# 故障演练平台

一、平台现状：

### 1.1 现状：

目前的故障演练平台支持appCode+noah环境维度的consumer端调用接口方法级别的RuntimeException异常和超时的故障注入。依赖于noah环境手动执行流程完成故障演练。

### 1.2 问题：

~~手动执行效率较低~~

故障注入涉及的流程需要人工梳理，耗时

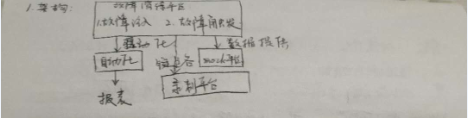
异常单一，不能满足各种异常的模拟

## 二、故障演练平台

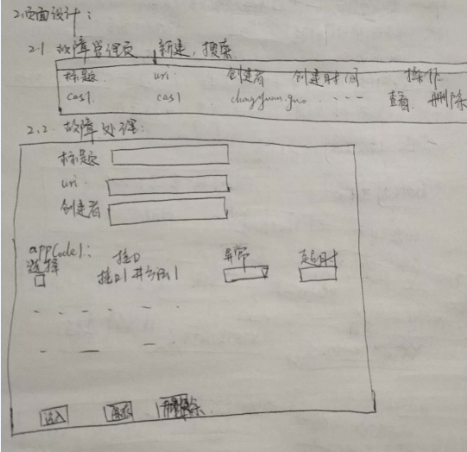
### 2.1 目标：

实现链路维度特定异常或超时的故障注入（涉及dubbo、http）~~，自动演练并将结果形成文档，方便后续跟进解决强弱依赖问题。~~

2.2 系统架构：



### 2.3 交互设计



注：~~最终效果依赖于自动化平台，~~测试过程可手动完成。

## 三、功能点划分

|  |  |  |
| --- | --- | --- |
| 功能 | 实现细节 | 完成时间 |
| 方案确定 | 故障演练平台方案确定 | 2018.12.26前 |
| 故障演练功能完善 | agent底层迁移 – dubbo |  |
| http接口支持 |  |
| 特定异常支撑 |  |
| 故障注入可视化页面 | 链路以appCode维度进行展示（后端+前端） |  |
| 故障注入新增 |  |
| 故障注入删除 |  |
| 故障注入 | 故障注入对应项目 |  |
| 故障注入管理 | 故障注入管理页面 |  |
| mock平台联调 | 故障演练和mock平台联调 |  |
| ~~自动化联调~~ | ~~故障演练和自动化联调~~ |  |

### 四、已有接口待确认

|  |  |  |
| --- | --- | --- |
| 录制平台 | 录制平台的链路数据 | 调研结果 |
| 自动化平台 | 自动化平台的触发机制 | 不可用 |
| mock平台现有功能 | appCode+环境+接口+方法级别的异常、超时的注入接口 | 支持 |
| 接口信息自动同步 | 支持 |
| 本次开启关闭上一次开启 | 支持 |
| 演练过程的记录 | 支持 |