JEESZ REST 服务接口文档

项目代号：JEESZ-2014-09-15

2014-10-02

目　　录

1、 引言

1.1、 REST 介绍

REpresentational State Transfer (REST) 是一种架构原则，其中将 web 服务视为资源，可以由其 URL 唯一标识。RESTful Web 服务的关键特点是明确使用 HTTP 方法来表示不同的操作的调用。
REST 的基本设计原则对典型 CRUD 操作使用 HTTP 协议方法：
POST - 创建资源
GET - 检索资源
PUT – 更新资源
DELETE - 删除资源
REST 服务的主要优势在于：
它们是跨平台 (Java、.net、PHP 等）高度可重用的，因为它们都依赖基本 HTTP 协议。
它们使用基本的 XML，而不是复杂的 SOAP XML，使用非常方便。

基于 REST 的 web 服务日益成为后端企业服务集成的首选方法。与基于 SOAP 的 web 服务相比，它的编程模型简单，而本机 XML（而不是 SOAP ）的使用减少了序列化和反序列化过程的复杂性，并且不再需要其他作用相同的第三方库。

1.2、 编写目的

编写本文的目的是为了将系统功能进行模块化、服务化，将用户的操作以服务的方式提供。系统与系统之间遵循服务规范，将系统与系统之间的交互转为定制化服务交互，以实现系统与系统之间的集成。

1.3、 编写原则

可寻址性（Addressability）REST 中的所有东西都基于*资源* 的概念。资源与 OOP 中的对象或其他名词不同，它是一种抽象，必须可以通过 URI 寻址或访问。

接口一致性（Interface uniformity）与 SOAP 或其他标准不同，REST 要求用来操纵资源的方法或动词不是任意的。这意味着 RESTful 服务的开发人员只能使用 HTTP 支持的方法，比如 GET、PUT、POST、DELETE 等等。因此不需要使用 WSDL 等服务描述语言。

无状态（Statelessness）为了增强可伸缩性，服务器端不存储客户机的状态信息。这使服务器不与特定的客户机相绑定，负载平衡变得简单多了。这还让服务器更容易监视、更可靠。

具象（Representational）客户机总是与资源的某种具象交互，绝不会直接与资源本身交互。同一资源还可以有多个具象。理论上说，持有资源的具象的任何客户机应该有操纵底层资源的足够信息。

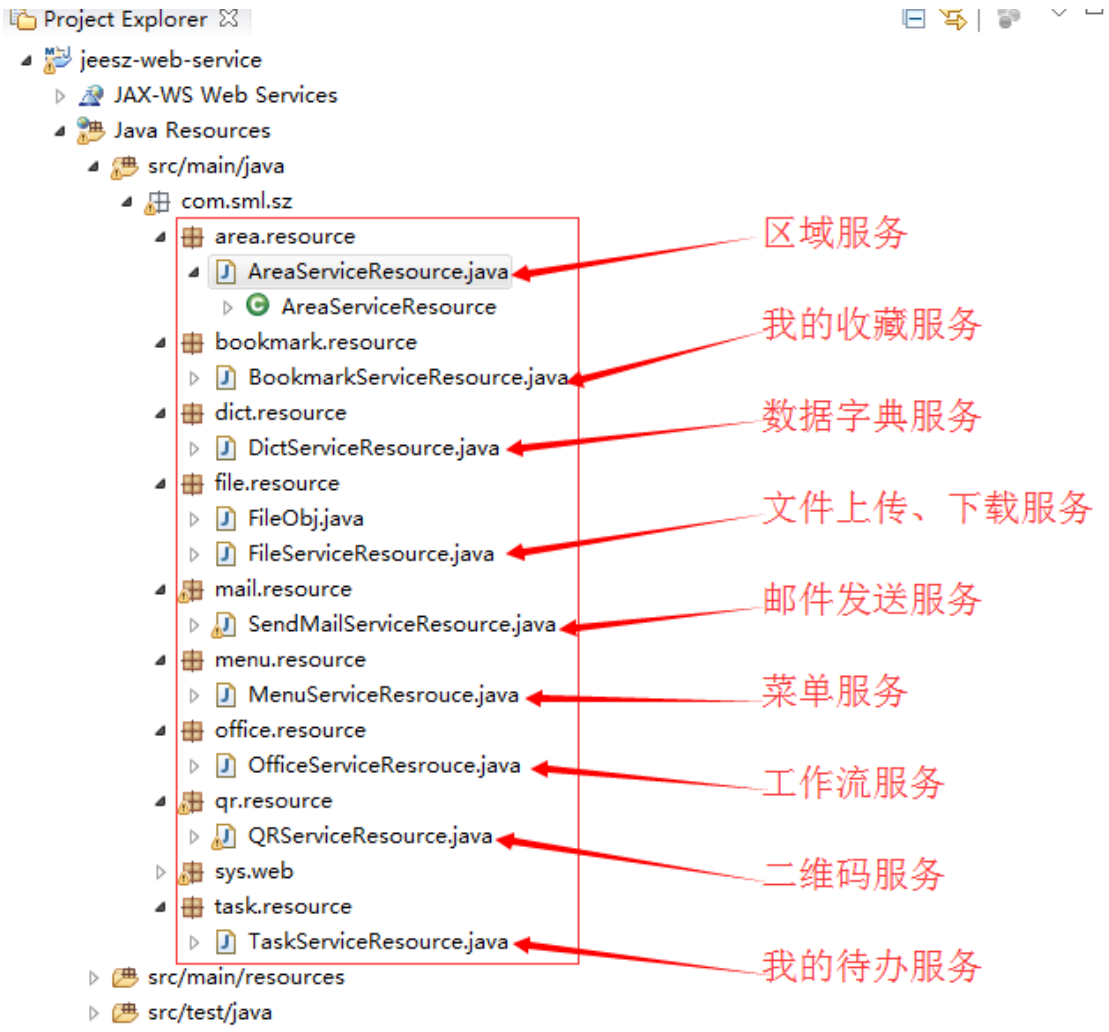
连通性（Connectedness）任何基于 REST 的系统都应该预见到客户机需要访问相关的资源，应该在返回的资源具象中包含这些资源。例如，可以以超链接的形式包含特定 RESTful 服务的操作序列中的相关步骤，让客户机可以根据需要访问它们。

1.4、 变更历史

序号	变更人员	变更时间	变更内容
1	Jeesz	2014-10-02	我的收藏服务 3.1、3.2、4.3、5.4(仅仅以我的收藏部分服务为实例)，用户根据自己的业务编写相关的服务文档即可

2、 服务使用说明

2.1、 当前系统已经提供的服务



2.2、 GET 方式调用服务

说明：

1. 请求方式包括：GET (这里以 area 服务为实例，GET 对应每一个服务 Resource 中的@RequestMapping(value = "treeData", method = RequestMethod.GET))
2. 请求 URL：rest 服务请求地址,对应 XXXServiceResource.java 的 mapping 配置中的 value @RequestMapping(value = "treeData", method = RequestMethod.GET))
3. 其中 GET 请求只包含了请求方式和请求的 URL，返回的结果以 json 格式返回给客户端

2.3、 POST、DELETE、UPDATE 方式调用服务

说明：

1. 请求方式选择 POST、DELETE、UPDATE(这里以保存收藏功能为例（PUT 请求），对应每一个服 Resource 中的@RequestMapping(value = "save", method = RequestMethod.PUT))
2. Json 参数： 其中 POST、DELETE、UPDATE 可能传递参数通过 json，也可能通过路径直接拼接参数，这边以传递 json 到服务端为实例，对应服务端代码：
   public JSONObject save(@RequestBody JSONObject obj, BookmarkTag bookmarkTag) {
3. 请求 URL：rest 服务请求地址,对应 XXXServiceResource.java 的 mapping 配置中的 value @RequestMapping(value = " save", method = RequestMethod.PUT))

4. 返回的结果以 json 格式返回给客户端

3、 服务列表（这边以我的收藏服务为例）

3.1、 添加标签分类

请求方式	PUT
服务 URL	http://localhost:8080/jeesz-service-web/rest/bookmark/save
路径参数?*描述	无
参数类型(Type)	application/json
参数描述	{“name”:”jeesz”} 备注： name：标签名称
返回值类型(Type)	application/json
描述	添加成功： {result:”添加成功”, ”name”:”jeesz”,” bookmarkTagId”:”123456789”} 提示：返回结果由用户根据自己的业务去扩充

3.2、 删除分类标签

请求方式	DELETE
服务 URL	/bookmark/delete? bookmarkTagId =xxxx

路径参数?*描述	bookmarkTagId：分类标签 id
参数类型(Type)	String
返回值类型(Type)	application/json
描述	添加成功：{“result”:”删除成功”} 提示：返回结果由用户根据自己的业务去扩充

3.3、 更新分类标签

请求方式	POST
服务 URL	/bookmark/update
路径参数?*描述	无
参数类型(Type)	application/json
参数描述	{“id”:”标签 id”,“name”:”标签 name”} 备注： id：标签 id name：标签名称
返回值类型(Type)	application/json
描述	添加成功：{“result”:”更新成功”} 提示：返回结果由用户根据自己的业务去扩充

3.4、 获取分类标签列表

请求方式	GET
服务 URL	/bookmark/list?pageNo=1&pageSize=3
路径参数?*描述	pageNo：当前页 pageSize：每页显示多少条
参数类型(Type)	无
参数描述	通过 request 获取参数（根据自己的业务，可以通过其他方式获取，如路径参数?*、路径拼接参数等）
返回值类型(Type)	application/json
描述	{"pageNo":1,"pageSize":1,"count":42, "list":[{"id":"de0163b614b34c0ba99590e8e63b9e3e", "isNewRecord":false,"createDate":"2016-02-28 21:40:36","updateDate":"2016-02-28 21:40:36","bookmarktagname":"jeesz"}]