
02.前后端交互规范

最后修改者 平 陶

打开 2019-05-31 15:19:59

内容列表

- 2. 前后端交互规范 3
 - 2.1 API交互返回数据规范 3
 - 2.2 HTTP 请求规范 3
 - 2.3 API版本规范 3
 - 2.4 API命名规范 3

2. 前后端交互规范

2.1 API交互返回数据规范

先说一个重点：不允许定义无返回值void的API方法，至少返回Boolean；

采用前后端分离的RESTful架构风格，前端调用后端API不管有无异常都要以HTTP 200状态返回，将返回的数据进行包装，从包装的数据体中解析是正常还是异常。

```
{
  "status": "200",    -- 状态码，非200为异常
  "data": {},         -- 数据主体
  "message": "",      -- 异常消息文本
  "extends": {}       -- 扩展数据体，用于如异常、字典、工作流等在主体API返回后还需扩展的数据
}
```

返回status非200则为异常，10000以下为系统内部异常，前端应结合实际情况酌情显示相应的提示信息；

返回status状态码在10000以上的异常码为业务异常码，应提示相应的异常提示信息。

2.2 HTTP 请求规范

- GET (SELECT)：查询；从服务器取出资源（一项或多项）；
- POST (CREATE)：新增；在服务器新建一个资源；
- PUT (UPDATE)：更新；在服务器更新资源（客户端提供改变的属性）；
- DELETE (DELETE)：删除；从服务器删除资源。

2.3 API版本规范

- 应该将API的版本号放入URL，统一使用大版本号及只有1位数的版本号，api小版本变更保持只做加法的原则。大版本号变更应是整体统一变更。

api定义示例：

http://api.example.com/v1/

http://example.com/api/v1

2.4 API命名规范

- 网址中不能有动词，只能有名词，而且所用的名词往往与数据库的表格名对应
- 数据库中的表都是同种记录的“集合”（collection），所以API中的名词也应该使用复数
- 尽量不要用大写，单词间使用下划线‘_’
- url 层级大于或等于三层，则使用‘?’带参数

API定义示例：

```
GET      /zoos: 列出所有动物园
POST     /zoos: 新建一个动物园
GET      /zoos/ID: 获取某个指定动物园的信息
PUT      /zoos/ID: 更新某个指定动物园的信息（提供该动物园的全部信息）
PATCH   /zoos/ID: 更新某个指定动物园的信息（提供该动物园的部分信息）
DELETE   /zoos/ID: 删除某个动物园
GET      /zoos/ID/animals: 列出某个指定动物园的所有动物
DELETE   /zoos/ID/animals/ID: 删除某个指定动物园的指定动物
DELETE   /zoos/ID/animals?id=: 删除某个指定动物园的指定动物
```