# Jersey介绍

Jersey RESTful 框架是开源的RESTful框架, 实现了JAX-RS (JSR 311 & JSR 339) 规范。它扩展了JAX-RS 参考实现， 提供了更多的特性和工具， 可以进一步地简化 RESTful service 和 client 开发。尽管相对年轻，它已经是一个产品级的 RESTful service 和 client 框架。与Struts类似，它同样可以和hibernate,spring框架整合。

Jersey 是一个非常不错的框架。对于请求式服务，对于GET,DELETE请求，你甚至只需要给出一个URI即可完成操作。

# Jersey入门

## 导包

导入jersey下的所有包

|  |
| --- |
|  |

## Servelt

|  |
| --- |
| @Path(**"/user"**) **public class** UserServlet {  ObjectMapper **mapper** = **new** ObjectMapper();   @GET  @Path(**"/list"**)  @Produces(MediaType.***APPLICATION\_JSON***)  **public** String list() **throws** Exception {  List<User> list = **new** ArrayList<User>();  list.add(**new** User(1,**"gyf"**,**"123"**));  list.add(**new** User(2,**"cdd"**,**"123"**));  list.add(**new** User(3,**"lkq"**,**"123"**));  **return mapper**.writeValueAsString(list);  }   @GET  *//@Path("/getUserById/{id}")* @Path(**"/{id}"**)  @Produces(MediaType.***APPLICATION\_JSON***)  **public** String userById(@PathParam(**"id"**) String id) **throws** Exception {  User user = **new** User(Integer.*parseInt*(id),**"cdd"**,**"123"**);  **return mapper**.writeValueAsString(user);  }   @POST  @Path(**"/add/{json}"**)  @Consumes(MediaType.***APPLICATION\_JSON***)*//请求数据是json格式* @Produces(MediaType.***APPLICATION\_JSON***)  **public** String add(String json) **throws** Exception {   System.***out***.println(json);*//{"username":"gyf","password":"123"}   //json转对象* User user = **mapper**.readValue(json,User.**class**);  System.***out***.println(user);   **return "success"**;  }  } |

## Web.xml配置servlet

|  |
| --- |
| <**servlet**>  <**servlet-name**>jersey-restful</**servlet-name**>  <**servlet-class**>org.glassfish.jersey.servlet.ServletContainer</**servlet-class**>  <**init-param**>  <**param-name**>jersey.config.server.provider.packages</**param-name**>  <**param-value**>com.gyf.web.servlet</**param-value**>  </**init-param**> </**servlet**>  <**servlet-mapping**>  <**servlet-name**>jersey-restful</**servlet-name**>  <**url-pattern**>/rest/\*</**url-pattern**> </**servlet-mapping**> |

## 访问规则

http://localhost:8080/rest/user/list

http://localhost:8080/rest/user/3

http://localhost:8080/rest/user/add/json

## 掌握Test RESTful Web Service的使用

|  |
| --- |
|  |
|  |

## application.wadl

查看网站的所有资源

http://localhost:8080/rest/application.wadl

