

Koa2 为单页面应用 Vue Angular 提供 api 接口 前后端分离 RESTful API 设计指南

主讲教师：（大地）

合作网站：<http://www.itying.com/>

中文文档：<http://www.itying.com/koa>

一、Koa2 RESTful API 设计指南

网络应用程序，分为前端和后端两个部分。当前的发展趋势，就是前端设备层出不穷（手机、平板、桌面电脑、其他专用设备.....）。

因此，必须有一种统一的机制，方便不同的前端设备与后端进行通信。这导致 API 构架的流行，甚至出现"API First"的设计思想。RESTful API 是目前比较成熟的一套互联网应用程序的 API 设计理论

一个好的 RESTful API 我们从以下几个方面考虑。

1. 协议：建议使用更安全的 https 协议
2. 域名：尽量部署在专属域名下面，比如 <https://a.itying.com> <https://api.itying.com>
3. 应该将 api 的版本号放入 URI 中：
 - (1) 比如：<https://a.itying.com/api1/newslst> <https://a.itying.com/api2/newslst>
 - (2) 比如：<https://a1.itying.com> <https://a2.itying.com>
4. 路径：在 RESTful 架构中，每个网址代表一种资源（resource），所以网址中建议不能有动词，只能有名词，而且所用的名词往往与数据库的表格名对应。一般来说，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。
5. http 请求数据的方式：（7 个 HTTP 方法：GET/POST/PUT/DELETE/PATCH/HEAD/OPTIONS）

GET（SELECT）：从服务器取出资源（一项或多项）。

POST（CREATE）：在服务器新建一个资源。

PUT（UPDATE）：在服务器更新资源（客户端提供改变后的完整资源）。

DELETE（DELETE）：从服务器删除资源。

还有三个不常用的 HTTP 请求方式。

HEAD: 获取资源的元数据。

OPTIONS: 获取信息，关于资源的哪些属性是客户端可以改变的。

PATCH (UPDATE): 在服务器更新资源（客户端提供改变的属性）。

6、过滤方式、请求数据方式、返回数据、安全。