《分布式计算》

RESTful Webservice 编程

学 院 名 称 ： 数据科学与计算机学院

成　　员 ： 陈伟宸

时　　间 ： 2016 年 9 月 17 日

1. 输入是 URL，输出是 XML

2. 需要申明 HTTP 协议方法和 URI；

这些申明为：

@GET

@Path("/customers/{id}/")

协议的方法对应@GET，协议的地址对应@Path

3.

```
D:\CURL\winssl>curl.exe http://localhost:9000/myservice/customers/123 -v
*   Trying 127.0.0.1...
* TCP_NODELAY set
* Connected to localhost (127.0.0.1) port 9000 (#0)
> GET /myservice/customers/123 HTTP/1.1
> Host: localhost:9000
> User-Agent: curl/7.50.2
> Accept: */*
>
< HTTP/1.1 200 OK
< Date: Sun, 25 Sep 2016 09:33:06 GMT
< Content-Type: text/xml
< Content-Length: 105
< Server: Jetty(9.2.15.v20160210)
<
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><Customer><id>123</id><name>John</name></Customer>* Curl_http_done: called premature == 0
* Connection #0 to host localhost left intact
```

4. GET 方法的查询字符串在 URL 中的，而 POST 方法的查询字符串实在 HTTP 消息主体中的

5. 会，当数据量增大时，由于 map 会维护数据，增加新用户的时间会边长

6. IoC 将需要的资源从主体中分离，在需要时向主体中注入，使得主体不需要考虑资源的生成与维护，降低了耦合程度

7. 代码见 IOC_test.zip，程序运行结果如下：

```
Problems  Javadoc  Declaration  Search  Console
<terminated> QuizProgram (1) [Java Application] C:\Program Files\Java\jdk1.8.0_91\bin\javaw.exe (2016年9月25日 下午10:24:39)
九月 25, 2016 10:24:39 下午 org.springframework.context.support.ClassPathXmlApplicationContext prepareRefresh
信息: Refreshing org.springframework.context.support.ClassPathXmlApplicationContext@1f17ae12: startup date [Sun Sep 25 22:24:39 CST 2016];
九月 25, 2016 10:24:39 下午 org.springframework.beans.factory.xml.XmlBeanDefinitionReader loadBeanDefinitions
信息: Loading XML bean definitions from class path resource [beans.xml]
九月 25, 2016 10:24:39 下午 org.springframework.beans.factory.support.DefaultListableBeanFactory preInstantiateSingletons
信息: Pre-instantiating singletons in org.springframework.beans.factory.support.DefaultListableBeanFactory@43a25848: defining beans [spring
Are you new to Struts?
```

```
Problems  Javadoc  Declaration  Search  Console
<terminated> QuizProgram (1) [Java Application] C:\Program Files\Java\jdk1.8.0_91\bin\javaw.exe (2016年9月25日 下午10:26:37)
九月 25, 2016 10:26:37 下午 org.springframework.context.support.ClassPathXmlApplicationContext prepareRefresh
信息: Refreshing org.springframework.context.support.ClassPathXmlApplicationContext@1f17ae12: startup date [Sun Sep 25 22:26:37 CST 2016];
九月 25, 2016 10:26:37 下午 org.springframework.beans.factory.xml.XmlBeanDefinitionReader loadBeanDefinitions
信息: Loading XML bean definitions from class path resource [beans.xml]
九月 25, 2016 10:26:38 下午 org.springframework.beans.factory.support.DefaultListableBeanFactory preInstantiateSingletons
信息: Pre-instantiating singletons in org.springframework.beans.factory.support.DefaultListableBeanFactory@43a25848: defining beans [spring
Are you new to Spring?
```

8. 在标签

<bean id="customerBean" class="demo.jaxrs.server.CustomerService" />

中，class 属性申明了根资源

9. Hello 类代码：

```
package demo.jaxrs.server;

import javax.ws.rs.GET;
import javax.ws.rs.Path;
import javax.ws.rs.PathParam;
import javax.ws.rs.Produces;

@Path("/hello/{username}")
public class Hello {

	@GET
	@Produces("text/xml")
	public String sayHello(@PathParam("username") String userName)
{
		return "hello " + userName;
	}
}
```

beans.xml 如下：

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:jaxrs="http://cxf.apache.org/jaxrs"
  xsi:schemaLocation="
    http://www.springframework.org/schema/beans
    http://www.springframework.org/schema/beans/spring-beans.xsd
    http://cxf.apache.org/jaxrs
    http://cxf.apache.org/schemas/jaxrs.xsd">

  <!-- do not use import statements if CXFServlet init parameters
link to this beans.xml -->

  <import resource="classpath:META-INF/cxf/cxf.xml" />
  <import resource="classpath:META-INF/cxf/cxf-servlet.xml" />
```

```
  <jaxrs:server id="customerService" address="/service1">
    <jaxrs:serviceBeans>
      <ref bean="customerBean" />
    </jaxrs:serviceBeans>
    <!-- 引入json 输入/输出支持 -->
    <jaxrs:providers>
      <bean
class="org.codehaus.jackson.jaxrs.JacksonJaxbJsonProvider"/>
      <bean
class="org.apache.cxf.jaxrs.provider.JAXBElementProvider"/>
    </jaxrs:providers>
  </jaxrs:server>

  <jaxrs:server id="helloService" address="/service2">
    <jaxrs:serviceBeans>
      <ref bean="helloBean" />
    </jaxrs:serviceBeans>
    <!-- 引入json 输入/输出支持 -->
    <jaxrs:providers>
      <bean
class="org.codehaus.jackson.jaxrs.JacksonJaxbJsonProvider"/>
      <bean
class="org.apache.cxf.jaxrs.provider.JAXBElementProvider"/>
    </jaxrs:providers>
  </jaxrs:server>

  <bean id="helloBean" class="demo.jaxrs.server.Hello" />
  <bean id="customerBean" class="demo.jaxrs.server.CustomerService"
/>
</beans>
```

curl 测试结果如下：

```
D:\CURL\winssl>curl.exe http://localhost:8080/service1/customerservice/customers/123
{"id":123,"name":"John"}
D:\CURL\winssl>curl.exe http://localhost:8080/service2/hello/Monkey
hello Monkey
```

10. Web Application Description Language

11. 由 Accept 域决定

12. Java API for RESTful Web Services

13. @Produces 用于指定输出格式，@Consumes 用于指定输入媒体的类型

小结

1. 学习了 RESTful Webservice 基本编程
2. 学习了 CXF 与 Spring 的集成
3. 理解了 IOC 和 DI 的思想
4. 学习了 Spring 的基本用法
5. 学习了 JAX-RS application 编程