

[Home](#) [All Tutorials](#) [Java Core](#) [JSF](#) [Spring](#) [Hibernate](#) [Struts](#) [Android](#) [Others](#)

Jersey Hello World Example



Posted on July 18, 2011 , Last modified : August 29, 2012

By [mkyong](#)

[Jersey](#), reference implementation to develop RESTful web service based on the [JAX-RS \(JSR 311\)](#) specification.

In this tutorial, we show you how to develop a simple hello world REST web application with **Jersey**.

Technologies and Tools used in this article:

1. Jersey 1.8
2. JDK 1.6
3. Tomcat 6.0
4. Maven 3.0.3
5. Eclipse 3.6

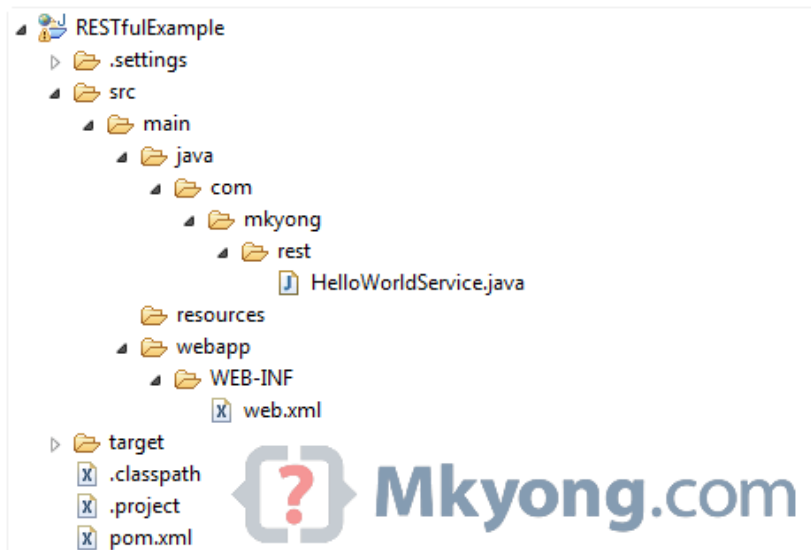
Note

If you want to know what and how REST works, just search on Google, ton of available resources.

1. Directory Structure

This is the final web project structure of this tutorial.

Recent Posts



PHÒNG KHÁM ĐA KHOA NHÂN ÁI
 KHÁM BỆNH DA DÀY, KHÔNG ỔNG SOI, KHÔNG ĐAU
 CHUYÊN ĐIỀU TRỊ BỆNH **DA DÀY, ĐƯỜNG RUỘT**

- 709 Giải Phóng HN -
0438-647-991

TƯ VẤN

2. Standard Web Project

Create a standard Maven web project structure.

```
mvn archetype:generate -DgroupId=com.mkyong.rest -DartifactId=RESTfulExample
-DarchetypeArtifactId=maven-archetype-webapp -DinteractiveMode=false
```

Note

To support Eclipse, use Maven command :

```
mvn eclipse:eclipse -Dwtpversion=2.0
```



Ant - How to create a Java Project



Create a fat Jar file - Maven Assembly Plugin



Maven - Exclude log4j.properties in Jar file



Maven - Create a fat Jar file - One-JAR example



Java - Cron job to run a jar file

Popular Tutorials



Android Tutorial



JSF 2.0 Tutorial



Spring Tutorial



Maven Tutorial



Hibernate Tutorial

3. Project Dependencies

Jersey is published in Java.net Maven repository. To develop Jersey REST application , just declares “**jersey-server**” in Maven `pom.xml` I.

File : pom.xml

```
<project ...>

    <repositories>
        <repository>
            <id>maven2-repository.java.net</id>
            <name>Java.net Repository for Maven</name>
            <url>http://download.java.net/maven/2/</url>
            <layout>default</layout>
        </repository>
    </repositories>

    <dependencies>

        <dependency>
            <groupId>com.sun.jersey</groupId>
            <artifactId>jersey-server</artifactId>
            <version>1.8</version>
        </dependency>

    </dependencies>

</project>
```

4. REST Service

Simple REST service with Jersey.

```
package com.mkyong.rest;

import javax.ws.rs.GET;
import javax.ws.rs.Path;
import javax.ws.rs.PathParam;
import javax.ws.rs.core.Response;

@Path("/hello")
public class HelloWorldService {

    @GET
    @Path("/{param}")
    public Response getMsg(@PathParam("param") String msg) {

        String output = "Jersey say : " + msg;

        return Response.status(200).entity(output).build();

    }

}
```

5. web.xml

In `web.xml`, register `com.sun.jersey.spi.container.servlet.ServletContainer`, and puts your Jersey service folder under `init-param`, `com.sun.jersey.config.property.packages`.

File : `web.xml`

```
<web-app id="WebApp_ID" version="2.4"
    xmlns="http://java.sun.com/xml/ns/j2ee"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee
http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd">
    <display-name>Restful Web Application</display-name>
```

```
<servlet>
  <servlet-name>jersey-servlet</servlet-name>
  <servlet-class>
    com.sun.jersey.spi.container.servlet.ServletContainer
  </servlet-class>
  <init-param>
    <param-name>com.sun.jersey.config.property.packages</param-name>
    <param-value>com.mkyong.rest</param-value>
  </init-param>
  <load-on-startup>1</load-on-startup>
</servlet>

<servlet-mapping>
  <servlet-name>jersey-servlet</servlet-name>
  <url-pattern>/rest/*</url-pattern>
</servlet-mapping>

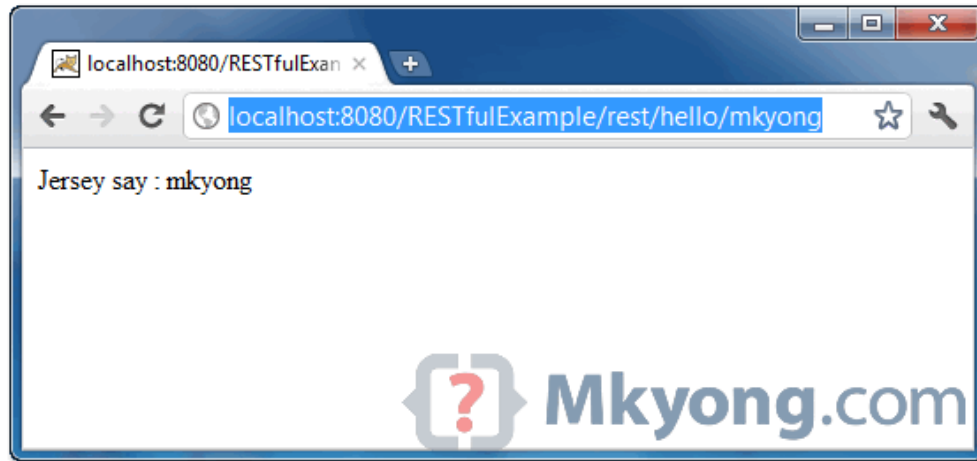
</web-app>
```

6. Demo

In this example, web request from “**projectURL/rest/hello/**” will match to “**HelloWorldService**”, via `@Path("/hello")`.

And the “**{any values}**” from “**projectURL/rest/hello/{any values}**” will match to parameter annotated with `@PathParam`.

URL : `http://localhost:8080/RESTfulExample/rest/hello/mkyong`



Download Source Code

Download it – [JAX-RS-Hello-World-Jersey-Example.zip](#) (6 KB)

References

1. [Jersey Official Website](#)
2. [RESTEasy hello world example](#)
3. [REST with Java \(JAX-RS\) using Jersey – Tutorial](#)
4. [IBM : RESTful Web services: The basics](#)
5. [RESTful Web Services](#)

Tags : [hello world](#) [jax-rs](#) [jersey](#)



mkyong

Founder of Mkyong.com and HostingCompass.com, love Java and open source stuff. Follow him on [Twitter](#), or befriend him on [Facebook](#) or [Google Plus](#). If you like my tutorials, consider making a [donation to this charity](#), thanks.

Related Posts

- ▶ [RESTful Java client with Jersey client](#)
- ▶ [JSON example with Jersey + Jackson](#)
- ▶ [File upload example in Jersey](#)
- ▶ [Jersey : The ResourceConfig instance does not contain any root resource classes](#)
- ▶ [ClassNotFoundException : com.sun.jersey.spi.container.servlet.ServletContainer](#)

Popular Posts

- ▶ [Top 8 Java People You Should Know](#)
- ▶ [Top 20 Java Websites](#)
- ▶ [Top 5 Free Java eBooks](#)
- ▶ [Top 10 Java Regular Expression Examples](#)
- ▶ [Top 5 Open Source Q&A Systems](#)

You might also like following tutorials :

