Introduction to Web Services (SOAP and REST) – 2.0
Lab Book

Document Revision History

Date	Revision No.	Author	Summary of Changes
May 2016	1.0	Yukti A Valecha	Created new lab book as per revised course contents

Table of Contents

Getting Started 4

Overview 4

Setup Checklist for Web Services 2.0 4

Instructions 4

Lab 1. Web Services (JAX-WS) 5

Lab 2. Web Services (JAX-RS) 6

Appendices 8

Appendix A: Naming Conventions 8

Appendix B: Table of Figures 9

Getting Started

Overview

This lab book is a guided tour for learning Introduction to Web Services (SOAP and REST) version 2.0. It comprises 'To Do' assignments.

Setup Checklist for Web Services 2.0

Here is what is expected on your machine in order for the lab to work.

Minimum System Requirements

- Intel Pentium 90 or higher (P166 recommended)
- Microsoft Windows 95, 98, or NT 4.0, 2k, XP.
- Memory: 32MB of RAM (64MB or more recommended)
- Internet Explorer 6.0 or higher
- Wildfly Server 8.0.

Please ensure that the following is done:

- A text editor like Notepad or Eclipse is installed.
- JDK 1.8 is installed. (This path is henceforth referred as <java_install_dir>)
- Wildfly is installed but not started.

Instructions

- For all naming conventions refer Appendix A. All lab assignments should refer coding standards.
- Create a directory by your name in drive <drive>. In this directory, create a subdirectory webservices_assgn. For each lab exercise create a directory as lab <lab number>.
- You may also look up the on-line help if necessary.

Lab 1. Web Services (JAX-WS)

Goals	Understand the process of creating and Consuming a Java Web Service
Time	50 minutes

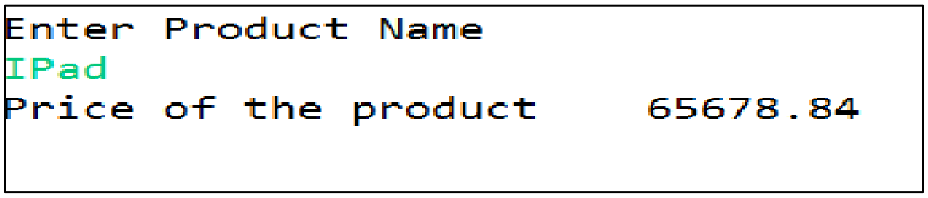
1. Refer below Java files:
    - Product.java
    - ProductDB.java

Create a Product Web service that will accept the product name and return the price of the product to the web service consumer.

If the product is not available then the web service should return a message to the consumer specifying that the product is not available.

Note: Make use of the static DB given in ProductDB.java

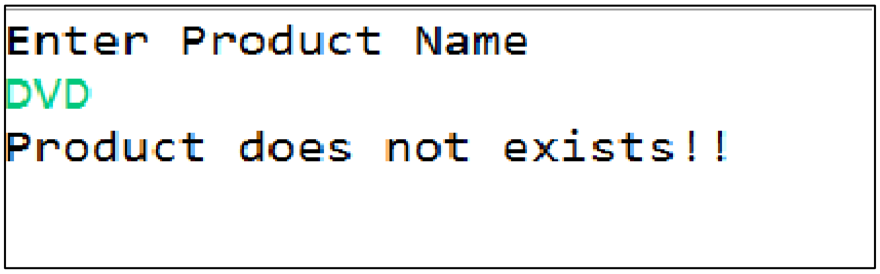
Refer below screen shots:



```
Enter Product Name
IPad
Price of the product     65678.84
```

Figure 1 : Screenshot

If the user enters a product that does not exists then, refer below screen shot



```
Enter Product Name
DVD
Product does not exists!!
```

Figure 2 : Screenshot

Lab 2. Web Services (JAX-RS)

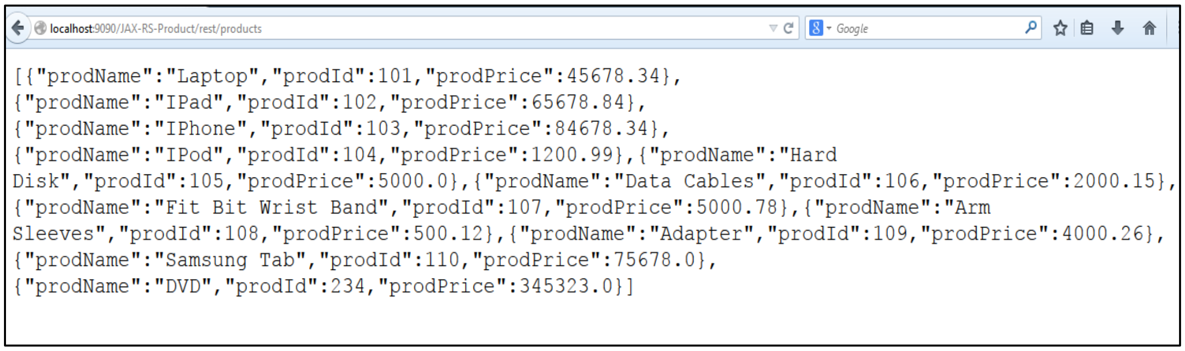
Goals	Understand the process of creating and Consuming a RESTful Java Web Service
Time	90 minutes

1. Refer below Java files:
   [Product.java]
   [ProductDB.java]

   Create a Product RESTful web service that will display all products to the Web service consumer.

   Note: Make use of the static DB given in ProductDB.java

   Refer below screen shot:



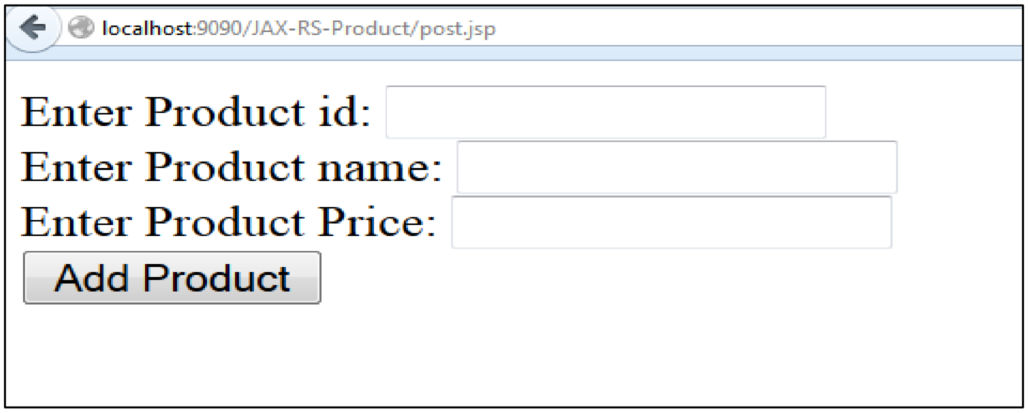
localhost:9090/JAX-RS-Product/rest/products

```
[{"prodName":"Laptop","prodId":101,"prodPrice":45678.34},
{"prodName":"IPad","prodId":102,"prodPrice":65678.84},
{"prodName":"IPhone","prodId":103,"prodPrice":84678.34},
{"prodName":"IPod","prodId":104,"prodPrice":1200.99},{"prodName":"Hard
Disk","prodId":105,"prodPrice":5000.0},{"prodName":"Data Cables","prodId":106,"prodPrice":2000.15},
{"prodName":"Fit Bit Wrist Band","prodId":107,"prodPrice":5000.78},{"prodName":"Arm
Sleeves","prodId":108,"prodPrice":500.12},{"prodName":"Adapter","prodId":109,"prodPrice":4000.26},
{"prodName":"Samsung Tab","prodId":110,"prodPrice":75678.0},
{"prodName":"DVD","prodId":234,"prodPrice":345323.0}]
```

Figure 3 : Screenshot

The consumer should also be able to add a product to the existing list of products.

Refer below screen shots:

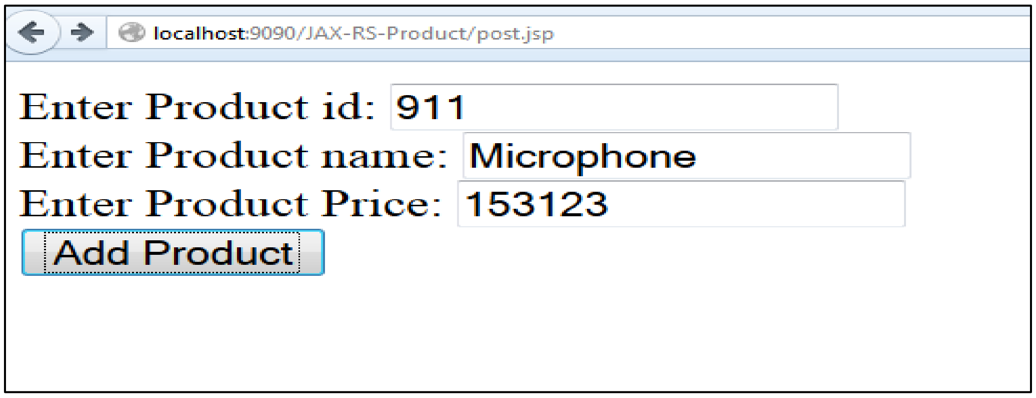


localhost:9090/JAX-RS-Product/post.jsp

Enter Product id:
Enter Product name:
Enter Product Price:
Add Product

Figure 4 : Input Screen

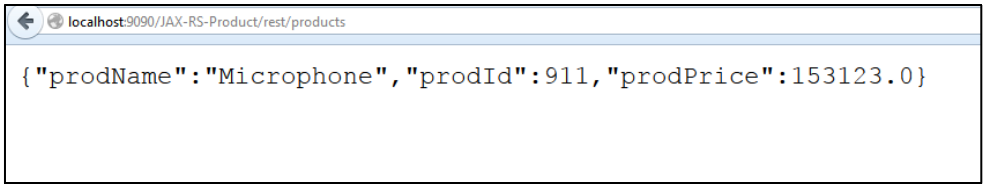
After entering product details, refer below screen shot



Enter Product id: 911
Enter Product name: Microphone
Enter Product Price: 153123

Figure 5 : Input Screen with data

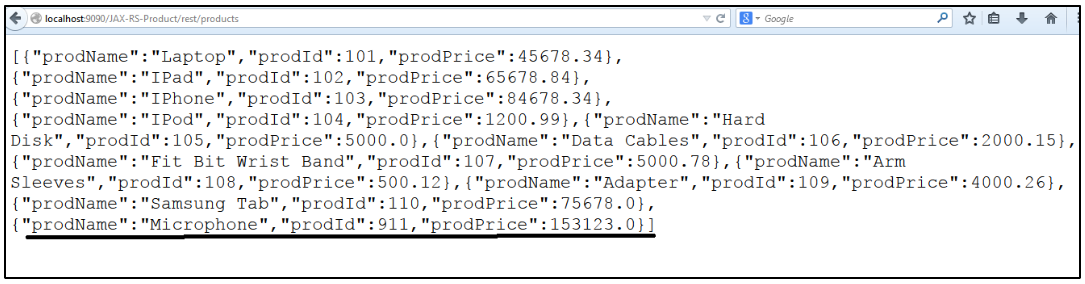
Product is added to existing list of products



```
{"prodName":"Microphone","prodId":911,"prodPrice":153123.0}
```

Figure 6 : Screen shot

Now, new product is added to existing list of products



```
[{"prodName":"Laptop","prodId":101,"prodPrice":45678.34},
{"prodName":"IPad","prodId":102,"prodPrice":65678.84},
{"prodName":"IPhone","prodId":103,"prodPrice":84678.34},
{"prodName":"IPod","prodId":104,"prodPrice":1200.99},{"prodName":"Hard
Disk","prodId":105,"prodPrice":5000.0},{"prodName":"Data Cables","prodId":106,"prodPrice":2000.15},
{"prodName":"Fit Bit Wrist Band","prodId":107,"prodPrice":5000.78},{"prodName":"Arm
Sleeves","prodId":108,"prodPrice":500.12},{"prodName":"Adapter","prodId":109,"prodPrice":4000.26},
{"prodName":"Samsung Tab","prodId":110,"prodPrice":75678.0},
{"prodName":"Microphone","prodId":911,"prodPrice":153123.0}]
```

Figure 7 : Screen shot

Appendices

Appendix A: Naming Conventions

Package names are written in all lower case to avoid conflict with the names of classes or interfaces. Companies use their reversed Internet domain name to begin their package names—for example, com.cg.learning.mypackage for a package named learning.mypackage created by a programmer at cg.com.
Packages in the Java language itself begin with **java. Or javax**.

Classes and interfaces the first letter should be capitalized, and if several words are linked together to form the name, the first letter of the inner words should be uppercase (a format that's sometimes called "camelCase").
For classes, the names should typically be nouns. For example:
Dog
Account
PrintWriter
For interfaces, the names should typically be adjectives like
Runnable
Serializable
Methods The first letter should be lowercase, and then normal camelCase rules should be used. In addition, the names should typically be verb-noun pairs. For example:
getBalance
doCalculation
setCustomerName
Variables Like methods, the camelCase format should be used, starting with a lowercase letter. Sun recommends short, meaningful names, which sounds good to us. Some examples:
buttonWidth
accountBalance
myString
Constants Java constants are created by marking variables static and final. They should be named using uppercase letters with underscore characters as separators:
MIN_HEIGHT

Appendix B: Table of Figures

Figure 1 : Screenshot 5
Figure 2 : Screenshot 5
Figure 3 : Screenshot 6
Figure 4 : Input Screen 6
Figure 5 : Input Screen with data 7
Figure 6 : Screen shot 7
Figure 7 : Screen shot 7