

REST初步(REST 接口的统一性)

(参考http://blog.csdn.net/linux2_scdn/article/details/52578273, 222

与http://blog.csdn.net/Linux2_SCDN/article/category/6422929, http://www.ruanyfeng.com/blog/2014/05/restful_api.html (比较清晰))

另HTTP状态码 (<https://baike.baidu.com/item/HTTP%E7%8A%B6%E6%80%81%E7%A0%81/5053660?fr=aladdin>)

1. REST是一种思想(面向资源ROA 即Resource-Oriented-Architecture, 面向资源的架构), 是一种编程风格:

- 基于HTTP方法
- 采用URI显式定义作用域(呈 目录结构 e.g. /zoos/ID/animals)
- 关注于资源而不是客户端-服务端之间的方法(与RPC显著区别)

2. REST接口的统一性

命名成的GET或其他方式并不代表代码里就一定是遵循该方式编程的, 例如;GET是获取资源, 如果在代码中有了写的操作, 那么该代码就不是REST风格的, 也不应该用GET方式请求. 因此, 该方式的命名只是一种标志性的统一, 只有代码也遵循该请求方式的内涵才是具备接口统一性特点的REST编程, 才能具有各个方式的特点

- GET 读取资源

特点: 安全性 (不改变服务端的资源状态), 幂等性 (多次读取同一资源, 得到相同的数据)

- PUT 添加和更新资源

特点: 不安全, 幂等的. (可以用于区分添加资源的时候使用PUT还是POST)

用于更改: 多次使用同一份数据更新服务端的同一资源, 应该得到相同的结果. 所以用PUT, 但是有人会想到时间字段, 但此时已经不是同一份数据了. (注: 该同一份数据指的是数据真正用于更新操作到数据库的数据, 并非是客户端传入的数据)

用于添加: 因为PUT是幂等的, 所以只有一种情况下会在添加的时候用PUT, 就是插入数据的时候不改变用于插入数据库资源的数据 (注: 事实上我们一般的添加操作都是插入到数据库时自动生成该条数据的自增主键, 此时已经改变了原来的数据, 所以是不能用PUT的, 只有多次插入同一数据并且返回的数据是完全相同的时候才能用PUT, 所以添加的时候基本上不用)

- DELETE 删除资源

特点: 不安全. 幂等的 (注: 多次删除同一份资源, 总是得到相同的数据, 这里可以加深对幂等性的了解, 幂等性中为什么说“多次删除”的同一份资源, 在第一次删除后就已经不是同一份了也就是说在幂等性中所说的同一份就是绝对的同一份, 这里的多次是指多次在完全相同的环境下重新操作)

- POST 添加资源

特点: 不安全, 非幂等性

带有写操作的方式都不是安全的 (这里的“安全”只针对于服务器端的资源状态, 不同于系统安全).