components in a graphical flow and how it works.

1. Go to the allrouter.mflow file and drag the **File** Inbound Endpoint. Double-click on the **File** Endpoint and configure it. Provide a full path name.

⊻ Mule - allrouter/flows/AllRou	🖌 Endpoint Properties	
File Edit Navigate Search Project	File (Inbound Endpoint)		
i 📑 • 🗄 🗟 🖻 i 🏇 • 💽 •	The File Endpoint allows Mule applications to read or write files on the local file system.		
🖹 🔛 Mule			
洋 Package E 🙁 🍃 Type Hier	General Advanced References Documentation		
E	C Display		Filter:
🗉 🔛 ajex	Display Name: File		Ju
alrouter	- Dath Information		Select
JRE System Library [Java5t Mule Runtime [Mule Server	Path in on addition		Contraction Endpoints 🗠
- 2 src/main/java	Paul:		🔎 Ajax
🥮 src/test/java	Move to Pattern:		😑 Database
src/main/resources	Move to Directory:		😑 FTP
□ → flows	Polling Information		🔁 File
AlRouter.mflow	Polling Frequency;		Generic
mappings	File Age:	_	() HTTP
mule-project.xml			IMAP
😑 😂 choice_router	File Name Filter		Scopes 🗠
🗈 🛋 JRE System Library [JavaSt	File Name Regex Filter:		Async
Mule Runtime [Mule Server			Composite Source
Src/test/java			Flow
- 🥮 src/main/resources			O Foreach
Contest/resources			Mossage Envictor
Choice Router.mflow			
- 🗁 mappings			Colo Rom
🗈 🗁 src			Sub now
mule-project.xmi			Components
€ Customfilter	(?) ОК	Cancel	Transformers
🗉 😂 cxf			Pilters
datamapper			Flow Control
external webservice			Error Handling
🗉 🚰 file transfer			Cloud Connectors
	Message Flow Global Elements Configuration XML		
	8	e 🚼 @	😣 M 🛛 a 🔚 🖉 a 📮

Understanding Flows, Routers, and Services -

2. Drag the Echo component onto the canvas; it's used for display purposes.



- ₩ Mule allrouter/flows/AllRouter.mflow Mule Studio File Edit Navigate Search Project Run Window Help 🔁 • 🔛 🗟 🗄 🕸 • 🔕 • 🔮 🞯 • 🛯 🖉 🔗 • 🗄 📾 🗄 🧏 • 🖓 • 🖓 • 🖓 🔛 🔛 Mule 😫 Package E 🛛 🐮 Type Hier 🛛 🖓 Choice Router 🛛 😭 *AllRouter 😒 - -8 🕒 🔄 🎽 Filter: Ju ^ Select A JRE System Library [JavaSE-1.6]
 A Mule Runtime [Mule Server 3.3.0 endpoints 皹 Ajax 进 src/main/java src/test/java
 src/main/resources 😑 Database 😑 FTP 🕮 src/test/resources 😑 File ■ Intersection and the section of the section o Generic 🗁 mappings flow: AllRouterFlow1 📝 HTTP src 🖂 IMAP choice_router
 A JRE System Library [JavaSE-1.6] Copes ([[] +])) Echo Async 💼 🔜 Mule Runtime [Mule Server 3.3.0 FileTransfer All src/main/java
 src/test/java 🛃 Composite Source Flow src/main/resources 1 Q Foreach 😬 src/test/resources Flows ••• Message Enricher 🕒 Poll 🗁 mappings Sub flow src Components 🗄 🔁 corn-expression a Transformers 🗉 🔛 customfilter Filters 🗄 📂 cxf datamapper
 dropbox_integration Flow Control arror Handling Corpose_integration
 Corpose_integration
 Corpose_integration
 Corpose_integration
 Corpose_integration
 Corpose_integration Cloud Connectors Message Flow Global Elements Configuration XML > ∎\$ 80 i e 🔝 @ 😡 M 🛛 e 🔚 🗍 e 🚍
- 3. Drag the **Choice** Router/Flow Control onto the canvas; it's used for sending messages to multiple targets.

Understanding Flows, Routers, and Services -

4. Drag the **File** Outbound Endpoint onto the canvas; the **File** Endpoint is chosen as the destination path.



5. Double-click on the **File** Endpoint to configure it. Click on the ... button and configure the destination path.



Understanding Flows, Routers, and Services -

6. Drag the second **File** Outbound Endpoint onto the canvas. Double-click to configure it. Configure it in the same way as the previous one.



7. Drag the third **File** Endpoint. As you send the file to multiple targets, you use three **File** Outbound Endpoints. The file is sent to three different locations.

Y Mule - allrouter/flows/AllRou		×	💶 🗗 🔀
Mule - alirouter/flows/AliRou Fie Edt Navigate Search Project Mule - alirouter/flows/AliRou Package E	File (Outbound Endpoint) The File Endpoint allows Mule applications to read or write files on the local file system. General Advanced References Documentation Display Name: File Path: C:Documente and Gattineol assards all Docklaps (Output Pattern: Browse For Folder Image: Online Training Mule Output Output: OK	Cancel	Filter: Select Filter: Select Filter: Select Filter: Filter: Select S
			Pilters
external_webservice	Message Flow Global Elements Configuration XML		Cloud Connectors
1 •		i # 🖹 @	🕒 M # 🗄 # 🗐
8. Drag the **Logger** component onto the canvas. Double-click on the **Logger** component to configure it. It's used for displaying a log on the console.

⊻ Mule - allrouter/flows/AllRou	Y Pattern Properties		🗖 🗗 🔽
File Edit Navigate Search Project	Logger The Logger Component performs logging using an expression that determines what should be logged. By default the current messages is logged using the DEBUG level.		
Package E	General Documentation		
ajex ⇒ 🛃 allrouter	Display Display Name: Logger		Filter: Ju
■ ■ JRE System Library [JavaSt ■ ■ Mule Runtime [Mule Server	Generic Message: All File transfer Successfully		Contraction Contra
- (# src/test/java - (# src/main/resources - (# src/test/resources	Level: INFO Category:		Catabase
German Flows MillRouter.mflow mappings			Carlie Ceneric
core router			
Be System Library [JavaSt			Async
src/main/java			Flow
Choice Router.mflow			Message Enricher
- 🗁 mappings B 🔁 src			Sub flow
corn-expression	?	Cancel	Components
Cxf datamapper drapbox integration			Flow Control
Groupox_integration Groupox_integration Group external_webservice Group file transfer	×		Cloud Connectors
 	Message Flow Global Elements Configuration XML	e 🖹 @	🔍 VI # 📴 # 🗨

-348-

How it works...

In this section you will learn how to deploy the application.

1. To deploy the application code in the Mule server, go to **Run As** | **Mule Application**; the Mule server will deploy your application.



 Once you have successfully deployed the application, you will see the following output on your console. You will also see a log on the console, which states that all the files are transferred to four different locations.



Configuring the Choice Router/Flow Control

Choice Router allows us to route a request to a specific path based on an expression. If the expression matches, it will move forward to the next Endpoint. In this recipe, you will see an example of how the Choice Router/Flow Control works.



Getting ready

In this example, you will use the File Inbound Endpoint, Choice Router, and the File Outbound Endpoint.

1. Open Mule Studio and enter the workspace name as shown in the following screenshot:

Y Workspace Launcher	X
Select a workspace Mule Studio stores your projects in a folder called a workspace. Choose a workspace folder to use for this session	
Workspace: Et/MuleCookBook	Browse
Use this as the default and do not ask again	
	OK Cancel

2. To create a new project, go to **File** | **New** | **Mule Project**. Enter the project name, choice_router, and click on **Next** and then on **Finish**. Your new project is created; you now have to start the implementation.

₩ Mule - Mule Studio			
File Edit Source Refactor Navigate S	Search Project F	Run Window Help	
New	Alt+Shift+N 🕨	🗠 Mule Project	
Open File		/ Java Project	
Close	Ctrl+W	Project	
Close All	Ctrl+Shift+W	📝 Mapping Flow	
🕌 Save	Ctrl+S	😪 Mule Flow	
📓 Save As		🖶 Package	
i Save All	Ctrl+Shift+S	🞯 Class	
Revert		🞯 Interface	
Move		() Enum	
Rename	F2	Annotation	
& Refresh	F5	B Source Folder	
Convert Line Delimiters To	•	Ender	
🚔 Print	Ctrl+P	File	
Switch Workspace	•	Intitled Text File	
Restart		Evanala	
≥ Import		- Example	
Export		📑 Other	Ctrl+N
101 Evport discremito		-	
		-	
Properties	Alt+Enter	_	
1 FlowRef.mflow [flowref/flows]			
2 package.html [activity/src/main/]			
3 Activity.mflow [activity/flows]	1		
4 stringToNumber.java [usdconverter]	-1	-	
Exit			



How to do it...

In this section, you will see how to configure the File Inbound Endpoint, Choice Router, and the File Outbound Endpoint.

1. Go to the Choice Router.mflow file and drag the **File** Inbound Endpoint onto the canvas. Double-click and configure it, and provide the path.

Y Mule - choice_router/flows/C	Y Endpoint Properties	X	🔳 🗖 🗾
File Edit Navigate Search Project	File (Inhound Endpoint)		
i 📑 • 🔛 🗟 🖻 i 🏇 • 🔘 •	The File Endpoint allows Mule appl	ications to read or write files on the local file system	
😭 🕎 Mule	The file Endpoint allows fildle appr	catoris to read of white hies of the local hie system.	
Rackage E	Consultant allocat		
	General Advanced References	Documentation	
	Display		Filter: Ju
ter ajex ter index allrouter	Display Name: File		Select
	Path Information		Christian I.
😟 🛋 JRE System Library [JavaSE	Path: C:\Docu	ments and Settings\azazdesai\Desktop\Input	Endpoints
🕀 🛋 Mule Runtime [Mule Server	Maura ka Dakkarau		Scopes
src/main/java	Move to Pattern:		Components
src/test/java	Move to Directory:		Carl Transformers
- # src/test/resources	Polling Information		Carl Filters
🖃 🗁 flows	Polling Frequency:		🚰 Flow Control 🛛 🗠
Choice Router.mflow	File Age		C All
i mappings	The Age.		Choice
mule-project.xml	File Name Filter		
🗈 😂 corn-expression	File Name Regex Filter:	+ × °	Aggregator
🖶 🔛 customfilter			Collection Splitter
🗄 🗁 cxf			Custom Aggregator
datamapper			C First Successful
external webservice			Message Chunk
🗉 🔛 file_transfer			Aggregator
🗈 😂 filter			Message Chunk Splitter
			Resequencer
Helionide Helionide			Round Robin
i 🔁 jms			Calibra
🗉 🔛 json	?	OK Cancel	Spiccer
🗈 😂 jsp			
B S logicfilter			
object-to-xml			Crror Handling
🗄 🔁 quartz			Cloud Connectors
<	Message Flow Global Ele	ments Configuration XML	
: •		😪 i i a 🔝 🤅	0 😡 M 🛛 (# 📇 🗍 (# 🖳)

-352

- Y Mule choice_router/flows/Choice Router.mflow Mule Studio _ 7 × File Edit Navigate Search Project Run Window Help 🔛 🔛 Mule ቹ Package E 🕺 🔋 Type Hier 🗖 🗖 🥁 *Choice Router 😣 - -8 □ 🕏 🌣 Filter: Ju ^ lack Select Carl Endpoints 🖶 🛋 JRE System Library [JavaSE-1.6] Copes 🚰 🖶 🔜 Mule Runtime [Mule Server 3.3.0 # src/main/java
 # src/test/java Components arransformers 进 src/main/resources flow: Choice_RouterFlow1 📝 Filters # src/test/resources Flow Control e All 🔁 mappings Choice corn-expression Collection Aggregator FILE 🗉 🔛 customfilter Default File Choice Cxf datamapper dropbox_integration EVII Custom Aggregator 🚭 First Successful Message Chunk Aggregator constant and the second s Message Chunk Splitter Bowref
 Bowref Resequencer 뜫 Round Robin 5plitter error Handling Cloud Connectors 🗄 🔁 quartz Message Flow Global Elements Configuration XML ∎⇔ i e 🔝 @ 😡 M 🛛 i e 🚍 🛛 e 🚍 8
- 2. Drag the **Choice** Router onto the canvas. This Flow Control will be configured later.

3. Drag the **File** Outbound Endpoint onto the canvas; double-click and configure it, and provide the path. You have to create a folder called Output. Inside the Output folder, you need to create three different folders: Text, PPT, and ZIP. Firstly, you have to provide the path for the ZIP folder.

Y Mule - choice_router/flows/C	👻 Endpoint Properties 🛛 🔀	- 2 🛛
File Edit Navigate Search Project	File (Outbound Endpoint)	
🗄 📑 📲 🔚 🚔 🗄 🏇 • 💽 •	The File Endpoint allows Mule applications to read or write files on the local file system.	
🖹 🔛 Mule		
😫 Package E 🛛 🍃 Type Hier	General Advanced References Documentation	
E	C Display	Filter:
🗉 🔛 ajex	Display Name: OnlyZIP File	N Salast
	Path Information	18 pelett
BE System Library []avaSE	Dath: C:\Documents and Settings\azazdesai\Decktop\Output)ZIP	Carl Endpoints
Aule Runtime [Mule Server		Copes 6
🕮 src/main/java	Output Pattern:	Components
😕 src/test/java		Carl Transformers
src/main/resources		Pilters
src/test/resources		Flow Control 🛛 🖘
Choice Router.mflow		
🗁 mappings		Chaire
🗷 🗁 src	🙂 🥌 119	
mule-project.xml	🗄 🧰 OrdersData	Collection
corn-expression	🖃 🧰 Output	Collection Splitter
tu internet	Dep PPT	Custom Aggregator
🖶 🔛 datamapper		
🗉 🔛 dropbox_integration		Pirsc buccessi di
external_webservice	710	Angregator
H Ster	Folder:	Message Chunk
Hiller	Make New Folder OK Cancel	Splitter
🛓 🔁 hellomule		Resequencer
🗷 🔛 http-security_filter		🞨 Round Robin
ims 📄 🔁		Splitter
ison	OK Cancel	
logicfilter		
ametransfer		Concer Wandling
😐 🔛 object-to-xml		
🗄 🔁 quartz	Macrona Elow Clobal Elemente Configuration VM	Cloud Connectors
	Inessage now global clements Conniguration And	

-354-

4. Drag the second **File** Outbound Endpoint onto the canvas. Configure it just like the previous one, but this time you have to provide the path for the Text folder.



5. Drag the third **File** Output Endpoint onto the canvas. Configure it just as the first one and provide the path for the PPT folder.



-356

6. Drag the **Logger** component onto the canvas. Double-click and configure it. If an expression doesn't match, that file will be sent to the **Logger** component.

Yule - choice_router/flows/C	YPattern Properties	3 – – 🗠 🛛
File Edit Navigate Search Project	Logger The Logger Component performs logging using an expression that determines what should be logged. By default the current messages is logged using the DEBUG level.	
Package E 🐹 🔋 Type Hier	General Documentation	
e	(Display	Filter:
🗉 🔛 ajex	Display Name: Logger	
🗈 😂 allrouter		l≳ Select
choice_router		Carl Endpoints
JRE System Library [Javast	Message: Invalid File Type	Copes
role Ruidine [mule Server	Level: INFO	
🗁 src/test/java	Category:	
- 🥮 src/main/resources		Filters
src/test/resources		Carbon Carbon
Hows		
		K CO
E Src		Choice
mule-project.xml		Collection
🗈 😂 corn-expression		Aggregator
E Customfilter		Collection Splitter
to CXF		Custom Aggregator
dropbox integration		C First Successful
🗉 🔛 external_webservice		Message Chunk
		Message Chunk
🗄 🔁 flowref		Splitter
😟 🔛 hellomule		Resequencer
🗈 😂 http-security_filter		J 🕓 Round Robin
itti izen		Splitter
n r r r r son	OK Cancel	
🗄 🥁 logicfilter		-
🗉 🖻 nametransfer		Error Handling
🗈 😂 object-to-xml		Cloud Connectors
⊞ 🗁 quartz	Message Flow Global Elements Configuration XMI	Cloud Connectors
: .	😪 :	@ 🖳 M # 🔠 # 🗳

7. Double-click on the Choice Router to configure it. You will write an expression for all the files. The # [message.inboundProperties['originalFilename']. endsWith('.zip')] expression is for ZIP files. Through this expression, only ZIP files are transferred to the destination folder.





Your final flow will look like the following screenshot:



How it works...

In this section, you will get an idea of deploying an application in Mule Studio.

1. To deploy the application code in the Mule server, go to **Run As** | **Mule Application**; the Mule server will deploy your application. At the same time you should run the Tomcat server in Eclipse.



-360

2. From the log on the console, you will see that all files are transferred to the specific folder. Only XML files are not transferred because we didn't use the XML file expression. That's why it shows Invalid File Type as the error description.



Configuring the Splitter Flow Control

A **Splitter** splits incoming messages into parts using the configured expression, which in turn is fed into the next message processor. In this recipe, you will see how to use the Splitter Flow Control.

361

Getting ready

In this section, you will use the File Inbound Endpoint, the Splitter, and the File Outbound Endpoint.

1. Open Mule Studio and enter the workspace name as shown in the following screenshot:

Y Workspace Launcher	×
Select a workspace	
Mule Studio stores your projects in a folder called a workspace. Choose a workspace folder to use for this session.	
Workspace: E://MuleCookBook	Browse
Use this as the default and do not ask again	
	OK Cancel

2. To create a new project, go to **File** | **New** | **Mule Project**. Enter the project name, splitter, and click on **Next** and then on **Finish**. Your new project is created, so you can now start the implementation.

🖌 Mule - Mule Studio				
File Edit Source Refactor Navigat	te Search Project	Run Window Help		
New	Alt+Shift+N	🕨 🚰 Mule Project		
Open File		🏄 Java Project		
Close	Ctrl+W	📬 Project		- F
Close All	Ctrl+Shift+W	Apping Flow		
Save	Ctrl+S	Mule Flow		
Save As		豑 Package		
😭 Save All	Ctrl+Shift+S	🞯 Class		
Revert		💕 Interface		
Move		🞯 Enum		
Rename	F2	Annotation		
🔁 Refresh	F5	Source Folder		
Convert Line Delimiters To		Java Working Set		
Print	Ctrl+P	Folder		
Switch Workspace		Untitled Text File		
Restart		Example		
Import		Ther	Ctrl+N	
C Export		_		
Export diagram to				
Properties	Alt+Enter	_		
1 FlowRef.mflow [flowref/flows]				
2 package.html [activity/src/main/	.]			
3 Activity.mflow [activity/flows] 4 StringToffumber jour [undergrowth]	tord 1			
+ pointigromaniper, java [asaconvert	.cr/]	-		
Exit An outline is not available.				
∎*				i a 🔛 @ 🕓 M 🗉
362				

How to do it...

In this section, you will see how to configure Splitter, File Inbound Endpoint, and File Outbound Endpoint.

1. Go to the splitter.mflow file and drag the **File** Endpoint onto the canvas. Double-click and configure it, and then provide an XML file path. You split a city name in that XML file.

The following code snippet is the Shipping.xml file:

```
<shiporder xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"</pre>
orderid="555-66-7777">
  <orderperson>Derek Adams</orderperson>
  <shipto>
    <name>Azaz Desai</name>
    <address>123 Test Drive</address>
    <city>Ahmedabad</city>
    <country>India</country>
  </shipto>
  <item>
    <title>Laptop</title>
    <note>Some piece of Mac crap!</note>
    <quantity>1</quantity>
    <price>99.97</price>
  </item>
  <item>
    <title>Memory Chips</title>
    <note>1 GB</note>
    <quantity>4</quantity>
    <price>49.99</price>
  </item>
</shiporder>
```

Here you configure the input folder path.

You Mule - splitter/flows/Splitter	Y Endpoint Properties		💶 🗗 🔀
File Edit Navigate Search Project	File (Inbound Endpoint)		
i 📫 • 🔛 🕼 🗁 i 🏇 • 💽 •	The File Endpoint allows Mule applications to read or write files on the local file system.		
🖹 🕎 Mule	······································		
📮 Package E 🕺 🍃 Type Hier	General Advanced References Documentation		
			Elitory 8
	Direlau Nameu File]]]	Ju
i i gittinoi	Display Name. The		Select
🗉 🔛 object-to-xml	Path Information		C Endpoints
🖶 😂 quartz	Path: D:\OrdersData\Input		
📱 🔛 quartz_scheduler	Move to Pattern:		Concernante
restbasedwebservice			
* 😂 scrint	Move to Directory:		Transformers
soap-jax-ws			Filters
🗉 🔛 spellcheck	Polling Frequency: 1000		🚰 Flow Control 🛛 🗠
😑 🔛 splitter	File (ne) 500		Kali 🔁
H A JRE System Library [JavaSi			Choice
Mule Runtime [Mule Server	File Name Filter		Collection
- # src/test/java	File Name Regex Filter:	- 🗣 🗙 🐂	Aggregator
- 🕮 src/main/resources			Collection Splitter
- 🥮 src/test/resources			Custom Aggregator
🖹 🧁 flows			Eirst Successful
Splitter.mflow			BU Message Chunk
The sec			Aggregator
mule-project.xml			Message Chunk
🗉 😥 spring			Splitter
studioconnector			Resequencer
Esttransformer			Kound Robin
transformer		Cancel	5plitter
twiterintegration			
🗴 🔛 usdconverter			·
🗉 🔂 web-services			Error Handling
🗈 🗁 webservice-cxf			Cloud Connectors
<	Message Flow Global Elements Configuration XML		
i •	🔂 i	: # 🖹 @	🔍 И 🕴 е 🔚 🕴 е 🕒

- Y Pattern Properties - - X Splitter 📬 • 🔛 🕼 🗁 🗄 🏇 • 🔘 The simple splitter splits incoming message into parts using the configured expression passing on each part in turn to the next message processor. 🔛 🔛 Mule - -洋 Package E 🛛 🏌 🏌 Type Hier General Documentation 8 Display Filter: Ju logicfilter Display Name: Splitter Select Generic andpoints
 quartz

 quartz_scheduler

 log

 restbasedwebservice
 ~ Enable Correlation: IF_NOT_SET a Scopes + × • Message Info Mapping: Components 🗈 🔁 restwebservice #[xpath:shiporder/shipto/city] a Transformers 🛓 📂 script script soap-jax-ws cellcheck Filters Expression: Flow Control IIA 🤝 🛓 🔜 JRE System Library [Java3 Choice 🗉 🛋 Mule Runtime [Mule Server Collection Aggregator - 🍅 src/main/iava Brc/main/jara
 src/test/java
 src/main/resources Collection Splitter est/test/resources Custom Aggregator Flows CCC First Successful Message Chunk Aggregator 🗁 mappings src
 mule-project.xml Message Chunk Splitter spring Resequencer testtransformer 뜫 Round Robin F Splitter b b twiiter
 b b twiiterintegration ? OK Cancel ſ sdconverter
 web-services a Error Handling 🗄 🔁 webservice-cxf Cloud Connectors Message Flow Global Elements Configuration XML < > .∎≎ 8 je 🔝 🥘 😥 M 🛛 je 🔡 🕴 e 📃
- 2. Drag the **Splitter** Flow Control onto the canvas. Double-click and configure it. To split the city name, you write an expression # [xpath:shiporder/shipto/city].

3. Drag the **File** Outbound Endpoint on the canvas. Double-click and configure it, and provide a destination path.

∑ Mule - splitter/flows/Splitter.	¥ Endpoint Properties		🔳 🗗 🗾
File Edit Navigate Search Project	File (Duthound Endpoint)		
i 📫 • 🔛 🗟 🖻 i 🏇 • 🔘 •	The File Endpoint allows Mule applications to read or write files on the local file system		
😭 💟 Mule	The time chappene allows made applications to read or write hies on the local file system.		
Package E 💥 🍃 Type Hier	General Advanced References Desumentation		
			Elhan B
	Display		Filter: Ju
	Display Name: File		Select
de de la constance de la	← Path Information		
🗉 😂 quartz	Path- C:\Documents and Settings\azazdesai\Desktop\Output		
🗊 🗁 quartz_scheduler		2	Copes Scopes
🗉 😂 restbasedwebservice	Output Pattern:		Components
🗈 🗁 restwebservice			Carl Transformers
🗄 🔛 script			C Filters
🗄 📅 soap-jax-ws			Elow Control
spelicheck			
🖶 🥣 spinter			< All
A Mule Runtime [Mule Server			Choice
🗁 src/main/java			Collection
🕮 src/test/java			Aggregator
- 🕮 src/main/resources			Collection Splitter
- 🥮 src/test/resources			Custom Aggregator
🖻 🧁 flows			Eirst Successful
Splitter.mflow			Maggana Chunk
mappings			Aggregator
mule-project.xml			Message Chunk
B R spring			Splitter
🗉 🔁 studioconnector			Resequencer
🖻 🗁 testtransformer			Round Robin
🗄 😂 transformer			Splitter
🗈 🗁 twiiter	OK Can	:el	
E 🔁 twiiterintegration			
web-services			Carlor Handling
	✓		Cloud Connectors
<	Message Flow Global Elements Configuration XML		
i □ *	😢 ::	8 @	🕒 И 🕴 е 🔚 🕴 е 🗐

Wule - splitter/flows/Splitter.mflow - Mule Studio File Edit Navigate Search Project Run Window Help - - X 🔛 🔛 Mule 😫 Package E 🕴 👔 Type Hier 💛 🗖 💟 *File Transfer 🛛 💟 *Splitter 🕄 8 □ 🕏 🌣 Filter: Ju Iogicfilter
 Image: Barbon State
 ^ lack Select Carl Endpoints duartz
 quartz_scheduler Copes 🚰 🗉 🛃 restbasedwebservice Components i 🔁 🔁 restwebservice arransformers 🗉 🔁 script scipt soap-jax-ws spellcheck Filters Elow Control e All JRE System Library [JavaSE-1.6]
 Mule Runtime [Mule Server 3.3.0 flow: SplitterFlow1 📝 Choice Collection Aggregator src/main/java
src/test/java FILE ▶ 🔡 FILE ≻ Brc/main/resources
 Brc/test/resources File Splitter File EVII Custom Aggregator 🚭 First Successful Message Chunk Aggregator - C> mappings mappings
 src
 mule-project.xml
 spring
 studioconnector
 testtransformer Message Chunk Splitter Resequencer 뜫 Round Robin transformer 5plitter Witerintegration
 Usdconverter web-services error Handling Cloud Connectors Message Flow Global Elements Configuration XML < > ₽ i # 🖹 @ 😣 M 🛛 i # 📰 🗍 # 🗐 🚓 🗄

Your final flow will look similar to the one shown in the following screenshot:

How it works...

In this section, you will see how to deploy the application using Mule Studio.

1. To deploy the application code in the Mule server, go to **Run As** | **Mule Application**; the Mule server will deploy your application.

Y Mule - splitter/flows/Splitter.mflow - Mule Studio	- B 🛛				
File Edit Source Refactor Navigate Search Project Run Window Help					
: *** * 🗐 陶 白 : 恭 * 🕼 * 0 * 0 # 0 * : @ 🛷 * : @ : り ~ 気 ~ や ク ~ : ◇ ひ 🗙					
Packane F X P Type Hier V *File Transfer					
	Filter: JU				
	Select				
😟 🔂 object-to-xml	Endpoints				
B 😂 quartz	Scoper				
B ∉ quartz_scheduler	Components				
	Transformers				
B C script	Carliners				
Den F3	Pilers				
B → Spelicheck Open with Show to Alt+Shift+W	Flow Control 🗠				
R Splitter Jiowall Antonictw					
B A Mule Runtim Copy Ctrl+C	Choice				
src/main/jay Ecopy Qualified Name					
	Collection Solitter				
Generatives Splitter File					
Build Path	Einek Susseanful				
Splitter., Refactor Alt+Shift+T					
mappings 🔁 Import	Aggregator				
🖬 🌝 St. 🔛 mule-project 🗠 Export	Message Chunk				
🖶 📅 spring 🤣 Refresh F5	Besequencer				
ter je studioconnector Assign Working Sets	Paued Babia				
B → Costa also inter B → Costa also inter B → Costa also inter					
🖷 😂 twilter 🛛 Run As 💦 👌 1 Run on Server 🛛 Alt+Shift+X, R	Splitter				
🕒 🗁 twitterintegration					
Configuration					
	Error Handling				
Compare with P	Cloud Connectors				
Mule Management Console					
🗄 🗅 Splitter.m 🔀 CloudHub 🧼 👘 🔛 @	😡 и 🛛 е 🗄 🗍 е 🗳				

-368

2. Once you have successfully deployed the application, you will see the following output on your console. Use the path highlighted in the following screenshot to see that only the city name will be written into the file.



369

10 Configuring Cloud Connectors

In this chapter, you will learn what a Cloud Connector is. We will also look at the following recipes:

- ► Configuring the Twitter Cloud Connector
- Configuring the DropBoxIntegration folder

Introduction

Through Cloud Connectors, easy integration of your third-party web APIs is possible. Cloud Connectors are mainly used for integration purposes. In this chapter, you will see how to integrate third-party APIs through different connectors.

Configuring the Twitter Cloud Connector

The **Twitter Cloud Connector** is used for integrating the Twitter API. Through this API, you will perform different operations such as updating the status on your Twitter account, and you can retweet the message, search and show statuses. In this recipe, you will see how to send a tweet using Mule Studio.

Configuring Cloud Connectors -

Getting ready

In this example, you will use the HTTP Endpoint and the Twitter Cloud Connector.

1. Open Mule Studio and enter the workspace name as shown in the following screenshot:



2. To create a new project, go to **File** | **New** | **Mule Project**. Enter the project name, Twitter, click on **Next** and then on **Finish**. Your new project is created, and you now have to start the implementation.

🖌 Mule - Mule Studio					
File Edit Source Refactor Nav	vigate Search Project	Run Window Help			
New	Alt+Shift+N	🕈 🔄 Mule Project		* 🌾 🔶 * 🔶 *	
Open File		/ Java Project			
Close	Ctrl+W	Project			
Close All	Ctrl+Shift+W	Apping Flow			
U Save	Ctrl+S	Mule Flow			
Save As		🖶 Package			
Save All	Ctrl+Shift+S	🞯 Class			
Revert		🗊 Interface			
Move		🞯 Enum			
Rename	F2	Annotation			
🗞 Refresh	F5	Source Folder			
Convert Line Delimiters To	1	Java Working Set			
Print	Ctrl+P	Folder			
Switch Workenace		Line Tile			
Restart		Indiana lext File			
N. *		- 📑 Example			
Import		🃬 Other	Ctrl+N		
Export					
🔟 Export diagram to					
Properties	Alt+Enter				
1 FlowRef.mflow [flowref/flows]				
2 package.html [activity/src/ma 3 Activity mflow [activity/flowc]	in/]				
4 StringToNumber.iava [usdcon	verter/]				
		-			
Arr outline is not available.					
. .					i e 💦 @ 🙆 M 🛛
372					

How to do it...

In this section, you will see how to configure the Twitter Cloud Connector and how to use it in a flow.

1. Go to the twitter.mflow file and click on the Global Elements tab. Go to Cloud Connectors | Twitter.



Configuring Cloud Connectors —

 Once you click on the Twitter Cloud Connector, you will see a similar screen on your system. Here, you generate the values for the Access Key:, Access Secret:, Consumer Key:, and Consumer Secret: fields.

Mule - twiiter/flows/twiiter.n	🖌 Global Element I	Properties		- 7 🗙
File Edit Navigate Search Project : ➡ ♥ ♥ ♥ ♥ ♥ ♥ ♥ ♥ ♥ ♥ E ♥ ♥ Mule	Twitter Global Twitter configur	ation information.		
Mule Package E P	Global Twitter configur General Name: General Access Key: Access Secret: Consumer Key: Consumer Key: Use SSL I	ation information. Inwitter 114742097-fCy26ovprbjNh1zQtzh8QH6YO23LUXJ370 983D182RLzK8zfVEerCYV09iy28aeBtSQtyqtQEY g5Op3z74O36vcnpTeXW6A mwjUegMGansmF6WAgVyYQyjw6NFvGSJ3gFIXUNo	SV8	Create Edit Delete
Crest/java Src/test/java Src/test/java Src/test/jesources Twitter [v2.4] Twitter [v2.4] Twitter [v2.4] Twitter [v2.4] Twitter [v2.4] Src Twitter [v2.4]		Global Elamente Confine valion VM	OK Cancel	
: •			: # 🖹 @	🛯 и је 🗄 је 🗐

-374-

3. Go to https://dev.twitter.com/apps. Here you can click on **Create application**. Then you have to fill up the application form. Once you create the application, you will find a key in that application. Paste that key into your **Twitter** Cloud Connector.

	ev.twitter.com/apps/3479973/show				<u>×</u> 🛯
🍠 Developers	API Health Blog Discuss	ions Documentation	Search	٩	9 -
our applications OAutris	ettings, keep the Consumer secret als	ecret. This key should never be r	ютал-теараріе ін уоргаррію	ation:	
ccess level	Read, write, and direct me About the application perr	essages nission model			
Consumer key	g50p3z740J6vcnpTeX	16A			
Consumer secret	mwjuegNGansmF6WAgVy	YQyjw6NFvGSj3gFIXUNo			
Request token URL	https://api.twitter	.com/oauth/request_toke	n		
uthorize URL	https://api.twitter	.com/oauth/authorize			
ccess token URL	https://api.twitter	.com/oauth/access_token			
allback URL	None				
ign in with Twitter	No				
Your access token Use the access token stri ccount. Do not share you	ig as your "oauth_token" and the acce Foauth_token_secret with anyone.	ss token secret as your "oauth	_token_secret" to sign re	quests with your own Twitte	r
Your access token Ise the access token stri ccount. Do not share you .ccess token	ig as your "oauth_token" and the acce roauth_token_secret with anyone. 114742097-ifCyZ6ovpbjNł	ss token secret as your "oautb 11zQtzh8QIH6YO2JcUXj3rDGV8	_token_secret" to sign re	quests with your own Twitte	r
Your access token lee the access token stri ccount. Do not share you access token access token secret	ig as your "oauth_token" and the acco roauth_token_secret with anyone. 114742097-ifCyZGovprbjNl 9B3D1B2RLzKi8zfivEerC`	iss token secret as your "oauth 11 v1zQtzh8QIH6YO2JcUXjI3r0GV8 vV09iy2BaeBtSQtyqtQEY	token_secret" to sign re	quests with your own Twitte	r

375

Configuring Cloud Connectors ——

4. Click on the **Message Flow** tab and drag the **HTTP** Endpoint onto the canvas. To configure the Endpoint, double-click on it. Enter the port number.

⊻ Mule - twiiter/flows/twiiter.n	Y Endpoint Properties	X
File Edit Navigate Search Project	HTTP (Inbound Endpoint)	
<u>□ • </u>	The HTTP Endpoint allows Mule application to connect to web resources through the HTTP transport protocol.	
🗈 🔛 Mule	This endpoint can also implement security through HTTP5.	
洋 Package E 😫 🍃 Type Hier	General Advanced References HTTP Settings Documentation	
E	Display	Filter:
🗈 😂 jms	Display Name: HTTP	JU
ison	Euclasse Deltance	Select
i i i i i i i i i i i i i i i i i i i	Exchange Patterns	Contraction Contractica Contra
	One-way • request-response	Copes Scopes
🛓 🔁 object-to-xml	Basic Settings	Components
🗷 🔛 quartz	Enable HTTPS 0	Carl Transformers
🗈 😂 quartz_scheduler	Enabling HTTPS will require configuring a HTTPS Connector	C Filters
restbasedwebservice	Host: localhost	Elow Control
script	Port: 8088	Error Handling
🗄 🔂 soap-jax-ws	Pathy	Cloud Connectors
🗉 🔛 spellcheck	rau	
🗈 🗁 splitter		Boxnet
🗉 😸 spring		CMIS
studioconnector testtrapsformer		Cloudhub
transformer		Oropbox
🖨 🔁 twiiter		🔚 Github
💷 🛋 JRE System Library [JavaSt		HBase
🗉 🛋 Mule Runtime [Mule Server		Takasak
src/main/java		in racce
- C src/resc/java		Y Jira
- 🕮 src/test/resources		Magento
🗉 🛋 Twitter [v2.4]		O Metanga
🖻 🧁 flows	(?) OK Cancel	Mongo DB
twiiter.mflow		Netsuite
mappings src		Objectstore
mule-project.xml		P Paypal
🗄 🔛 twiiterintegration	V Harrow Flaw Cick of Flawards Configuration 988	, alba
	Message How Global Liements Configuration XML	
0 ∎	😥 i 🖉 🚼 🤅) 🗛 M 🛛 🖉 🖷 🔛

-376-

- Whule twiiter/flows/twiiter.mflow Mule Studio File Edit Navigate Search Project Run Window Help _ 7 🛛 🔛 🔛 Mule ቹ Package E 🖄 🔋 Type Hier 🗖 🗖 💟 *twiiter 🖄 - -8 ₽ 🕏 🎽 Filter: ims ipson ipso Ju ^ lack Select Carl Endpoints Copes 🚰 Components arransformers guartz_scheduler
 extrastice Filters Festbasedwebser
 restwebservice
 Festwebservice
 Festweb flow: twiiterFlow1 📝 Control a Error Handling HTTP Cloud Connectors box Boxnet 🚯 CMIS 🚾 Cloudhub 💿 Dropbox transformer *(*) 👼 Github TRE System Library [JavaSE-1.6]
 Mule Runtime [Mule Server 3.3.0 HBase - 🥮 src/main/java - 🕮 src/test/java 🙆 Intacct 👻 Jira src/main/resources
 src/test/resources 🝈 Magento srctest/resources
 srctest/resources 💿 Metanga 🚺 Mongo DB Netsuite Generation
 Src
 Mule-project.xml Sobjectstore 😰 Paypal 🗈 🔛 twiiterintegration Message Flow Global Elements Configuration XML Σ a 🖹 @ 😣 M 🗄 a 🗄 R æ E
- 5. Drag the Twitter Cloud Connector onto the canvas. To configure it, double-click on it.

Configuring Cloud Connectors -

6. Once you click on the Connector, you will see a similar screen, as shown in the following screenshot, on your system. You will select the configuration reference name that was created before in the **Global Elements** tab—Twitter. Then, you select operation **Update Status**. In the **Status**: textbox, write an expression to send a tweet to your account, # [header:INBOUND:mymessage].

⊻ Mule - twiiter/flows/twiiter.n	🖌 Pattern Properties	
File Edit Navigate Search Project	Twitter Twitter Integration	
Package E Type Hier	General Documentation Display Display Name: Twikter Generic	Filter: 5 Select
logicfilter logicfilter logicfilter logicfilter logict-to-xml logict-to-xml logicfilter	Config Reference: Twitter Coperation: Update status Ceneral	Components
Ger quartz_scheduler Ger restbasedwebservice Ger restwebservice Ger script	#[header:INBOUND:mymessage] Status:	C Filters
sop-jax-ws splitter splitter splitter splitter splitter splitter splitter splitter	In Reply To: Coordinates Latitude:	Cloud Connectors \infty
	Longitude:	Corpopox Construite HBase
- (src/main/java - (src/test/java - (src/main/resources - (src/test/resources		Intacct In
Twitter [v2.4] Flows Wiiter.mflow mappings	OK Cancel	Mongo DB Netsuite
mule-project.xml	Message Flow Global Elements Configuration XML	Paypal
: □*	😥 i	@ 🕒 M 🛛 🖶 🗄 🖉

-378

7. Drag the Expression transformer; it will convert a payload into a string.



Configuring Cloud Connectors -

8. Double-click on the **Expression** transformer to configure it. Write an expression to convert it into a string, payload.toString().



-380

How it works...

In this section, you will have a look at how to deploy the application in Mule Studio.

1. To deploy the application code in the Mule server, go to **Run As** | **Mule Application**; the Mule server will deploy your application.

Mule - twiiter/flows/twiiter.mflow - Mule Studio	- 7 🛛
File Edit Source Refactor Navigate Search Project Run Window Help	
*** *	
Package E 23 New	
Open F3	Filter:
	Select
isp Copy Ctrl+C	Carl Endpoints
	Copes Scopes
B Cri+V	Components
🕆 😂 quartz 🔀 Delete Delete	Transformers 🗠
Build Path	Append String
ter Construences and the structure of t	Attachment
	Body to Parameter Map
spelicheck	••• Byte Array to MIME
Construction of the second sec	Byte Array to Object
studioconnecto Vilidate	Byte Array to Serializable
B → transformer Valuade	Byte Array to String
E wilter Debug As	Z DataMapper
B A JRE System Profile As	est Email to String
Run Configurations	Expression
Grand Statistics Compare With	File to Pute Away
espace with	cel File to byte Array
# src/test/res 20 Cloudlub	••• File to String
	Groovy
Visite in a second seco	Pilters
😂 mappings	Control
B 🗁 src	Caror Handling
mule-project.xml	Cloud Connectors
Message Flow Global Elements Configuration XML	
🗄 🗅 twitter.mflow - twitter/flows	2 @ 🕼 И 🛛 е 🔚 🗍 е 📮

Configuring Cloud Connectors _____

2. Once you have successfully deployed the application, you will see the following output on your console. You will see a log on the console, which states that all files are transferred to different locations.

Y Mule - twiiter/flows/twiiter.mflow - Mule Studio		J X
File Edit Navigate Search Project Run Window Help		
i 📬 • 🖫 🗟 i 🏇 • 🗨 • 🍇 • i 🖶 🞯 •	i 🕭 🖋 • i 💷 i 🖢 - 🖓 - 🖘 🔶 • - i 🖑 🗞 🗶	
🖺 📔 Mule		
📮 Console 🛛	🔳 🗶 💥 📴 🚝 💭 😁 🗂 🗁 🗂 🗖	
twiiter [Mule Application] C:\Program Files\Java\jre6\bin\javaw.ex	xe (Apr 27, 2013 4:50:38 PM)	-
,		Ju
INFO 2013-04-27 16:50:56,484 [main] or	g.mule.lifecycle.AbstractLifecycleManager: Starting connector: connector.http.mule	
INFO 2013-04-27 16:50:56,484 [main] or	g.mule.twitter.agents.DefaultSplashScreenAgent:	
+++++++++++++++++++++++++++++++++++++++	+++++++++++++++++++++++++++++++++++++++	
+ DevKit Extensions (1) used in this ap	plication	
Twitter 2.7.2 (DevKit 3.3.1 +		
+ Build UNNAMED.1297.150f2c9)+		
	+	
***************************************	******	
INFO 2013-04-27 16:50:56,484 [main] or	g.mule.lifecycle.AbstractLifecycleManager: Starting model: _muleSystemModel	
INFO 2013-04-27 16:50:56,500 [main] or	g.mule.construct.riowconstructLifecycleAnager: starting flow: twitteriowi	
INFO 2013-04-27 16:50:56,500 [main] of	g.mule.processor.sedastageLitecyclemanager: Starting Service: twitterflowi.stagel	
INFO 2013-04-27 16:50:56 546 [main] or	g.mule transport service Defmettor. Registering Hasenet. Conternet on engine	
INFO 2013-04-27 16:50:56,562 [main] or	g.mule.lifecycle.MhstractLifecycleManager: Initialising: 'null'. Object is: HttnMe	
INFO 2013-04-27 16:50:56.625 [main] or	g.mule.lifecycle.AbstractLifecycleManager: Starting: 'null'. Object is: HttnMessac	
INFO 2013-04-27 16:50:56,640 [main] or	g.mule.module.management.agent.WrapperManagerAgent: This JVM hasn't been launched	
INFO 2013-04-27 16:50:56,703 [main] or	g.mule.module.management.agent.JmxAgent: Attempting to register service with name:	
INFO 2013-04-27 16:50:56,703 [main] or	g.mule.module.management.agent.JmxAgent: Registered Endpoint Service with name: Mu	
INFO 2013-04-27 16:50:56,703 [main] or	g.mule.module.management.agent.JmxAgent: Registered Connector Service with name Mu	
INFO 2013-04-27 16:50:56,718 [main] or	g.mule.module.launcher.application.DefaultMuleApplication: Reload interval: 3000	
INFO 2013-04-27 16:50:56,718 [main] or	g.mule.DefaultMuleContext:	
******	****************	
* Application: twiiter	*	
* OS encoding: Cp1252, Mule encoding: U	TF-8 *	
*	*	
* Agents Running:	*	
* DevKit Extension Information	*	
* JMX Agent	7	
TNEO 2012 04 22 16.50.56 210 (eessessessessessesses and the termine the second seco	
inro 2013-04-27 16:50:56,718 [main] or	g.mute.mouute.tauncher.DeploymentService:	
***************************************		2

3. Open your browser and paste the URL in it http://localhost:8088/?mymessage =HelloMule. You will receive a JSON output. When you receive a similar output on your browser (as shown in the following screenshot), your message will be successfully sent to your Twitter account:



4. You can now see the message displayed on your Twitter account:




Configuring the DropBoxIntegration folder

Dropbox is a file hosting service operated by Dropbox Inc. It is a free service that lets you bring your photos, documents, and videos anywhere and share them easily. Through Dropbox Connectors, you can access the Dropbox API and you can also insert, upload, download, and delete files and folders.

Getting ready

In this example, we will use the HTTP Endpoint, the Dropbox Connector, and the Choice Router. Create a new project by performing the following steps:

1. Open Mule Studio and enter the workspace name as shown in the following screenshot:

Workspace Launcher	×
Select a workspace	
Mule Studio stores your projects in a folder called a workspace. Choose a workspace folder to use for this session.	
Workspace: E:\MuleCookBook	Browse
Use this as the default and do not ask again	
	OK Cancel



2. To create a new project, go to **File** | **New** | **Mule Project**. Enter the project name, Dropbox_Integration, click on **Next** and then on **Finish**. Your new project is created, and you now have to start the implementation.

🎽 Mule - Mule Studio					
File Edit Source Refactor Naviga	ate Search Project	Run Window Help			
New	Alt+Shift+N	🗠 🚘 Mule Project		- () () - () -	
Open File		🏄 Java Project			
Close	Ctrl+W	🎦 Project			
Close All	Ctrl+Shift+W	📌 Mapping Flow			
🕌 Save	Ctrl+S	Mule Flow			
📓 Save As		🖶 Package			
in Save All	Ctrl+Shift+S	🞯 Class			
Revert		🗊 Interface			
Move		🕼 Enum			
Rename	F2	Annotation			
Refresh	F5	Source Folder			
Convert Line Delimiters To		 Bava working Set Solder 			
👜 Print	Ctrl+P	E File			
Switch Workspace		Inc I			
Restart					
an Import		Example			
Export		📑 Other	Ctrl+N		
den Consult dis more to		-			
Export diagram to		-			
Properties	Alt+Enter				
1 FlowRef.mflow [flowref/flows]					
2 package.html [activity/src/main/]				
3 Activity.mflow [activity/flows]					
4 StringToNumber.java [usdconve	rter(]	_			
Exit					
Arroddine bried available.		_			
: =0					

How to do it...

In this section, you will see how to configure the Dropbox Connector and how to use it in a flow.

1. Go to the Dropbox_Integration.mflow file. Click on the **Global Elements** tab and go to **Cloud Connectors** | **Dropbox**.



-386

Y Mule - dropbox_integration/f	🖌 Global Element	t Properties		3 🔳 🖉 🛛
File Edit Navigate Search Project	Dropbox			
📬 • 🔛 🗟 📄 🏇 • 🔘 •	Global Dropbox confi	iguration information.		
😭 🔛 Mule		-		
🔋 Package E 🛛 🔋 Type Hier	General Oauth D	ocumentation		
E	Generic			
🕀 🔂 ajex	Name:	Dropbox		JU
🗈 😂 allrouter				Create
E Choice_router	General			Edit
Contrexpression	Content Server:			
🗄 🔁 cxf	Server:			Delete
🕀 🔛 datamapper	Port:	8081		
G aropbox_integration	App Key:	ppr2sh8h1is39ks		
IRE System Library [Java5] IRE System Library [Java5] IRE Server	App Secret:	h3v5hzqz9x6lula		
 # # src/main/java 		-		
🕮 src/test/java	🗌 Debug 🤨 🔛			
src/main/resources				
Beforenced Libraries				
References cibraries				
DropBox Integration.ml				
- 🧀 mappings				
🗈 🔂 src				
mule-project.xmi				
🗊 🔁 filter				
🗄 😂 flowref				
E E hellomule			· · · · · · · · · · · · · · · · · · ·	-
the first trace	2		OK Cancel	1
🗄 🔁 json	\odot			J
🗉 🔛 jsp				_
🗈 😂 logicfilter				
🗄 🗁 nametransfer	-			
Conject-co-xmi	Message Flo	w Global Elements Configuration XML		
		6	: e 💽	@ 🕓 M 🗄 # 😫 🗍 # 🗳

2. Configure the **Dropbox** Connector; here, you add the values for **Port:**, **App Key:**, and **App Secret:** fields.

3. Click on the **Oauth** tab. Here, you can fill in the **Domain:**, **Local Port:**, and **Remote Port:** fields.

⊻ Mule - dropbox_integration/	Global Element Properties		🔳 🗗 🔀
File Edit Navigate Search Project	Dropbox Global Dropbox configuration information.		
Package E Alexage E	General Oauth Caliback Config Domain: localhost Local Port: 8081 Remote Port: 8081 Path:	ancel	Create Edit Delete
	en 1	s 🚼 @	🔍 И 🛛 е 🔚 🗍 е 🖨

4. To generate the values of the **App Key:** and **App Secret:** fields, go to the URL https://www.dropbox.com/developers. Click on **Create new app**. Once you have created an app, you will see the app key and the app secret.

← → C 🔒 https://ww	w.dropbox.com/developers/app_info/111588		👷 📴 😑
😻 Dropbox		🎁 Get free space!	Azaz Desai 🔻 🥈
Developer home Apps console Create new app Dropbox Chooser	MuleESB App General information		
 Sync API Core API Reference 	App name App status App key	MuleESB App Development (Apply for production status) ppr2sh8h1is39ks	
Forums API support	App secret Access type Number of users	h3v5hzqz9x6lula Full Dropbox 0.of 5.(Lolink all users)	
	Additional information	Increase limit to 100 users.	
	Website Description (max 1500 characters)	MuleESB	

- 🍟 Mule Mule Studio File Edit Source Refactor Navigate Search Project Run Window Help - 💼 • 🔚 🕞 👌 + 🍫 • 🔕 • 🛛 🖶 🞯 • 🛛 🍅 🔗 • 🛛 💷 🗄 🦕 - 🎘 - 🦘 🔶 - 🔶 🔛 🔛 Mule 😫 Package E 🕺 🔋 Type Hier 📃 🗖 Ð 🕒 🔄 Ju 🗉 📂 ajex 🗄 🔁 allrouter choice_router 🗄 😂 corn-expression ■ estimation
 ■ estimation
 ■ custom
 🗉 📴 datamapper dropbox_integration
 JRE System Library [JavaSE-1.6] 🛓 🛋 Mule Runtime [Mule Server 3.3.0 estimation of the second se - 🍅 srcitestiresources 🗉 🔜 Referenced Libraries E Contraction Flows 🗁 mappings 😑 🔑 src 😑 😥 main 😑 🥟 app 🛓 🗁 docroot bome.html Index.css
 TropBox Integration.
 Inde-app.properties
 mule-apply.propertie 🕞 test Mule-project.xml i 😥 😥 external webservice 🗉 🔂 file_transfer 🗄 📂 filter > 8 i e 🖹 @ 😣 VI 🕴 e 🔚 🕴 e 🚍 home.html - dropbox_integration/src/main/app/docroot
- 5. Go to src/main/app and create a docroot folder. Inside this folder create two files: home.html and index.css.

6. In the docroot folder, click on the home.html file. The following code snippet is present inside the home.html file:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<link rel="stylesheet" type="text/css" href="index.css" />
<meta http-equiv="Content-Type" content="text/html;
charset=UTF-8">
<title>Insert title here</title>
</head>
<body>
<script type="text/javascript">
function processElements ( elements, style) {
```

-390

```
for ( var i = 0; i < elements.length; i++) {</pre>
      elements[i].style.display=style;
    }
  }
  function updateOptions(value) {
  var hideElements = document.getElementsByClassName('hidden');
  var showElements = document.getElementsByClassName(value);
  processElements(hideElements, 'none');
  processElements(showElements, 'block');
  }
  </script>
  <!--onsubmit="this.action=document.getElementById('op').
options[document.getElementById('op').selectedIndex].value;"-->
  <form action=/in method="post">
    Welcome to AttuneInfocom!!! <br /> <br />
    Operation: <select id="op" name="op"
onchange="updateOptions(this.options[this.selectedIndex].value);">
                <option value="selectoption">--Select Option--<///i>
option>
                     <option value="createF">Create Folder</option>
                    <option value="delete">Delete</option>
                </select><br /><br />
    <div class="hidden upFile createF delete downFile
list getLink" id="dropboxPath">Path:<input type="text"</pre>
name="dropboxPath" /></div>
    <input type="submit" value="Submit"/>
  </form>
</body>
</html>
```

```
7. Click on the index.css file. The following is the code snippet in the index.css file:
   form {
   background: -webkit-gradient(linear, bottom, left 175px,
   from(#CCCCCC), to(#EEEEEE));
   background: -moz-linear-gradient(bottom, #CCCCCC, #EEEEEE 175px);
   margin:auto;
   position:relative;
   width:350px;
   height:350px;
   font-family: Tahoma, Geneva, sans-serif;
   font-size: 14px;
   font-style: italic;
   line-height: 24px;
   font-weight: bold;
   color: #09C;
   text-decoration: none;
   -webkit-border-radius: 10px;
   -moz-border-radius: 10px;
   border-radius: 10px;
   padding:10px;
   border: 1px solid #999;
   border: inset 1px solid #333;
   -webkit-box-shadow: 0px 0px 8px rgba(0, 0, 0, 0.3);
   -moz-box-shadow: 0px 0px 8px rgba(0, 0, 0, 0.3);
   box-shadow: 0px 0px 8px rgba(0, 0, 0, 0.3);
   }
   textarea#feedback {
   width:375px;
   height:150px;
   }
   textarea.message {
   display:block;
   }
```

-392

```
input.button {
width:100px;
position:absolute;
right:20px;
bottom:20px;
background:#09C;
color:#fff;
font-family: Tahoma, Geneva, sans-serif;
height:30px;
-webkit-border-radius: 15px;
-moz-border-radius: 15px;
border-radius: 15px;
border: 1p solid #999;
}
input.button:hover {
background:#fff;
color:#09C;
}
textarea:focus, input:focus {
border: 1px solid #09C;
}
img,a {
display:none;
}
#obj {
display:none;
}
.hidden {
display:none;
}
```

8. Go to the dropboxIntegration.mflow file and drag the HTTP Endpoint onto the canvas. First, you have to authorize that application. To configure the Endpoint, double-click on it. Enter the port number and the pathname.

Y Mule - dropbox_integration/f	Y Endpoint Properties		🔳 🗗 🔀
File Edit Navigate Search Project	HTTP (Inbound Endpoint)	1	
[1] • ∐ @ ≙ ! ≫ • Q •	The HTTP Endpoint allows Mule application to connect to web resources through the HTTP transport protocol.		
	This endpoint can also implement security through HTTPS.	-	
洋 Package E 🛛 🏌 🏌 Type Hier	General Advanced References HTTP Settings Documentation		
Ξ.	Display	1	Filter:
🕀 🛋 Mule Runtime [Mule Server	Display Name: HTTP		N Select
- C src/main/java	Exchange Patterns		
# src/main/resources	🔘 one-way 💿 request-response		Endpoints 🗠
src/test/resources		11.	Ajax
Referenced Libraries			Database
DropBox Integration.mf	Enabling HTTPS will require configuring a HTTPS Connector		FTP
🗁 mappings	Host: Joralbost		File
🖨 🔂 src	Parti 9091		Generic
and	Polt, 0001		() HTTP
🖨 🗁 docroot	Path: auth		
home.html			Georges 🗠
DropBox Integr			Async 🔤
mule-app.prope			🙁 Composite Source
i mule-deploy.pro			Flow
test			Q Foreach
external_webservice			Message Enricher
🗉 🔛 file_transfer			O Poll
tilter i ⊡ ⊖ flowref			Sub flow
🗈 🔛 hellomule		1	Components
the security_filter			a Transformers
The second seco	OK Cancel	J	🚰 Filters
🗷 🗁 jsp			Control
🗎 🗁 logicfilter			🚰 Error Handling
mametransfer metransfer metransfer		>	Cloud Connectors
K	Message Flow Global Elements Configuration XML		
	🚓 i i 🖉 🔣	@	🔍 M 🛛 e 🔚 🖉 e 🗐



9. Drag the **Dropbox** Cloud Connector onto the canvas. To configure the Connector, double-click on it, select the configuration reference **Dropbox**, and select the operation **Authorize**.

Y Mule - dropbox_integration/f	Y Pattern Properties	🔳 🗗 🔀
File Edit Navigate Search Project : □: • □: • □: • · · · · · · · · · · · · · · · · · ·	Dropbox Dropbox Cloud Connector. The Dropbox Connector will allow to use the Dropbox REST API. Almost every operation that can be done via the API can be done thru this connector.	
	operation that can be done via the API can be done thru this connector.	Filter: Filter: Select Components Components Filters Filters Filters Cloud Connectors Cloud C
ter ter fins ter f	OK Cancel OK Cancel Kessage Flow Global Elements Configuration XML	Objectstore Paypal Redis Salesforce
: •		

10. To create the second flow, drag the **HTTP** Endpoint onto the canvas. To configure the Endpoint, double-click on it. Enter the port number and the pathname.

Y Mule - dropbox_integration/f	🖌 Endpoint Properties 🛛 🛛 🔀	
File Edit Navigate Search Project	HTTP (Inbound Endpoint) The HTTP Endpoint allows Mule application to connect to web resources through the HTTP transport protocol. This endpoint can also implement security through HTTPS.	
Package E	General Advanced References HTTP Settings Documentation Display Name: HTTP Display Name: HTTP Exchange Patterns One-way @ request-response Back Settings Enable HTTPS @ Enable HTTPS will require configuring a HTTPS Connector Host: locahost Port: 8081 Path: home	Filter: Filter: Filter: Filter: Filter: Filter: Components Filters
politicer ⇒ constraints ⇒ constra	A	Paypal Redis Salesfarce
I e mule/flow/choice/whe	en/processor-chain 😥 🗄 🖉 🐉 🧔	0 🖳 VI 🗄 8 🔚 🗄 8 🚍

11. After the **HTTP** Endpoint configuration, you have to add the following line of code:

<http:static-resource-handler resourceBase="\${app.home}/docroot" defaultFile="home.html"></http:static-resource-handler>

Through this handler, you call the home.html page in the docroot folder.

🗄 📂 hellomule	40		
🗄 😂 http-security filter	410	<flow doc:name="HTML" name="HTML"></flow>	
🗉 😥 ims	42	<pre><http:inbound-endpoint <="" exchange-pattern="request-response" host="localhost" pre=""></http:inbound-endpoint></pre>	
🗉 🗁 ison	43	<pre><http:static-resource-handler defaultfil<="" pre="" resourcebase="\${app.home}/docroot"></http:static-resource-handler></pre>	
₽ 🔁 jsp	44		

-396

12. To create the third flow, drag the **HTTP** Endpoint onto the canvas. To configure it, double-click on the Endpoint. Enter the port number and the pathname.

Y Mule - dropbox_integration/f	🖌 Endpoint Properties	3	🔳 🗗 🗙
File Edit Navigate Search Project	HTTP (Inbound Endpoint)		
EI•∐@≙:≫•O•	The HTTP Endpoint allows Mule application to connect to web resources through the HTTP transport protocol.		
	This endpoint can also implement security through HTTPS.		
🚦 Package E 🛛 🎦 Type Hier	General Advanced References HTTP Settings Documentation		
•	Display	^	Filter:
🗄 🛋 Mule Runtime [Mule Server	Display Name: HTTP		N Select
ta angle an	CExchange Patterns		1% pelect
src/test/java			Carl Endpoints
src/test/resources			Copes Copes
Referenced Libraries	Basic Settings		Components
🗆 🥭 flows	Enable HTTPS 0		arransformers
mannings	Enabling HTTPS will require configuring a HTTPS Connector		Carl Filters
B Src	Host: localhost		Control
🖃 🐎 main	Port: 8081		🚰 Error Handling
🖃 🗁 app	Path: in		Cloud Connectors 🛛 🗠
home.html			Chudhub
index.css			
			Copbox
mule-app.prope			👼 Github
test			HBase
mule-project.xml			🔕 Intacct
🖲 🔛 external_webservice			👻 Jira
E Sther			Magento
Theref			💽 Metanga
🗊 🔁 hellomule			Mongo DB
🗊 😂 http-security_filter			Netsuite
I ison	OK Cancel		Chiectstore
isp		-	D Paynal
🗊 😂 logicfilter			Bodic
🗈 😂 nametransfer		×	Colorform
🛎 🗁 object-to-xmi	Message Flow Global Elements Configuration XML	-	Datesterce
E mule/flow/bttp:static	resource-bandler	@ 🚯	

- Mule dropbox_integration/flows/DropBox Integration.mflow Mule Studio _ P 🗙 File Edit Navigate Search Project Run Window Help 🖹] 🕸 • 🚺 • 🍇 •] 🖶 🞯 •] 🤔 🖋 •] 🖿] ½ - 🖓 - 🤤 // 🖓 • 🖉 🖉 📑 🕶 🔡 🔛 🔛 Mule 😫 Package E 🕱 Type Hier 🦳 🗖 🗈 home.html 🔯 index.css 🛛 😭 DropBox Integration 🔀 - -8 □ 🔄 🏹 ^ flow: Processor 📝 Filter: Ju 🗉 🛋 Mule Runtime [Mule Server 3.3.0 🔼 Select Carl Endpoints src/main/resources Copes 2 src/test/resources 😟 🛋 Referenced Libraries Components 😑 🗁 flows arransformers Y DropBox Integration.mflow 🔁 Filters 🗁 mappings •+∎ Logger 츦 Flow Control 😑 😥 src 😑 😥 main HTTP Body to Parameter Map Logger Choice a Error Handling app Cloud Connectors 🚊 🗁 docroot home.html
 index.css Cloudhub 💿 Dropbox DropBox Integration. mule-app.properties 👼 Github HBase 🗁 test mule-project.xml 💿 Intacct file_transfer 🏆 Jira 🝈 Magento i iter
 i iter
 i iter
 i iter 💽 Metanga Hellomule
 Http-security_filter 🚺 Mongo DB 1 Netsuite 🗉 🔛 jms flow: Authorize 🕃 Objectstore 🗉 📂 ison ⇒ json
 jsp
 ⊖ jsp
 ⊖ logicfilter
 nametransfer
 ⊖ object-to-xml 😰 Paypal 📔 Redis C Salesferce Message Flow Global Elements Configuration XML > **■**♥ 80 e 🖹 @ 😣 M 🕴 e 🗄 ə 📃 e mule/flow/http:static-resource-handler
- 13. Drag the **Body to Parameter Map** transformer onto the canvas. There is no need to configure this Endpoint.

- 14. Drag the **Logger** component onto the canvas. It is used for displaying the log on the console.
- 15. Drag the **Choice** Router onto the canvas; you can configure it later.

- 398

16. Drag the **Dropbox** Connector onto the canvas. To configure the Connector, double-click on it. Select the configuration reference name, select the operation **Create folder**, and enter the path # [payload ['dropboxPath']].

Y Mule - dropbox_integration/f	¥ Pattern Properties	X	🔳 🗗 🗾
File Edit Navigate Search Project	Dropbox Dropbox Cloud Connector. The Dropbox Connector will allow to use the Dropbox REST API. Almost every operation that can be done via the API can be done thru this connector.	7	
Mule Package E 23 Type Her Package E 23 Typ	General Display Display Name: Create Folder General Image: Create Folder General Image: Create folder General Image: Create folder Image: Create folder Image: Create folder General Image: Create folder Image: Create folder Image: Create folder Image: Create		Filter: Select Components Components Filters
bigcfilter bigcfilter bigcfilter bigcfilter bigcfilter bigcfilter bigcfilter	Konger State K	>	Paypal Redis Galesforce
: 📑 🛛 🕑 mule/flow/http:static	-resource-handler 😥 🗄 🕾 📘	@	😣 VI e 🔠 e 🖳

17. Drag the **Object to JSON** transformer onto the canvas; there is no need to configure it. This transformer is used to convert an object to the JSON format.

18. Drag the Dropbox Connector onto the canvas. To configure the Connector, double-click on it. Select the configuration reference name, select the operation Delete, and enter the path #[payload['dropboxPath']].

Y Mule - dropbox_integration/f	🖌 Pattern Properti	ies	×	🗖 🗖 🔀
File Edit Navigate Search Project	Dropbox Dropbox Cloud Connec operation that can be	tor. The Dropbox Connector will allow to use the Dropbox REST API. Almost every done via the API can be done thru this connector.		
Mule Mule Package E Package	Dropbox Cloud Conner operation that can be a General Documentati Display Name: Dele Generic Config Reference: Operation: General Path:	tor. The Dropbox Connector will allow to use the Dropbox REST API. Almost every done via the API can be done thru this connector.		Filter: Select Select Sopes Counts Sopes Counts Filters Filters Filters Filters Cloud Connectors Cloudhub Cloudhub Countable Cloudhub Select Select Select Sopes Cloudhub Select Select Select Select Sopes Sop
	(?)	Cichal Elements Configuration XM	Cancel	Image: Constraint of the second se
P mule/flow/http:static	recource-bandler		: e 🔊 a	

19. Drag another **Object to JSON** transformer onto the canvas; there is no need to configure it. This transformer is used to convert an object to the JSON format.

-400



Your third flow will look like this:

20. This is your authorized flow. You have to first run this flow:





21. The following flow is your second flow. Once you have authorized an app, you need to run this flow. This flow will call the home.html page.

flow: HTML 🕼	111	
	~	

How it works...

In this section, you will see how to deploy the application.

 To deploy the application code in the Mule server, go to Run As | Mule Application; the Mule server will deploy your application.





2. Open a browser and paste the following URL in it: http://localhost:8081/auth. Once you enter the URL, the page will redirect you to the Dropbox site, and here you have to click on **Allow** to provide access to the app.

	. Get free space!	Azaz Desai 🔻
The app MuleESB App would like to		
connect with your Dropbox.		
 This app will have access to your entire Dropbox. 		
Please make sure you trust this app before proceeding.		
 You're currently signed in as azaz_d@yahoo.com. If you meant to conn- another account, you can sign out. 	ect from	
Allow	Deny	

3. Now you can run the second flow, http://localhost:8081/home/. Here you need to select the operation. Suppose you have selected the **Create Folder** operation. You then have to enter the folder name, DropBoxIntegration, in the **Path:** textbox.

me to AttuneInfocomiii		
tion:Select Option Select Option Create Folder Delete		
	it on a select Option- Select Option- Create Folder Delete	me to AttuneInfocom!!! tion:Select Option Select Option Create Folder Delete



4. Once you click on the **Submit** button, you will see an output, similar to the following screenshot, in your browser:



5. Check your Dropbox account. You will see that a folder has been created on your account named DropBoxIntegration.

← → C 🔒 https://www.	.dropbox.com/home			ন্দ্র 🞯	≡
			\\ Get free space!	Azaz Desai 🔻	~
Image: Specific Scheme Sche	🛟 Dropbox	J 🖬 🖬 🖌	Q Search Dropbox		
	Name 🔺	Kind	Modified		Ξ
🌈 Sharing	Camera Uploads	folder			
Links Events	ContactsKit	folder		Ø	
Get Started	DropBoxIntegration	folder		Ø	

-404

Index

A

Advanced tab, Servlet Endpoint 199 agent 12 AJAX 181 **AJAX Endpoint** about 133, 182 used, for sending messages asynchronously 182-197 All Router 30, 339 All Router/Flow Control about 339 configuring 340-350 allrouter.mflow file 341 And filter 249 Append String transformer 201 architecture, Mule ESB application layer 11 diagrammatic representation 11 integration layer 11 transport layer 11 Asynchronous Java and XML. See AJAX

В

bean, evaluator expressions 260

С

canvas 23 catch exception properties configuring 283 Catch Exception Strategy about 282 catch exception properties, configuring 283 configuring 283

use case 283 using 282 **Choice Exception Strategy** about 284 configuring 284, 285 defining 284 use case 285 using 285 Choice Router 30, 339, 350 **Choice Router/Flow Control** configuring 351-361 Choice Router.mflow file 352 **Cloud Connectors** about 30.371 Salesforce 31 Twitter 31 command prompt used, for running Mule application 57 components configuring 44, 45 custom filters 47 Groovy component 101 Java component 100 Javascript component 101 Python component 100 Ruby component 100 Script component 100 components, Mule Echo 27 Logger 27 REST 27 SOAP 27 components, Mule Studio canvas 23 package tree 21

palette 22 components, palette components about 26 general components 26 script components 26 web service components 26 configuration, All Router/Flow **Control 340-350** configuration, Choice Router/Flow Control 351-361 configuration, Generic Endpoint 134, 135 configuration, HTTP Endpoint 135-144 configuration, IMAP Endpoint for e-mails retrieval 145-147 configuration, JDBC Endpoint 147-163 configuration, Servlet Endpoint 198-200 configuration, Splitter Flow Control 362-369 connectors 12 content-type property 123 custom expression evaluator creating 260 **Custom filter** about 29 creating 47, 48, 273 Custom filter project creating 274, 275 deploying 279 Echo component, configuring 277, 278 HTTP Endpoint, configuring 276 custom message sources 114 custom transformer about 28, 226 creating 227-232 working 234, 235

D

database connection JDBC Endpoint, using for 147-163 DataMapper transformer about 201, 235 configuring 237-243 working 246, 247 Documentation tab, Servlet Endpoint 197 Dropbox 384 Dropbox Connector configuring 386-395

using, in flow 396-401 working 402-404 **DropBoxIntegration folder** configuring 384, 385 DropBox_Integration project application code, deploying in Mule server 301-303 Choice Router, configuring 294, 295 Cloud Connector, configuring 296 creating 290 Dropbox connector, configuring 291 home.html file 297 HTTP Endpoint, configuring 292-297 index.css file 299 JUnit test case, creating 305 static resources handler, adding 297

Ε

Echo component about 27,48 used, for displaying message payload 48 using 54 Echo project creating 49 deploying 55, 56 Echo component, using 51 executing, command prompt used 57 flow, creating 50 HTTP Endpoint, configuring 50 Logger component, configuring 52, 53 Logger component, using 51 Eclipse download link 16 Mule IDE, setting up 15 Endpoint 13, 133 **Endpoints, palette components** about 25 File Endpoint 25 FTP Endpoint 25 Generic Endpoint 26 HTTP Endpoint 26 Inbound Endpoint 25 JMS Endpoint 26 Outbound Endpoint 25 VM Endpoint 26

Enterprise Service Bus. See ESB ESB about 8 applications, connecting to 9 features 8 functionality 8 evaluator expressions bean 260 exception type 260 groovy 260 header payload type 260 regex 260 types 260 wildcard 260 exception 258, 281 **Exception filter** about 29, 258 using 258 working 259 expression filter 29 expressions about 260 JXPath expressions 261 OGNL expressions 261 XPath expressions 261 Extensible Markup Language (XML) technology 311 external_webservice project creating 333 deploying 337, 338 HTTP Endpoint, configuring 334 SOAP component, configuring 335 external web services calling, SOAP component used 329-336

F

File Endpoint about 25, 164 used, for implementing File Transport channel 164-181
File Transport channel implementing, File Endpoint used 164-181
filtering performing 258
filter reference 30

filters

about 13.29 custom filter 29 exception filter 29 expression filter 29 filter reference 30 message property filter 29 payload filter 30 regular expression filter 30 wildcard filter 30 flow 13 Flow Controls 339 **Flow Reference** about 57 used, for executing another flow 57-68 **FlowRef project** creating 58 deploying 69, 71 Flow Reference component, configuring 66 HTTP Endpoint, configuring 61, 62 Java component, configuring 63, 67 FTP Endpoint 25, 134

G

general components 26 General tab, Servlet Endpoint 198 Generic Endpoint about 26, 133 configuring 134, 135 getwelcomeMsg() 73 global configuration 12 Global Endpoints 12 global message processor 12 Groovy component about 101 using 101

Η

Hello World application deploying, on Mule 31-41
HelloWorld project creating 124 custom component, creating 124, 126 deploying 130, 132

HTTP Endpoint, configuring 127 Java component, configuring 128, 129 hiMule() method 86 HTTP Endpoint about 26, 133, 135 configuring 135-144

I

IMAP 145 IMAP Endpoint about 133, 145 configuring, for e-mails retrieval 145-147 IMAP/POP3 connector 145 Inbound Endpoints 25, 113, 133 inbound properties 122, 224 incoming events or messages handling, Message filter used 261 integration 12 Internet Message Access Protocol. *See* IMAP invocation properties, message property scopes 123

J

Java component about 46, 100 configuring 100 Javascript component 101 Java transformer 201 **JAX-WS** services creating 313-320 JDBC Endpoint about 147 configuring 147-163 using, for database connection 147-163 JMS Endpoint 26, 133 JSON-to-Object transformer about 202 configuring 202 database, configuring 209 Database Endpoint, configuring 210 using 202-208 working 212, 213 JUnit about 289 used, for testing in Mule ESB 289

JUnit4TestAdapter 289 JUnit test case deploying, in Mule 308, 310 JXPath expressions 261

L

Logger component 27 Logic Filter project creating 250 deploying 256, 257 HTTP Endpoint, configuring 251 Java component, configuring 255 Logic filter, configuring 253 logic filters And filter 249 configuring 249 Or filter 249 Payload Type filter 249 Wildcard filter 249

Μ

Message filter about 261 configuring 262 Message Property window 263 used, for handling incoming events or messages 261 working 263 message processor about 114 example 114 one-way exchange pattern 112 request-response pattern 112 Message Properties transformer about 223 configuring 224, 225 inbound properties 224 outbound properties 224 session properties, adding 225 message property filter 29 message property scopes about 122 inbound properties 122 invocation properties 123 outbound properties 123 session properties 123

Message Property window inbound 263 invocation 263 outbound 263 session 263 messages filtering 260 sending, AJAX Endpoint used 182-197 message sources about 112 custom message sources 114 Inbound Endpoints 113 polls 113 using 112 **Messaging Exception Strategies** about 282 Catch Exception Strategy 282 Choice Exception Strategy 284 Default Exception Strategy 282 Reference Exception Strategy 286 Rollback Exception Strategy 288 types 282 models 13 Mule logic filters, configuring 249 web services, proxying 312 Mule components about 23 configuring 24 Java component 23 palette components 24 simple component 23 types 23 Mule configuration about 11 agent 12 connectors 12 Endpoints 13 filters 13 flow 13 global configuration 12 Global Endpoints 12 global message processor 12 integration 12 model 13 service component development 11 service orchestration 11

services 13 Spring beans 12 transformer 13 Mule configuration, in Eclipse performing 19, 20 muleCookBook method 59 Mule ESB about 7,10 advantages 10 architecture 11 capabilities 10 testing, with JUnit 289 working 11 Mule Expression Language (MEL) 52 Mule IDE setting up 13-17 Mule IDE Standalone 3.3 downloading 15 **Mule server** Hello World application, deploying 31-38 MuleSoft **URL 13** Mule Studio about 20 components 21 downloading 20 environment variable, setting 21 installing 20

0

Object-to-Xml transformer 28 Object-to-XML transformer about 214 configuring 214-221 working 222 OGNL expressions 261 Or filter 249 Outbound Endpoint 25, 133 outbound properties 123, 224

Ρ

package tree about 21 graphical flow, creating 21 palette 22



palette components about 24 Cloud Connectors 30 components 26 Endpoints 25 filters 29 routers 30 transformers 28 payload filter 30 polls 113 POP3 145 POP3 Endpoint 145 Post Office Protocol Version 3. See POP3 protocol binding 312, 313 Python component about 100 using 100

R

Reference Exception Strategy about 286 configuring 286, 287 References tab, Servlet Endpoint 200 RegEx filter 260 regular expression filter 30 Representational State Transfer, See REST REST about 322 used, for publishing RESTful web service 72 restbasedwebservice project about 323 deploying 328 HTTP Endpoint, configuring 325 Logger component, configuring 326 REST component, configuring 327 **REST component** about 27 used, for creating web service 322-325 **RESTful web service** about 72 creating 73 publishing, REST used 72 **REST** project creating 73 deploying 81-83 flow, creating 75, 76

Logger component, configuring 80 RESTful web service, creating 77, 78 **Rollback Exception Strategy** configuring 288 working 289 **routers** about 30 all router 30 choice router 30 **Ruby component 100**

S

Salesforce Cloud Connector 31 sayHi() method 124 Script component about 26, 100 using 100 Script project creating 102 deploying 109-112 Groovy component, configuring 107, 108 HTTP Endpoint, configuring 104 Logger component, configuring 105, 106 Script Transformer 28 service component development 11 service orchestration 11 services 13 Servlet Endpoint Advanced tab 199 configuring 198, 200 Documentation tab 197 General tab 198 References tab 200 used, for listening to events from servlet requests 197, 199 session properties, message property scopes 123 Singleton object 100 SMTP Endpoint 133 SOAP used, for publishing SOAP web service 84 SOAP-based web service creating 84 **SOAP component 27** soap-jax-ws project creating 314

deploying 321, 322 HTTP Endpoint, configuring 317 Java component, configuring 319 SOAP component, configuring 318 **SOAP** project creating 85 deploying 94-97 flow, creating 88-91 HTTP Endpoint, configuring 89 Java component, configuring 92, 93 SOAP component, configuring 90 **SOAP** web service publishing, SOAP used 84 Splitter 339, 361 **Splitter Flow Control** configuring 362-369 splitter.mflow file 363 Spring beans 12 Spring object 100 STDIO component 114 StudioConnector project creating 115, 118 deploying 121, 122

Т

Transformer Ref 28 transformers about 13, 28, 201 custom transformer 28 Object-to-Xml transformer 28 Script transformer 28 Transformer Ref 28 Xml-to-Object Transformer 29 XSLT Transformer 29 transformers, types Append String 201 DataMapper 201 Java 201 XSLT 201 **Twitter Cloud Connector** about 31, 371 configuring 372-378 Expression transformer, configuring 379, 380 working 381-383

V

Variable transformer 223, 226 VM Endpoint 26, 134

W

web service about 311 creating, REST component used 322-327 proxying 312 types 312 web service components 26 Web Services Description Language 97 Wildcard filter about 30, 264 configuring 264 Wildcard Filter project creating 265-267 deploying 272 HTTP Endpoint, configuring 269 Java transformer, configuring 271 Wildcard filter, configuring 270 WS-Proxy web service 312 WS-Security 312

X

Xml-to-Object Transformer 29 XPath expressions 261 XSLT Transformer 29



Thank you for buying Mule ESB Cookbook

About Packt Publishing

Packt, pronounced 'packed', published its first book "*Mastering phpMyAdmin for Effective MySQL Management*" in April 2004 and subsequently continued to specialize in publishing highly focused books on specific technologies and solutions.

Our books and publications share the experiences of your fellow IT professionals in adapting and customizing today's systems, applications, and frameworks. Our solution based books give you the knowledge and power to customize the software and technologies you're using to get the job done. Packt books are more specific and less general than the IT books you have seen in the past. Our unique business model allows us to bring you more focused information, giving you more of what you need to know, and less of what you don't.

Packt is a modern, yet unique publishing company, which focuses on producing quality, cuttingedge books for communities of developers, administrators, and newbies alike. For more information, please visit our website: www.packtpub.com.

About Packt Open Source

In 2010, Packt launched two new brands, Packt Open Source and Packt Enterprise, in order to continue its focus on specialization. This book is part of the Packt Open Source brand, home to books published on software built around Open Source licences, and offering information to anybody from advanced developers to budding web designers. The Open Source brand also runs Packt's Open Source Royalty Scheme, by which Packt gives a royalty to each Open Source project about whose software a book is sold.

Writing for Packt

We welcome all inquiries from people who are interested in authoring. Book proposals should be sent to author@packtpub.com. If your book idea is still at an early stage and you would like to discuss it first before writing a formal book proposal, contact us; one of our commissioning editors will get in touch with you.

We're not just looking for published authors; if you have strong technical skills but no writing experience, our experienced editors can help you develop a writing career, or simply get some additional reward for your expertise.

[PACKT] Open source &





JBoss ESB Beginner's Guide

ISBN: 978-1-84951-658-7

Paperback: 320 pages

A comprehensive, practical guide to developing service-based applications using the Open Source JBoss Enterprise Service Bus

- 1. Develop your own service-based applications, from simple deployments through to complex legacy integrations
- 2. Learn how services can communicate with each other and the benefits to be gained from loose coupling
- Contains clear, practical instructions for service development, highlighted through the use of numerous working examples



SOA Made Simple

ISBN: 978-1-84968-416-3

Paperback: 292 pages

Discover the true meaning behind the buzzword that is 'Service Oriented Architecture'

- 1. Get to grips with clear definitions of 'Service' and 'Architecture' to understand the full SOA picture
- 2. Read about SOA in simple terms from Oracle ACE Directors for SOA and Middleware in this book and e-book
- 3. A concise, no-nonsense guide to demystifying Service Oriented Architecture

Please check www.PacktPub.com for information on our titles





Drools JBoss Rules 5.X Developer's Guide

ISBN: 978-1-78216-126-4

Paperback: 338 pages

Define and execute your business rules with Drools

- 1. Learn the power of Drools as a platform for writing your business rules
- 2. Integrate Drools into your Java business application using the Spring framework
- 3. Through real-world examples and solutions, you will be taken from novice to expert



JBoss Weld CDI for Java Platform

ISBN: 978-1-78216-018-2

Paperback: 122 pages

Learn CDI concepts and develop modern web applications using JBoss Weld

- 1. Learn about dependency injection with CDI
- 2. Install JBoss Weld in your favorite container
- 3. Develop your own extension to CDI
- 4. Decouple code with CDI events
- 5. Communicate between CDI beans and AngularJS

Please check www.PacktPub.com for information on our titles