

Restful API设计最佳实践

RESTFul

REST(Representational State Transfer), 表现层状态转移。

它首次出现在2000年Roy Fielding的博士论文中，Roy Fielding是HTTP规范的主要编写者之一。

表现层是资源的表现层，对于网络中的资源就需要URI(Uniform Resource Identifier)来指向。REST指的是资源的状态变化。

注：RESTFul没有标准。

1、协议

使用HTTP或者HTTPS。对外若有安全性要求，可以使用HTTPS。但是内部服务间调用可以使用HTTP或HTTPS。

2、HTTP方法

HTTP请求中的方法表示执行的**动作**

常用方法（动词）	说明
GET	获取资源
POST	创建新的资源
PUT	更新资源
PATCH	部分更新资源
DELETE	删除资源

3、使用名词

URL指向**资源**，在URL路径的描述中，只需要出现名词，而不要出现动词。动词由HTTP方法提供。

不要单复数混用，建议名词使用复数。

Restful的核心是资源，URL应该指向资源，所以应该是使用名称表达，而不是动词表达。

方法	路径	说明
GET	/posts	返回所有文章
GET	/posts/10	返回id为10的文章
POST	/posts	创建新的文章
PUT	/posts/10	更新id为10的文章
DELETE	/posts/10	删除id为10的文章
PATCH	/posts/10	部分更新id为10的文章数据

不要出现下面的访问资源的路径

```

1 /getAllPosts
2 /addPost
3 /updatePost
4 /delPost
5 上面URL中动作和资源混在一起，不好

```

GET方法只是获取资源，而不是改变资源状态。改变资源请使用POST、PUT、DELETE等方法。

例如，使用 `GET /posts/10` 就可以获取资源了，但是却使用 `GET /posts/10/del` 或 `GET /posts/10?v=del`，本意是想删除。但这样不好，GET方法请求只为获取资源，不要改变资源状态。

子资源的访问

方法	路径Endpoint	说明
GET	/posts/10/authors	返回id为10的文章的所有作者
GET	/posts/10/authors/8	返回id为10的文章的作者中id为4的

4、集合功能

过滤 Filtering

指定过滤条件 `GET /posts?tag=python`

排序 Sorting

指定排序条件。有很多种设计风格，例如使用+表示asc，-表示desc。 `GET /posts?sort=+title,-id` 或 `GET /posts?sort=title_asc,id_desc`

分页 Pagination

一般情况下，查询返回的记录数非常多，必须分页。 `GET /posts?page=5&size=20`

5、状态码

使用HTTP响应的状态码表示动作的成功与否。

2XX表示用户请求被服务器端成功的处理；4XX表示用户请求的错误；5XX表示服务器端出错了

Status Code	说明	Method	说明
200	OK	GET	成功获取资源
201	CREATED	POST、PUT、PATCH	成功创建或修改
204	NO CONTENT	DELETE	成功删除资源
400	Bad Request	ALL	请求中有错误，例如 GET时参数有问题 PUT时提交的数据错误等
401	Unauthorized	ALL	权限未通过认证
403	Forbidden	ALL	有无权限都禁止访问该资源
404	Not Found	ALL	请求的资源不存在
500	Internal Server Error	ALL	服务器端错误

详细状态码参考<https://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html>

6、错误处理

在Restful API设计中，错误处理也非常重要。单单从状态码中无法详尽描述错误的信息。

1、返回消息

```
1 {  
2   "error":"User Not Found"  
3 }
```

2、从错误消息中了解到错误号、错误信息、错误描述等信息。甚至更详细的信息可以通过code查阅文档

```
1 {  
2   "code":10056,  
3   "message":"Invalid ID",  
4   "description":"More details":  
5 }
```

code 为 0 表示无错误。非0表示有错误，就要看message中的错误描述了。

7、版本

在较大项目开发时，由于业务升级，需要对接口调用做出改动，例如请求参数、返回状态码、json数据发生变化等。如果使用同一套接口，在原有接口上直接修改会导致原有调用者调用出问题。解决方法：

1. 完全兼容原有接口，虽然这样做可以，但是很难做到所有接口的完全兼容设计。实在不行的，老的接口保留不动，建立新的API接口以供调用

2. 对已经发布使用的接口规定版本号访问。新增的、改进的、删除的API重新发布新的版本。项目开发时，指定版本和接口即可。老项目可以不必升级到这个新版本。

强烈要求使用版本，版本号使用简单数字，例如v2。

2种风格

- <http://api.magedu.com/v1/posts/10> 这种风格会跨域，适合较大的项目
- <http://www.magedu.com/api/v1/posts/10>

8、返回结果

方法	路径	说明
GET	/posts	返回所有文章的列表
GET	/posts/10	返回id为10的文章对象
POST	/posts	创建新的文章并返回这个对象
PUT	/posts/10	更新id为10的文章并返回这个对象
DELETE	/posts/10	删除id为10的文章返回一个空对象
PATCH	/posts/10	部分更新id为10的文章数据并返回这个对象

返回数据一律采用JSON格式。

