API Testing

Content

Application coverage

Technology

Project structure

Bamboo plan

Worth mentioning

Application coverage - Simple tracking API

simple-tracking?referrer=&url=https%3A%2F%2Fw...

Headers Preview Response Cookies Timing

▼ General

Request URL: https://clickout-staging.cuponation.es/default/simple-tracking?referrer=&url=https%3A%2F%2Fwww.staging.cuponation.es%2F&browserName=chrome&browserMajorVersion=60.0&t=1505857490526&clientId=e916c61feaf5e98bc9b770a02f24eefc&device_source=desktop

Request Method: GET

Status Code: 🟢 200 OK

Remote Address: 52.58.151.71:443

Referrer Policy: no-referrer-when-downgrade

▼ Response Headers   view source

Cache-Control: no-cache, no-store, must-revalidate

Connection: keep-alive

Content-Length: 108

Content-Type: image/png

Date: Tue, 19 Sep 2017 21:46:44 GMT

Expires: 0

Pragma: no-cache

Server: nginx/1.4.6 (Ubuntu)

Set-Cookie: customerHash=3ed0b8ff-3a13-4aeb-bf71-d1249d4f177b; Version=1; Path=/; Domain=.cuponation.es; Max-Age=157680000

Set-Cookie: idSession=33dbacf5-b1d0-422c-8c66-5651adf2effe; Version=1; Path=/; Domain=.cuponation.es; Max-Age=1800

X-Powered-By: Undertow/1

X-Robots-Tag: noindex, nofollow

▼ Request Headers   view source

Accept: image/webp,image/apng,image/*,*/*;q=0.8

Accept-Encoding: gzip, deflate, br

Accept-Language: en-US,en;q=0.8

Connection: keep-alive

Cookie: customerHash=3ed0b8ff-3a13-4aeb-bf71-d1249d4f177b; idSession=33dbacf5-b1d0-422c-8c66-5651adf2effe; _ga=GA1.2.109125002.1505857245; _gid=GA1.2.4679369.1505857245; _gat=1

Host: clickout-staging.cuponation.es

Referer: https://staging.cuponation.es/

User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_5) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/60.0.3112.113 Safari/537.36

▼ Query String Parameters   view source   view URL encoded

referrer:

url: https://www.staging.cuponation.es/

browserName: chrome

browserMajorVersion: 60.0

t: 1505857490526

clientId: e916c61feaf5e98bc9b770a02f24eefc

Global Savings Group

JAVA 8

Rest-Assured

Mongo

MySQL

Main parts:

    Methods,Helpers

    Abstract Tests Classes

    TestData

```
▼ 📁 src
  ▼ 📁 main
    ▼ 📁 java [main] sources root
      ▼ 📁 com
        ▼ 📁 cuponation
          ▼ 📁 clickout
            ▼ 📁 apitesting
              ▼ 📁 config
                  🅒 Api
                  🅒 Methods
                  🅒 TestData
              ▶ 📁 exceptions
              ▶ 📁 jdbc
              ▶ 📁 mailTestReport
              ▶ 📁 mongo
              ▶ 📁 reports
              ▶ 📁 retryAnalyzer
              ▶ 📁 services
              ▼ 📁 utils
                  🅒 Config
                  🅒 Helpers
    ▶ 📁 resources [main] resources root
  ▼ 📁 test
    ▼ 📁 java
      ▼ 📁 com
        ▼ 📁 clickout
          ▼ 📁 apitesting
            ▶ 📁 deals
            ▶ 📁 p0Tests
            ▶ 📁 p1Tests
                🅒 AbstractP0Test
                🅒 AbstractP1Test
                🅒 ConsistencyChecks
                🅒 DealsTest
```
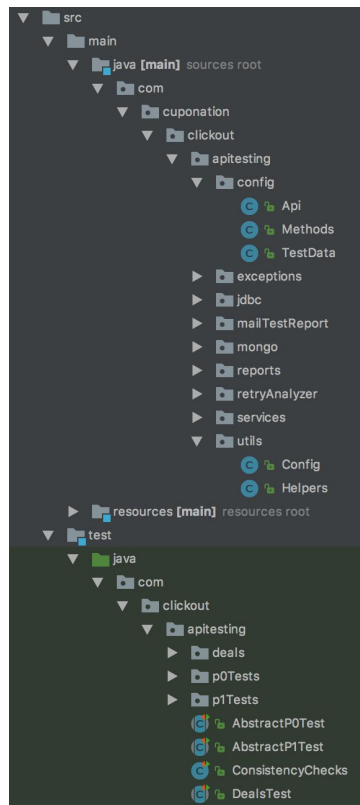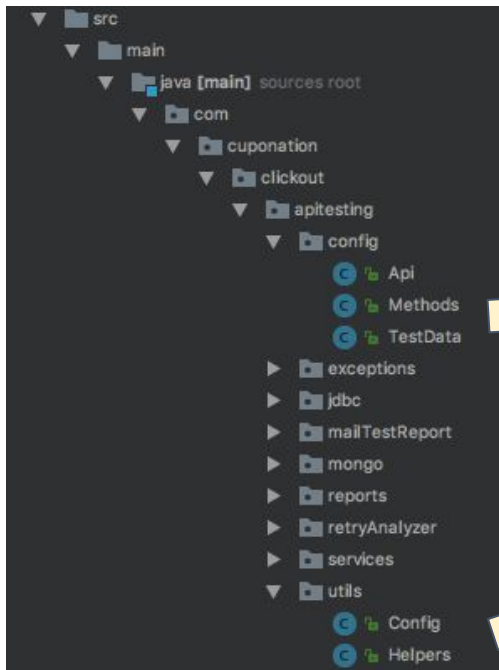
```java
public void basicTestClickout(String clientUID, String realDomain, String url, String countryIso) {
    ActiveVoucher voucher = VoucherService.getRandomExpiredVoucherWithA2ASUBID(clientUID, ActiveVoucher.class);
    String voucherId = voucher.getIdVoucher().toString();
    String affiliateURL = voucher.getAffiliateUrl();
    Long voucherFromMongo = voucher.getIdVoucher();


    Response visit = createVisit(realDomain, url, clientUID);//This method creates a visit

    String visitCustomerHash = helpers.extractCustomerHash(visit);//Extracts the value of the customerHash from the visit request
    String visitIdSession = helpers.extractIdSession(visit);//Extracts the value of the idSession

    Cookie visitCustomerHashCookie = visit.detailedCookie(TestData.CUSTOMER_HASH);//Extracts the customerHash cookie from the visit
    Cookie visitIdSessionCookie = visit.detailedCookie(TestData.ID_SESSION);//Extracts the idSession cookie from the visit

    Cookie visitAWSELB = visit.detailedCookie(TestData.AWSELB);

    if (visitAWSELB == null) {
        Response clickout = createClickoutWithoutLB(visitCustomerHashCookie, visitIdSessionCookie, realDomain, voucherId, clientUID, url);//This methods creates a clickout

        Cookie clickoutCustomerHashCookie = clickout.detailedCookie(TestData.CUSTOMER_HASH);//Extracts the customerHash cookie from the clickout
        Cookie clickoutIdSessionCookie = clickout.detailedCookie(TestData.ID_SESSION);//Extracts the idSession cookie from the clickout

        String clickoutCustomerHash = helpers.extractCustomerHash(clickout);//Extracts the value of the customerHash from  the clickout request
        String clickoutIdSession = helpers.extractIdSession(clickout);//Extracts the value of the idSession from  the clickout request

        String locationHeaderWithoutClickoutId = helpers.extractLeftPartLocationHeader(clickout);//Extracts the locationHeader without the clickoutId
        String affiliateURLWithoutA2ASUBID = helpers.extractLeftPartAffiliateURL(affiliateURL);//Extracts the affiliateURL without the A2ASUBID

        Integer startingIndexOfA2ASUBID = helpers.getStartingIndexOfAffiliateURL(affiliateURL);//Gets the starting index of A2ASUBID

        String redirectClickoutId = helpers.extractClickoutId(clickout);//Extracts the clickoutId from the locationHeaderRedirect

        String replacedA2ASUBIDWithClickoutId = clickout.getHeader(TestData.LOCATION).substring(startingIndexOfA2ASUBID, startingIndexOfA2ASUBID + 42);//Gets the value for the A2ASUBID p

        List<Visit> returnVisit = SQSService.getInstance().consumeMessagesDeserialized(SQSQueue.TEST_VISIT_SQS, Visit.class,  maxMessages: 1);//Get and delete sqsVisit messages

        String sqsVisitCustomerHash = returnVisit.get(0).getHash();//Extracts the sqsVisitCustomerHash
        String sqsVisitIdSession = returnVisit.get(0).getIdSession();//Extracts the sqsVisitIdSession
        String sqsVisitCountryIso = returnVisit.get(0).getCountryIso();//Extracts the sqsVisitCountryIso
        String sqsVisitClientUID = returnVisit.get(0).getClientUID();//Extracts the sqsVisitClientUID

        assertEquals(visitCustomerHashCookie, clickoutCustomerHashCookie, TestData.VISIT_CUSTOMER_HASH_VS_CLICKOUT_CUSTOMER_HASH_MESSAGE);//Compares the customerHash from the visit with
        assertEquals(visitIdSessionCookie, clickoutIdSessionCookie, TestData.VISIT_ID_SESSION_VS_CLICKOUT_ID_SESSION_MESSAGE);//Compares the idSession from the visit with the clickout
        assertEquals(locationHeaderWithoutClickoutId, affiliateURLWithoutA2ASUBID, TestData.LOCATION_HEADER_WITHOUT_CLICKOUT_ID_VS_AFFILIATE_URL_WITHOUT_A2ASUBID_MESSAGE);//Compares
        assertEquals(redirectClickoutId, replacedA2ASUBIDWithClickoutId, TestData.CLICKOUT_REDIRECT_VS_REPLACED_A2ASUBID_WITH_CLICKOUT_ID_MESSAGE);//Compares the clickoutId with the rep
        assertEquals(visitCustomerHash, sqsVisitCustomerHash, TestData.EXTRACTED_VISIT_CUSTOMER_HASH_VALUE_VS_SQS_VISIT_HASH_MESSAGE);//Compares the customerHash from the visit with the
        assertEquals(visitIdSession, sqsVisitIdSession, TestData.EXTRACTED_VISIT_ID_SESSION_VS_SQS_VISIT_ID_SESSION_MESSAGE);//Compares the idSession from the visit with the sqsIdSession
        assertEquals(countryIso, sqsVisitCountryIso, TestData.COUNTRY_ISO_VS_SQS_VISIT_COUNTRY_ISO_MESSAGE);//Compares the countryIso from the visit with the sqsCountryIso
        assertEquals(clientUID, sqsVisitClientUID, TestData.CLIENT_UID_VS_SQS_VISIT_CLIENT_UID_MESSAGE);//Compares the clientUID from the visit with the sqsClientUID
```

Project structure - Each client has its own class where the data is stored

Project structure - How the test looks

```java
public abstract class AbstractP0Test {

    Methods methods = new Methods();

    MysqlCon mysqlCon = new MysqlCon();

    protected abstract String getClientUID();

    protected abstract String getConfigDomain();

    protected abstract String getURL();

    protected abstract String getCountryIso();

    private String getRealDomain() {
        String envDomain = System.getenv( name: "DOMAIN");
        return Strings.isNullOrEmpty(envDomain) ? getConfigDomain() : envDomain;
    }

    @BeforeSuite
    public void cleanSQS() throws InterruptedException {
        int count = methods.consumeAllVisitSqsMessages();
        Logger.getLogger(AbstractP1Test.class).info(count + TestData.VISIT_SQS_MESSAGES_DELETED);

        int number = methods.consumeAllCLickoutSqsMessages();
        Logger.getLogger(AbstractP1Test.class).info(number + TestData.CLICKOUT_SQS_MESSAGES_DELETED);
    }

    @Test(retryAnalyzer = ReRunTest.class)
    public void dbTest() throws SQLException, InterruptedException {
        methods.consumerTest(getClientUID(), getRealDomain(), getURL(), getCountryIso());
    }

    @Test(retryAnalyzer = ReRunTest.class)
    public void basicTest() {
        methods.basicTestClickout(getClientUID(), getRealDomain(), getURL(), getCountryIso());
    }

    @Test(retryAnalyzer = ReRunTest.class)
    public void basicTest2() {
        methods.basicTestClickout(getClientUID(), getRealDomain(), getURL(), getCountryIso());
    }

    @Test(retryAnalyzer = ReRunTest.class)
    public void directClickoutTest() {
        methods.directClickout(getClientUID(), getRealDomain(), getURL(), getCountryIso());
    }

    @Test(retryAnalyzer = ReRunTest.class)
    public void directClickoutTest2() {
        methods.directClickout(getClientUID(), getRealDomain(), getURL(), getCountryIso());
    }
}
```
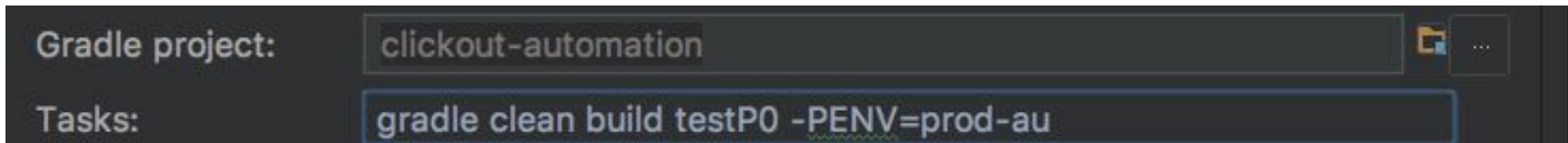
Running parameters

Gradle project:	clickout-automation
Tasks:	gradle clean build testP0 -PENV=prod-au

Package com.clickout.apitesting.p0Tests.production.brasil

all > com.clickout.apitesting.p0Tests.production.brasil

20	0	0	3m34.16s
tests	failures	ignored	duration

100%
successful

Classes

Class	Tests	Failures	Ignored	Duration	Success rate
BrasilCuponationClickoutTestId	5	0	0	1m8.50s	100%
ClimaTempoClickoutIdTest	5	0	0	49.613s	100%
FlipItBrasilClickoutIdTest	5	0	0	40.526s	100%
IGBRClickoutIdTest	5	0	0	55.520s	100%

Global Savings Group

Bamboo plan

Tests - from 1 to 10 mins (region)

Cron job - Every 2 hours

Retry

Logging

Error messages