Consuming a Web Service (SOAP and RESTful) in Java

Cheat Sheet For Consuming Services in Java

This document will provide a user the capability to create an application to consume a sample web service (Both SOAP and RESTful) in Java using the following technologies

1> Eclipse IDE

Karan Moodbidri and Amey Gawde
7/25/2013
**Prerequisites:**

Before we start with the process of consuming web services

Eclipse IDE (Kepler preferred - Can be Downloaded from here - [Link](#))

**Consuming a Web Service in Java**

- Open Eclipse IDE
- Click on New
- Go to the Pop up Option 'Other'
- Search for Dynamic Web Project

![Image of Eclipse IDE with a search bar for Dynamic Web Project](image-url)
- Select The same options as in the image

- Click Next
- Click Next
- Click on Project
- Click on New
- Click On other
- Select Web Service Client
- Enter WSDL Link in the Textbox
- Click Next
- Click Finish
Now we need to write Code to Consume the Service.

- Click on Project
- Right Click on Java Resources in the Project Tree Structure
- Click on New
- Click on Class
- Select the Same options as given in the Picture
- Click on Finish
- You will have a java program open in front of you
- Go to java resources
- Go into src
- Check the name of the package other than the package you created
- Go into the package
- Remember to Check the name of the file with Proxy in the end.
- Now import the WS package into your java program
Now Create an object of the proxy class you previously checked
Call the function of the web service with the object of the proxy class and print it.

Code should look like this:

```java
public static void main(String[] args) throws RemoteException {
    // TODO Auto-generated method stub
    WeatherSoapProxy wc = new WeatherSoapProxy();

    System.out.println(wc.getCityForecastByZIP("14623").getForecastResult()[0].getTemperatures().getMorningLow());
}
```

Run the program
You have successfully Created a web client for the SOAP service.
Creating Client for RESTful Services

- Open Eclipse IDE
- Click on New
- Go to the Pop up Option 'Other'

- Search for Dynamic Web Project
➢ Select The same options as in the image

➢ Click Next
➢ Click Next
➢ Click Finish
➢ Click on Project
➢ Right Click on Java Resources in the Project Tree Structure
➢ Click on New
➢ Click on Class
➢ Select the Same options as given in the Picture
Click on Finish
You will have a java program open in front of you
You will need to include the following two packages
import com.sun.jersey.api.client.Client;
import com.sun.jersey.api.client.WebResource;
Follow the code given Below:

Note: The String held by variable apiCall should be the URL to the RESTful Service.

Client client = Client.create();
String apiCall = null;
apiCall = "http://www.exampleRestApi/q=";
WebResource webResource = client.resource(apiCall);
String outputString = webResource.get(String.class);
System.out.println(outputString);

You have created a client for your RESTful Service.

We can Consume a REST based Service without using the Jersey API. I will be providing the instructions Below.
We would first create a normal Java project.
For that, click File
Click New
Click Java Project

Enter the Name of the project
Click Finish
Right click on Project
Click on New
Click on Class
Select the same options as given in the picture
➢ Click on Finish
➢ Now I would go through the code
➢ First we will create a URL variable to store the URL for the Rest Service.

Example code:

```java
URL url = new URL("http://www.omdbapi.com/?s=Titanic");
```

➢ We open a HTTP Connection on that URL. We set the requestMethod to "GET" and we connect to the Rest Service.

Example Code:

```java
HttpURLConnection connection = (HttpURLConnection)url.openConnection();
connection.setRequestMethod("GET");
connection.connect();
```

➢ We now to have read data from the rest service input stream

Example Code:

```java
BufferedReader bufferReader = new BufferedReader(new InputStreamReader(connection.getInputStream()));
```
Then we read the data from the stream into a String buffer to print it onto the console.

**Example Code:**

```java
String str;
StringBuffer stringBuffer = new StringBuffer();
while ((str = bufferReader.readLine()) != null) {
    stringBuffer.append(str);
    stringBuffer.append("\n");
}
System.out.println(stringBuffer.toString());
```

We have successfully consumed a service. The code entire code is as below:

```java
URL url = new URL("http://www.omdbapi.com/?s=Titanic");
HttpURLConnection connection = (HttpURLConnection)url.openConnection();
connection.setRequestMethod("GET");
connection.connect();
BufferedReader bufferReader = new BufferedReader(new InputStreamReader(connection.getInputStream()));
String str;
StringBuffer stringBuffer = new StringBuffer();
while ((str = bufferReader.readLine()) != null) {
    stringBuffer.append(str);
    stringBuffer.append("\n");
}
System.out.println(stringBuffer.toString());
```