RESTful API 设计最佳实践

2016-04-17 REST (/categories/#REST) REST (/tags/#REST) API (/tags/#API)

Web API 近几年变得越来越火，而简洁的 API 设计在多后端系统交互应用中也变得尤为重要。通常，会使用 RESTful API 来作为我们的 Web API 。本文介绍了几种简洁 RESTful API 设计的最佳实践。

使用的名词而不是动词

使用名词来定义接口

HTTP 请求方法在RESTful Web 服务中的典型应用

资源	GET	PUT	POST	DELETE
一组资源的**URI**，比如 `http://www.waylau.com/resources/`	**列出** URI，以及该资源组中每个资源的详细信息（后者可选）。	使用给定的一组资源**替换**当前整组资源。	在本组资源中**创建/追加**一个新的资源。 该操作往往返回新资源的URL。	**删除** 整组资源。
单个资源的**URI**，比如 `http://www.waylau.com/resources/142`	**获取** 指定的资源的详细信息，格式可以自选一个合适的网络媒体类型（比如：XML、JSON等）	**替换/创建** 指定的资源。并将其追加到相应的资源组中。	把指定的资源当做一个资源组，并在其下**创建/追加**一个新的元素，使其隶属于当前资源。	**删除** 指定的元素。

不应该使用动词：

```
/getAllResources
/createNewResource
/deleteAllResources
```

GET 方法和查询参数不能改变资源状态

如果要改变资源的状态，使用 PUT, POST 和 DELETE。下面是错误的用 GET 方法来修改 user 的状态：

```
GET /users/711?activate 或
GET /users/711/activate
```

使用名词复数

不要混淆名词的单复数。保持简单，只用复数名词定义所有资源。

```
/cars 代替 /car
/users 代替 /user
/products 代替 /product
/settings 代替 /setting
```

使用子资源来表达资源间的关系

```
GET /cars/711/drivers/  返回 711 号 car 的所有 driver 列表
GET /cars/711/drivers/4 返回 711 号 car 的 4 号 driver
```

使用 HTTP header 来序列化格式

客户端、服务端都需要知道互相之间的通讯格式。这些格式可以定义在 HTTP header 里面：

- Content-Type：定义了请求格式
- Accept：定义了接收相应的格式列表

使用 HATEOAS 约束

HATEOAS（Hypermedia as the engine of application state）是 REST 架构风格中最复杂的约束，也是构建成熟 REST 服务的核心。它的重要性在于打破了客户端和服务器之间严格的契约，使得客户端可以更加智能和自适应，而 REST 服务本身的演化和更新也变得更加容易。 在介绍 HATEOAS 之前，先介绍一下 Richardson 提出的 REST 成熟度模型。该模型把 REST 服务按照成熟度划分成 4 个层次：

- 第一个层次（Level 0）的 Web 服务只是使用 HTTP 作为传输方式，实际上只是远程方法调用（RPC）的一种具体形式。SOAP 和 XML-RPC 都属于此类。
- 第二个层次（Level 1）的 Web 服务引入了资源的概念。每个资源有对应的标识符和表达。
- 第三个层次（Level 2）的 Web 服务使用不同的 HTTP 方法来进行不同的操作，并且使用 HTTP 状态码来表示不同的结果。如 HTTP GET 方法来获取资源，HTTP DELETE 方法来删除资源。
- 第四个层次（Level 3）的 Web 服务使用 HATEOAS。在资源的表达中包含了链接信息。客户端可以根据链接来发现可以执行的动作。

从上述 REST 成熟度模型中可以看到，使用 HATEOAS 的 REST 服务是成熟度最高的，也是推荐的做法。对于不使用 HATEOAS 的 REST 服务，客户端和服务器的实现之间是紧密耦合的。客户端需要根据服务器提供的相关文档来了解所暴露的资源和对应的操作。当服务器发生了变化时，如修改了资源的 URI，客户端也需要进行相应的修改。而使用 HATEOAS 的 REST 服务中，客户端可以通过服务器提供的资源的表达来智能地发现可以执行的操作。当服务器发生了变化时，客户端并不需要做出修改，因为资源的 URI 和其他信息都是动态发现的。

下面是一个 HATEOAS 的例子：

```
{
  "id": 711,
  "manufacturer": "bmw",
  "model": "X5",
  "seats": 5,
  "drivers": [
   {
    "id": "23",
    "name": "Stefan Jauker",
    "links": [
     {
     "rel": "self",
     "href": "/api/v1/drivers/23"
    }
   ]
  }
 ]
}
```

提供过滤、排序、字段选择、分页

过滤：

```
GET /cars?color=red
GET /cars?seats<=2
```

排序：

```
GET /cars?sort=-manufactorer,+model
```

字段选择:

```
GET /cars?fields=manufacturer,model,id,color
```

分页：

```
GET /cars?offset=10&limit=5
```

API 版本化

版本号使用简单的序号，并避免点符号，如2.5等。正确用法如下：

```
/blog/api/v1
```

充分使用 HTTP 状态码来处理错误

HTTP状态码（HTTP Status Code）是用以表示网页服务器 HTTP 响应状态的3位数字代码。它由 RFC 2616 规范定义的，并得到 RFC 2518、RFC 2817、RFC 2295、RFC 2774、RFC 4918 等规范的扩展。

在设计 API 处理错误时，应该充分使用 HTTP 状态码，而不是简单的抛出个 “500 – Internal Server Error（内部服务器错误）”

所有的异常都应该有个错误的 payload 作为映射，下面是一个例子：

```
{
  "errors": [
   {
    "userMessage": "Sorry, the requested resource does not exist",
    "internalMessage": "No car found in the database",
    "code": 34,
    "more info": "http://dev.mwaysolutions.com/blog/api/v1/errors/12345"
   }
  ]
}
```

参考引用

- 《REST 实战》 (https://github.com/waylau/rest-in-action)
- http://martinfowler.com/articles/richardsonMaturityModel.html (http://martinfowler.com/articles/richardsonMaturityModel.html)
- https://www.crummy.com/writing/speaking/2008-QCon/act3.html (https://www.crummy.com/writing/speaking/2008-QCon/act3.html)
- https://en.wikipedia.org/wiki/HATEOAS (https://en.wikipedia.org/wiki/HATEOAS)
- https://en.wikipedia.org/wiki/List_of_HTTP_status_codes (https://en.wikipedia.org/wiki/List_of_HTTP_status_codes)

← Hibernate 性能优化法则 (/tips-to-boost-your-hibernate-performance/) ↕ DB2 所有数据库表、表字段注释乱码问题的排查及解决方案 → (/db2-chinese-remarks-gibberish/)

Way Lau

Software Engineer and *Full Stack Developer*, now work and live in Hangzhou, China. Detail (https://waylau.com/resume)

Contact

- waylau521@gmail.com (/cdn-cgi/l/email-protection#0f786e76636e7a3a3d3e4f68626e6663216c6062)
- waylau521 (http://weibo.com/waylau521)
- waylau (https://github.com/waylau)
- waylau521 (https://twitter.com/waylau521)
- waylau521 (https://facebook.com/waylau521)
- RSS (/feed.xml)

Latest Comments

Recently Visitors

Donate

Code licensed under MIT (LICENSE),documentation under CC BY-NC-SA 3.0 CN (https://creativecommons.org/licenses/by-nc-sa/3.0/cn/)
Copyright © 2009-2017 柳伟卫/老卫/Way Lau's Personal Site - 关注编程、系统架构、性能优化 (https://waylau.com)
Powered by jekyll-bootstrap-blog (https://github.com/waylau/jekyll-bootstrap-blog)
粤ICP备16084618号-1 (http://www.miibeian.gov.cn/)