
Retrofit & RESTful

1. Retrofit 简介

Retrofit 是 Square 公司基于 RESTful 风格推出的网络框架封装，Retrofit 是基于 OKHttp 的网络请求框架的二次封装，其本质仍是 OKHttp。

2. RESTful 简介

RESTful 架构风格最初由 Roy T. Fielding (HTTP/1.1 协议专家组负责人) 在其 2000 年的博士学位论文中提出。HTTP 就是该架构风格的一个典型应用。从其诞生之日起开始，它就因其可扩展性和简单性受到越来越多的架构师和开发者们的青睐。时至今日，RESTful 架构风格已成为企业级服务的标配。

REST 即 Representational State Transfer 的缩写，可译为“表现层状态转化”。

REST 最大的几个特点为：资源、统一接口、URI 和无状态。

3. 资源

所谓“资源”，就是网络上的一个实体，或者说是网络上的一个具体信息。它可以是一段文本、一张图片、一首歌曲、一种服务，总之就是一个具体的存在。资源总要通过某种载体反应其内容，文本可以用 txt 格式表现，也可以用 HTML 格式、XML 格式，甚至可以采用二进制格式来表现；图片可以用 JPG 格式表现，也可以用 PNG 格式表现。JSON 是现在最常用的资源表示格式之一。

4. 统一接口

RESTful 架构风格规定，数据的元操作，也就是数据的增删改查操作，分别对应于 HTTP 的 GET，POST 等方法，这样就统一了数据操作的接口，仅通过 HTTP 方法，就可以完成对数据的所有增删改查工作。

即：

GET (SELECT): 从服务器取出资源。

POST (CREATE): 在服务器新建一个资源。

PUT (UPDATE): 在服务器更新资源。

DELETE (DELETE): 从服务器删除资源。

5. 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。要想获取这个资源, 只要访问它的 URI 就可以了, 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 即 URL。

6. 无状态

所谓无状态的, 即所有的资源, 都可以通过 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改变。例如: 查询员工的工资。如果查询工资是需要登录系统, 进入查询工资的页面, 执行相关操作后, 获取工资的多少, 则这种情况是有状态的, 因为查询工资的每一步操作都依赖于前一步操作, 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 即可得到指定员工的工资, 则这种情况是无状态的, 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。