# 什么是RESTful API？

2017年08月17日 16:20:21

阅读数：11500

要弄清楚什么是RESTful API,首先要弄清楚什么是REST。REST -- REpresentational State Transfer，英语的直译就是“表现层状态转移”。如果看这个概念，估计没几个人能明白是什么意思。那下面就让我来用一句人话解释一下什么是RESTful:**URL定位资源，用HTTP动词（GET,POST,PUT,DELETE)描述操作。**

**Resource：资源，即数据。  
     Representational：某种表现形式，比如用JSON，XML，JPEG等；  
     State Transfer：状态变化。通过HTTP动词实现。**

所以RESTful API就是REST风格的API。 那么在什么场景下使用RESTful API呢？在当今的互联网应用的前端展示媒介很丰富。有手机、有平板电脑还有PC以及其他的展示媒介。那么这些前端接收到的用户请求统一由一个后台来处理并返回给不同的前端肯定是最科学和最经济的方式，RESTful API就是一套协议来规范多种形式的前端和同一个后台的交互方式。

     RESTful API由后台也就是SERVER来提供前端来调用。前端调用API向后台发起HTTP请求，后台响应请求将处理结果反馈给前端。也就是说RESTful 是典型的基于HTTP的协议。那么RESTful API有哪些设计原则和规范呢？

     1，资源。首先是弄清楚资源的概念。资源就是网络上的一个实体，一段文本，一张图片或者一首歌曲。资源总是要通过一种载体来反应它的内容。文本可以用TXT，也可以用HTML或者XML、图片可以用JPG格式或者PNG格式，JSON是现在最常用的资源表现形式。

     2，统一接口。RESTful风格的数据元操CRUD（create,read,update,delete）分别对应HTTP方法：GET用来获取资源，POST用来新建资源（也可以用于更新资源），PUT用来更新资源，DELETE用来删除资源，这样就统一了数据操作的接口。

     3，URI。可以用一个URI（统一资源定位符）指向资源，即每个URI都对应一个特定的资源。要获取这个资源访问它的URI就可以，因此URI就成了每一个资源的地址或识别符。一般的，每个资源至少有一个URI与之对应，最典型的URI就是URL。

     4，无状态。所谓无状态即所有的资源都可以URI定位，而且这个定位与其他资源无关，也不会因为其他资源的变化而变化。有状态和无状态的区别，举个例子说明一下，例如要查询员工工资的步骤为第一步：登录系统。第二步：进入查询工资的页面。第三步：搜索该员工。第四步：点击姓名查看工资。这样的操作流程就是有状态的，查询工资的每一个步骤都依赖于前一个步骤，只要前置操作不成功，后续操作就无法执行。如果输入一个URL就可以得到指定员工的工资，则这种情况就是无状态的，因为获取工资不依赖于其他资源或状态，且这种情况下，员工工资是一个资源，由一个URL与之对应可以通过HTTP中的GET方法得到资源，这就是典型的RESTful风格。

     说了这么多，到底RESTful长什么样子的呢？

GET:http://www.xxx.com/source/id 获取指定ID的某一类资源。例如GET:http://www.xxx.com/friends/123表示获取ID为123的会员的好友列表。如果不加id就表示获取所有会员的好友列表。

POST:http://www.xxx.com/friends/123表示为指定ID为123的会员新增好友。其他的操作类似就不举例了。

     RESTful API还有其他一些规范。1：应该将API的版本号放入URL。GET:http://www.xxx.com/v1/friend/123。或者将版本号放在HTTP头信息中。我个人觉得要不要版本号取决于自己开发团队的习惯和业务的需要，不是强制的。2：URL中只能有名词而不能有动词，操作的表达是使用HTTP的动词GET,POST,PUT,DELETEL。URL只标识资源的地址，既然是资源那就是名词了。3：如果记录数量很多，服务器不可能都将它们返回给用户。API应该提供参数，过滤返回结果。?limit=10：指定返回记录的数量、?page=2&per\_page=100：指定第几页，以及每页的记录数。