REST培训教程

# Jar准备

asm-3.3.1.jar

grizzly-http-all-2.2.1.jar

jersey-bundle-1.12.jar

jsr311-api-1.1.1.jar

javax.servlet-3.1.jar  
org.restlet.jar

上面6个jar文件

# 环境搭建

JDK版本必须是1.6版本build 1.6.0\_24-b07

培训的tomcat版本：apache-tomcat-6.0.33

将上述jar文件发布到项目的WEB-INF\lib目录下

在web.xml中添加

<servlet>

<display-name>JAX-RS REST Servlet</display-name>

<servlet-name>JAX-RS REST 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.ailk.toptea.sysmgr.upload.rest</param-value>

</init-param>

<load-on-startup>4</load-on-startup>

</servlet>

<servlet-mapping>

<servlet-name>JAX-RS REST Servlet</servlet-name>

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

</servlet-mapping>

注意：

com.sun.jersey.config.property.packages是我们开发rest程序的包路径

JAX-RS REST Servlet名字可以根据自己的设计起

# 实例

## 创建工程

我们在ecplise中创建一个rest.sample工程，注意工程使用的jdk一定要是jdk1.6的版本，创建用户库名称使用rest并将上述依赖的jar文件导入库中。

创建包com.sample.rest

创建类SampleRest

编写代码

package com.sample.rest;

import javax.ws.rs.GET;

import javax.ws.rs.Path;

@Path("sample")

public class SampleRest {

}

@Path就是HTTP地址

下面介绍具体的例子

## PATH的用户

测试方法：直接地址栏键入地址

### 无path的方法

@GET

public String getString(){

return "rest success";

}

在IE地址栏中键入<http://127.0.0.1:10000/sample1/rest/sample>

我们可以看到

rest success

### 有path的方法

@GET

@Path(“path1”)

public String getString1(){

return "rest success path";

}

在IE地址栏中键入 http://127.0.0.1:10000/sample1/rest/sample/path1

我们可以看到

rest success path

## 几种请求方式

测试方法教程中提供了restform.html文件可以测试（无参数的测试可以通过改文件测试）

### GET方式

在上介绍path的时候已经说明，就不再具体说明了

### POST方式

@POST

**public** String getStringByPost(){

**return** "rest success post";

}

restform.html中输入<http://127.0.0.1:10000/sample1/rest/sample/>

结果返回

rest success post

### DELETE方式

暂时没有使用

### PUT方式

暂时没有使用

## 参数传递方式

测试方式我们利用jersey中的http请求类模仿url请求

### @QueryParam

URL后面跟参数方式获取

例子

@GET

@Path("param1")

**public** String getStringByParam1(@QueryParam("name") String name){

**return** name;

}

测试方法

**public** **static** **void** testGetStringByQueryParam (){

Client client = Client.*create*();

WebResource webResource2 = client.resource("http://127.0.0.1:10000/sample1/rest/sample/param1");

MultivaluedMap queryParams = **new** MultivaluedMapImpl();

queryParams.add("name", "test name");

String a = webResource2.queryParams(queryParams).get(String.**class**);

System.*out*.print(a);

}

### @FormParam

表单方式

例子

@GET

@Path("param2")

**public** String getStringByParam2(@FormParam("name") String name){

**return** name;

}

测试方法

**public** **static** **void** testGetStringByFormParam (){

Client client = Client.*create*();

WebResource webResource2 = client.resource("http://127.0.0.1:10000/sample1/rest/sample/param2");

MultivaluedMap queryParams = **new** MultivaluedMapImpl();

queryParams.add("name", "form name");

String a = webResource2.post(String.**class**,queryParams);

System.*out*.print(a);

}

### @PathParam

路径方式

例子

1. @GET
2. @Path("/book/{isbn}")
3. **public** String getBook(@PathParam("isbn") String id) {
4. // search my database and get a string representation and return it
5. }

测试方法

**public** **static** **void** testGetStringByPathParam (){

Client client = Client.*create*();

WebResource webResource2 = client.resource("http://127.0.0.1:10000/sample1/rest/sample/param3/dsdsdsldskl");

String a = webResource2.get(String.**class**);

System.*out*.print(a);

}

### @MatrixParam

路径方式

例子

@GET

@Path("/param4")

**public** String getStringByParam4(@MatrixParam("name") String name, @MatrixParam("author") String author) {

**return** name+author;

}

测试

**public** **static** **void** testGetStringByMatrixParam (){

Client client = Client.*create*();

WebResource webResource2 = client.resource("http://127.0.0.1:10000/sample1/rest/sample/param4;name=EJB;author=BillBurke");

String a = webResource2.get(String.**class**);

System.*out*.print(a);

}

### @Context 动态方式获取

@GET

@Path("/param5")

**public** String getStringByParam5(@Context UriInfo info) {

String from = info.getQueryParameters().getFirst("from");

String to = info.getQueryParameters().getFirst("to");

List<String> orderBy = info.getQueryParameters().get("orderBy");

**return** "form="+from+";to="+to+";orderBy-length="+orderBy.size();

}

测试

**public** **static** **void** testGetStringByContext (){

Client client = Client.*create*();

WebResource webResource2 = client.resource("http://127.0.0.1:10000/sample1/rest/sample/param5?from=100&to=200&orderBy=age&orderBy=name");

String a = webResource2.get(String.**class**);

System.*out*.print(a);

}

### @DefaultValue初始值

@GET

@Path("/param6")

**public** String getStringByParam6( @DefaultValue("sddsds") @QueryParam("name") String name) {

**return** name;

}

测试

**public** **static** **void** testGetStringByDefaultValue (){

Client client = Client.*create*();

WebResource webResource2 = client.resource("http://127.0.0.1:10000/sample1/rest/sample/param6");

String a = webResource2.get(String.**class**);

System.*out*.println(a);

MultivaluedMap queryParams = **new** MultivaluedMapImpl();

queryParams.add("name", "test name");

a = webResource2.queryParams(queryParams).get(String.**class**);

System.*out*.println(a);

}

### @HeaderParam头参数

@GET

@Path("/param7")

**public** String getStringByParam7( @Context HttpHeaders headers) {

String userAgent = headers.getRequestHeader("user-agent").get(0);

**return** userAgent;

}

测试

**public** **static** **void** testGetStringByHeaderParam(){

Client client = Client.*create*();

WebResource webResource2 = client.resource("http://127.0.0.1:10000/sample1/rest/sample/param7");

String a = webResource2.get(String.**class**);

System.*out*.print(a);

}

### 自定义类参数

@POST

@Path("/param8")

**public** String getStringByParam8(User user) {

**return** user.getId()+""+user.getName();

}

测试

**public** **static** **void** testGetStringByObject(){

Client client = Client.*create*();

WebResource webResource2 = client.resource("http://127.0.0.1:10000/sample1/rest/sample/param8");

User u = **new** User();

u.setId("panhua");

u.setName("潘华");

String a = webResource2.post(String.**class**,u);

System.*out*.print(a);

}

## 返回方式

### String

上述方式全部为String返回就不再举例

### JSON

@GET

@Path("/param9")

@Produces("application/json")

**public** User getStringByParam9() {

User u = **new** User();

u.setId("panhua");

u.setName("潘华");

**return** u;

}

测试

**public** **static** **void** testGetJson(){

Client client = Client.*create*();

WebResource webResource2 = client.resource("http://127.0.0.1:10000/sample1/rest/sample/param9");

String a = webResource2.get(String.**class**);

System.*out*.print(a);

}

### XML

@GET

@Path("/param10")

@Produces("application/xml")

**public** User getStringByParam10() {

User u = **new** User();

u.setId("panhua");

u.setName("潘华");

**return** u;

}

测试

**public** **static** **void** testGetXML(){

Client client = Client.*create*();

WebResource webResource2 = client.resource("http://127.0.0.1:10000/sample1/rest/sample/param10");

String a = webResource2.get(String.**class**);

System.*out*.print(a);

}

### 对象

可以用json和XML直接放回对象

测试方法

**public** **static** **void** testGetObject(){

Client client = Client.*create*();

WebResource webResource2 = client.resource("http://127.0.0.1:10000/sample1/rest/sample/param9");

User a = webResource2.get(User.**class**);

System.*out*.print(a.getName());

}

## Spring的调用

@Context ServletContext context;

@GET

@Path("/param11")

@Produces("application/json")

**public** User getStringByParam11() {

WebApplicationContext ctx = WebApplicationContextUtils.*getRequiredWebApplicationContext*(context);

TestSpring ur = (TestSpring)ctx.getBean("testSpring");

**return** ur.getUser();

}

测试

**public** **static** **void** testGetObjectBySpring(){

Client client = Client.*create*();

WebResource webResource2 = client.resource("http://127.0.0.1:10000/sample1/rest/sample/param11");

User a = webResource2.get(User.**class**);

System.*out*.print(a.getName());

}