

环境准备

1. 扫码添加今日实战群，按照群公告设置环境

2. IDE/Postman/Java环境/Gradle

3. 请下载以下jar包到本地：

WireMock Jar: <http://repo1.maven.org/maven2/com/github/tomakehurst/wiremock-standalone/2.5.0/wiremock-standalone-2.5.0.jar>

4. 将会用到以下Repo，请拉到本地并且用IDE逐个打开下载依赖：

<https://github.com/Mikuu/SBD-single-wiremock>

<https://github.com/Mikuu/SBD-douban-book-api>

<https://github.com/Mikuu/SBD-simple-ui.git>

<https://github.com/Mikuu/SpringBootDemoAdmin.git>

<https://github.com/Mikuu/SpringBootDemoFrontend.git>

ThoughtWorks®

Chengdu BQMeetup

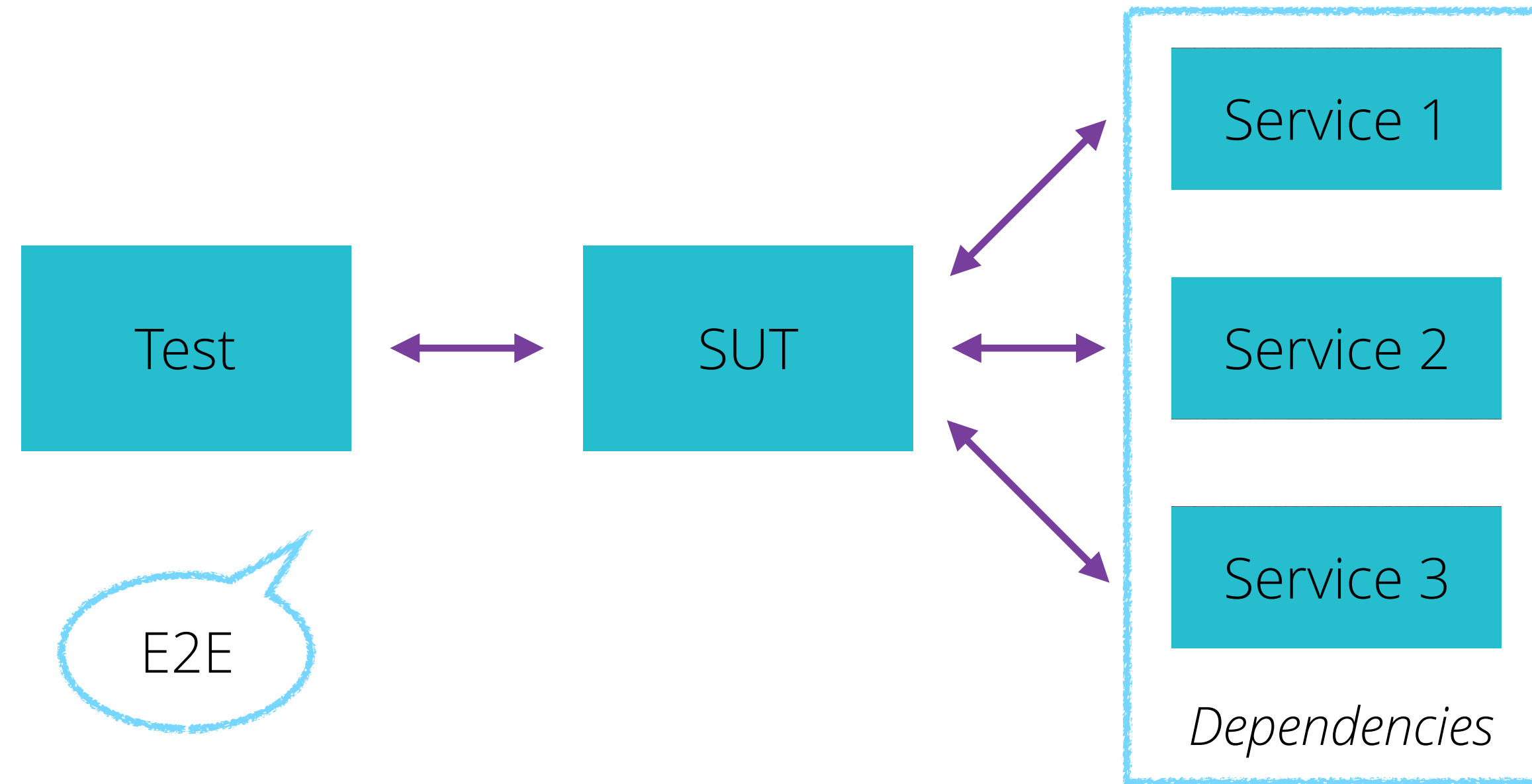
WIREMOCK

模拟API以进行快速、健壮、全面的测试

ThoughtWorks®

PART 1 - 简介

常见问题



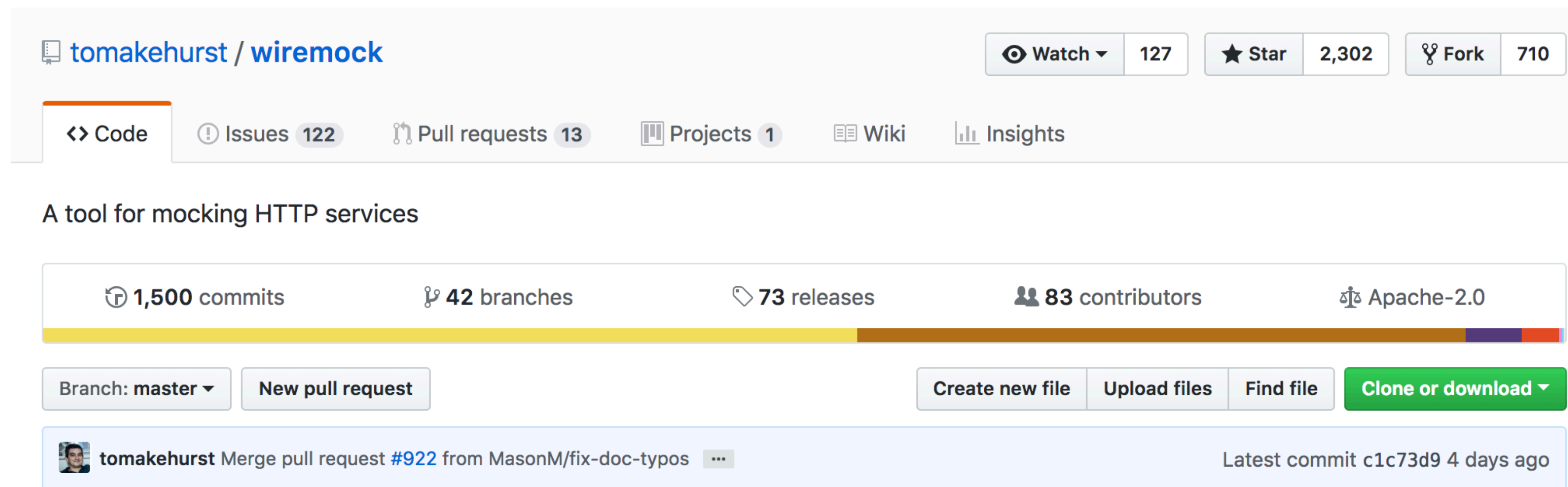
问题：

- 在依赖准备好之前，无法进行SUT的开发；
- 在依赖不稳定时，测试无法持续稳定地被执行；
- ○ ○ ○

WIREMOCK简介

WireMock是一个开源的测试工具，支持HTTP响应存根、请求验证、代理/拦截、记录和回放，它可以：

- 为Web/移动应用构建Mock Service
- 快速创建Web API原型
- 模拟Web Service中错误返回
- 录制HTTP请求和回放



The screenshot shows the GitHub repository page for tomakehurst/wiremock. At the top, the repository name is displayed with icons for Watch (127), Star (2,302), and Fork (710). Below this is a navigation bar with links to Code, Issues (122), Pull requests (13), Projects (1), Wiki, and Insights. The main description reads "A tool for mocking HTTP services". A statistics bar shows 1,500 commits, 42 branches, 73 releases, 83 contributors, and Apache-2.0 license. Action buttons include "Branch: master", "New pull request", "Create new file", "Upload files", "Find file", and "Clone or download". A recent activity bar at the bottom shows a merge pull request #922 by tomakehurst from MasonM/fix-doc-typos, with the latest commit c1c73d9 made 4 days ago.

tomakehurst / wiremock

Watch 127 Star 2,302 Fork 710

Code Issues 122 Pull requests 13 Projects 1 Wiki Insights

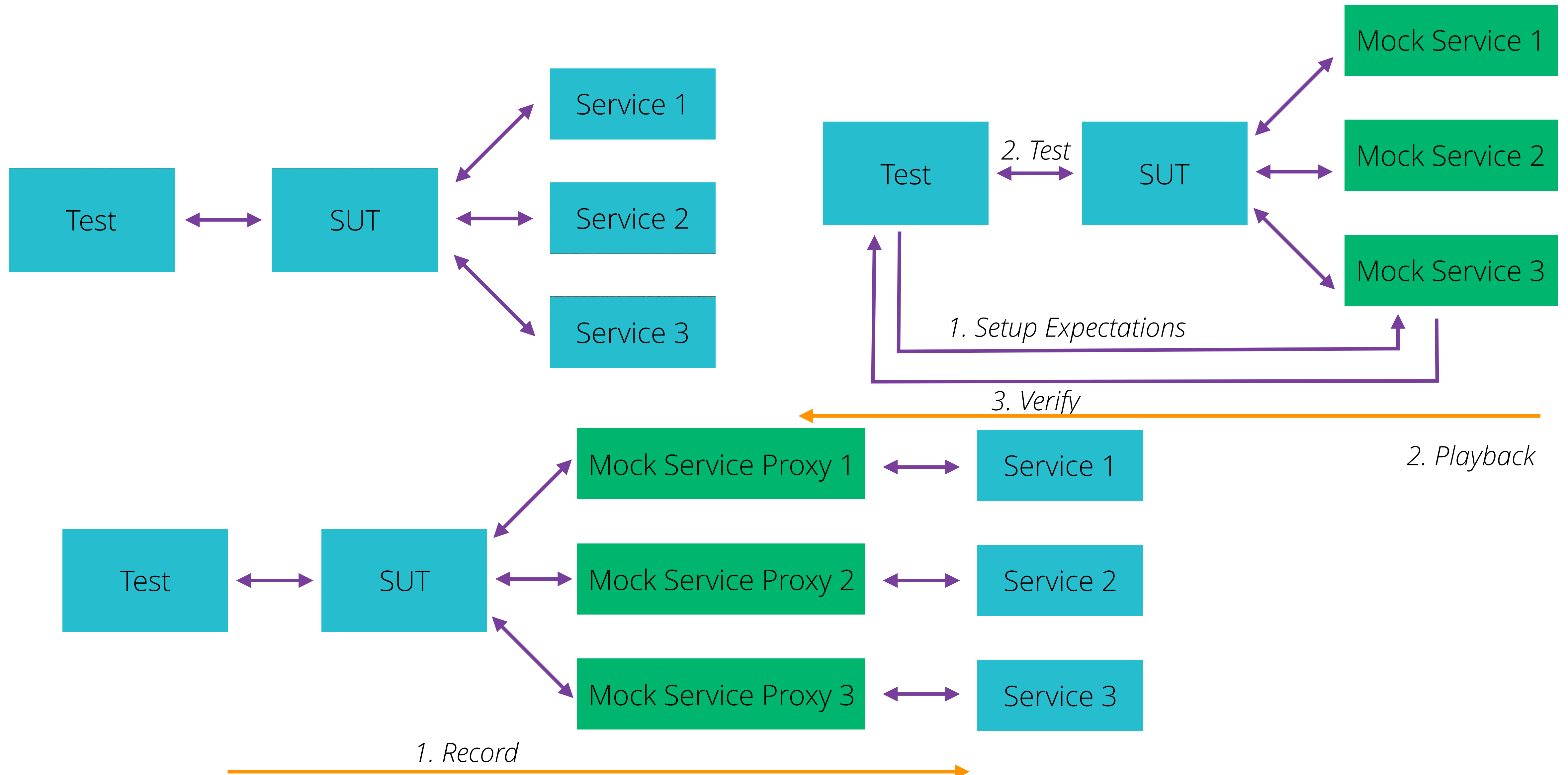
A tool for mocking HTTP services

1,500 commits 42 branches 73 releases 83 contributors Apache-2.0

Branch: master New pull request Create new file Upload files Find file Clone or download

tomakehurst Merge pull request #922 from MasonM/fix-doc-typos Latest commit c1c73d9 4 days ago

WIREMOCK使用场景

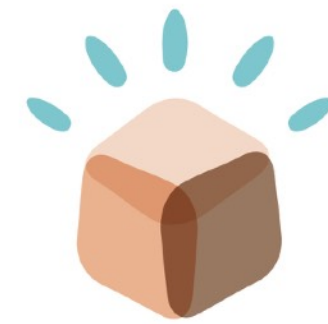


WIREMOCK、MOCO & MOCKSERVER

	WireMock	Moco	MockServer
Clients	Java	Java	Java、JavaScript、Ruby
Proxy	Record/Replay, Redirect, Intercept, Additional headers, Browser Proxy	Failover Playback, Redirect	Record requests and responses, Verify Requests, Port Forwarding, Web Proxying, HTTPS Tunneling Proxying, SOCKS Proxying
Simulating Faults	Random delays, Bad responses	Simply Latency	Delay
Template	Headers, Bodies, Proxy URLs	Content, Header, Query, Form, Cookie, Custom Variable, Redirect, File Name...	No
Stateful Behaviour	Yes	No	No
Other	Spring Boot	Easy to use	Docker

PART 2 - 基本用法

WIREMOCK SERVICE使用方式



Java



Standalone

下载 STANDALONE

<http://repo1.maven.org/maven2/com/github/tomakehurst/wiremock-standalone/2.5.0/wiremock-standalone-2.5.0.jar>

Initialise

```
private WireMockServer wireMockService = new WireMockServer(port); //port is 8080 by default  
wireMockService.start();
```

```
wireMockService.stop();
```

MOCK SERVICE IN JAVA

Registry Stub to mock service

```
stubFor(get(urlMatching("/api/demo"))  
    .willReturn(aResponse()  
        .withStatus(200)  
        .withStatusMessage("Everything was just fine!")  
        .withHeader("Content-Type", "text/plain")  
        .withBody("Stub For demo1 (Java)")));
```

PS: get / post / delete / put / options

MOCK SERVICE WITH STANDALONE

```
# Start mock service
```

```
>> java -jar wiremock-standalone-2.5.0.jar --port 2346
```

MOCK SERVICE WITH STANDALONE

Create stub in JSON

```
{  
  "request": {  
    "method": "POST",  
    "url": "/api/demo3",  
    "bodyPatterns": [{  
      "equalTo": "apple"  
    }]  
  },  
  "response": {  
    "status": 200,  
    "body": "Stub For Demo3(JSON)"  
  }  
}
```

PS: src/main/resources/Demo_PostWithReq.json

ThoughtWorks®

PART 3 - 实战

#1. start Application with local config

```
>> export SPRING_PROFILES_ACTIVE=local
```

#2. start Application with mock config

```
>> export SPRING_PROFILES_ACTIVE=mock
```

**POSTMAN >> [http://localhost:8080/douban/book/search?
query=Python&count=3](http://localhost:8080/douban/book/search?query=Python&count=3)**

Run API test

#3. Start mock service in Record mode in another terminal

```
>> cd mock
```

```
>> java -jar wiremock-standalone-2.5.0.jar --port 3000 --root-dir=mock-data --proxy-all="https://  
api.douban.com" --record-mappings --verbose
```

#4. Start mock service in Playback mode in another terminal

```
>> cd mock
```

```
>> java -jar wiremock-standalone-2.5.0.jar --port 3000 --root-dir=mock-data --verbose
```

Run Integration test

#1. start Application with local config

```
>> export SPRING_PROFILES_ACTIVE=local
>> gradlew bootRun
```

#2. start Application with mock config

```
>> export SPRING_PROFILES_ACTIVE=mock
>> gradlew bootRun
```

http://localhost:8081

中国铁路客户服务中心

唯一网站。截止目前，没有授权其他网站开展类似服务内容，敬请广大用户注意。

2017年11月23日 星期四

首页 | 客运服务 | 货运服务 | 行包服务 | 旅游超市 | 车站引导 | 铁路常识 | 站车风采 | 客户信箱

最新动态

为保障您顺畅购票，请下载安装**根证书**。

- 公告 (2015-12-01)
- 关于调整互联网、电话订票起售时间的公告 (2014-11-26)
- 哈尔滨铁路局关于部分旅客列车临时调整运行的公告 (2017-07-28)
- 广铁集团公司关于2017年7月31日至8月27日期间增开部分旅客列车的公告 (2017-07-28)
- 昆明铁路局关于7月28日、29日K692/3/2次列车恢复开行的公告 (2017-07-28)
- 太原铁路局关于7月28日北京西开Z55、K1115/8次恢复开行等事项的公告 (2017-07-28)
- 昆明铁路局关于暑运期间增开部分旅客列车的公告 (2017-07-28)

更多>>>

全文搜索: 请输入搜索条件

我要发货 中国铁路货运电子商务平台

货物快运

主要营业站受理服务电话

货运运费查询

货物追踪

相关链接

中央政府门户网站

出发地	目的地	7月29日	7月30日	7月31日	8月1日
深圳	襄阳	605	1064	1190	1392
深圳	信阳	595	1392	1673	2130
深圳	西安	211	1762	1742	2028
深圳	济南	252	832	1153	1248

#3. Start mock service in Record mode

in another terminal

```
>> cd mock
>> ./record.sh
```

#4. Start mock service in playback mode

in another terminal

```
>> cd mock
>> ./playback.sh
```

SPRING BOOT DEMO ADMIN

<https://github.com/Mikuu/SpringBootDemoAdmin.git>

```
## start backend service
```

```
>> gradlew bootRun
```

SPRING BOOT DEMO FRONTEND

<https://github.com/Mikuu/SpringBootDemoFrontend.git>

#1. start Application with local config

```
>> export SPRING_PROFILES_ACTIVE=local  
>> gradlew bootRun
```

<http://localhost:8001/SpringDemo>

The screenshot shows a web application interface for managing users. It has a light gray background with a white form area. The form is titled 'User' and contains three input fields: 'Name' with placeholder text 'Enter your name', 'Age' with placeholder text 'Enter your Age.', and 'Salary' with placeholder text 'Enter your Salary.'. Below the input fields are two buttons: 'Add' (blue) and 'Reset Form' (orange). Below the form is a section titled 'List of Users' which contains a table with four columns: 'ID', 'NAME', 'AGE', and 'SALARY'. The table is currently empty.

#2. start Application with mock config

```
>> export SPRING_PROFILES_ACTIVE=mock  
>> gradlew bootRun
```

#3. Start mock service in Record mode

In another terminal

```
>> cd mock  
>> ./record.sh
```

#4. Start mock service in Playback mode

In another terminal

```
>> cd mock  
>> ./playback.sh
```

THANK YOU

ThoughtWorks®