MockWebServer使用文档

目录

[1. MockWebServer介绍 1](#_Toc454356513)

[2. MockWebServer能帮我们做什么 1](#_Toc454356514)

[3. MockWebServer不能帮我们做什么 2](#_Toc454356515)

[4. MockWebServer环境配置 2](#_Toc454356516)

[5. MockWebServer一般步骤 2](#_Toc454356517)

[6. MockWebServer使用方法 3](#_Toc454356518)

[6.1. 添加预置的响应 3](#_Toc454356519)

[6.2. 模拟网速慢的情况 3](#_Toc454356520)

[6.3. RecordedRequest的使用 4](#_Toc454356521)

[6.4. Dispatcher的使用 4](#_Toc454356522)

# MockWebServer介绍

MockWebServer是square出品的跟随okhttp一起发布，用来Mock服务器行为的库。MockWebServer使用在**单元测试中**，专门用来测试http请求。其原理是启动了一个本地web server，监听了本地某个端口。所以在测试的时候，需要把请求的url替换为MockWebServer提供的地址，然后web server会handle请求，记录请求，根据预设的stubbing进行反馈。MockWebServer提供接口，让我们可以在发起请求之后，验证请求是否和预期的一样，比如验证header、path等等。

# MockWebServer能帮我们做什么

MockWebServer可以mock反馈，验证请求，一下是MockWebServer能帮我们做的事情

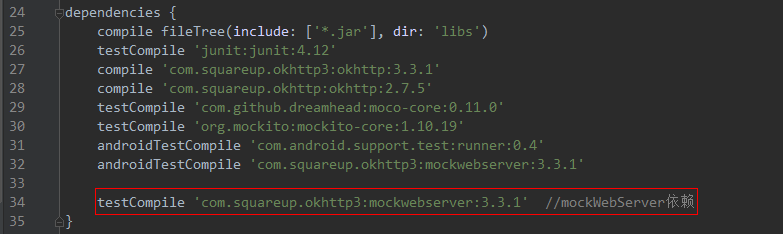
* 可以设置http response，设置response的header、body、status code等
* 可以记录接收到的请求，获取请求的body、header、method、path、HTTP version（在单元测试中很有用）
* 可以模拟网速慢的网络环境
* 提供Dispatcher，让mockWebServer可以根据不同的请求进行不同的反馈

# MockWebServer不能帮我们做什么

MockWebServer是在单元测试中启动一个web server的，主要用于测试验证。并不能像tomcat、moco一样，独立运行一个web server服务。如果需要长期运行一个web server，请选用moco或者tomcat等来搭建一个web server

# MockWebServer环境配置

Android Studio中使用MockWebServer很简单，只需要在build.gradle文件中加入依赖即可。如图：



添加依赖之后，点击sync,即可自动下载：



# MockWebServer一般步骤

1. 为mock server设置response
2. 客户端发起请求
3. 用mockWebServer记录的请求进行验证

下面是一个使用的例子：



# MockWebServer使用方法

## 添加预置的响应



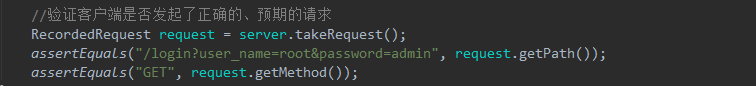
预置的相应，会按照添加的顺序依次返回给客户端。可以给MockResponse设置header、状态码、body。

## 模拟网速慢的情况



设置这个MockResponse返回的时候，以低速率传输。

## RecordedRequest的使用



按顺序从web server把接收到的request取出来，这是一个阻塞的方法，会一直等待到web server接收到了请求之后再返回。RecordedRequest可以帮助我们验证我们的请求客户端是否按预期生成了请求，可以验证的内容分别有：

* 请求method
* 请求path
* 请求header
* 请求body
* 请求HTTP version

## Dispatcher的使用

有时候我们希望web server能根据我们的请求返回不同的response，dispatcher可以帮助我们做到这个，使用方法如下：



开发者也可以根据不同的设定（如method、header等）放回不同的响应。