

Create Very Simple Jersey REST Service and Send JSON Data From Java Client

 crunchify.com/create-very-simple-jersey-rest-service-and-send-json-data-from-java-client

App Shah

November 28, 2013

```
@Path("/")
public class CrunchifyRESTService {
    @POST
    @Path("/crunchifyService")
    @Consumes(MediaType.APPLICATION_JSON)
    public Response crunchifyREST(InputStream incomingData) {
        StringBuilder crunchifyBuilder = new StringBuilder();
        try {
            BufferedReader in = new BufferedReader(new InputStreamReader(incomingData));
            String line = null;
            while ((line = in.readLine()) != null) {
                crunchifyBuilder.append(line);
            }
        } catch (Exception e) {
            System.out.println("Error Parsing: - ");
        }
        System.out.println("Data Received: " + crunchifyBuilder.toString());
    }
}
```

Recently I have to pass JSON data to REST Service and did not have any simple Client handy. But created very simple Java program which read JSON data from file and sends it to REST service.

Representational State Transfer (REST) has gained widespread acceptance across the Web as a simpler alternative to SOAP- and Web Services Description Language (WSDL)-based Web services. Key evidence of this shift in interface design is the adoption of REST by mainstream Web 2.0 service providers—including Yahoo, Google, and Facebook—who have deprecated or passed on SOAP and WSDL-based interfaces in favor of an easier-to-use, resource-oriented model to expose their services. In this article, Alex Rodriguez introduces you to the basic principles of REST.

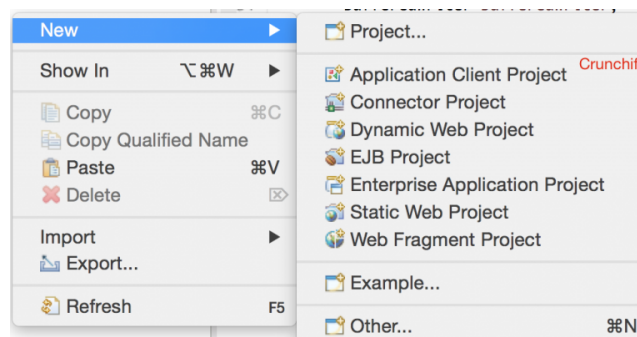
Let's start coding this:

1. Create RESTFul Web Service
 - Java file: CrunchifyRESTService.java
 - web.xml file
2. Create RESTService Client
 - 1. CrunchifyRESTServiceClient.java file

Another must read: [Spring MVC Example/Tutorial: Hello World – Spring MVC 3.2.1](#)

Step-1

In **Eclipse => File => New => Dynamic Web Project** . Name it as “ **CrunchifyTutorials** ”. Below tutorial also works with **Tomcat 8** .



New Dynamic Web Project Crunchify Tutorials

Dynamic Web Project
Create a standalone Dynamic Web project or add it to a new or existing Enterprise Application.

Project name:

Project location

☒ Use default location

Location:

Target runtime

Try to use Tomcat 7

Dynamic web module version

Step-2 Create Deployment Descriptor File

If you don't see `web.xml` (deployment descriptor) under `WebContent\WEB-INF\` then follow these steps.

Open `web.xml` and replace content with below contents:

`web.xml` file

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <web-app xmlns="http://java.sun.com/xml/ns/javaee" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
4      http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd"
5  version="3.0">
6  <display-name>CrunchifyRESTJerseyExample</display-name>
7  <welcome-file-list>
8  <welcome-file>index.html</welcome-file>
9  <welcome-file>index.htm</welcome-file>
10 <welcome-file>index.jsp</welcome-file>
11 <welcome-file>default.html</welcome-file>
12 <welcome-file>default.htm</welcome-file>
13 <welcome-file>default.jsp</welcome-file>
14 </welcome-file-list>
15 <servlet>
16 <servlet-name>Jersey Web Application</servlet-name>
17 <servlet-class>com.sun.jersey.spi.container.servlet.ServletContainer</servlet-class>
18 <load-on-startup>1</load-on-startup>
19 </servlet>
20 <servlet-mapping>
21 <servlet-name>Jersey Web Application</servlet-name>
22 <url-pattern>/api/*</url-pattern>
23 </servlet-mapping>
24 </web-app>
25

```

Step-3 Convert Project to Maven Project

Follow this tutorial: <https://crunchify.com/how-to-convert-existing-java-project-to-maven-in-eclipse/>. Here is my pom.xml file.

pom.xml file

```
1  <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
2  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
3  <modelVersion>4.0.0</modelVersion>
4  <groupId>CrunchifyTutorials</groupId>
5  <artifactId>CrunchifyTutorials</artifactId>
6  <version>0.0.1-SNAPSHOT</version>
7  <packaging>war</packaging>
8  <build>
9  <sourceDirectory>src</sourceDirectory>
10 <plugins>
11 <plugin>
12 <artifactId>maven-compiler-plugin</artifactId>
13 <version>3.1</version>
14 <configuration>
15 <source>1.8</source>
16 <target>1.8</target>
17 </configuration>
18 </plugin>
19 <plugin>
20 <artifactId>maven-war-plugin</artifactId>
21 <version>2.4</version>
22 <configuration>
23 <warSourceDirectory>WebContent</warSourceDirectory>
24 <failOnMissingWebXml>false</failOnMissingWebXml>
25 </configuration>
26 </plugin>
27 </plugins>
28 </build>
29 <dependencies>
30 <dependency>
31 <groupId>asm</groupId>
32 <artifactId>asm-all</artifactId>
33 <version>3.3.1</version>
34 </dependency>
35 <dependency>
36 <groupId>com.sun.jersey</groupId>
37 <artifactId>jersey-bundle</artifactId>
38 <version>1.14</version>
39 </dependency>
40 <dependency>
41 <groupId>org.json</groupId>
42 <artifactId>json</artifactId>
43 <version>20090211</version>
44 </dependency>
45 </dependencies>
46 </project>
```

Step-4

Create RESTful service: `CrunchifyRETSERVICE.java` . Here we will create two services:

1. `/api/crunchifyService` – POST call – we will use this with our test
2. `/api/verify` – GET call – just to make sure service started successfully

CrunchifyRETSERVICE.java

Java

```

1  package com.crunchify.tutorials;
2  /**
3   * @author Crunchify.com
4   *
5   */
6  import java.io.BufferedReader;
7  import java.io.InputStream;
8  import java.io.InputStreamReader;
9  import javax.print.attribute.standard.Media;
10 import javax.ws.rs.Consumes;
11 import javax.ws.rs.GET;
12 import javax.ws.rs.POST;
13 import javax.ws.rs.Path;
14 import javax.ws.rs.Produces;
15 import javax.ws.rs.core.MediaType;
16 import javax.ws.rs.core.Response;
17 @Path("/")
18 public class CrunchifyRESTService {
19     @POST
20     @Path("/crunchifyService")
21     @Consumes(MediaType.APPLICATION_JSON)
22     public Response crunchifyREST(InputStream incomingData) {
23         StringBuilder crunchifyBuilder = new StringBuilder();
24         try {
25             BufferedReader in = new BufferedReader(new InputStreamReader(incomingData));
26             String line = null;
27             while ((line = in.readLine()) != null) {
28                 crunchifyBuilder.append(line);
29             }
30         } catch (Exception e) {
31             System.out.println("Error Parsing: - ");
32         }
33         System.out.println("Data Received: " + crunchifyBuilder.toString());
34         // return HTTP response 200 in case of success
35         return Response.status(200).entity(crunchifyBuilder.toString()).build();
36     }
37     @GET
38     @Path("/verify")
39     @Produces(MediaType.TEXT_PLAIN)
40     public Response verifyRESTService(InputStream incomingData) {
41         String result = "CrunchifyRESTService Successfully started..";
42         // return HTTP response 200 in case of success
43         return Response.status(200).entity(result).build();
44     }
45 }
46
47
48
49
50
51
52
53

```

Step-5

Deploy project **CrunchifyTutorials** on Tomcat. Web project should be deployed without any exception.

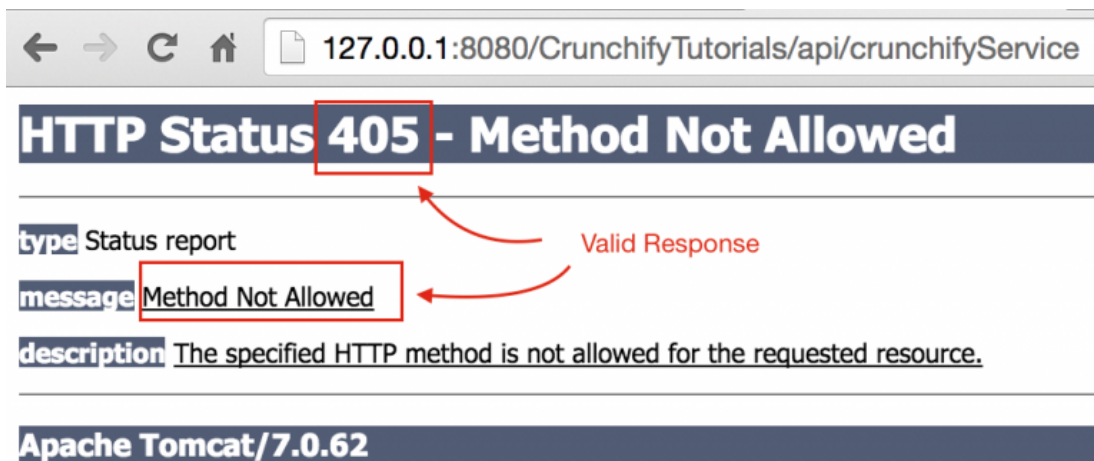
1. Right Click on **Servers** tab in Eclipse
2. Click on **Add and Remove...** Project
3. Add Project CrunchifyTutorials to right **Configured:** side.
4. Click on **Publish**
5. Click on **Start**

Step-6 Verify REST service

Rest service should be accessible using this URL: <http://127.0.0.1:8080/CrunchifyTutorials/api/verify>



If you try to access `http://127.0.0.1:8080/CrunchifyTutorials/api/crunchifyService` then you will see error code `405 - Method not allowed` – which is valid response. As you can see it's POST call and should expect some data with the request.



Let's move on.

Step-7

Copy below JSON content and put it under `C:\CrunchifyJSON.txt` file for windows or `/Users/<username>/Documents/CrunchifyJSON.txt` file if Macbook.

CrunchifyJSON.txt

```
1 {
2   "tutorials": {
3     "id": "Crunchify",
4     "topic": "REST Service",
5     "description": "This is REST Service Example by Crunchify."
6   }
7 }
```

Step-8

Create REST Call Client: `CrunchifyRESTServiceClient.java`.

Please change path to `CrunchifyJSON.txt` in below program.

CrunchifyRESTServiceClient.java

Java

```

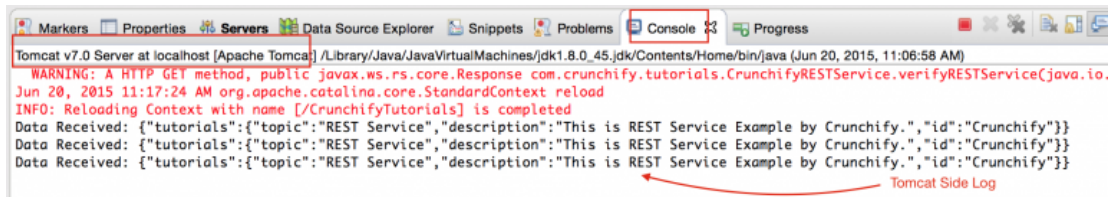
1  package com.crunchify.tutorials;
2  import java.io.BufferedReader;
3  import java.io.FileInputStream;
4  import java.io.InputStream;
5  import java.io.InputStreamReader;
6  import java.io.OutputStreamWriter;
7  import java.net.URL;
8  import java.net.URLConnection;
9  import org.json.JSONObject;
10 /**
11  * @author Crunchify.com
12  *
13  */
14 public class CrunchifyRESTServiceClient {
15     public static void main(String[] args) {
16         String string = "";
17         try {
18             // Step1: Let's 1st read file from fileSystem
19             // Change CrunchifyJSON.txt path here
20             InputStream crunchifyInputStream = new FileInputStream("/Users/<username>/Documents/CrunchifyJSON.txt");
21             InputStreamReader crunchifyReader = new InputStreamReader(crunchifyInputStream);
22             BufferedReader br = new BufferedReader(crunchifyReader);
23             String line;
24             while ((line = br.readLine()) != null) {
25                 string += line + "\n";
26             }
27             JSONObject jsonObject = new JSONObject(string);
28             System.out.println(jsonObject);
29             // Step2: Now pass JSON File Data to REST Service
30             try {
31                 URL url = new URL("http://localhost:8080/CrunchifyTutorials/api/crunchifyService");
32                 URLConnection connection = url.openConnection();
33                 connection.setDoOutput(true);
34                 connection.setRequestProperty("Content-Type", "application/json");
35                 connection.setConnectTimeout(5000);
36                 connection.setReadTimeout(5000);
37                 OutputStreamWriter out = new OutputStreamWriter(connection.getOutputStream());
38                 out.write(jsonObject.toString());
39                 out.close();
40                 BufferedReader in = new BufferedReader(new InputStreamReader(connection.getInputStream()));
41                 while (in.readLine() != null) {
42                     // 
43                 }
44                 System.out.println("\nCrunchify REST Service Invoked Successfully..");
45                 in.close();
46             } catch (Exception e) {
47                 System.out.println("\nError while calling Crunchify REST Service");
48                 System.out.println(e);
49             }
50             br.close();
51         } catch (Exception e) {
52             e.printStackTrace();
53         }
54     }
55 }
56
57
58
59
60
61
62
63
64

```

Step-9

Now let's run Client Program by right click on CrunchifyRESTServiceClient.java and you should see below two outputs

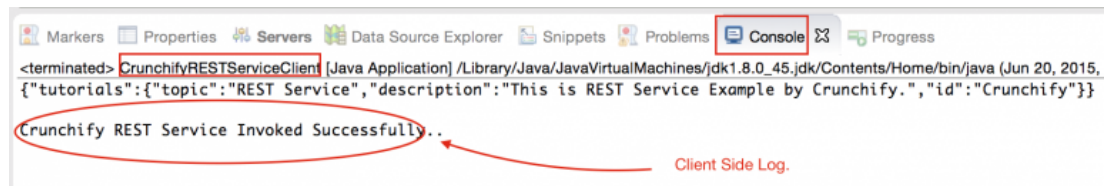
1) in Tomcat Console



```
Tomcat v7.0 Server at localhost [Apache Tomcat] /Library/Java/JavaVirtualMachines/jdk1.8.0_45.jdk/Contents/Home/bin/java (Jun 20, 2015, 11:06:58 AM)
WARNING: A HTTP GET method, public javax.ws.rs.core.Response com.crunchify.tutorials.CrunchifyRETSERVICE.verifyRETSERVICE(java.io.
Jun 20, 2015 11:17:24 AM org.apache.catalina.core.StandardContext reload
INFO: Reloading Context with name [/CrunchifyTutorials] is completed
Data Received: {"tutorials":{"topic":"REST Service","description":"This is REST Service Example by Crunchify.","id":"Crunchify"}}
Data Received: {"tutorials":{"topic":"REST Service","description":"This is REST Service Example by Crunchify.","id":"Crunchify"}}
Data Received: {"tutorials":{"topic":"REST Service","description":"This is REST Service Example by Crunchify.","id":"Crunchify"}}
```

Tomcat Side Log

2) in Local Client Console



```
<terminated> CrunchifyRETSERVICEClient [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_45.jdk/Contents/Home/bin/java (Jun 20, 2015,
{"tutorials":{"topic":"REST Service","description":"This is REST Service Example by Crunchify.","id":"Crunchify"}}
Crunchify REST Service Invoked Successfully..
```

Client Side Log.

Join the Discussion

Share & leave us some comments on what you think about this topic or if you like to add something.