

"RESTFUI API规范"

全称是Representational State Transfer

RESTful是一种定义Web API接口的设计风格，适用于前后端分离的应用模式中

"十条规范/也可以不遵守，但是要遵守最重要的***"

数据的安全保障

url链接一般都采用https协议进行传输

采用https协议，可以提高数据交互过程中的安全性

#接口特征表现

用api关键字标识接口url:

<https://api.baidu.com>

<https://www.baidu.com/api>

看到api字眼，就代表该请求url链接是完成前后台数据交互的

多数据版本共存

在url链接中标识数据版本

<https://api.baidu.com/v1>

<https://api.baidu.com/v2>

url链接中的v1、v2就是不同数据版本的体现（只有在一种数据资源有多版本情况下）

保持老版本app等应用可用

***数据即是资源，均使用名词（可复数）

接口一般都是完成前后台数据的交互，交互的数据我们称之为资源

<https://api.baidu.com/users>

<https://api.baidu.com/books>

<https://api.baidu.com/book>

一般提倡用资源的复数形式，在url链接中奖励不要出现操作资源的动词

错误示范：<https://api.baidu.com/delete-user>

特殊的接口可以出现动词，因为这些接口一般没有一个明确的资源

或是动词就是接口的核心含义

<https://api.baidu.com/place/search>

<https://api.baidu.com/login>

***资源操作由请求方式决定（method）

操作资源一般都会涉及到增删改查，提供请求方式来标识增删改查动作

<https://api.baidu.com/books>

get请求：获取所有书

<https://api.baidu.com/books/1>

get请求：获取主键为1的书

<https://api.baidu.com/books>

post请求：新增一本书书

<https://api.baidu.com/books/1>

put请求：整体修改主键为1的书

<https://api.baidu.com/books/1>

patch请求：局部修改主键为1的书

<https://api.baidu.com/books/1>

delete请求：删除主键为1的书

过滤，通过在url上传参的形式传递搜索条件

<https://api.example.com/v1/zoos?limit=10>

指定返回记录的数量

<https://api.example.com/v1/zoos?offset=10>

指定返回记录的开始位置

https://api.example.com/v1/zoos?page=2&per_page=100

指定第几页，以及每页的记录数

<https://api.example.com/v1/zoos?sortby=name&order=asc>

指定返回结果按照哪个属性排序

以及排序顺序

https://api.example.com/v1/zoos?animal_type_id=1

指定筛选条件

```
# 响应状态码
  响应状态码2xx
    200: 常规请求
    201: 创建成功
  响应状态码3xx
    301: 永久重定向
    302: 暂时重定向
  响应状态码4xx
    403: 请求无权限
    404: 请求路径不存在
    405: 请求方法不存在（是否支持）
  响应状态码5xx
    500: 服务器异常
```

```
# 错误处理，应返回错误信息，error当做key
{
  error: "无权限操作"
}
```

```
# 返回结果，针对不同操作，服务器向用户返回的结果应该符合以下规范
GET /collection      返回资源对象的列表（数组）
GET /collection/resource 返回单个资源对象
POST /collection     返回新生成的资源对象
PUT /collection/resource 返回完整的资源对象
PATCH /collection/resource 返回完整的资源对象
DELETE /collection/resource 返回一个空文档
```

```
# 需要url请求的资源需要访问资源的请求链接
```

```
# Hypermedia API, RESTful
```

API最好做到Hypermedia，即返回结果中提供链接，连向其他API方法，使得用户不查文档，也知道下一步应该做什么

```
{
  "status": 0,
  "msg": "ok",
  "results":[
    {
      "name":"肯德基(罗餐厅)",
      "img": "https://image.baidu.com/kfc/001.png"
    }
    ...
  ]
}
```

比较好的接口返回

```
# 响应数据要有状态码、状态信息以及数据本身
{
  "status": 0,
  "msg": "ok",
  "results": [
    {
      "name": "肯德基(罗餐厅)",
      "location": {
        "lat": 31.415354,
        "lng": 121.357339
      },
      "address": "月罗路2380号",
      "province": "上海市",
      "city": "上海市",
      "area": "宝山区",
      "street_id": "339ed41ae1d6dc320a5cb37c",
      "telephone": "(021)56761006",
      "detail": 1,
      "uid": "339ed41ae1d6dc320a5cb37c"
    }
    ...
  ]
}
```