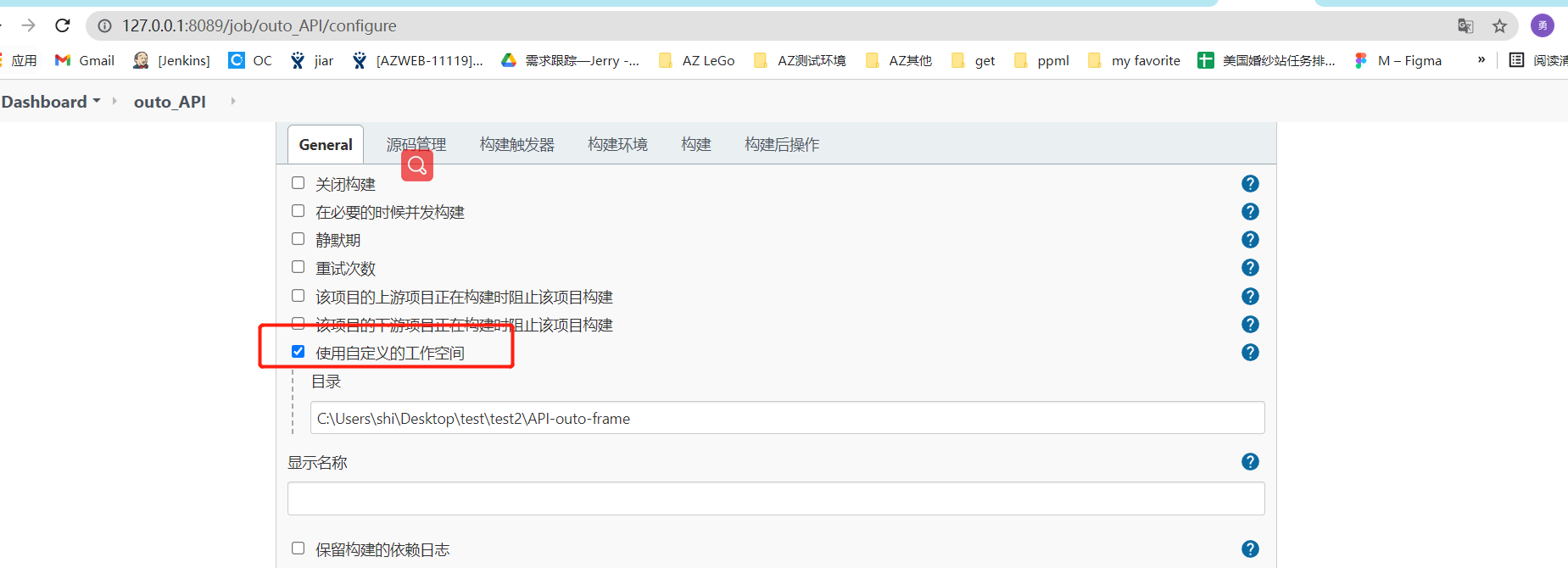
# Run Jenkins server

Java -jar Jenkins.war --httpPort=8089

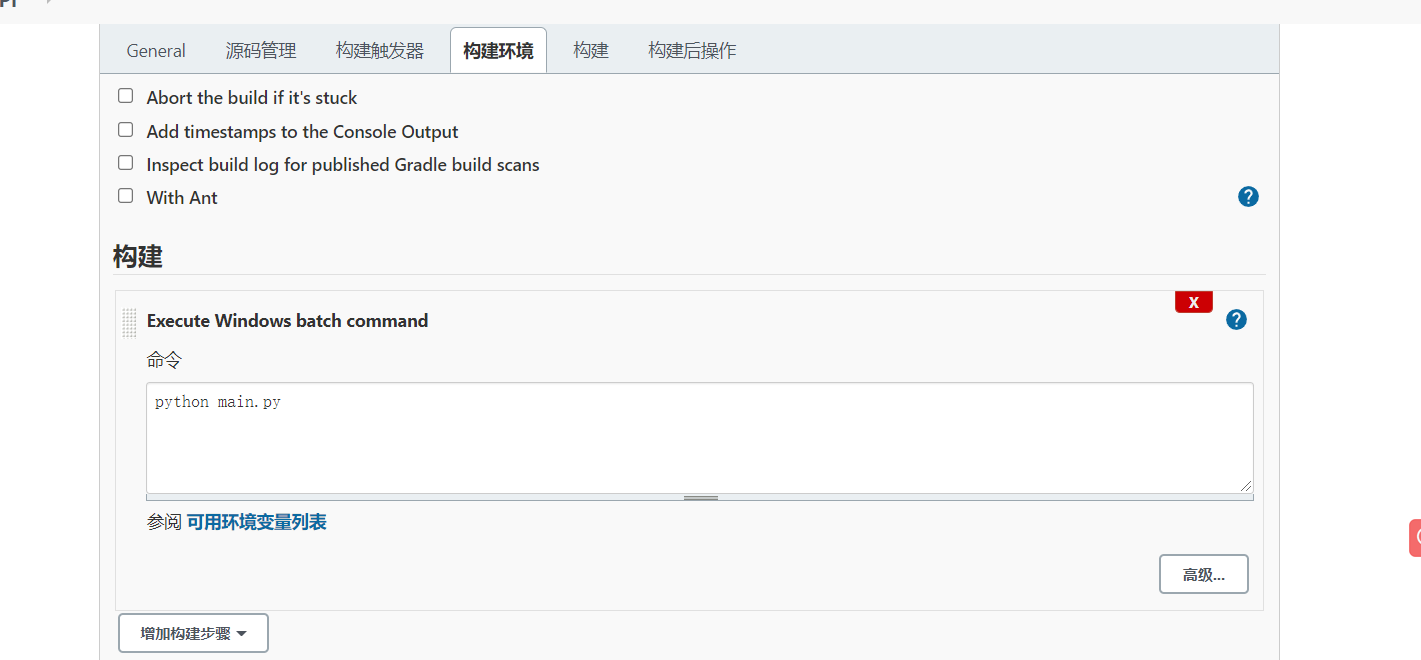
http://127.0.0.1:8089/



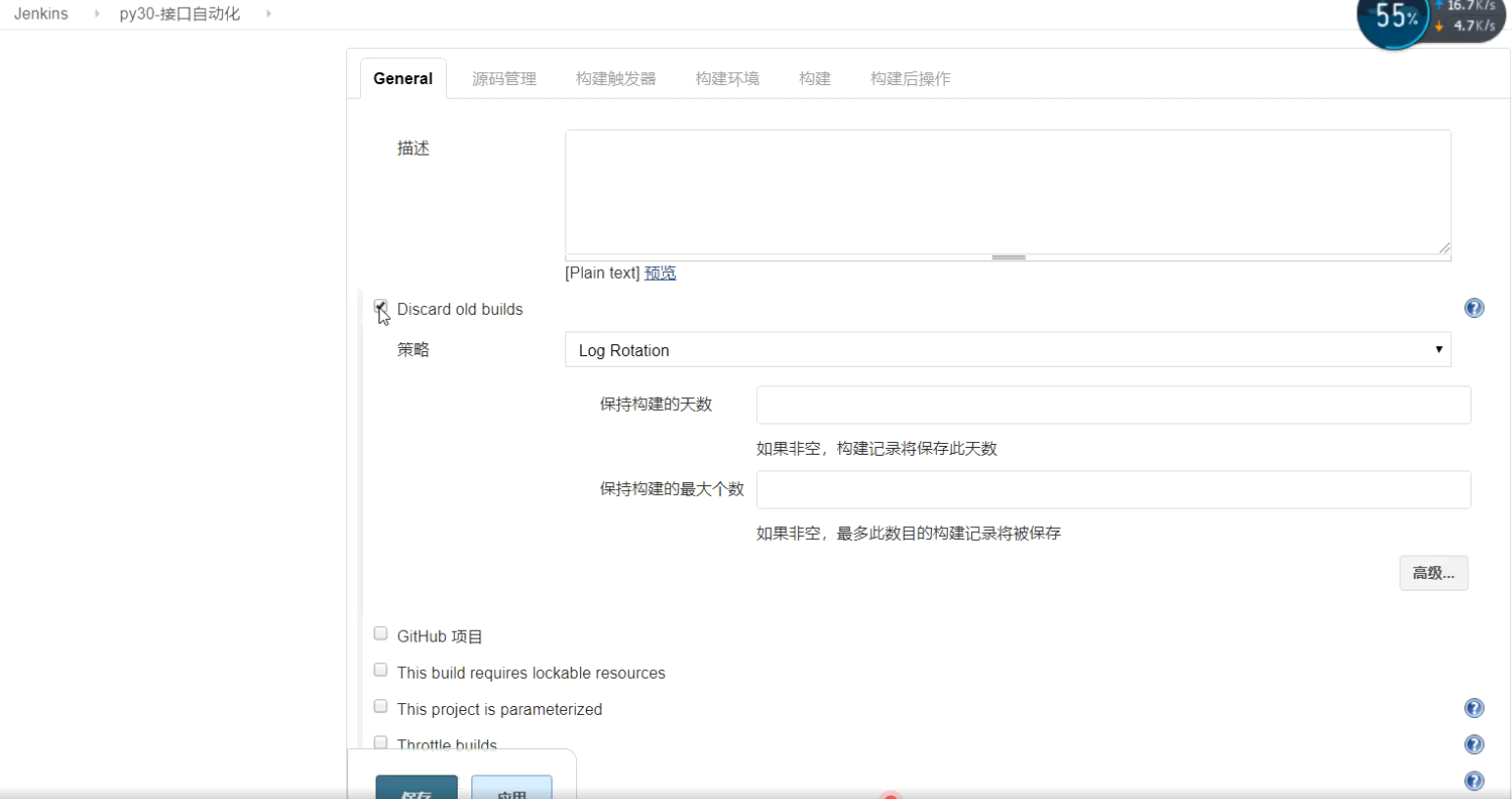
## 使用自定义空间



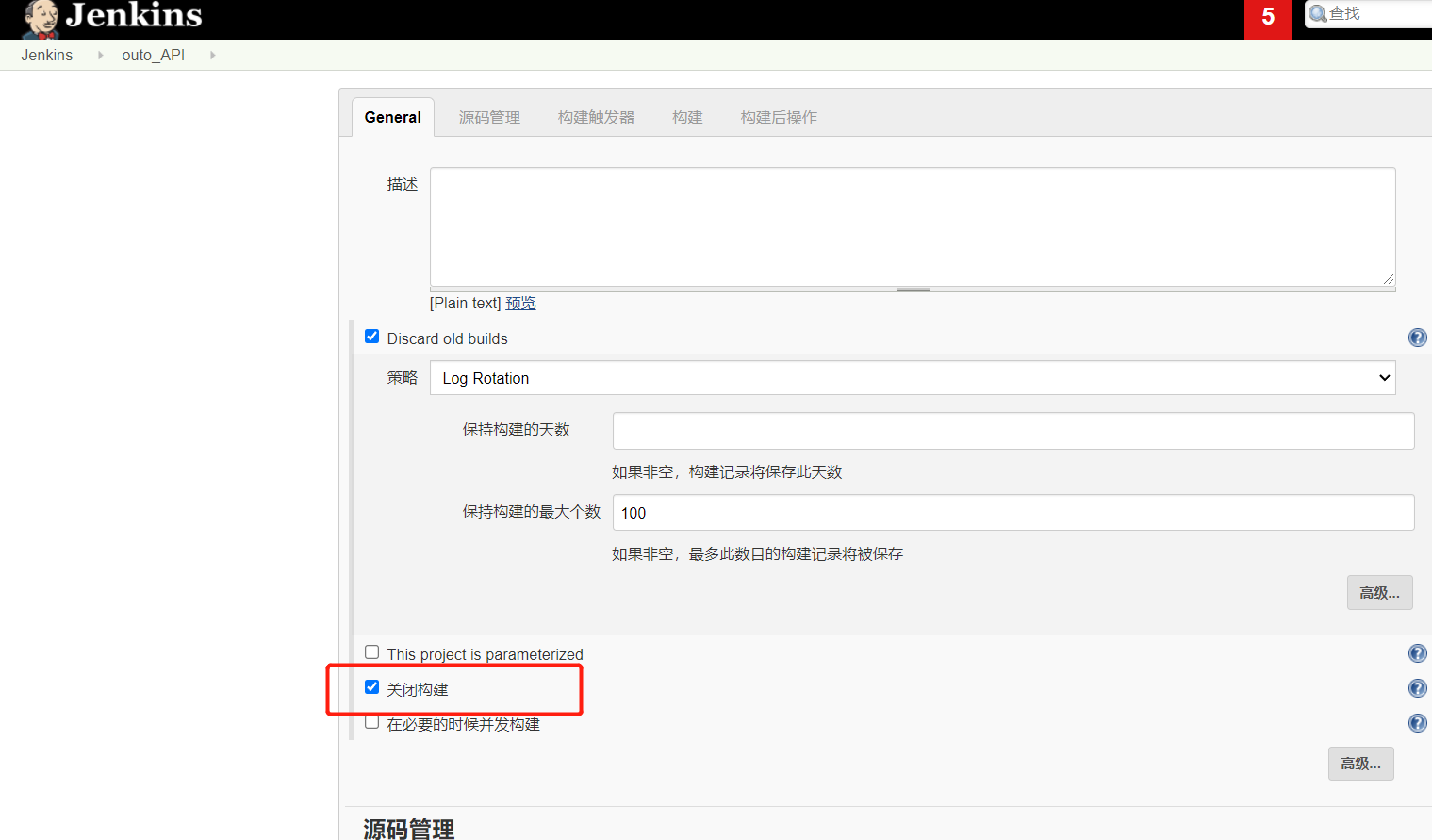
## 构建环境



## 管理、清除构建数据

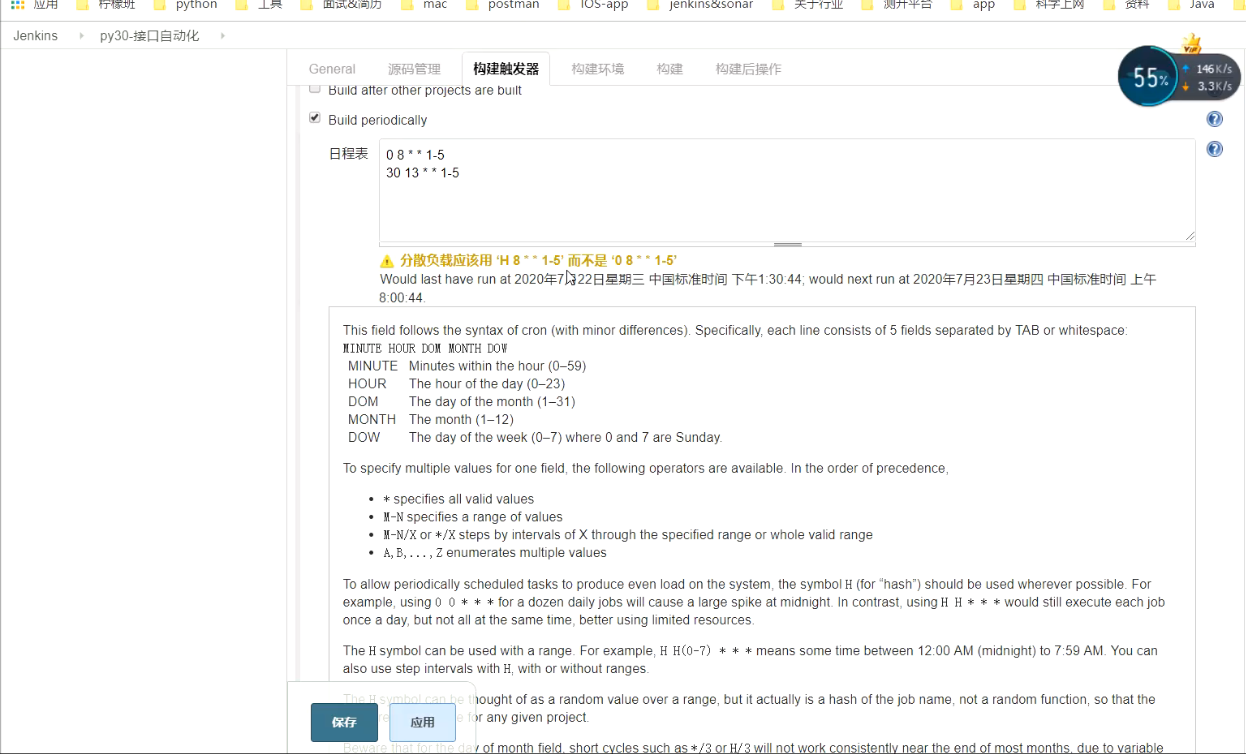


## 禁用job

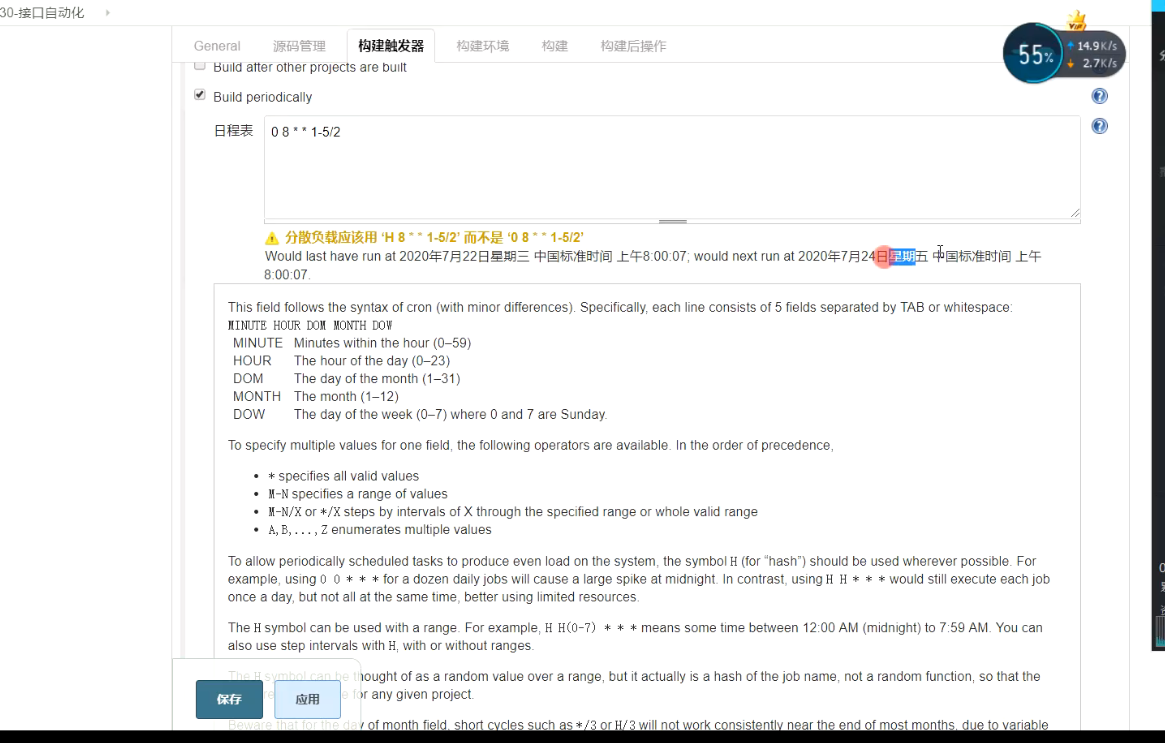


## 定时构建

周一至周五的 8点 和 下午1点

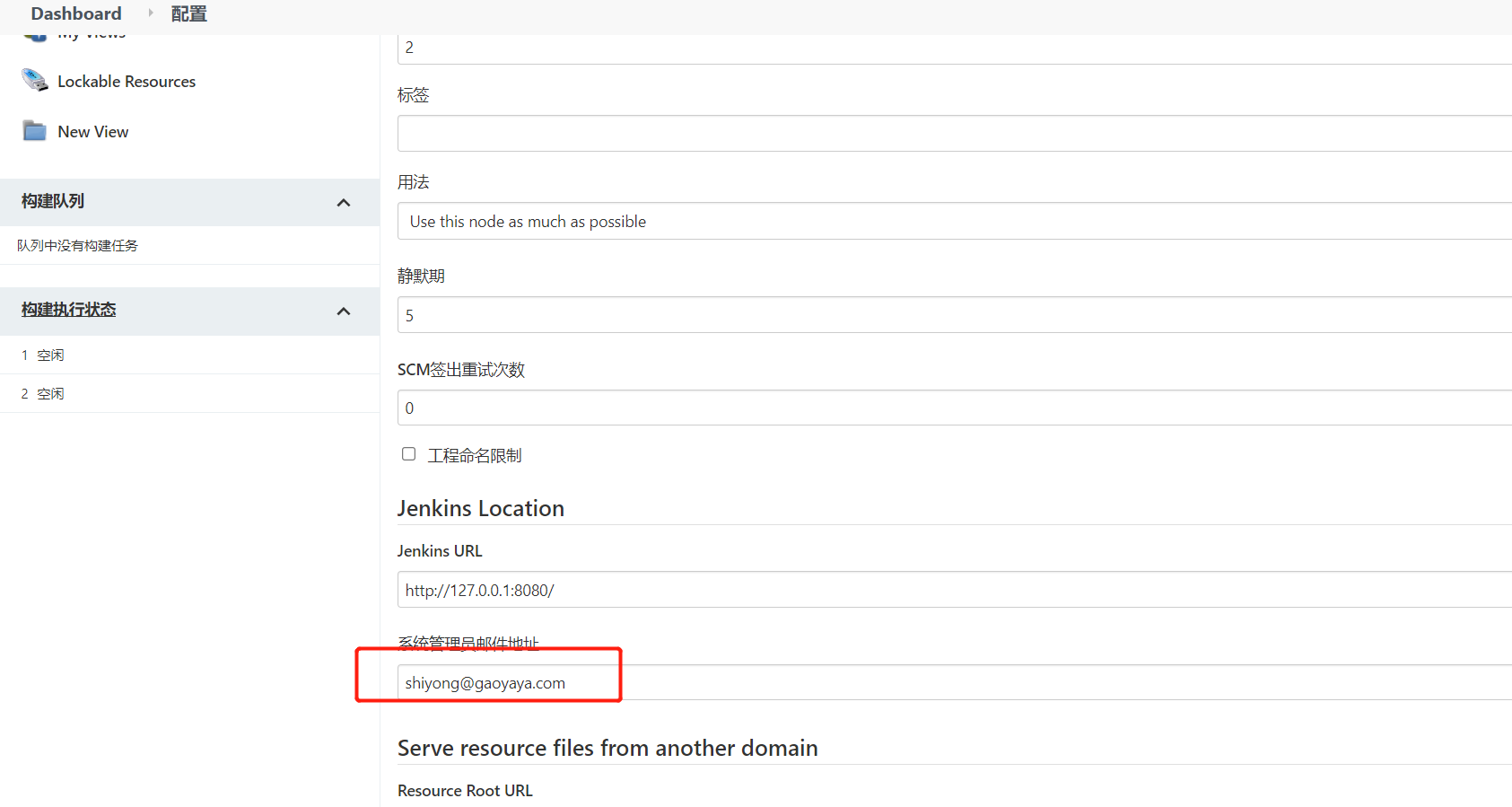


周三到周五



## 设置发件人邮件地址

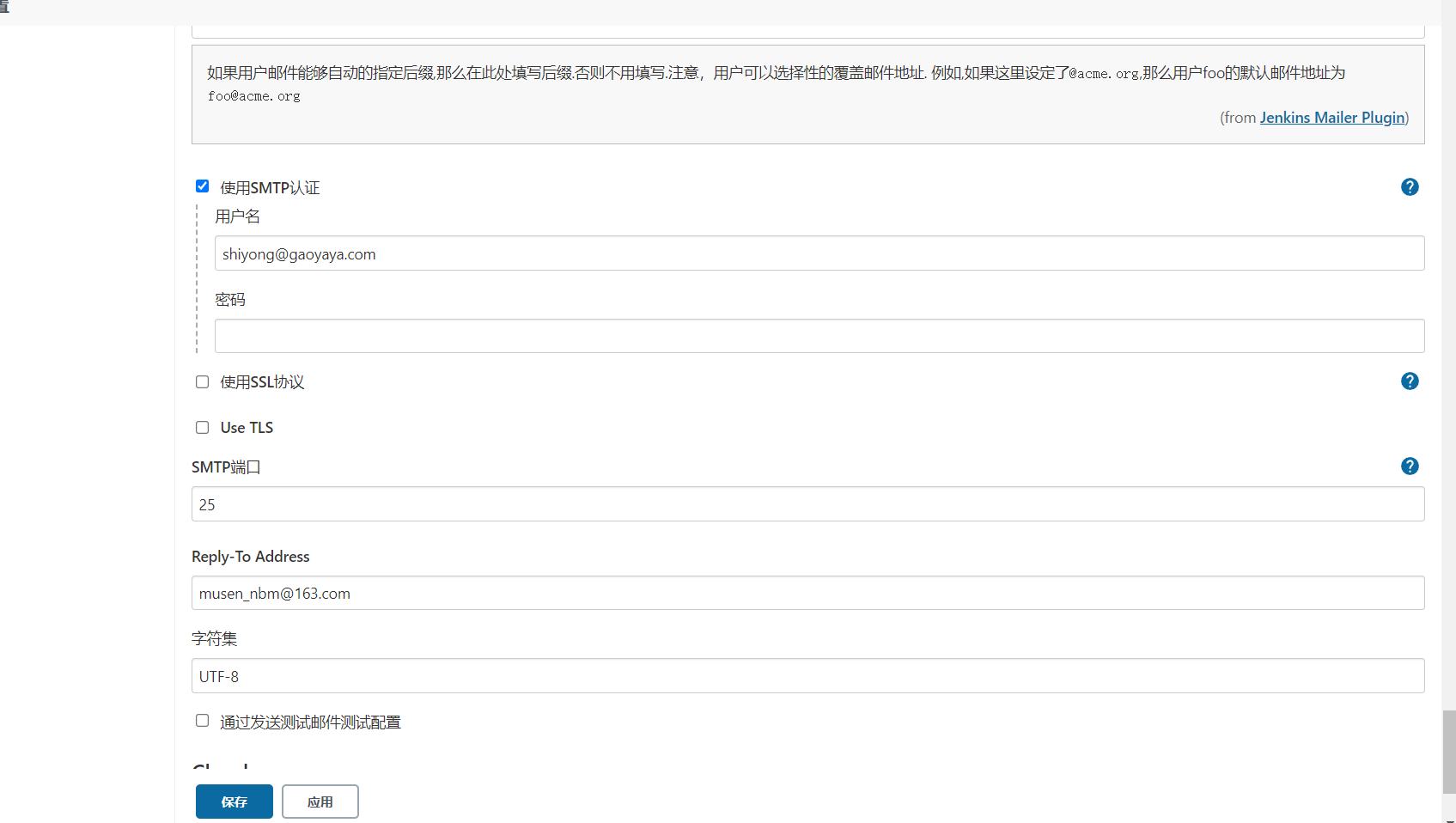
System config-jenkins locating



## 配置smtp邮件服务



## 配置第三方授权码



# 接口自动化

## 自动化理论

不能靠抓包来做接口自动化 费时间 字段要猜 辅助了解接口的业务

接口自动化是需要团队推动的

有些公司 自动化是跟迭代保持同步的（开发 实现接口逻辑的实现 ，测试开发接口自动化的用例 ，开发 接口开发完成后 跑下接口自动化用例看下有没有不通过的）

## 你是怎么管理自动化用例的

我搭建了一套测试框架，运用分层设计思想，数据驱动思想

## 框架分了哪些层

## Python json和字典的区别

Json 只是个格式化的字符串 只不过格式是以 key value形式的 是为了不同语言 数据交互定义的一种规范 ，字典是python的一种数据类型

编写自动化测试的前提 接口文档一定的要有 明确的正向输出 和一些关键的反向用例 返回什么都要了解

## 自动化的优点

自动化 效率高 节约时间

随着现在的项目越来越大 功能越来越多 手工已经点不赢了 不能保证产品的质量

## 自动化做些什么

接口自动化用于历史功能的回归

已经成熟稳定的模块需要做回归测试

## 自动化框架的好处

框架搭好后只需要写测试用例就可以了

## Swagger

：自动生成接口文档工具 很大公司再用 前提是开发人员一定要按照它的规则写 否则是生成不出来的

## 一天写几个接口？

要看接口的复杂程度，对接口的熟悉程度 前期是比较慢的 因为接口时要慢慢熟悉的 一天5个左右把

## 上家公司 写多少个接口自动化

50个以上的接口最好

## 接口自动化覆盖功能用例的多少合适

覆盖接口70%的用例 就比较可以的了

## 怎么管理你的参数的

使用excel数据存储

、

使用excel 存储数据做到 代发 数据分离的作用，好维护格式清晰

## 给你一个项目你怎么开展接口测试

### 1.需求分析-项目周期-核心功能-

需求分析 -了解/熟悉业务 ,明确项目的现状（哪些 项目/模块 适合做自动化测试）- 项目周期短的不适合 比如一个项目周期就半年做完就不做了（项目结束） 点完得了 没必要 )，区分 历史功能/稳定功能/新的功能/线上历史问题最多功能模块

### 2接口的了解 -接口文档

需要开发提供接口文档 不要求一次提供完成每周 提供20个 或者十几个，找个技术对接下 有问题问下

### 3.自动化框架/工具选择

Python ，第三方平台，jmeter，soapui， 选择框架要确保 公司最复杂的接口是能做的

框架结构

运行流程

#### 4.写接口用例

尽早做jenkins集成

定期汇报

自动化测试方案

测试报告-分析用例失败原因

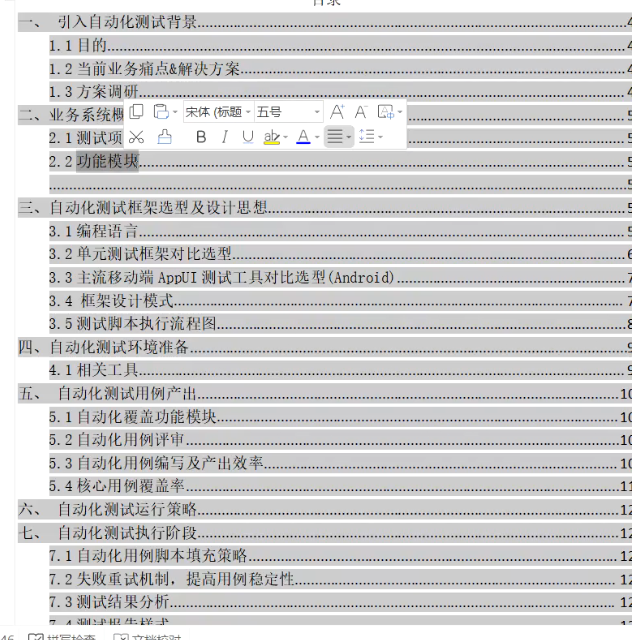
从开始做自动化 到一直做自动化 要记录自己发现的bug 写明日期

5.维护阶段

开发修复接口 同步修改用例

新增接口

遇到问题优化框架

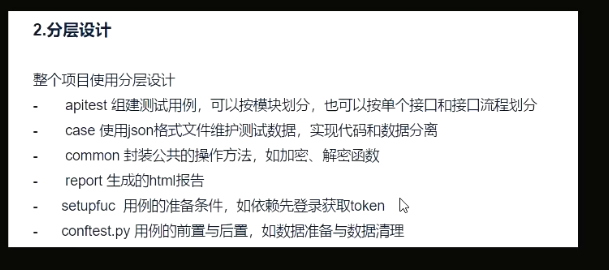


## 自动化测试方案？

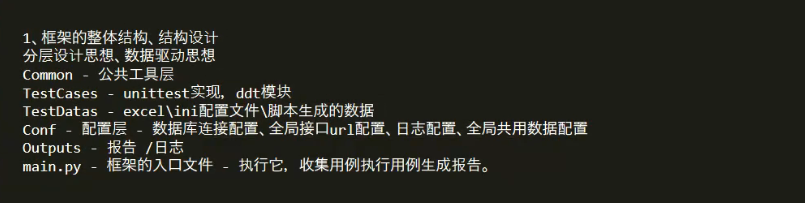
## 自动化框架提供了哪些功能？

1.编写用例（断言，前置，后置），执行用例，生成测试报告，日志输出，回溯问题

## 框架的整体结构时什么？



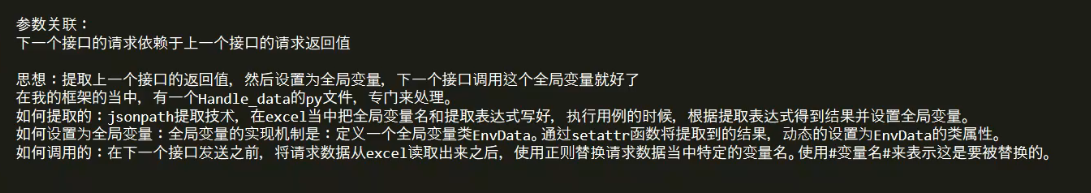
## 框架整体结构、结构设计



### 针对接口的处理

1. 数据库处理
2. 接口关联处理
3. 数据参数化（数据驱动思想）实现excel数据读取
4. 提取数据处理
5. 数据替换处理
6. 配置文件处理
7. http请求二次封装：鉴权处理，请求头处理，
8. 数据生成处理

### 参数关联

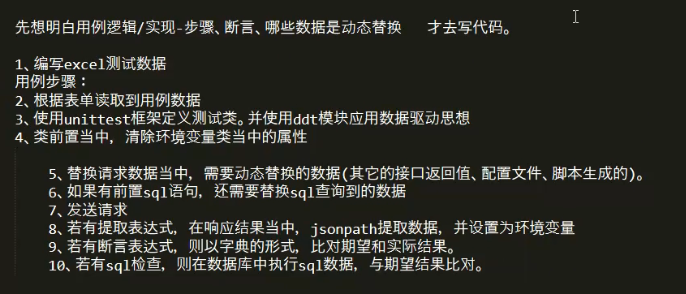


### 怎么保证用例的覆盖

深入理解需求

用例评审\讨论

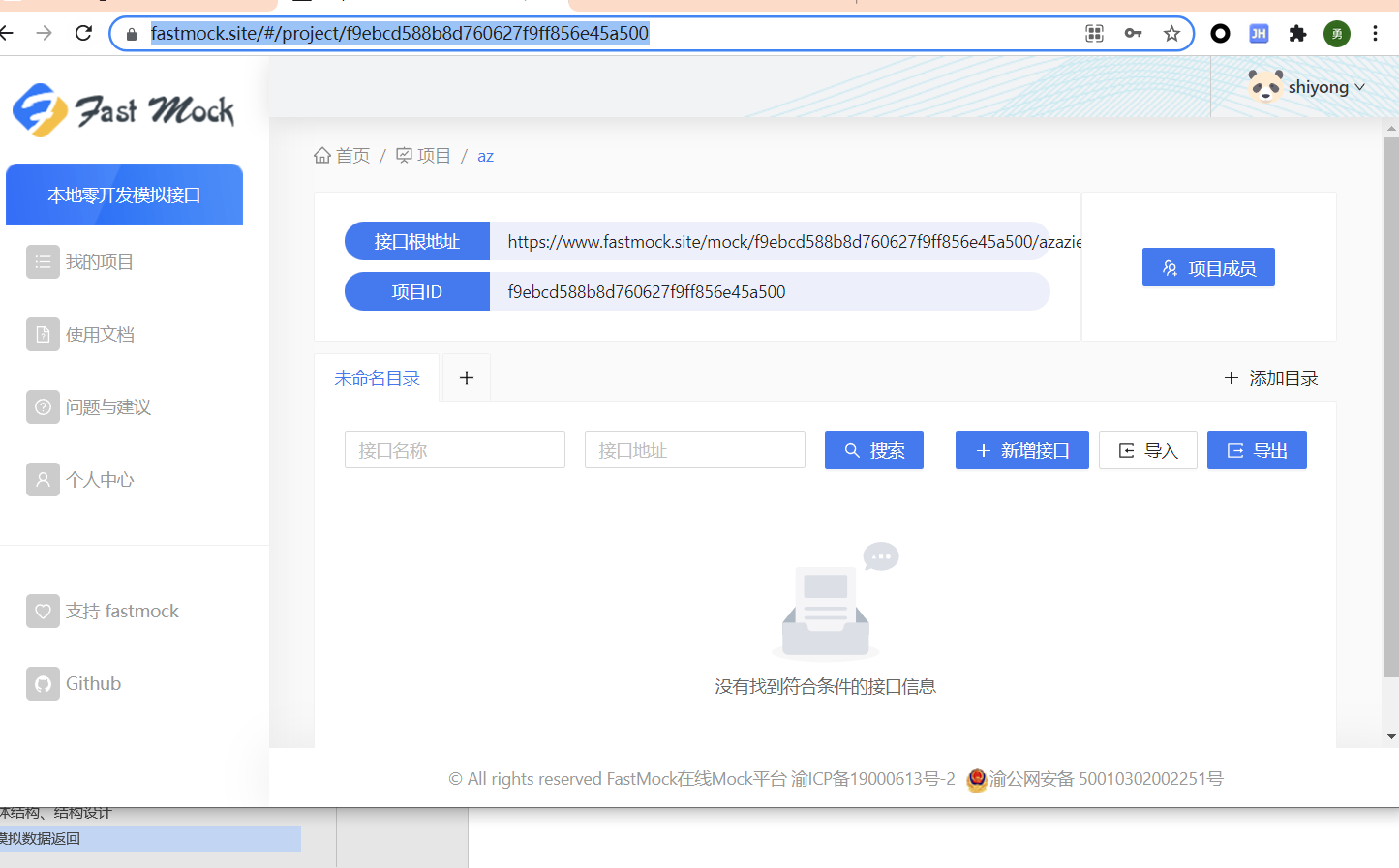
### 编写用例的过程



## Moke模拟数据返回

## Fast mock网站

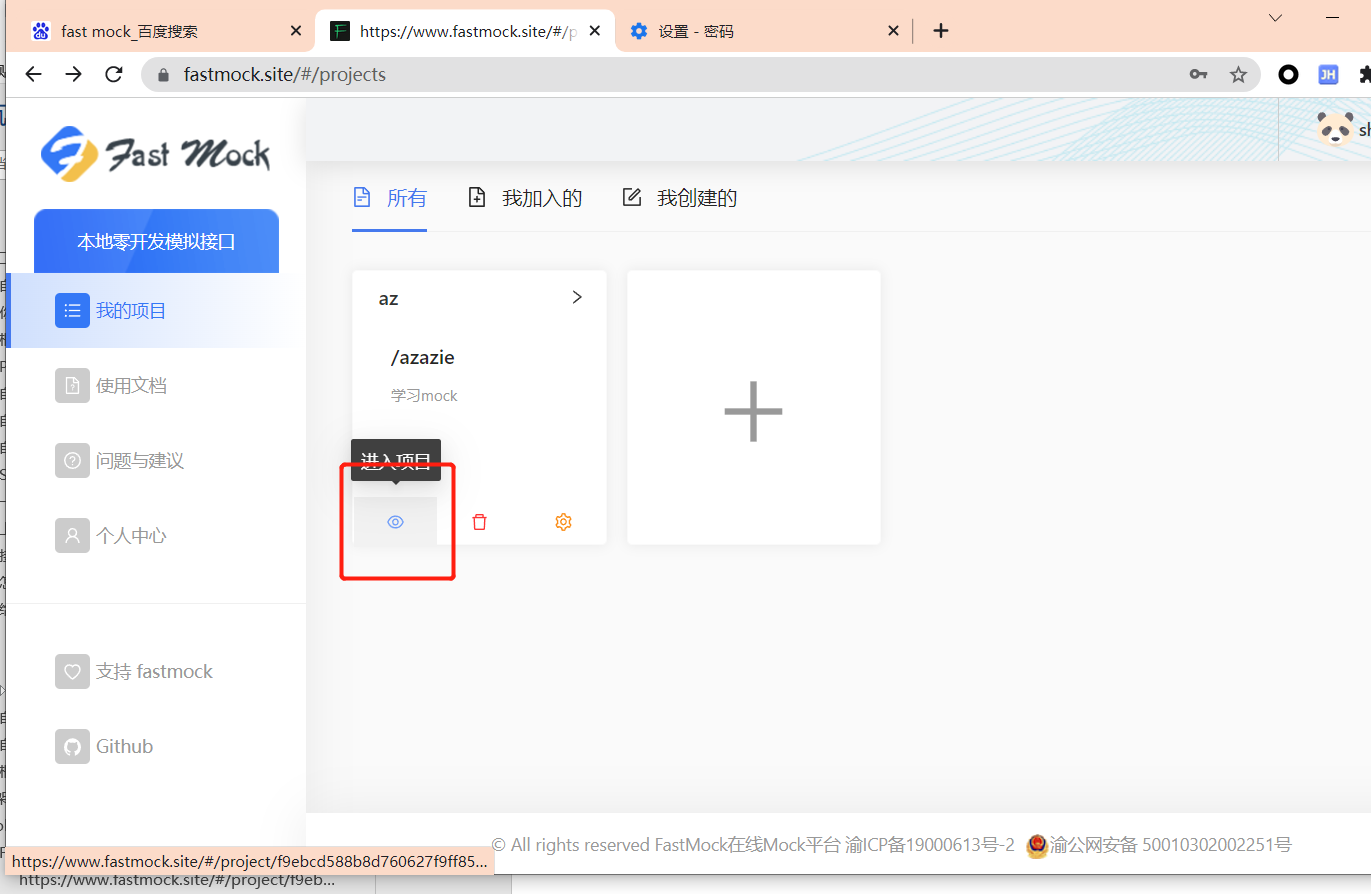
<https://www.fastmock.site/#/project/f9ebcd588b8d760627f9ff856e45a500>



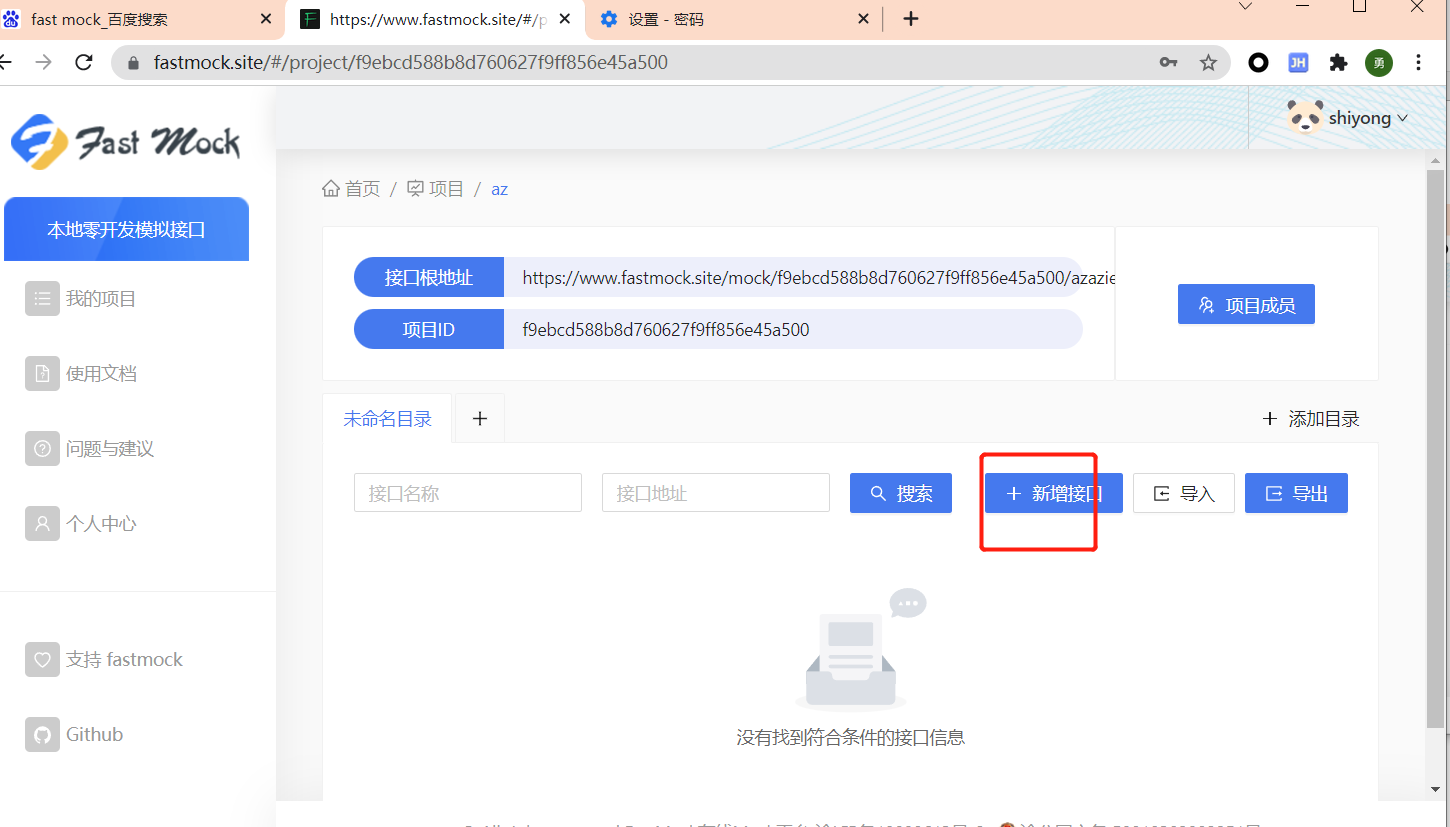
### 43秒

62-55.48 41.40

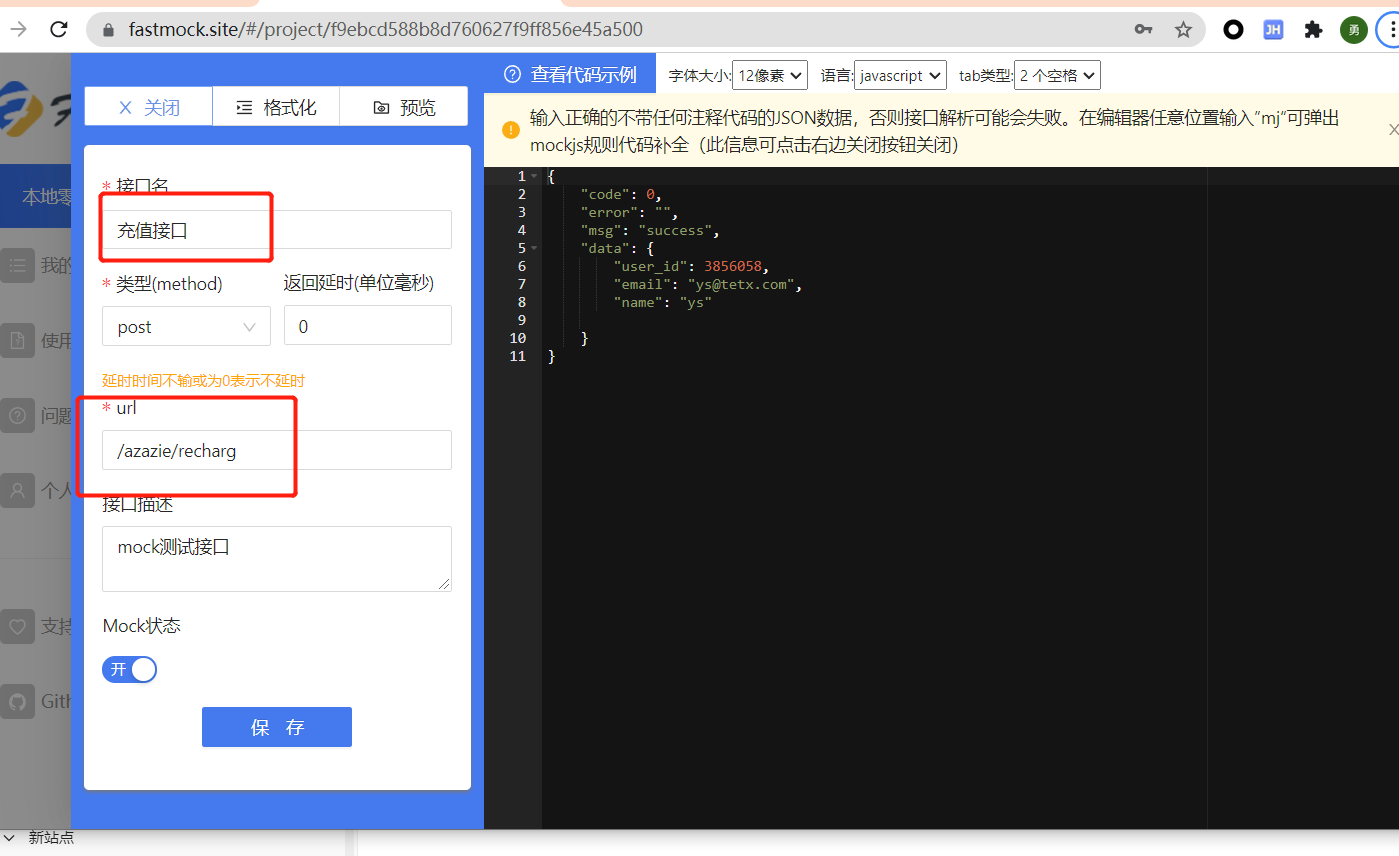
### 进入项目



### 新增接口

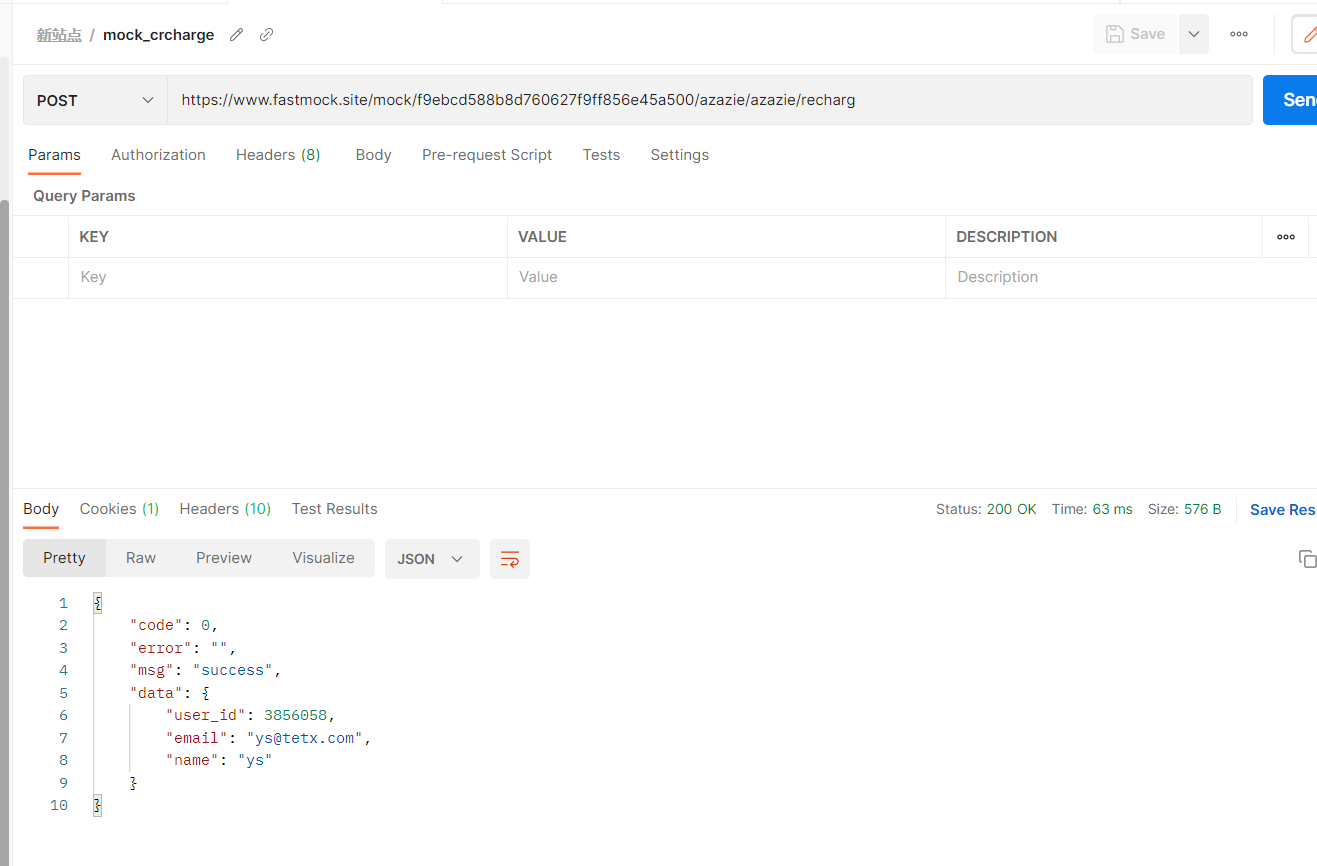


### 编写接口



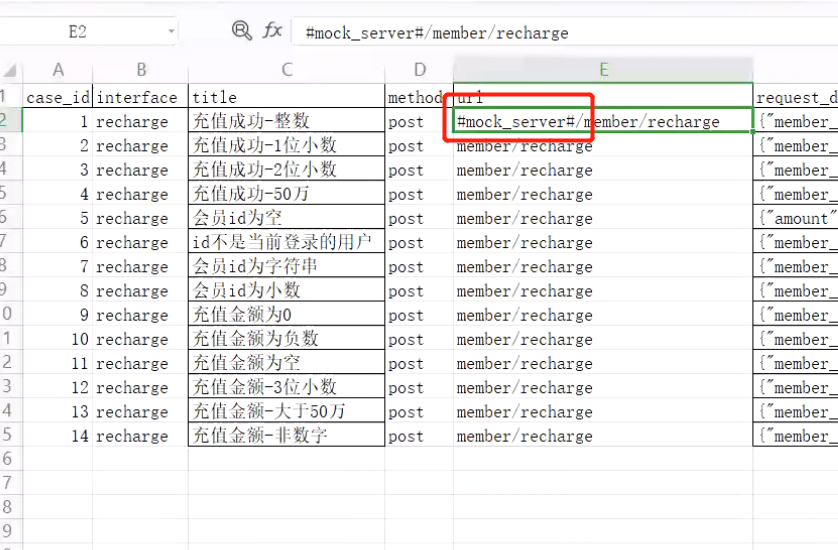
调用接口

跟地址 加接口地址

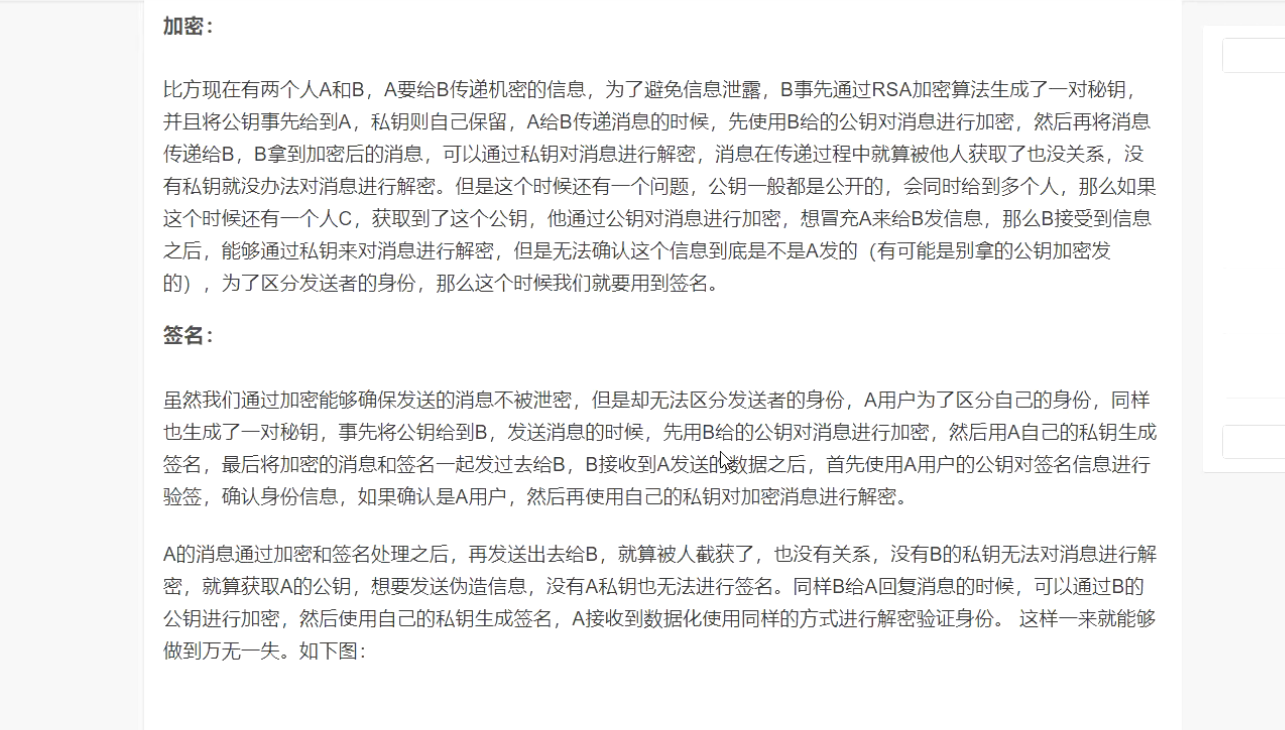


### Mock 接口应用到框架

url也要判断



## 接口ras 加密



自动化造数据

1. 创建一个filtergoos类
2. 创建一个调用列表页接口返回整个列表商品数据的方法

这个方法会遍历接口中的商品数据列表 字典中取出商品goodsId catId goodsName

把商品数据 生成新字典 添加到列表中返回出来

1. 调用列表页接口

Api

## Common

### Handle\_excel

导入 openpyxl import load\_workbook

创建excel对象 传入文件路径

选择表单 获取shell数据

方法1：遍历 excel title 获取casetitle

方法2：遍历 除title行 所有行，再遍历每一行取出值 添加如list

调用title方法获取title +vale 返回字典

### Handle\_log

日志输出处理

创建日志类handlelog类 继承logging第三方库

对象属性初始化 logname,level,format,file

创建日志收集器 返回收集器对象

设置收集器级别

创建渠道

设置日志收集格式

渠道绑定 日志格式

收集器绑定渠道

If log对象 file属性有值就创建文件输出渠道

渠道绑定日志格式

设置日志收集级别

添加渠道

### Handle\_request

二次封装request 对请求做处理

## Case\_data

Excel参数化内容

Case\_name, method, url, case\_data, header, expect, check\_sql

Case data关键字替换处理

Case data 提取 处理

## mypath