

RESTEasy hello world example

By [mkyong](#) | June 30, 2011 | Updated : August 29, 2012 | Viewed : 431,515 times +1,753 pv/w

[RESTEasy](#), JBoss project, implementation of the [JAX-RS](#) specification. In this tutorial, we show you how to use RESTEasy framework to create a simple REST style web application.

Technologies and Tools used in this article:

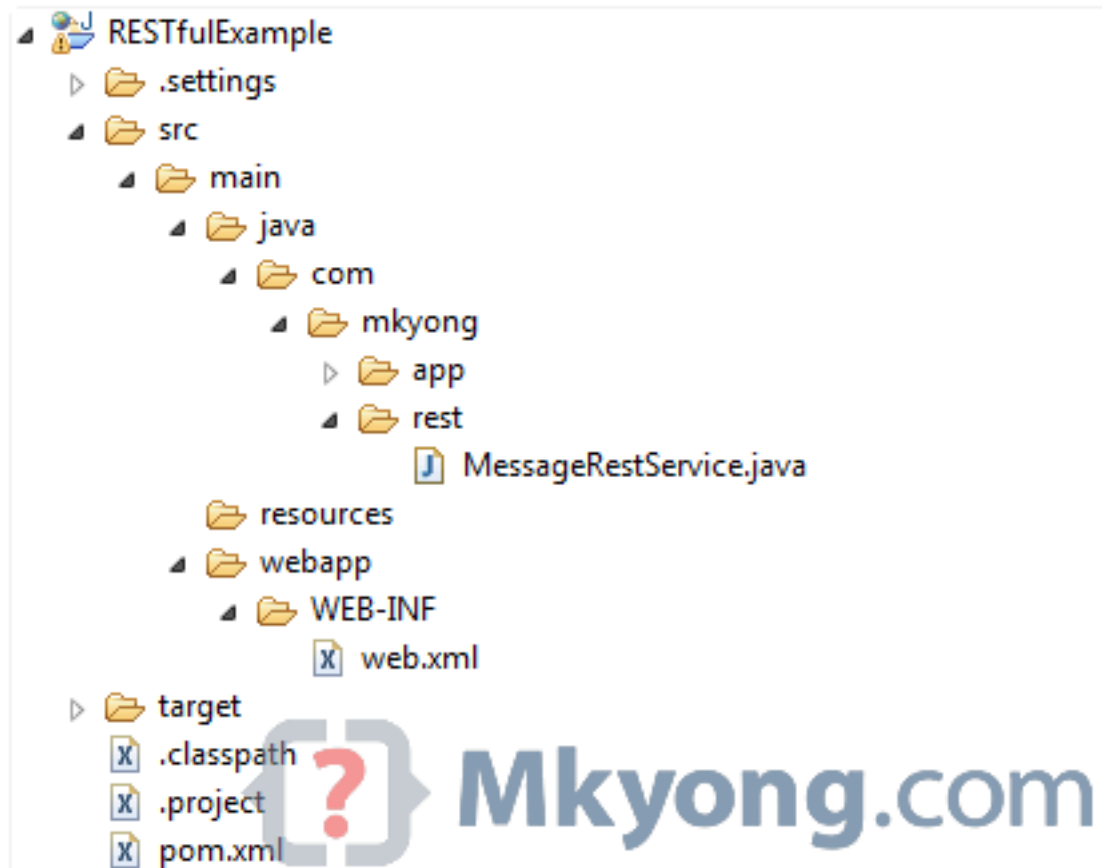
1. RESTEasy 2.2.1.GA
2. JDK 1.6
3. Maven 3.0.3
4. Eclipse 3.6

What's REST?

Read [this](#), [this](#) and [this](#) to understand what's REST.

1. Directory Structure

Review final directory structure of this tutorial. Just a standard web project structure.





2. Standard Web Project

Create a standard Maven web project structure.

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

Convert to Eclipse web project.

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



3. Project Dependencies

Declares JBoss public Maven repository and “**resteasy-jaxrs**” in your Maven `pom.xml` file. That’s all you need to use **RESTEasy**.

File : pom.xml

```
<project ...">

    <repositories>
        <repository>
            <id>JBoss repository</id>
            <url>https://repository.jboss.org/nexus/content/groups/public-jboss/</url>
        </repository>
    </repositories>

    <dependencies>

        <dependency>
            <groupId>org.jboss.resteasy</groupId>
            <artifactId>resteasy-jaxrs</artifactId>
            <version>2.2.1.GA</version>
        </dependency>

    </dependencies>
</project>
```

4. REST Service

A simple REST service. See demo at the end of the article, it should be self-explanatory.

```
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("/message")
public class MessageRestService {

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

        String result = "Restful example : " + msg;

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

    }

}
```

5. web.xml

Now, configure listener and servlet to support RESTEasy. Read this [JBoss documentation](#) for detail explanation.

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>

  <!-- Auto scan REST service -->
  <context-param>
    <param-name>resteasy.scan</param-name>
    <param-value>true</param-value>
  </context-param>

  <!-- this need same with resteasy servlet url-pattern -->
  <context-param>
    <param-name>resteasy.servlet.mapping.prefix</param-name>
    <param-value>/rest</param-value>
  </context-param>

  <listener>
    <listener-class>
      org.jboss.resteasy.plugins.server.servlet.ResteasyBootstrap
    </listener-class>
  </listener>
```

```
<servlet>
    <servlet-name>resteasy-servlet</servlet-name>
    <servlet-class>
        org.jboss.resteasy.plugins.server.servlet.HttpServletDispatcher
    </servlet-class>
</servlet>

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

</web-app>
```

Note

You need to set the “**resteasy.servlet.mapping.prefix**” if your servlet-mapping for the resteasy servlet has a url-pattern other than “/*”.

In above example, the resteasy servlet url-pattern is “**/rest/***”, so you have to set the “resteasy.servlet.mapping.prefix” to “**/rest**” as well, otherwise, you will hit resource not found error message.

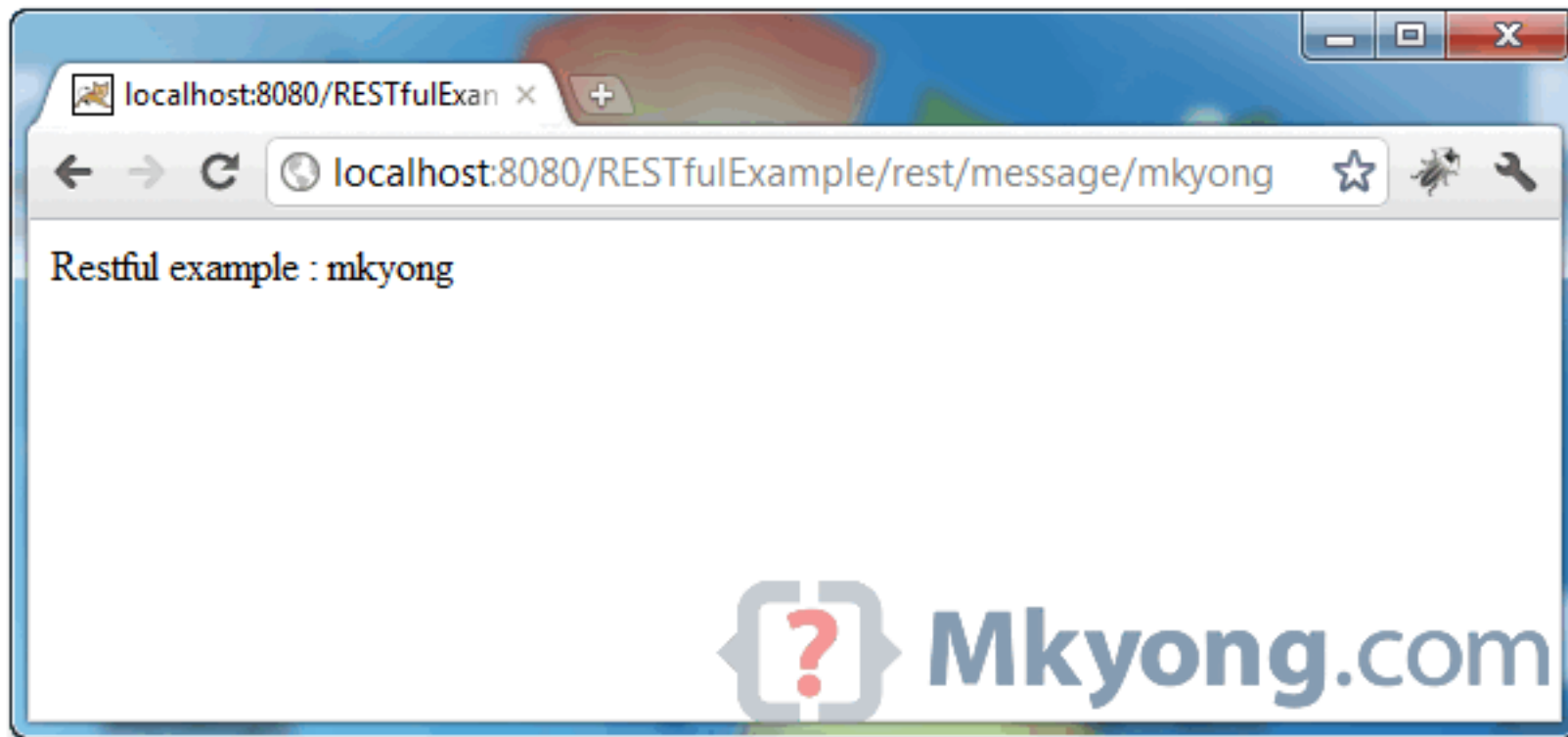
Note

Remember to set “**resteasy.scan**” to true, so that RESTEasy will find and register your REST service automatically.

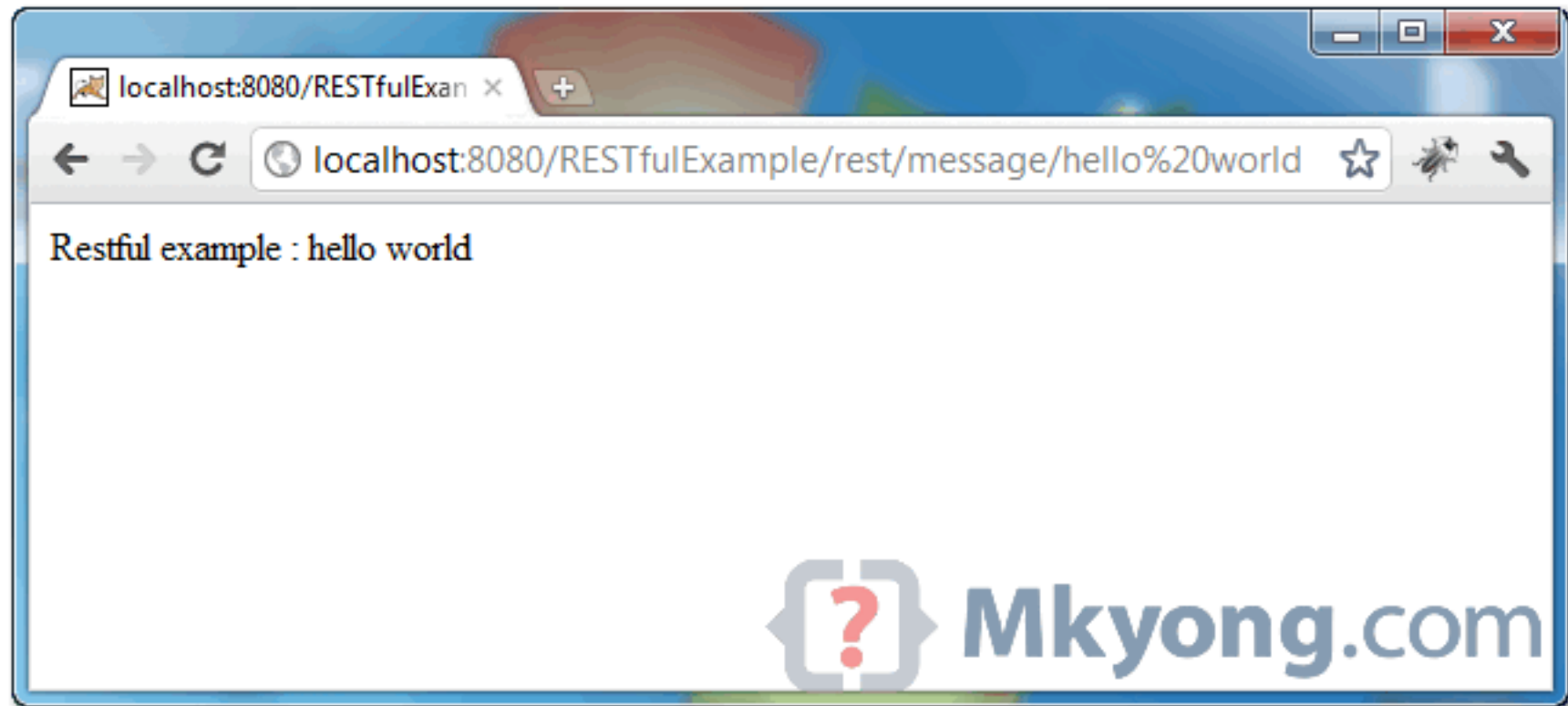
6. Demo

In this example, web request from “**projectURL/rest/message/**” will match to “**MessageRestService**“, and “**projectURL/rest/message/{any values}**” will match to **@PathParam** parameter.

Test 1 : <http://localhost:8080/RESTfulExample/rest/message/mkyong>



Test 2 : <http://localhost:8080/RESTfulExample/rest/message/hello%20world>



Alternative REST Service Registration

In above example, you are register REST service via “**ResteasyBootstrap**” listener. Here i show you another way.

Create a class and extends `javax.ws.rs.core.Application` , and add your REST service manually.

```
package com.mkyong.app;
```

```

import java.util.HashSet;
import java.util.Set;
import javax.ws.rs.core.Application;
import com.mkyong.rest.MessageRestService;

public class MessageApplication extends Application {
    private Set<Object> singletons = new HashSet<Object>();

    public MessageApplication() {
        singletons.add(new MessageRestService());
    }

    @Override
    public Set<Object> getSingletons() {
        return singletons;
    }
}

```

File : web.xml , no more listener, configure your application class like below :

```

<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>

    <context-param>

```

```
<param-name>resteasy.servlet.mapping.prefix</param-name>
<param-value>/rest</param-value>
</context-param>

<servlet>
  <servlet-name>resteasy-servlet</servlet-name>
  <servlet-class>
    org.jboss.resteasy.plugins.server.servlet.HttpServletDispatcher
  </servlet-class>
  <init-param>
    <param-name>javax.ws.rs.Application</param-name>
    <param-value>com.mkyong.app.MessageApplication</param-value>
  </init-param>
</servlet>

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

</web-app>
```

Done.

Download Source Code





Download it – [RESTEasy-Hello-World-Example.zip](#) (7 KB)

References

1. [RESTEasy Framework](#)
2. [RESTEasy installation and configuration](#)
3. [IBM : RESTful Web services: The basics](#)
4. [RESTful Web Services](#)
5. [Wiki : Representational State Transfer](#)

Tags :

[hello world](#)

[jax-rs](#)

[resteasy](#)

Share this article on

[Twitter](#)

[Facebook](#)

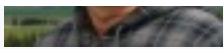
[Google+](#)

About the Author



mkyong

Founder of [Mkyong.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 make a donation to [these charities](#).

Related Posts

374k

[JAX-RS @QueryParam example](#)

16k

[RESTEasy Unable to scan WEB-INF for JAX-RS annotations ?](#)

5k

[ClassNotFoundException : org.jboss.resteasy.plugins.providers.multipart.MultipartInput](#)

23k

[RESTEasy - Could not find message body reader for type: multipart/form-data](#)

133k

[File upload example in RESTEasy](#)

45k

[RESTEasy - Could not find MessageBodyWriter for response object of type:xx of media type: application/xml](#)

48k

[XML example with RESTEasy + JAXB](#)

171k

[RESTful Java client with RESTEasy client framework](#)

96k

[RESTEasy + Spring integration example](#)

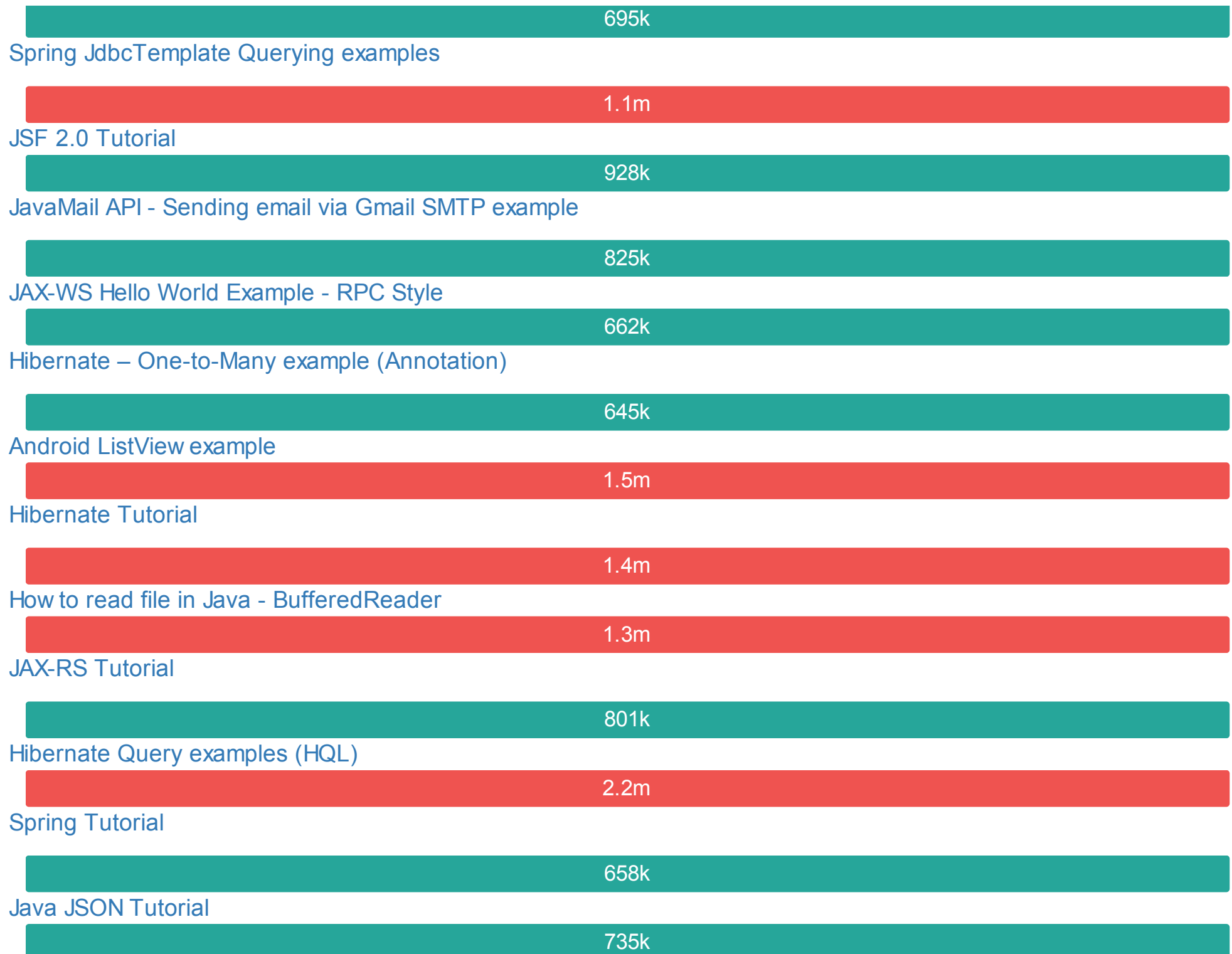
737k

[Jersey hello world example](#)



Popular Posts

	838k
log4j.properties example	
	713k
Android custom dialog example	
	1.1m
Android Tutorial	
	794k
What is Tomcat default administrator password ?	
	943k
How to convert String to Date - Java	



Spring MVC hello world example

1.1m

Android spinner (drop down list) example

1.6m

How to convert Java object to / from JSON (Jackson)

Comments

Sorry, the browser you are using is not currently supported. Disqus actively supports the following browsers:

- [Firefox](#)
- [Chrome](#)
- [Internet Explorer 11+](#)
- [Safari](#)



[Steve](#) • 3 years ago

Thanks for this. It worked fine for me.

In case anyone wants to build the war using gradle rather than maven create the following build.gradle file & to it:

```
apply plugin: 'java'
apply plugin: 'war'
```

```
repositories {
    mavenCentral()
}
```

```
dependencies {
    providedCompile 'javax.servlet:servlet-api:2.5'
```





Mkyong

53,216 likes



Like Page

Be the first of your friends to like this



Rising Posts (100k-500k pv)

+9.2k

Java 8 forEach examples

+8.2k

Java - Convert String to int

+5.7k

Java 8 Streams filter examples

+4.5k

Java - Read a file from resources folder

+4.2k

Java enum example

+4.1k

Log4j hello world example

+3.8k

Java 8 Lambda : Comparator example

+3.7k

How to skip Maven unit test

+3.4k

How to set java_home on Windows 10?

+3.3k

Spring @PropertySource example

Rising Posts (10k-99k pv)

+6.3k

Spring Boot Tutorials

+5.7k

Java 8 Tutorials

+4k

Spring Boot Hello World Example - JSP

+3.6k

Apache POI – Reading and Writing Excel file in Java

+2.6k

Java 8 – Stream Collectors groupingBy examples

+2.5k

Spring Boot Hello World Example - Thymeleaf

+2.4k

JUnit + Spring integration example

+2.3k

Spring Boot + Spring Data JPA + Oracle example

+1.9k

Spring Boot @ConfigurationProperties example

+1.9k

Java - How to override equals and hashCode

Rising Posts (<10k pv)

+2.9k

Java 8 Streams map() examples

+1.2k

Spring Boot JDBC + MySQL + HikariCP example

+1.1k

Java 8 Optional In Depth

+1k

Spring Boot + Spring Data + Elasticsearch example

+891

Spring Boot + Spring Data MongoDB example

+887

Spring Boot Ajax example

+729

Java - Convert comma-separated String to a List

+711

Spring MVC - How to get client IP address

+692

Spring Boot + Spring Data JPA + Java 8 date and time (JSR310)

+684

Spring Boot file upload example - Ajax and REST

Favorites Links

[Android Getting Started](#)

[Google App Engine – Java](#)

[Spring 2.5.x Documentation](#)

[Spring 3.2.x Documentation](#)

[Spring 4.1.x Documentation](#)

[Java EE 5 Tutorial](#)

[Java EE 6 Tutorial](#)

[Java EE 7 Tutorial](#)

[Java 6 API](#)

[Java 7 API](#)

[Java 8 API](#)

[Oracle J2SE Tutorials](#)

[JSF Home Page](#)
[JSP Home Page](#)
[Maven Central Repository](#)
[Gradle User Guide](#)
[Hibernate ORM](#)
[JAX-WS Home Page](#)
[JAX-RS Home Page \(Jersey\)](#)
[Tomcat 8 Documentation](#)

Partners & Bookmarks

[Java Code Geeks](#)
[TestNG Founder](#)
[DZone](#)

About Mkyong.com

Mkyong.com is for Java and J2EE developers, all examples are simple and easy to understand, and well tested in my development environment.

Mkyong.com is created, written by, and maintained by Yong Mook Kim, aka Mkyong. It is built on [WordPress](#), hosted by [Liquid Web](#), and the caches are served by CloudFlare CDN.

