

# 网络API接口设计

目前许多的Web应用和移动应用都使用了前后端分离的开发模式，前后端分离简单的说就是前端或移动端通过网络API接口和后台进行交互。API是应用程序的编程接口的缩写；网络API通常指的是基于一个URL（统一资源定位符）可以访问到的资源，也就是说通过这个URL我们可以让服务器对某个资源进行操作并返回操作的结果，复杂的业务逻辑被隐藏在简单的API接口中。URL的通用格式如下所示：

```
协议://用户名:口令@主机:端口/路径1/.../路径N/资源名
```

说明：URL中的用户名（有可能不需要提供用户名）、口令（有可能不需要提供口令）、端口（有可能使用默认端口）、路径（资源有可能直接位于根路径下）并不是必需的部分，可以根据需要进行设置。

网络API通常基于HTTP或HTTPS进行访问，基于HTTP/HTTPS最大的好处就在于访问起来非常的简单方便，而且可以跨语言、跨应用进行访问和互操作。

## 设计原则

### 关键问题

为移动端或者PC端设计网络API接口一个非常重要的原则是：根据业务实体而不是用户界面或操作来设计。如果API接口的设计是根据用户的操作或者界面上的功能设置来设计，随着需求的变更，用户界面也会进行调整，需要的数据也在发生变化，那么后端开发者就要不停的调整API，或者给一个API设计出多个版本，这些都会使项目的开发和维护成本增加。

下面是某个网站开放API的接口，可以看出API的设计是围绕业务实体来进行的，而且都做到了“见名知意”。

评论	
comments/show	获取某条微博的评论列表
comments/by_me	自己的评论列表
comments/to_me	收到的评论列表
comments/mentions	@了自己的评论列表
comments/create	创建一条评论
comments/destroy	删除一条评论
comments/reply	回复一条评论

需要说明的是，上面的API接口并不是REST风格的。REST是一种网络应用架构风格，被认为最适合分布式的网络应用。关于REST的知识，可以阅读阮一峰老师的[《理解RESTful架构》](#)以及[《RESTful API设计指南》](#)，当然这两篇文章大家也要批判的阅读，因为上面阐述的观点并不完全正确，有些内容甚至是自相矛盾的。

API接口返回的数据通常都是JSON或XML格式，我们这里不会讲述XML的知识，因为这种格式几乎已经被淘汰掉了。对于JSON格式的数据，我们需要做到不要返回null这的值，因为这样的值一旦处置失当，会给前端和移动端开发带来不必要的麻烦（因为开发者有可能会使用强类型语言）。要解决这个问题可以从源头入手，在设计数据库的时候，尽量给每个字段都加上“not null”约束或者设置合理的默认值约

束。

其他问题

- 1. 更新提示问题：设计一个每次使用系统首先要访问的API，该API会向移动端返回系统更新的相关信息，这样就可以提升用户更新App了。
- 2. 版本升级问题：API版本升级时应该考虑对低版本的兼容，同时要让新版本和旧版本都能够被访问，可以在URL中包含版本信息或者在将版本号放在HTTP(S)协议头部，关于这个问题有很多的争论，有兴趣的可以看看[stack overflow](#)上面对这个问题的讨论。
- 3. 图片尺寸问题：移动端对于一张图片可能需要不同的尺寸，可以在获取图片时传入尺寸参数并获取对应的资源；更好的做法是直接使用云存储或CDN（直接提供了图片缩放的功能），这样可以加速对资源的访问。

文档撰写

下面以设计评论接口为例，简单说明接口文档应该如何撰写。

评论接口

全局返回状态码

返回码	返回信息	说明
10000	获取评论成功	
10001	创建评论成功	
10002	无法创建评论	创建评论时因违反审核机制而无法创建
10003	评论已被删除	查看评论时评论因不和谐因素已被删除
10004	.....	.....

1. GET /articles/{article-id}/comments/

开发者：王大锤  
最后更新时间：2018年8月10日  
标签：v 1.0  
接口说明：获取指定文章的所有评论  
使用帮助：默认返回20条数据，需要在请求头中设置身份标识（key）  
请求参数：

参数名	类型	是否必填	参数位置	说明
page	整数	否	查询参数	页码，默认值1
size	整数	否	查询参数	每次获取评论数量（10~100），默认值20
key	字符串	是	请求头	用户的身份标识

响应信息：

```
{
  "code": 10000,
  "message": "获取评论成功",
  "page": 1,
  "size": 10,
  "totalPage": 35,
  "contents": [
    {
      "userId": 1700095,
      "nickname": "王大锤",
      "pubDate": "2018年7月31日",
      "content": "小编是不是有病呀",
      /* ... */
    },
    {
      "userId": 1995322,
      "nickname": "白元芳",
      "pubDate": "2018年8月2日",
      "content": "楼上说得好",
      /* ... */
    }
  ]
  /* ... */
}
```

## 2. POST /articles/{article-id}/comments

开发者：王大锤

最后更新时间：2018年8月10日

标签：v 1.0

接口说明：为指定的文章创建评论

使用帮助：暂无

请求参数：

参数名	类型	是否必填	参数位置	说明
userId	字符串	是	消息体	用户ID
key	字符串	是	请求头	用户的令牌
content	字符串	是	消息体	评论的内容

响应信息：

```
{
  "code": 10001,
  "message": "创建评论成功",
  "comment": {
    "pubDate": "2018年7月31日",
    "content": "小编是不是有病呀"
    /* ... */
  }
  /* ... */
}
```

提示：如果没有接口文档撰写经验，可以使用在线接口文档编辑平台RAP2或YAPI来进行接口文档撰写，也可以参考我的[《FTX（租房项目）接口文档》](#)来了解如何撰写接口文档。