



链滴

浅谈 Restful

作者: [sirwsl](#)

原文链接: <https://ld246.com/article/1590396823438>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

在开始写之前，个人认为在编程中，对于java来说是一切事物皆对象，在Restful架构风格中核心思想资源，可以含糊的说为“一切链接皆资源”，接下来说说restful

一、restful是什么？

Rest与Restful 简单理解为同一回事，基于REST构建的API就是Restful风格，REST 就是一种设计 API 的模式。最常用的数据格式是 JSON。Rrestful是什么一种分布式系统应用层解决方案、能够实现clien与server的解耦。Restful是一种面向资源的一种如那件架构风格，他可以降低开发的复杂性，提高系的可伸缩性。

restful面向资源接口设计以及操作抽象简化了开发者的不良设计，同时极大限度利用资源。

二、设计规范：

协议：使用https

域名：尽量将api部署在专有域名下

版本：放在url或者是header中

路径：每一个地址都是表示一种资源，不能有动词，只有名次，一般来说API中名词为复数

过滤信息： ? Limit=10 ? Offset10 ? Page=1 ? Sortby=name (排序)

三、请求方式：

面向资源的接口设计，抽象操作作为基础的CURD

Method	CRUD	Detail
GET 项或多项)	READ	从服务器取出资源 (
POST 资源	CREATE	在服务器新建一
PUT (客户端提供改变后的完整资源)	UPDATE	在服务器更新资
PATCH 源 (客户端提供改变的属性)	UPDATE	在服务器更新
DELETE 源	DELETE	从服务器删除
HEAD		获取资源的元数据
OPTIONS 哪些属性是客户端可以改变的		获取信息，关于资源

四、Spring中的应用：

```
//@RequestMapping(value =("/{id}", method = RequestMethod.GET)
@GetMapping(value="/{id}")
public String toUpdate(@PathVariable("id") Integer id, Student stu) {}
```

Method	or1	or2
GET stMapping(method = RequestMethod.GET)	@GetMapping()	@Requ
POST stMapping(method = RequestMethod.POST)	@PostMapping()	@Requ
PUT Mapping(method = RequestMethod.PUT)	@PutMapping	@Reques
PATCH uestMapping(method = RequestMethod.PATCH)	@DeleteMapping	@Re
DELETE questMapping(method = RequestMethod.DELETE)	@DeleteMapping	@R
HEAD hod = RequestMethod.HEAD)		@RequestMapping(me
OPTIONS method = RequestMethod.OPTIONS)		@RequestMapping

五、返回结果：

方式	路径	说明
Get	/demos	返回资源对象列表
Get	/demos/1	返回单个资源对象
Post	/demos	返回新生成资源对象
Put 象	/demos/1	返回完整的资源
Patch	/demos/1	返回修改的属性
Delete 体	/demos/1	返回空的相

六、AJAX请求

```
$.ajax( {
    type : "DELETE",
    url : "/test/demo/" + id,
    dataType : "json",
    success : function(data) {
        alert(data);
        location.href = "/test/demo/1";
    }
});
```

PS:

1. SOAP安全性高于REST，但是REST效率与易用性高于SCOP
2. Post是不幂等方法，get是幂等方法。幂等性：发送100次与发送一次的引起的边界效应一致也就

结果一致

3. Get、Head、Options安全方法，只是获取数据，不具有边界效应

♡heart **给梦想一点时间，每天进步一点点**♡heart