

基于RestAssured等整合的一套API自动化测试框架。

8 commits

1 branch

0 packages

0 releases

1 contributor

Branch: master

New pull request

Create new file

Upload files

Find file

Clone or download

MeYoung Update README.md ...

Latest commit f9bdb7a on 19 Mar

src	测试报告中添加测试步骤	4 months ago
.gitignore	第一次提交	5 months ago
README.md	Update README.md	2 months ago
pom.xml	添加RPC自动化测试	4 months ago

README.md

帮助文档地址：<https://www.yuque.com/miyang.meyoung/luban> git地址：[https://github.com/MeYoung/Luban\\_API\\_Test](https://github.com/MeYoung/Luban_API_Test)

## 优点

- case分级 (blocker, critical, normal, minor, trivial)
- 数据驱动
- 简化数据库交互
- 参数化 (多环境支持)
- 对象对比、两个Json对比
- 指定需要运行的case (冒烟, 回归, 全量等。。。)
- 详细的测试报告, 可以回溯每个case的历史执行记录, 并报告展示每个请求请求
- 支持任意次数的失败重跑
- 支持多种并发方式, 自由配置的并发数
- 自由不受限, 易扩展
- 支持Dubbo RPC测试
- 支持报告自动记录测试步骤
- 等等

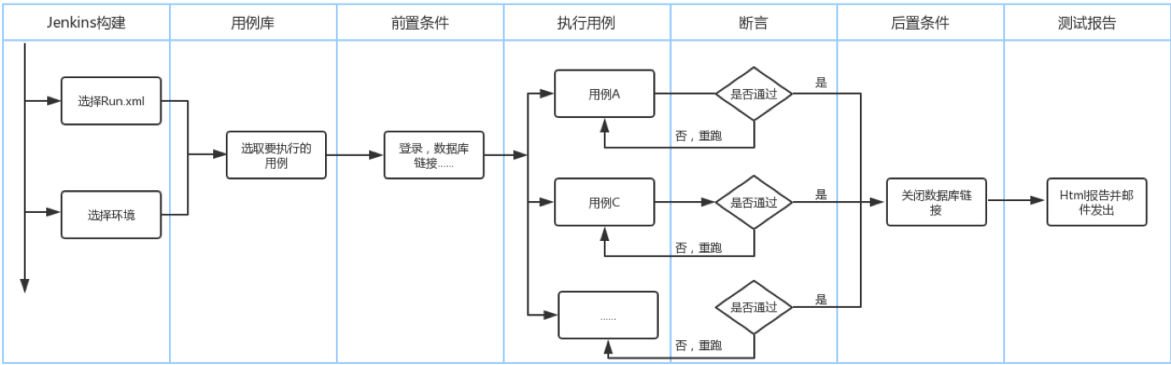
## 缺点

- 上手难度相对较大

## 技术架构

- API调用框架: RestAssured、Dubbo
- Case管理: TestNG
- Json解析: fastjson、JsonPath
- Log打印: slf4j logback
- 持久层框架: Mybatis 、Mapper
- 测试报告: allure2
- CI: jenkins 工程

## 整体流程



Jenkins参数化构建

工程 Dapi\_Test

This build requires parameters:

CaseFile 

dapi/RunDapi.xml

选择运行的case

env 

test

环境选择

UserName 

2019060071

PassWord 

Zc123456

cid 

999100

platform 

android

选择平台

Build

Run.xml配置

```
<!--配置Case集合名 开发方式 开发个数-->
<suite name="All Test" parallel="tests" thread-count="3">

    <parameter name="env" value="{env}"/>
    <parameter name="userName" value="{userName}"/>
    <parameter name="password" value="{password}"/>
    <parameter name="cid" value="{cid}"/>
    <parameter name="platform" value="{platform}"/>

    <test verbose="1" name="Store 模块" annotations="JDK">
        <packages>
            <package name="com.lucky.dapi.store"/>
        </packages>
    </test>

    <test verbose="2" name="member 模块" annotations="JDK">
        <classes>
            <class name="com.lucky.dapi.member.LoginTest">
                <methods>
                    <!--排除失败的用例-->
                    <exclude name="loginPassError"/>
                    <!--要跑的用例-->
                    <include name="loginPassEmpty"/>
                </methods>
            </class>
        </classes>
    </test>

    <test verbose="3" name="postman 模块" annotations="JDK">
        <packages>
            <package name="com.lucky.dapi.postman"/>
            <package name="com.lucky.dapi.base"/>
        </packages>
    </test>

    <listeners>
        <!--失败重跑-->
        <listener class-name="com.lucky.common.testing.RetryListener"/>
    </listeners>
</suite>
```

前置条件

```

@Parameters({"env", "userName", "password", "cid", "platform"})
@BeforeSuite
public void dLogin(String env, String userName, String password, String cid, String platform) {
    读取配置文件
    EnvReaderUtils envReaderUtils = new EnvReaderUtils( fileName: env + ".properties");
    String cellerCode = envReaderUtils.getValue( key: "dapi.cellCode." + platform);
    String apiVersion = envReaderUtils.getValue( key: "dapi.apiVersion." + platform);
    String secKey = envReaderUtils.getValue( key: "dapi.secKey." + platform);
    String origin = envReaderUtils.getValue( key: "dapi.origin." + platform);
    String deviceId = envReaderUtils.getValue( key: "dapi.deviceId." + platform);
    String baseUrl = envReaderUtils.getValue( key: "dapi.url.base");
    String dbEhr = envReaderUtils.getValue( key: "db.ehr");
    String dbOperation = envReaderUtils.getValue( key: "db.operation");
    String dbDispatch = envReaderUtils.getValue( key: "db.dispatch");
    String loginPath = DapiPath.LOGIN;
    // 加密方式请求
    MApiClient client = Dlogin( loginUrl: baseUrl + loginPath, userName, password, deviceId, origin, cellerCode, apiVersion, secKey);
    String uid = client.getUid();

    数据链接
    SqlSession sessionOperation = SQLExecuteUtils.run(dbOperation);
    SqlSession sessionEhr = SQLExecuteUtils.run(dbEhr);
    SqlSession sessionDispatch = SQLExecuteUtils.run(dbDispatch);
}

```

## Case

```

@Slf4j
public class PositioninfoTest extends DRequest {
    String apiPath = DapiPath.BASE_POSITION_INFO;
    PositioninfoParam positioninfoParam = new PositioninfoParam();

    TBaseCityService tBaseCityService=new TBaseCityService();

    @Severity(SeverityLevel.BLOCKER)
    @Test(description = "获取请求IP和位置信息")
    public void positioninfoTest() {
        // 从数据库查询一条数据
        TBaseCity tBaseCitySql=tBaseCityService.getXiamenInfo();
        positioninfoParam.setLon(tBaseCitySql.getLongitude());
        positioninfoParam.setLat(tBaseCitySql.getLatitude());

        // 发起请求
        String reqStr = dPost(apiPath, positioninfoParam);

        // 结果校验
        assertSuccessDapi(reqStr);
        JsonPath jsonPath=new JsonPath(reqStr);
        log.info(tBaseCitySql.toString());
        AssertUtils.assertEquals(jsonPath.getString( path: "content.cityCode"),String.valueOf(tBaseCitySql.getId()));
        AssertUtils.assertEquals(jsonPath.getString( path: "content.cityName"),tBaseCitySql.getName());
    }

    @Data
    class PositioninfoParam {
        /**
         * 经度
         */
        private String lon;
        /**
         * 纬度
         */
        private String lat;
    }
}

```

用例描述和用例级别

数据库动态获取数据

发起请求

结果和数据库校验

入参

## 后置条件

```

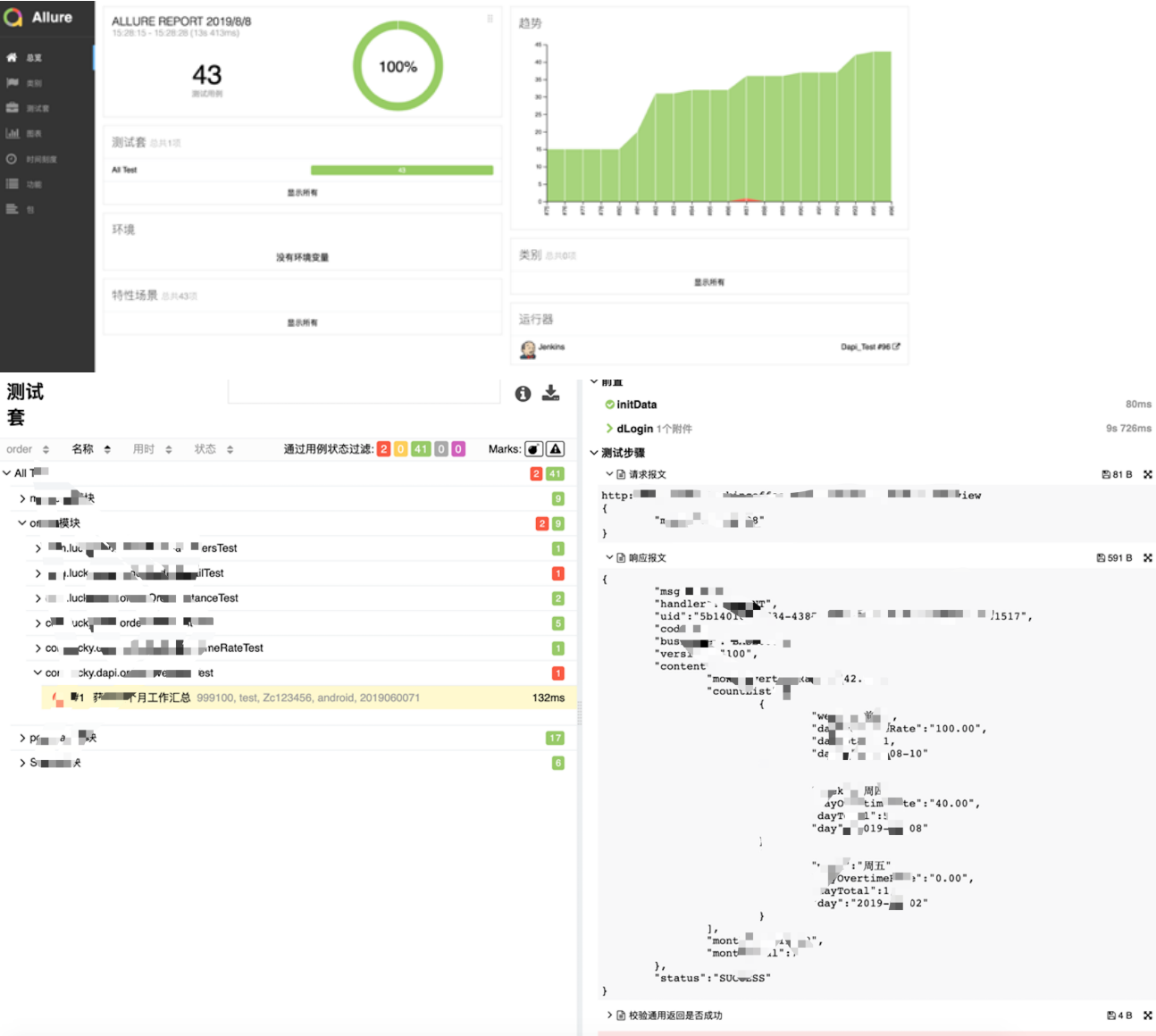
@AfterSuite
public void closed() {
    BaseData baseData = BaseDataUtils.getBaseData();
    if (baseData.getSessionOperation() != null) {
        baseData.getSessionOperation().close();
    }

    if (baseData.getSessionDispatch() != null) {
        baseData.getSessionDispatch().close();
    }

    if (baseData.getSessionEhr() != null) {
        baseData.getSessionEhr().close();
    }
    log.info("关闭资源完成");
}

```

## 报告和邮件



Dapi\_Test测试报告 - Successful

构建信息

- 项目名称：Dapi\_Test
- 构建编号：第93次构建
- GIT 版本：29596084eafcaa29d2dc9d0df74814b506d49c52
- 触发原因：Started by timer
- 项目 Url：[http://10.104.130.97:8080/jenkins/job/Dapi\\_Test/](http://10.104.130.97:8080/jenkins/job/Dapi_Test/)
- 构建日志：[http://10.104.130.97:8080/jenkins/job/Dapi\\_Test/93/console](http://10.104.130.97:8080/jenkins/job/Dapi_Test/93/console)
- 构建 Url：[http://10.104.130.97:8080/jenkins/job/Dapi\\_Test/93/](http://10.104.130.97:8080/jenkins/job/Dapi_Test/93/)
- 生成HTML报告：[http://10.104.130.97:8080/jenkins/job/Dapi\\_Test/93/allure](http://10.104.130.97:8080/jenkins/job/Dapi_Test/93/allure)

Changes Since Last Successful Build:

- 历史变更记录：[http://10.104.130.97:8080/jenkins/job/Dapi\\_Test/changes](http://10.104.130.97:8080/jenkins/job/Dapi_Test/changes)

Changes for Build #93:

[shijin.huang]  
修改模板

[shijin.huang]  
add orderList

No tests ran.

构建日志 (最后 100行):

```
[...truncated 168.74 KB...]
  "orderList": {
    },
  },
  "status": "SUCCESS"
}
==> Parameters: 2019-08-07 00:00:00(String), 2019-08-07 23:59:59(String), 808614(Long),
245665(Integer), 70(String)
<== Total: 0
2019-08-07 07:48:51 | INFO | TestNG-tests-1 | com.lucky.client.DRequest | 请求URL:
http://dapiptest.luckincoffee.com/resource/d/order/orderOverTimeRate
2019-08-07 07:48:51 | INFO | TestNG-tests-1 | com.lucky.client.DRequest | DAPI Get
```