# 构建restful风格的maven项目

## 创建maven web 项目

1. 创建一个maven web 项目。

## 在 pom.xml 中添加 RESTful 依赖

<dependency>

<groupId>com.sun.jersey</groupId>

<artifactId>jersey-server</artifactId>

<version>1.17.1</version>

</dependency>

<dependency>

<groupId>com.sun.jersey</groupId>

<artifactId>jersey-core</artifactId>

<version>1.17.1</version>

</dependency>

<dependency>

<groupId>com.sun.jersey</groupId>

<artifactId>jersey-servlet</artifactId>

<version>1.17.1</version>

</dependency>

## 在 web.xml 中配置 RESTful

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<web-app xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xmlns=*"http://java.sun.com/xml/ns/javaee"* xsi:schemaLocation=*"http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app\_2\_5.xsd"*

id=*"WebApp\_ID"* version=*"2.5"*>

<welcome-file-list>

<welcome-file>index.jsp</welcome-file>

</welcome-file-list>

<display-name>Test Maven RESTful</display-name>

<servlet>

<servlet-name>RESTful Application</servlet-name>

<servlet-class>

com.sun.jersey.spi.container.servlet.ServletContainer

</servlet-class>

<init-param>

<param-name>

jersey.config.server.provider.packages

</param-name>

<param-value>com.itree</param-value>

</init-param>

</servlet>

<servlet-mapping>

<servlet-name>RESTful Application</servlet-name>

<url-pattern>/\*</url-pattern>

</servlet-mapping>

</web-app>

## 编写 RESTful 风格的类

package service;

import javax.ws.rs.GET;

import javax.ws.rs.Path;

import javax.ws.rs.PathParam;

import javax.ws.rs.Produces;

import javax.ws.rs.core.MediaType;

@Path(value = "service")

public class TestService {

@GET

@Path(value = "/{message}")

@Produces(MediaType.APPLICATION\_XML)

public String echoService(@PathParam("message") String message) {

return message;

}

}

## 项目打包部署

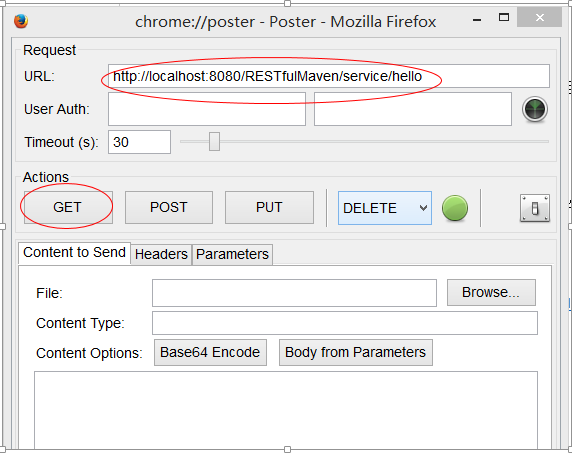
在项目上右击 run as 选择 maven install 将自动在项目下的 target 目录下生成一个 war ,j将它拷到 tomcat 下的 webapp 下。

## 启动tomcat

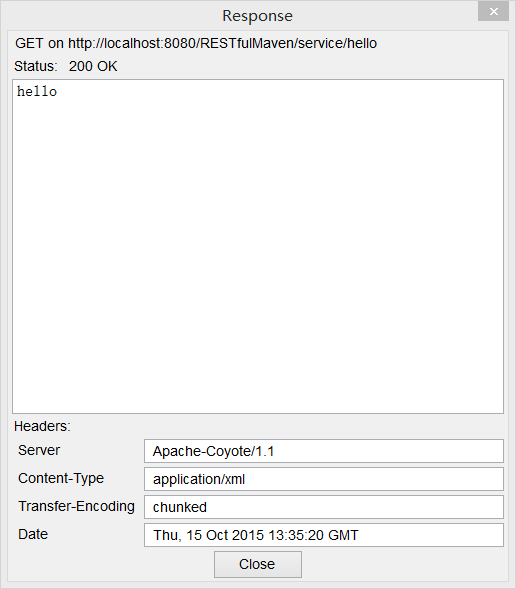
在tomcat根目录下的 bin 目录下 找到 statup.bat 文件，双击即可启动 tomcat 。

## 运行程序

1. 打开火狐浏览器的 poster ，输入接口地址：<http://localhost:8080/RESTfulMaven/service/hello>，点击 GET ，如下图所示：



1. 测试成功，结果如下：



## 附录：（网络资源）

1. **http 动词**
2. GET（SELECT）：从服务器取出资源（一项或多项）。
3. POST（CREATE）：在服务器新建一个资源。
4. PUT（UPDATE）：在服务器更新资源（客户端提供改变后的完整资源）。
5. PATCH（UPDATE）：在服务器更新资源（客户端提供改变的属性）。
6. DELETE（DELETE）：从服务器删除资源。
7. **URI 格式**

<protocol>://<service-name>/<ResourceType>/<ResourceID>

如本例中的url：<http://localhost:8080/RESTfulMaven/service/hello>。

1. **设计 URI 时要考虑的要点：**
2. 使用复数名词 - 使用复数名词定义资源。比如，我们使用 users 标识用户资源。
3. 避免使用空格 - 处理长资源名时使用下划线（\_）或者连字符（-），比如，用 authorized\_users 而不是 authorized%20users。
4. 使用小写字母 - 尽管 URI 不区分带小写，但是在 url 中使用小写字母是一种很好的做法。
5. 保持向后兼容 - 由于 Web 服务是一种公共服务，URI 一旦公开之后应该始终可用。这种情况下，要更新 URI，请使用 HTTP 状态码 - 300 重定向老的 URI 到新的 URI。
6. 使用 HTTP Verb - 始终使用 HTTP Verb，比如 GET，PUT 以及 DELETE 处理资源操作。在 URL 中使用操作名并不好。