**CART\_**接口自动化方案

自动化方案变更维护

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **序号** | **变更（+/-）说明** | **作者** | **版本号** | **日期** | **批准** |
| 1 | 草稿 | 魏建星 | V1.0.0 | 2016/12/05 |  |
| 2 |  |  |  |  |  |
|  |  |  |  |  |  |

**接口调研pom**

 <dependency>

                  <groupId>com.gome.pangu</groupId>

                  <artifactId>pangu-trading-client</artifactId>

                  <version>1.0.4-SNAPSHOT</version>

  </dependency>

**被测接口清单**：

**参考《购物车接口表》**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 接口名称 | Method数量 | 计划开始日期 | 计划结束日期 | 进度 | 问题 |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

购物车接口调用依赖关系

Cart trading

上层调用接口

下流依赖接口

Member

Invoice

Address

Checkout

Identificateion

ShortMessageService

GomeStoreShipping

Seller

Mobile service

Payment

Dragon

PointsClient

ShippingContainer

Keyt

GomeDo

VirtualAccount

Coupon

Price

ShoppingCartQuery

NPOP

CartPromotion

**MergeCart**

ShoppingCartCheck

Product

HTTP CART

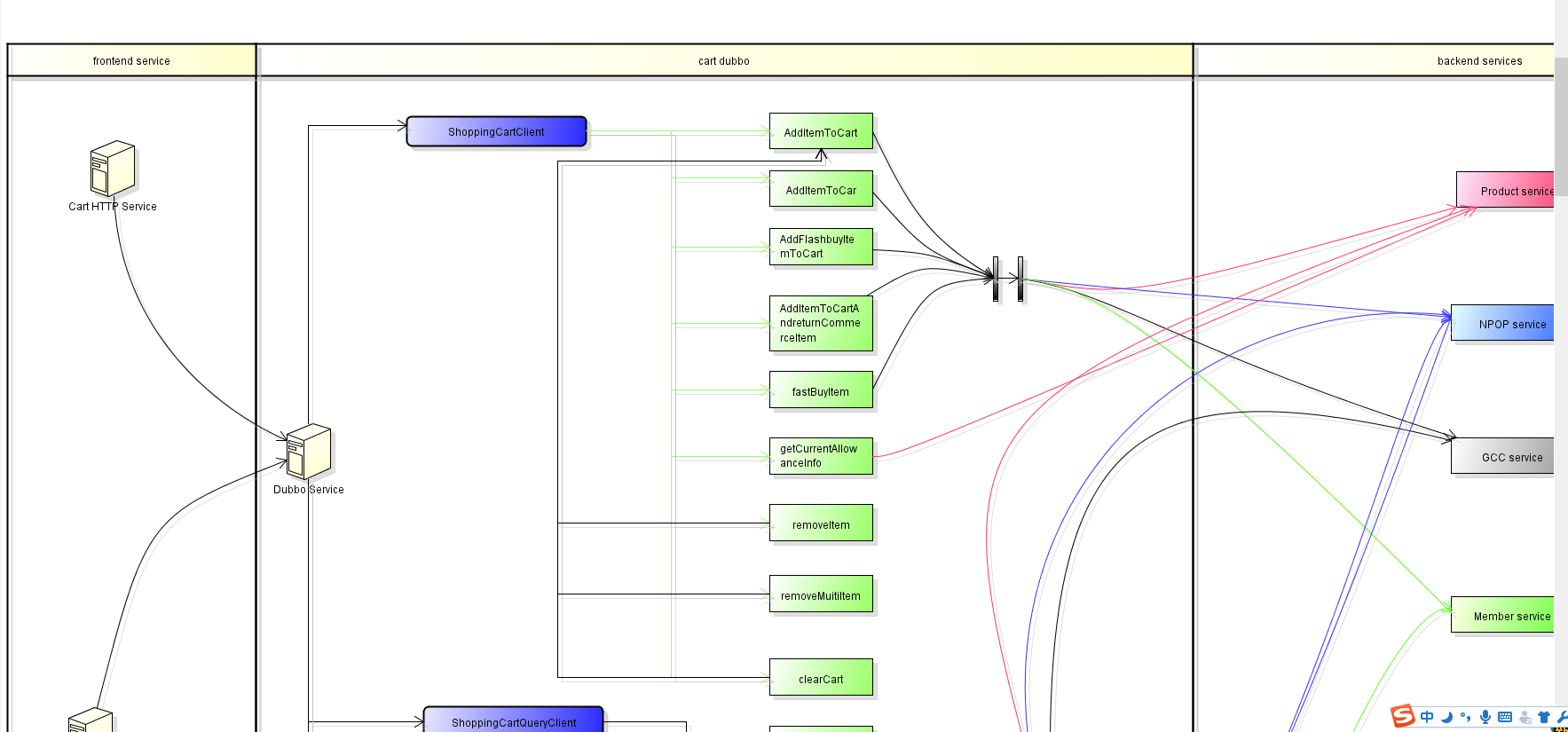
SafetyVerification

ShoppingCart

GCC

**CommitOrder**

Model Flowchart ：



**里程碑规划**

**//TODO**

## 一自动化脚本调用流程设计

### 1.1基本代码结构及用例维护

测试用例

测试脚本自动生成插件

CART被测接口服务

异步验证

同步验证

验证工具类 backend

接口方法调用适配器

TestNG 测试脚本

测试平台CI接口

测试业务验证

构建验证并生成测试报告

#### 1.1.1自动化数据维护

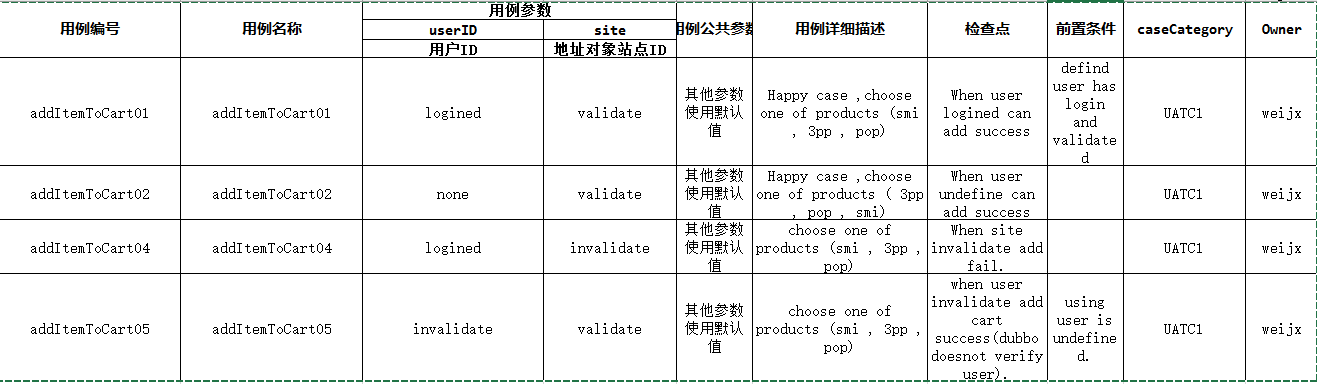
功能概述：

自动化数据主要是通过固定格式的excel语法设计用例，当前已经存在对应的maven插件对excel语法进行解析并生产testNG测试脚本。

**注意：**

**在设计用例时需考虑环境相关性，当前用例执行环境的差异性可用通过caseCategory区分.**

如：添加购物车



#### 1.1.2自动化业

#### 务验证层

功能概述：

测试框架业务处理层主要完成接口调用并根据测试用例的设计完成结果校验。根据自动化的处理流程其划分为三个模块：

##### 1.1.2.1用例装载模块

功能描述：

此模块为自动化框架自动化生产，其主要作用是装载自动化流程（调用被测接口封装模块，保存上下文环境，调用验证逻辑）并根据测试结果生产测试报告。

##### 1.1.2.2被测系统接口方法适配模块

功能描述：

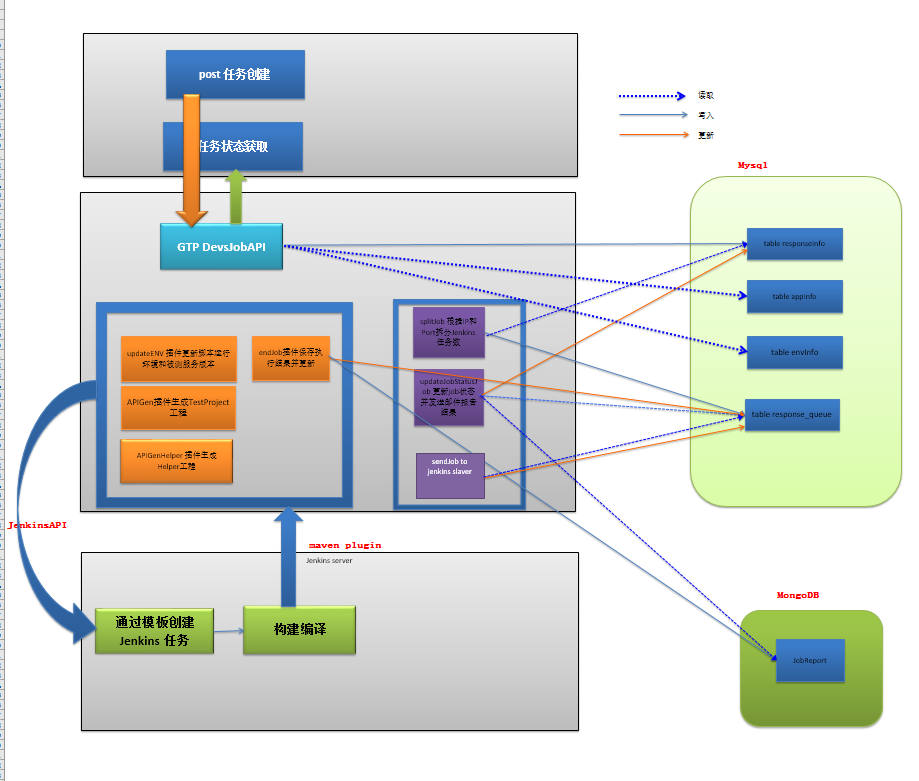
封装被测接口调用流程，并保持调用上下文。

##### 1.1.2.3被测系统接口验证模块

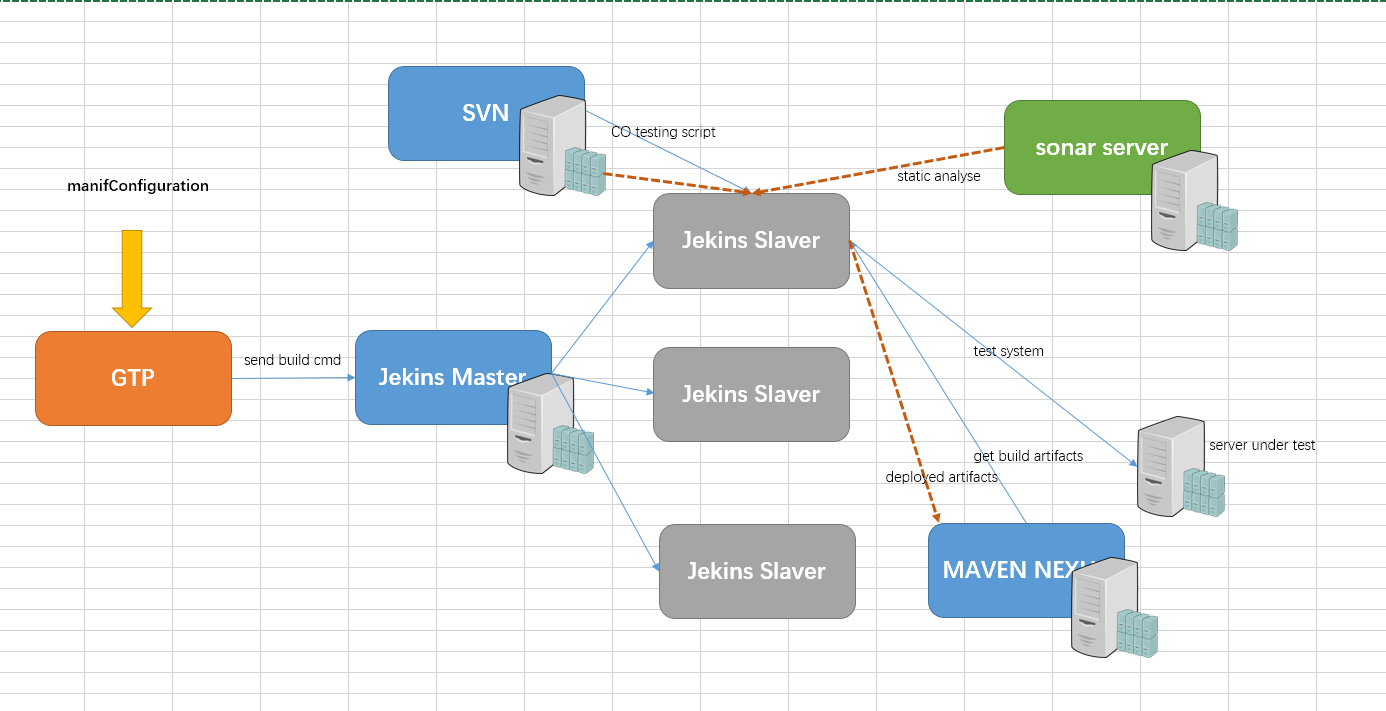
调用被测接口封装的验证逻辑，完成接口校验。

### 1.2 自动化脚本对外持续集成方案

此为自动化平台自有对外持续集成接口，便于每次代码checkout build后触发此流程对更新后的接口进行自动化校验。



（架构设计）



（自动化执行平台网络拓扑图）

## 二：自动化方案实现需求

### 2.1 接口描述

2.1.1 TODO 需要开发提供接口文档，其中文档中包含接口方法参数和返回值描述。

2.1.2 TODO 接口部署调用描述，包含接口POM，ZK配置

2.1.3 TODO 接口如果涉及到后台数据库更新信息验证，需提供数据库的读权限

### 2.2 自动化框架配置规范

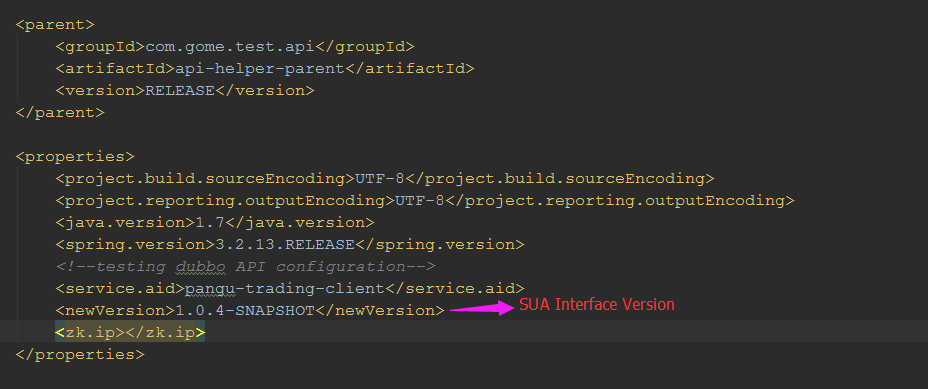
2.2.1 接口自动化基础规范及框架使用

参考: <http://wiki.ds.gome.com.cn/pages/viewpage.action?pageId=3670998>

2.2.2 脚本SVN维护规范

参考: 邮件《P0测试环境自动化测试注意事项》

2.2.3 接口版本维护方案



。

## 三：用例设计技巧：

### 测试粒度及自动化程度划分

功能保障

隔离性,速度

当前测试粒度位于中间,为了便于衡量自动化的程度，当前接口自动化划分如下三个等级：

|  |  |  |
| --- | --- | --- |
| 接口测试覆盖度等级 | 等级描述 |  |
| Level 1 | 涵盖接口冒烟用例 |  |
| Level 2 | 涵盖每个接口正向逆向用例，每个功能点有相应的用例覆盖 |  |
| Level 3 | 1. 当代码提交时能运行自动化用例 2. 主功能至少每天运行一次自动化测试 3. 所有测试点达到覆盖 4. CI |  |

### 使用的测试模式

**Category-Partition：** 基于输入输出的范畴划分

**Combinational Function Test：** 根据接口的业务逻辑对接口进行组合测试

### 用例设计流程

**1.1** 对于多参数的用例需考虑使用用例生产工具来降低用例数并达到一定的覆盖率（如PICT）。或使用以下黑盒测试用例设计技巧来设计用例：

* **空数据**: 验证当提交空数据时系统是否能正常响应，如不会抛空指针异常。
* **有效数据** : 正向用例，当输入有效数据时系统会按照接口描述进行相应。
* **无效数据** : 逆向用例，当输入无效数据时系统会按照接口描述进行相应。
* **错误的参数格式:** 当参数未按照接口描述进行入参时，接口能正常响应。
* **边界值校验**: 对参数的边界值进行校验。
* **等价类划分** : 对输入参数进行等价类划分.

**… …**

**Example**

回滚虚拟账号预授权

publicResultDO<Boolean>rbHandleVirtualAccountAuthorize(String userId, String txId, String applySite);

**参数说明：**

userId用户ID

txId事务ID，保证全局唯一性，建议可采用UUID

applySite站点

**返回结果说明：**

ResultDO<Boolean回滚是否成功 true成功，false失败>

接口测试之前，需要充分的了解接口的实现功能的业务逻辑、接口参数、接口返回值。功能业务逻辑：

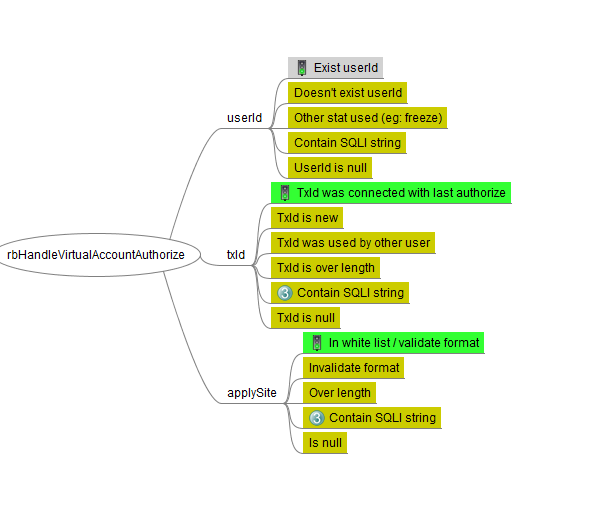
测试设计主要关注点

1. 接口中所有的入参都要写测试用例，对于机器来说你搞一个TC和多个TCes执行时间没有什么区别。避免hardcode
2. 每个入参的每个错误类型都要准备一个异常用例。如必须参数缺省、参数类型错误、参数 范围错误、参数超过最大位数、参数没有达到最小指定位数、参数的无效值（有效状态外）、参数的小数点超过规定长度、参数含有非法字、参数含有违禁字(如针对订单的评价是否包含反社会语句)、参数 的关联性检查（如：在用户输入的地址中区域地址不匹配情况：朝阳区软件园）等等。
3. 对于正常系的用例，要把所有入参的各种合法的有效值都执行到。所有入参的最大位可以用一个测试用例执行掉。
4. 测试设计时需致命测试的优先级。

对于搜索接口，应该把每个参数单独作为搜索条件来确认搜索结果是否正确，然后再确认多条件输入后的结果。

如public ResultDO<VirtualResult>findVirtualAccountByUserId(String userId,StringapplySite)，userId和applySite应分别作为一个用例来验证，userId+applySite作为多条件查询用例。

基于多参数用例设计MS PICT 示例：

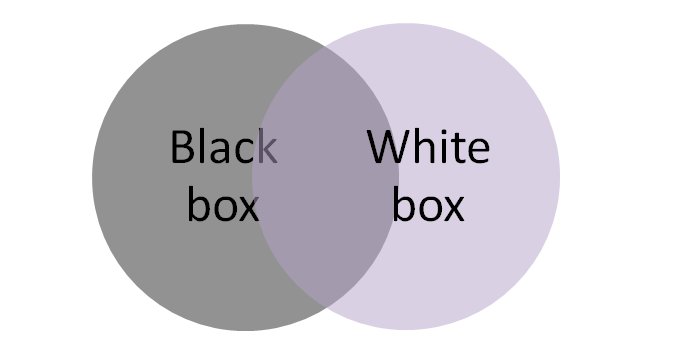


Pairwise generate =32:

----------------------------------------------------------------------------------------------------

|  |  |  |
| --- | --- | --- |
| userId | txId | applyStie |
| exist | indiv | invalidate |
| freeze | null | invalidate |
| SQLI | invalidate | whiteList |
| SQLI | null | SQLI |
| exist | overLength | null |
| SQLI | indiv | null |
| SQLI | overLength | invalidate |
| exist | SQLI | overLength |
| null | null | whiteList |
| freeze | overLength | whiteList |
| exist | validate | whiteList |
| freeze | validate | overLength |
| null | null | null |
| null | SQLI | invalidate |
| freeze | SQLI | null |
| null | invalidate | overLength |
| freeze | indiv | overLength |
| exist | invalidate | SQLI |
| null | overLength | overLength |
| freeze | invalidate | null |
| SQLI | null | overLength |
| null | indiv | whiteList |
| null | validate | invalidate |
| SQLI | validate | null |
| SQLI | SQLI | SQLI |
| null | SQLI | whiteList |
| freeze | overLength | SQLI |
| null | validate | SQLI |
| SQLI | indiv | SQLI |
| exist | invalidate | invalidate |
| exist | null | whiteList |

**Step2** 基于业务逻辑进行白盒（或灰盒）用例设计。



1 组合不同的接口进行业务逻辑校验，如调用插入接口创建一条数据，然后调用更新接口进行更新，最后再调用查询接口进行校验。另一情况是根据业务场景来创建接口调用来验证业务流程。

2 backend 校验，主要根据接口业务流最终状态来验证相关的持久化是否符合接口描述。**（需要开发提供DB 连接串信息，便于数据的提取进行对比校验）**

#### 基于后台数据验证的策略流程

当前被测接口对应测试人员来说是一个黑盒，测试人员不知道其逻辑及流程，对于被测接口根据其使用场景划分为：1查询接口， 2操作接口。

对于查询接口，其前置条件创建一是通过对应的操作接口另一是直接查询DB获取。

（被测系统理解模型）

Automation framework

Business Logic

Others

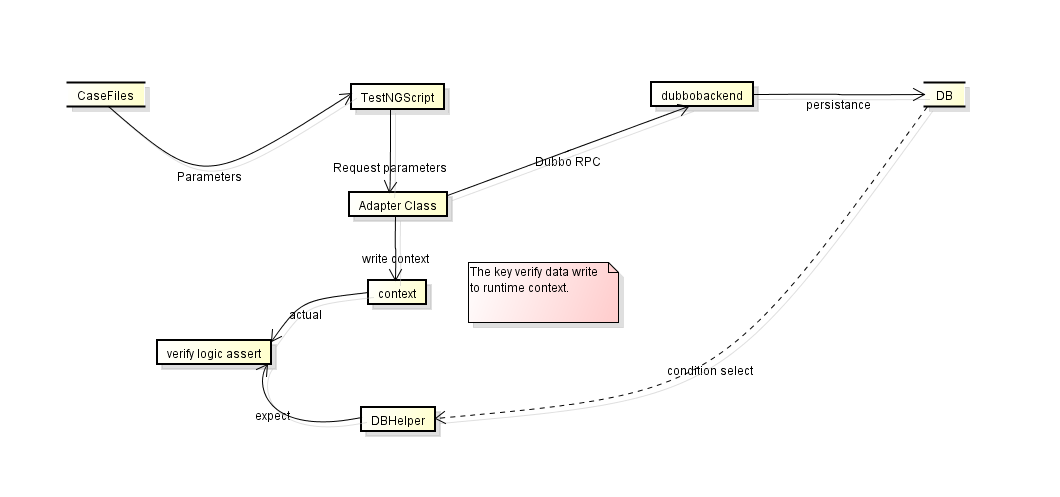
CART API

DB

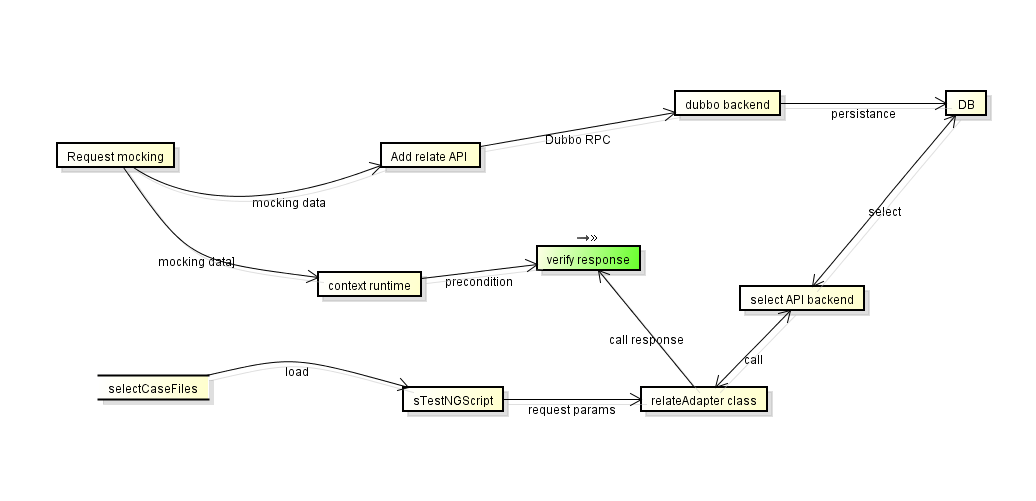
DB

DB

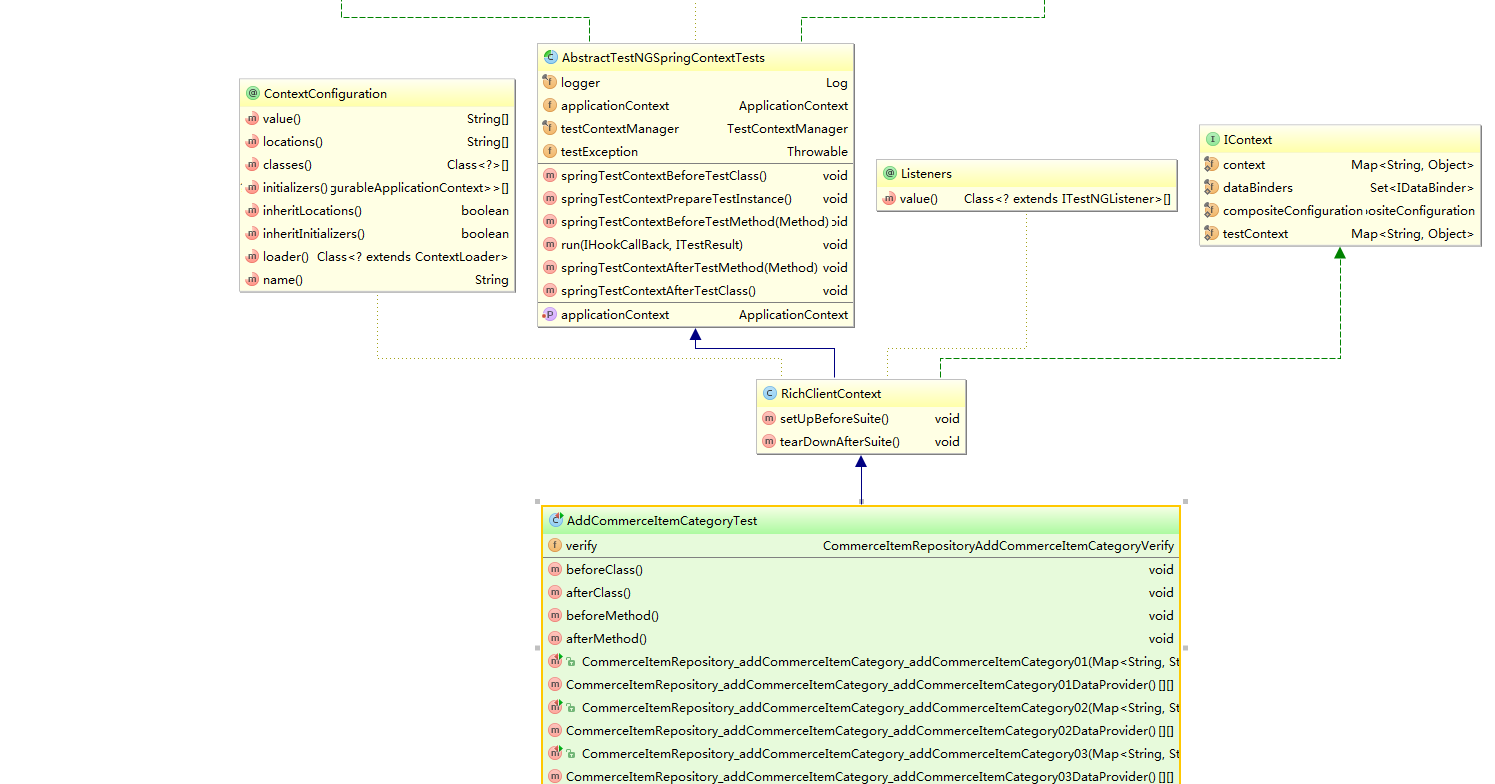
通过数据封装DAO来验证接口的dataflow diagram.适合操作类接口。对于查询接口其数据也可以通过提前写入然后调用查询接口和写入的数据进行对比。也可以使用条件查询语句获取数据同查询接口调用返回的数据进行对比。



查询类dataflow

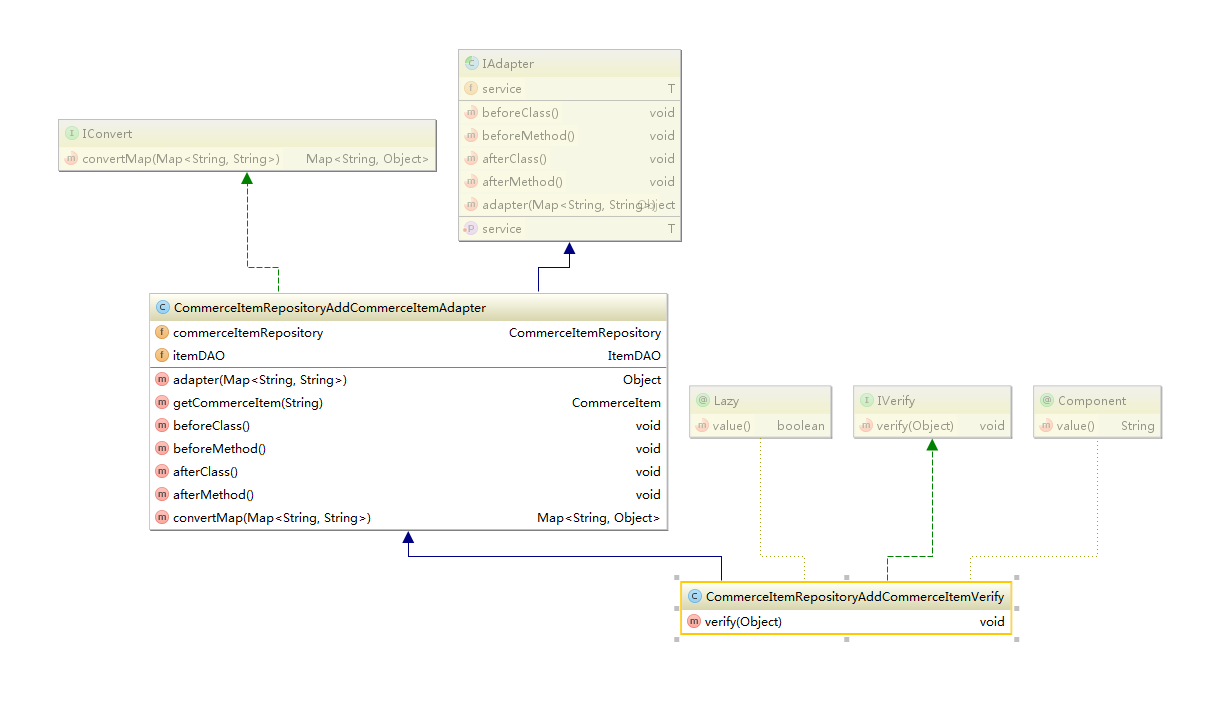


示例UML

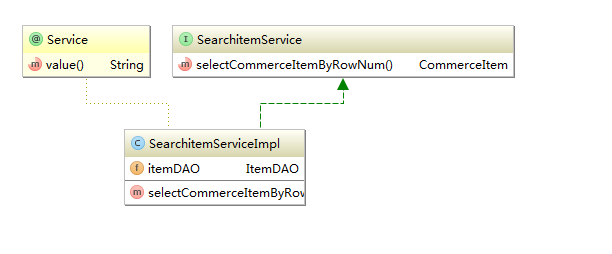


框架规则引擎插件负责把接口用例excel转换成testNG模板。TEDer根据用例逻辑完成接口的调用和验证逻辑代码。

实现\*\*\*Verify类的验证逻辑和\*\*Adapter类的接口调用逻辑。（如下）



DAO层用例校验数据和获取前置条件



#### 基于接口调用闭环流程方案

// cleanCart->addItemToCart->getItemFromCart->clentCart