**基于JAX-RS的RESTful风格web服务**

**Demo运行说明**

林志捷

2016-05-12

**【功能说明】**

该Demo定义了一个基于JAX-RS的RESTful风格的Web服务，演示如何通过HTTP的POST、DELETE、PUT、GET四种请求，通过RESTful风格的访问，来进行增删改查操作。

通过本Demo，可以基本了解基于JAX-RS的RESTful应用的基本实现，了解@Path、@GET、@POST、@DELETE、@PUT、@Produces、@ApplicationPath等标签的使用以及@PathParam、@QueryParam、@FormParam三种主要的参数绑定方式。

**【运行环境】**

WildFly 10.0.0Final

JDK 1.8+

Maven 3.1.1+

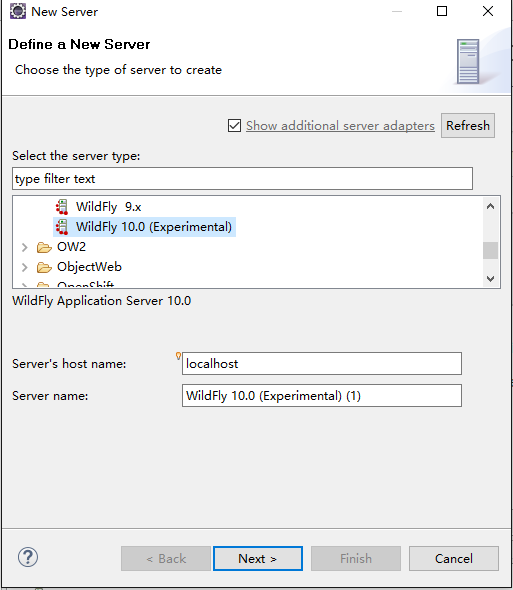
Eclipse J2EE Version with JBoss Tools

**【部署步骤】**

**1、创建Wildfly服务器**

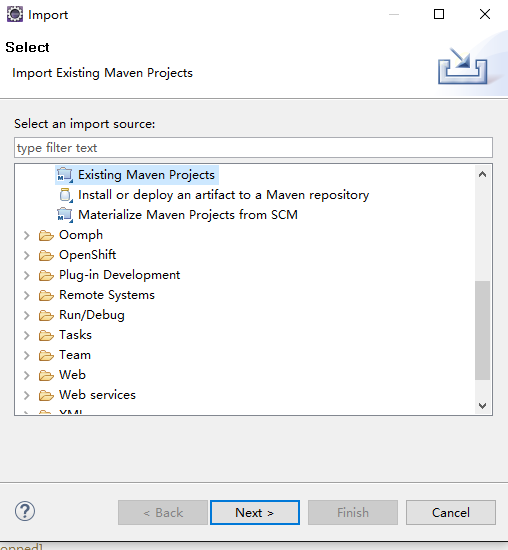
**（如果之前已有，则不用再创建了。）**

进入Eclipse，选择File->new->server，或是在Server的View中（一般使用Java EE的Perspective都会出现在界面下方），右键点击并选择new ->server，选择Wildfly10，一路next即可。



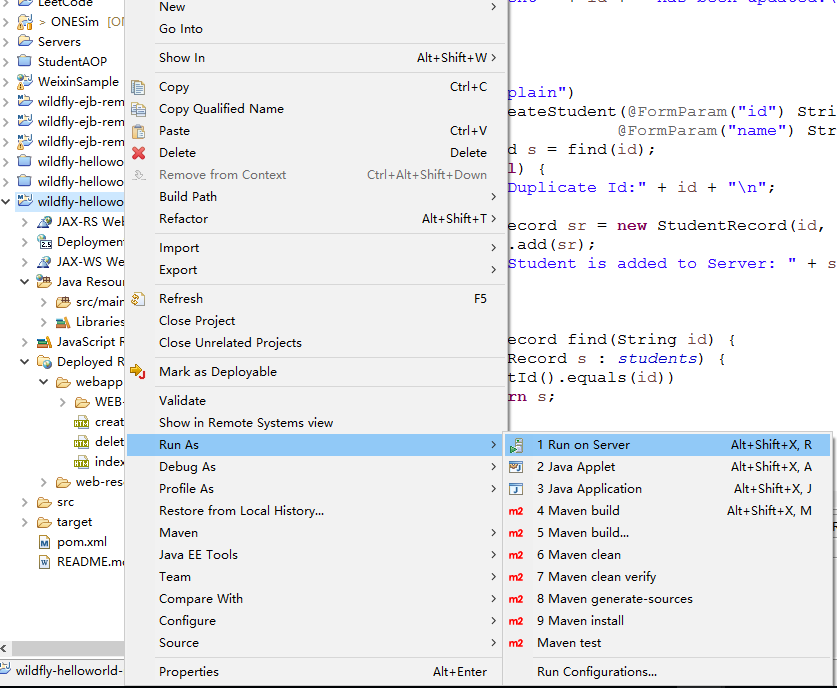
**2、导入工程**

导入时请注意，选择Existing Maven Projects，一路next、finish即可。

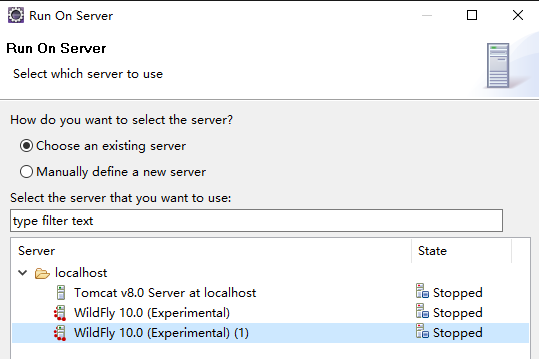


**3、编译部署**

右键点击项目 Run As -> Run on Server



选择刚才创建的服务器



**4、运行指南与程序分析**

本实验一共提供了5个REST接口，以下分别进行说明。

**4.1、查询接口（getAll、get）**

查询所对应的两个接口getAll与get，可直接通过以下URL直接通过浏览器进行访问：

【查询所有学生】

<http://localhost:8080/wildfly-helloworld-rs/rest/getAll>

【查询编号为id的学生，使用时将{id}替换为编号】

[http://localhost:8080/wildfly-helloworld-rs/rest/get/{id}](http://localhost:8080/wildfly-helloworld-rs/rest/get/%7bid%7d)

结果如图所示：



（getAll）



（get/001）

其中，对单个学生进行查询时使用的URL：……get/001，在代码实现中采用了@PathParam来获取参数，PathParam标签可以将位于“URL”路径中的参数绑定到相应的变量上，这里，“001”即是绑定的参数（源代码49行）。

**4.2、删除接口（delete）与修改接口（update）**

删除与修改所对应的两个REST接口，由于分别需要使用HTTP DELETE和HTTP PUT来进行请求，而在浏览器中直接创建这两种请求较为困难，故使用curl工具来模拟这两种请求，windows下可以使用的curl工具的可执行文件已经随本教程附带，各位通过命令行进入curl所在的目录下，即可直接使用curl命令（也可以通过配置环境变量的方式，让其全局可见），而对于使用类unix系统（如各个linux发行版，或Mac OS X）的同学，一般系统中已经自带了curl命令，直接使用即可。

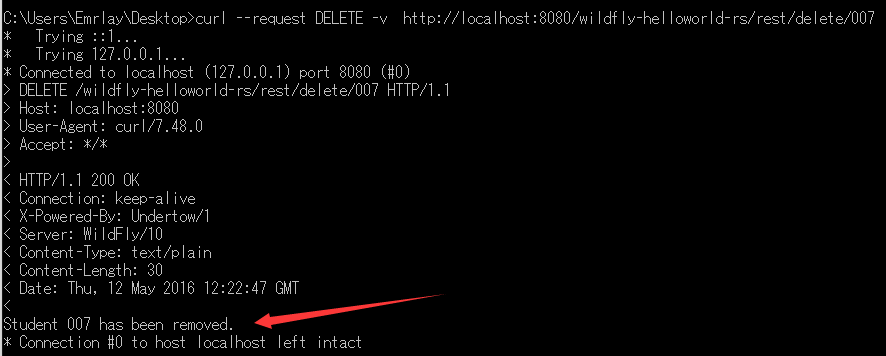
【删除编号为id的学生，使用时将{id}替换为编号】

[http://localhost:8080/wildfly-helloworld-rs/rest/delete/{id}](http://localhost:8080/wildfly-helloworld-rs/rest/delete/%7bid%7d)

【curl命令】

curl --request DELETE –v http://localhost:8080/wildfly-helloworld-rs/rest/delete/{id}

使用curl命令时，可以带上-v参数(verbose)，这样可以查看到request和response时的请求头，如图所示。



很显然，这个Demo中，我们依然使用了@PathParam来绑定参数。

【修改为id的学生姓名为name，使用时将{id}、{name}替换为编号、姓名】

[http://localhost:8080/wildfly-helloworld-rs/rest/update?id={id}&name={name}](http://localhost:8080/wildfly-helloworld-rs/rest/update?id=%7bid%7d&name=%7bname%7d)

【curl命令】

curl --request PUT -v [http://localhost:8080/wildfly-helloworld-rs/rest/update?id=004^&name=lasd](http://localhost:8080/wildfly-helloworld-rs/rest/update?id=004%5e&name=lasd)

需要格外注意，在windows使用时，这里的“&”符号之前需要带有转义符“^”，因为“&”是命令行下的特殊字符，用于拼接两个先后执行的命令（类unix使用的转义符是“\”）。



不同于前面的查询和删除，在修改的例子中，我们使用了@QueryParam标签来绑定参数，@QueryParam对应的是URL上，以“?”跟随的明文参数串，不同参数之间用“&”隔开，可以看到，这里我们有“id”和“name”两个参数（源代码85~86行）。

**4.3、增加接口（create）**

最后，增加学生的REST接口由于使用了HTTP POST请求，Demo中利用一个简单的html页面，通过提交表单来实现HTTP POST请求，在使用时，可以直接在浏览器中访问以下页面。

【增加学生】

<http://localhost:8080/wildfly-helloworld-rs/create.html>

访问后，将会返回html页面





在页面中，填入所需增加的学生id和姓名，点击submit，即将增加的学生信息以POST表单请求的形式提交到<http://localhost:8080/wildfly-helloworld-rs/rest/create>这个REST接口上，在该接口的实现中，我们使用了@FormParam来绑定参数，该标签可以绑定通过表单提交的参数（源代码71行）。

**4.4、代码中其他标签的说明**

@ApplicationPath标签指定了所有@Path标记的资源的“基位置”，它必须被标记在Application的子类上，在本demo中，它被标记在JAXActivator类上，该类没有实现，仅继承了Application类，是用来为@ApplicationPath提供标记的位置。

@Path标签定义了服务（资源）绑定的位置

@Produces标签定义了提供服务的MIME类型，在本demo中，为了简洁，均使用了文本（text/plain）类型

@GET、@POST、@DELETE、@PUT标签分别表示被该标签标注的资源响应的HTTP请求类型。