**1.Restful接口规范介绍**

REST：Representational State Transfer，

REST 指的是一组架构约束条件和原则。满足这些约束条件和原则的应用程序或设计就是 RESTful。

是一种软件架构风格、设计风格，而不是标准，只是提供了一组设计原则和约束条件。

它主要用于客户端和服务器交互类的软件。基于这个风格设计的软件可以更简洁，更有层次。

restful接口规范是用于在前端与后台进行通信的一套规范。使用这个规范可以让前后端开发变得更加轻松。

**2.适用场景：**一个系统的数据库数据，展现的平台有PC端、移动端、app端、ios端。

前端工程师：都遵循RESTful编程规范

后端工程师：都遵循RESTful编程规范

最终结果：开发效率高，便于管理

**3.协议：**用http或者https协议。

**4.数据传输格式：**

数据传输的格式应该都用json格式。

**5.url链接规则：**

url链接中，不能有动词，只能有名词。

并且对于一些名词，如果出现复数，那么应该在后面加s。

比如：获取新闻列表，应该使用`/news/`，而不应该使用/get\_news/

**6.HTTP请求方式：**

GET：从服务器上获取资源。

POST：在服务器上新增或者修改一个资源。

PUT：在服务器上更新资源。（客户端提供所有改变后的数据）

PATCH：在服务器上更新资源。（客户端只提供需要改变的属性）

DELETE：从服务器上删除资源。

**7.状态码：**

|  |  |  |
| --- | --- | --- |
| **状态码** | **原因描述** | **描述** |
| 200 | OK | 服务器成功响应客户端的请求。 |
| 400 | INVALID REQUEST | 用户发出的请求有错误，服务器没有进行新建或修改数据的操作 |
| 401 | Unauthorized | 用户没有权限访问这个请求 |
| 403 | Forbidden | 因为某些原因禁止访问这个请求 |
| 404 | NOT FOUND | 用户请求的url不存在 |
| 406 | NOT Acceptable | 用户请求不被服务器接收（比如服务器期望客户端发送某个字段，但是没有发送）。 |
| 500 | Internal server error | 服务器内部错误，比如遇到bug |