tutorialspoint

SIMPLY EASY LEARNING

www.tutorialspoint.com

https://www.facebook.com/tutorialspointindia

https://twitter.com/tutorialspoint

About the Tutorial

RESTful Web Services are basically REST Architecture based Web Services. In REST Architecture everything is a resource. RESTful web services are light weight, highly scalable and maintainable and are very commonly used to create APIs for web-based applications.

This tutorial will teach you the basics of RESTful Web Services and contains chapters discussing all the basic components of RESTful Web Services with suitable examples.

Audience

This tutorial is designed for Software Professionals who are willing to learn RESTful Web Services in simple and easy steps. This tutorial will give you great understanding on RESTful Web Services concepts and after completing this tutorial you will be at intermediate level of expertise from where you can take yourself at higher level of expertise.

Prerequisites

Before proceeding with this tutorial, you should have a basic understanding of Java Language, Text Editor, etc. Because we are going to develop web services applications using RESTful, so it will be good if you have understanding on other web technologies like HTML, CSS, AJAX, etc.

Copyright & Disclaimer

Table of Contents

1. RESTful Web Services – Introduction

What is REST?

REST stands for **RE**presentational **S**tate **T**ransfer. REST is a web standards based architecture and uses HTTP Protocol for data communication. It revolves around resources where every component is a resource and a resource is accessed by a common interface using HTTP standard methods. REST was first introduced by Roy Fielding in year 2000.

In REST architecture, a REST Server simply provides access to resources and the REST client accesses and presents the resources. Here each resource is identified by URIs/ Global IDs. REST uses various representations to represent a resource like Text, JSON and XML. JSON is now the most popular format being used in Web Services.

HTTP Methods

The following HTTP methods are most commonly used in a REST based architecture.

- **GET** - Provides a read only access to a resource.
- **PUT** - Used to create a new resource.
- **DELETE** - Used to remove a resource.
- **POST** - Used to update an existing resource or create a new resource.
- **OPTIONS** - Used to get the supported operations on a resource.

RESTFul Web Services

A web service is a collection of open protocols and standards used for exchanging data between applications or systems. Software applications written in various programming languages and running on various platforms can use web services to exchange data over computer networks like the Internet in a manner similar to inter-process communication on a single computer. This interoperability (e.g., between Java and Python, or Windows and Linux applications) is due to the use of open standards.

Web services based on REST Architecture are known as RESTful Web Services. These web services use HTTP methods to implement the concept of REST architecture. A RESTful web service usually defines a URI (Uniform Resource Identifier), which is a service that provides resource representation such as JSON and a set of HTTP Methods.

Creating RESTFul Web Service

In this tutorial, we will create a web service called **User Management** with the following functionalities:

Sr. No.	HTTP Method	URI	Operation	Operation Type
1	**GET**	/UserService/users	Get list of users	Read Only
2	**GET**	/UserService/users/1	Get User with Id 1	Read Only
3	**PUT**	/UserService/users/2	Insert User with Id 2	Idempotent
4	**POST**	/UserService/users/2	Update User with Id 2	N/A
5	**DELETE**	/UserService/users/1	Delete User with Id 1	Idempotent
6	**OPTIONS**	/UserService/users	List the supported operations in web service	Read Only

2. RESTful Web Services – Environment Setup

This tutorial will guide you on how to prepare a development environment to start your work with **Jersey Framework** to create RESTful Web Services. Jersey framework implements **JAX-RS 2.0** API, which is a standard specification to create RESTful Web Services. This tutorial will also teach you how to setup **JDK**, **Tomcat** and **Eclipse** on your machine before you the Jersey Framework is setup.

Setup Java Development Kit (JDK)

You can download the latest version of SDK from Oracle's Java site: Java SE Downloads. You will find the instructions for installing JDK in the downloaded files. Follow the given instructions to install and configure the setup. Finally set the **PATH** and **JAVA_HOME** environment variables to refer to the directory that contains **Java** and **Javac**, typically java_install_dir/bin and java_install_dir respectively.

If you are running Windows and installed the JDK in C:\jdk1.7.0_75, you would have to put the following line in your C:\autoexec.bat file.

```
set PATH=C:\jdk1.7.0_75\bin;%PATH%
set JAVA_HOME=C:\jdk1.7.0_75
```

Alternatively, on Windows NT/2000/XP, you could also right-click on My Computer → select Properties → then Advanced → then Environment Variables. Then, you would update the PATH value and press the OK button.

On Unix (Solaris, Linux, etc.), if the SDK is installed in /usr/local/jdk1.7.0_75 and you use the C Shell, you would put the following into your .cshrc file.

```
setenv PATH /usr/local/jdk1.7.0_75/bin:$PATH
setenv JAVA_HOME /usr/local/jdk1.7.0_75
```

Alternatively, if you use an Integrated Development Environment (IDE) like Borland JBuilder, Eclipse, IntelliJ IDEA, or Sun ONE Studio, compile and run a simple program to confirm that the IDE knows where you installed Java, otherwise do proper setup as given document of the IDE.

Setup Eclipse IDE

All the examples in this tutorial have been written using the Eclipse IDE. So, I would suggest you should have the latest version of Eclipse installed on your machine.

To install Eclipse IDE, download the latest Eclipse binaries from http://www.eclipse.org/downloads/. Once you downloaded the installation, unpack the binary distribution to a convenient location. For example, in C:\eclipse on windows, or /usr/local/eclipse on Linux/Unix and finally set the PATH variable appropriately.

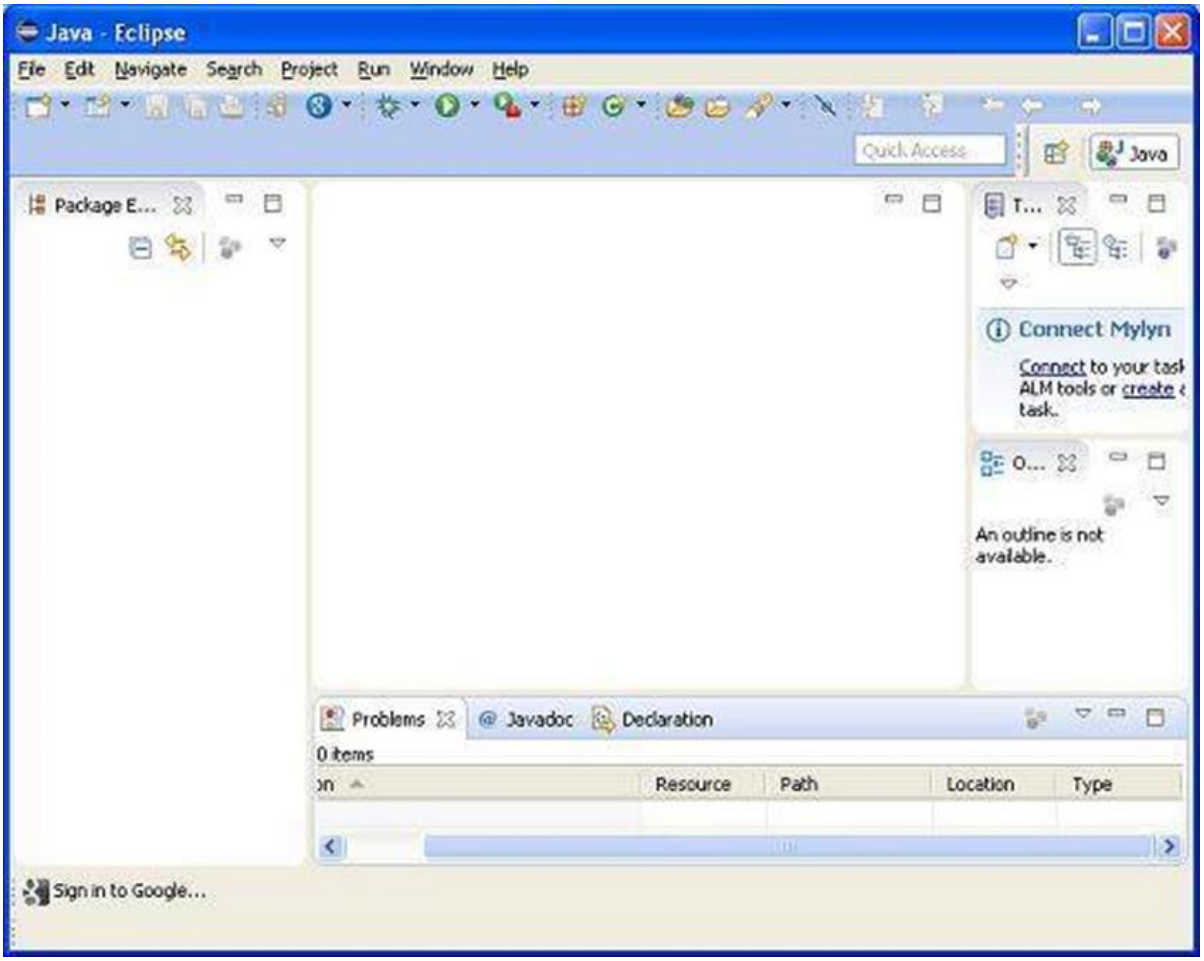
Eclipse can be started by executing the following commands on a windows machine, or you can simply double click on eclipse.exe

```
%C:\eclipse\eclipse.exe
```

Eclipse can be started by executing the following commands on Unix (Solaris, Linux, etc.) machine:

```
$/usr/local/eclipse/eclipse
```

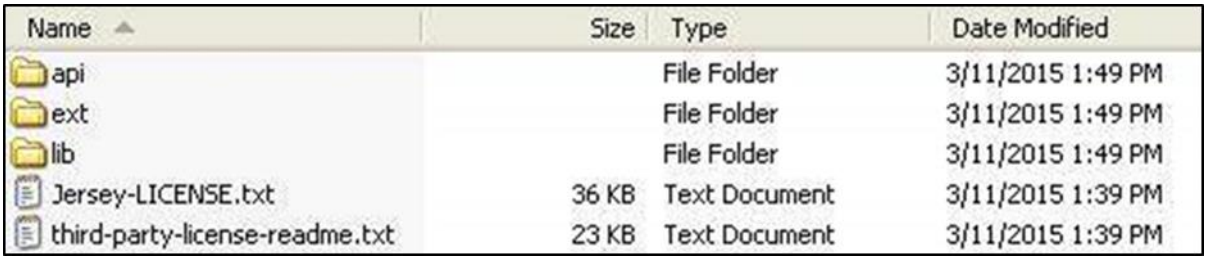
After a successful startup, if everything is fine, then your screen should display the following result:



Setup Jersey Framework Libraries

Now, if everything is fine, then you can proceed to setup the Jersey framework. Following are a few simple steps to download and install the framework on your machine.

- Make a choice whether you want to install Jersey on Windows, or Unix and then proceed to the next step to download the .zip file for windows and then the .tz file for Unix.
- Download the latest version of Jersey framework binaries from the following link – https://jersey.java.net/download.html.
- At the time of writing this tutorial, I downloaded **jaxrs-ri-2.17.zip** on my Windows machine and when you unzip the downloaded file it will give you the directory structure inside E:\jaxrs-ri-2.17\jaxrs-ri as shown in the following screenshot.



Name	Size	Type	Date Modified
api		File Folder	3/11/2015 1:49 PM
ext		File Folder	3/11/2015 1:49 PM
lib		File Folder	3/11/2015 1:49 PM
Jersey-LICENSE.txt	36 KB	Text Document	3/11/2015 1:39 PM
third-party-license-readme.txt	23 KB	Text Document	3/11/2015 1:39 PM

You will find all the Jersey libraries in the directories **C:\jaxrs-ri-2.17\jaxrs-ri\lib** and dependencies in **C:\jaxrs-ri-2.17\jaxrs-ri\ext**. Make sure you set your CLASSPATH variable on this directory properly otherwise you will face problem while running your application. If you are using Eclipse, then it is not required to set the CLASSPATH because all the settings will be done through Eclipse.

Setup Apache Tomcat

You can download the latest version of Tomcat from http://tomcat.apache.org/. Once you downloaded the installation, unpack the binary distribution into a convenient location. For example in C:\apache-tomcat-7.0.59 on windows, or /usr/local/apache-tomcat-7.0.59 on Linux/Unix and set CATALINA_HOME environment variable pointing to the installation locations.

Tomcat can be started by executing the following commands on a windows machine, or you can simply double click on startup.bat.

```
%CATALINA_HOME%\bin\startup.bat
```

or

```
C:\apache-tomcat-7.0.59\bin\startup.bat
```

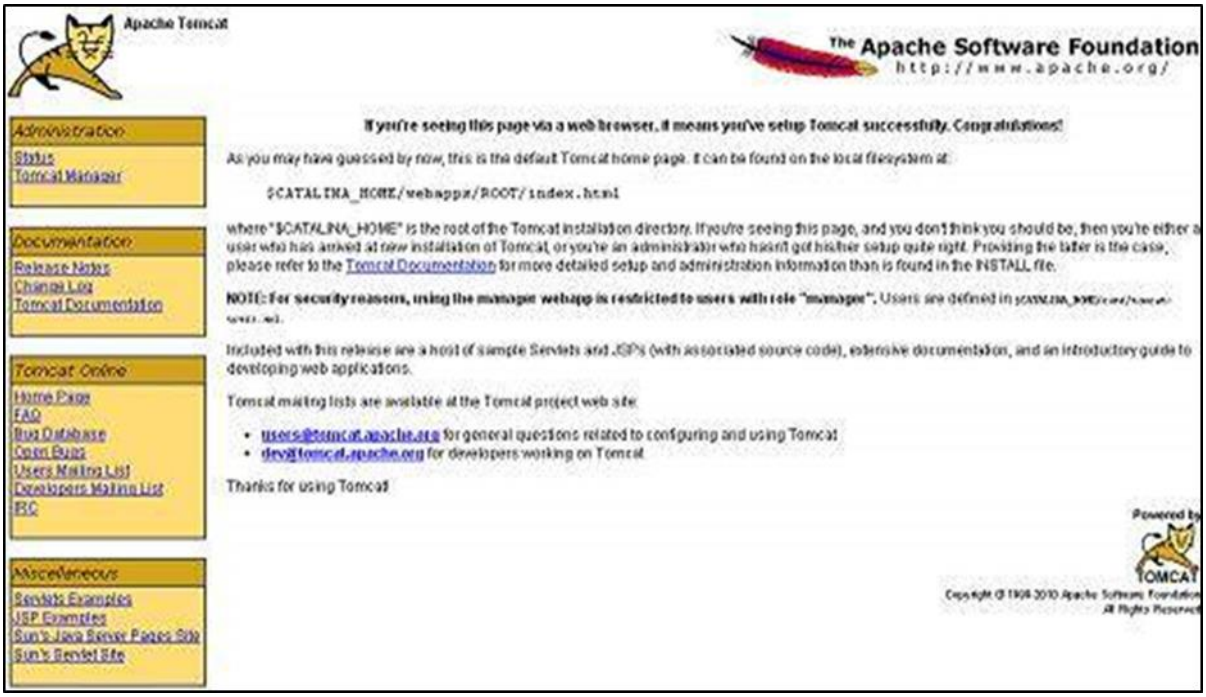
Tomcat can be started by executing the following commands on a Unix (Solaris, Linux, etc.) machine:

```
$CATALINA_HOME/bin/startup.sh
```

or

```
/usr/local/apache-tomcat-7.0.59/bin/startup.sh
```

After a successful startup, the default web applications included with Tomcat will be available by visiting **http://localhost:8080/**. If everything is fine then it should display the following result:



Further information about configuring and running Tomcat can be found in the documentation included on this page. This information can also be found on the Tomcat website: http://tomcat.apache.org.

Tomcat can be stopped by executing the following commands on a windows machine:

```
%CATALINA_HOME%\bin\shutdown
```

or

```
C:\apache-tomcat-7.0.59\bin\shutdown
```

Tomcat can be stopped by executing the following commands on Unix (Solaris, Linux, etc.) machine:

```
$CATALINA_HOME/bin/shutdown.sh
```

or

```
/usr/local/apache-tomcat-7.0.59/bin/shutdown.sh
```

Once you are done with this last step, you are ready to proceed for your first Jersey example which you will see in the next chapter.

End of ebook preview

If you liked what you saw…

Buy it from our store @ **https://store.tutorialspoint.com**