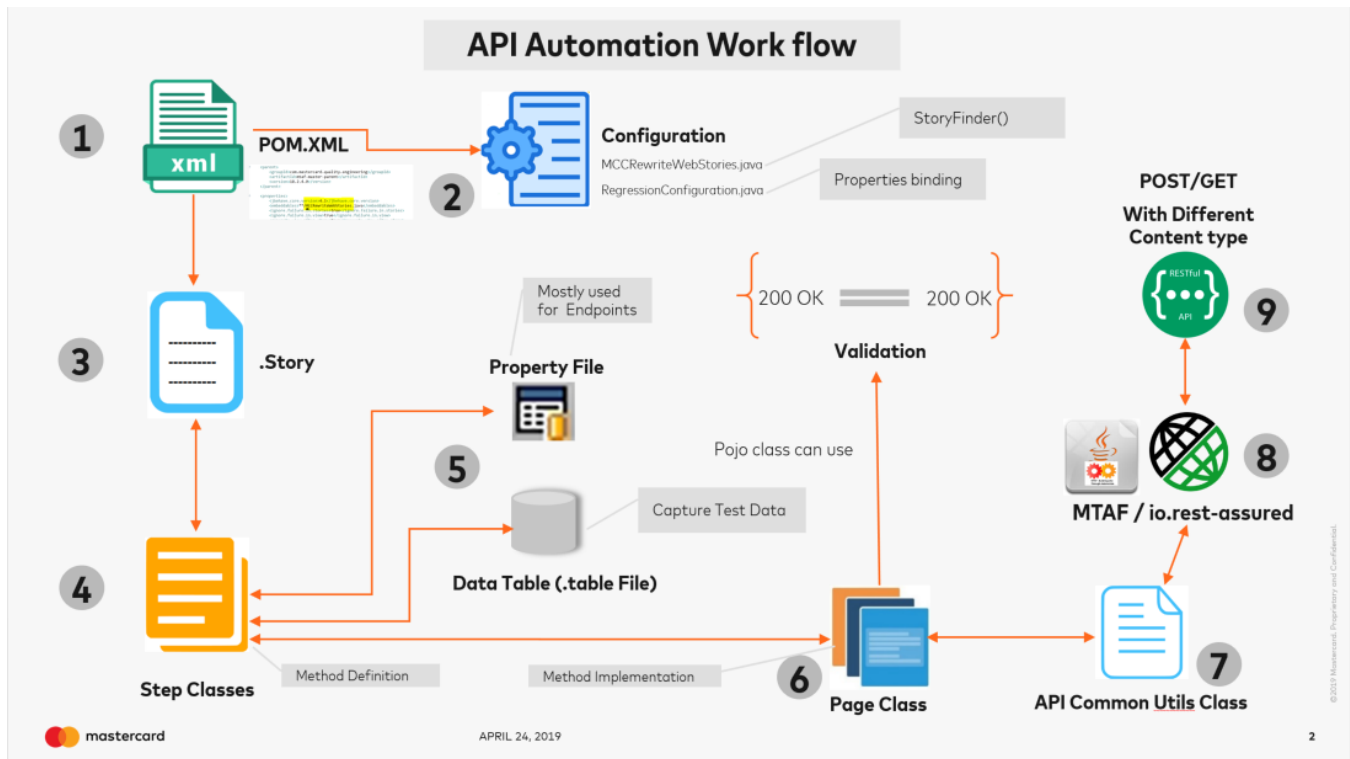


MCC API Test Automation Guide



1. **POM.XML** - Which contains all required dependency and MTAF-CORE for automation. These are the Mastercard MTAF-CORE. dependency. Example- `mtaf-ui-tools/mtaf-driver-factory/mtaf-page-bindings/mtaf-rest-tools`

All dependant .jars & maven plugins are controlled in the parent-pom file (pom.xml).

For more details MTAf Projects : [MTAF-CORE-PROJECTS](#)

2. Configuration

- **MCCRewriteWebStories.java** - This class called from Pom file and contains storyPaths() method. Method will find and map story as per VM argument.
- **RegressionConfiguration.java** - This class used for properties file binding.

3. Story Class - Declare JBehave .story class as provided path src/main/resources.

4. Step Class- Create or add method into .steps class i.e. [com.mastercard.quality.engineering.validation.steps](#)

- Map JBehave story steps and Java class steps with JBehave annotation.
- It is recommend to keep method definition in step class.
- Call test data from data table or property file and send to page class for validation.

5. Page Class -Create or add method into page class i.e. [com.mastercard.quality.engineering.validation.pages](#)

- Contains all validation methods implementation
- Contains all page element.
- Declare all page element above of the file as per standards, Don't declare in middle.
- Add dashboard report and try catch block in the validation methods.

6. API Common Utils class: [com.mastercard.quality.engineering.validation.commonUtils / APICommonUtils.java](#)

- Contains common POST and GET method which will call MTAf io.rest-assured methods.

Few Common methods

- `postRequestToAPIJson(String endpoint, Map<String, String> requestParams)`
- `postRequestToAPIJsonWithCookies(Set<Cookie> cookies, String endpoint, Map<String, String> requestParams)`
- `postRequestToAPI(Set<Cookie> cookies, String endpoint, Map<String, String> requestParams)`

- `getServiceResponseBody(Set<Cookie> cookies, String endpoint)`
- `getServiceResponseBody(String endpoint)`

API Automation Workflow

Step1 : Create JBehave Story Class

Story File

```
Scenario: Sign up through API
Meta:
@TestId TC401290
@signupAPI
@signupApiOnly
@signupapprove
@signupapprovetour
@Project 193123719952d
@TestCaseName Sign up through API-TC401290
Given the external user signup into MCC
When user hit the create user api call with update payload data
Then validating success response from create user API with code responseBody requestStatus

Examples:
TestData/E2EAPIAutomation/signup_dataAPI.table
```

Bind story name with
`@testCaseName` meta filter either
from top or from data table

Get data from external resource
i.e. data table

Compulsory Meta tag and format

- **Test case name** i.e. @TC10001
- **Owner** of the test case i.e. @e062443
- **Component** name i.e. @signincritical
- **Sprint and team** name i.e. @insprintWarrior18.4.2



mastercard

APRIL 24, 2019

3

© 2019 Mastercard. Proprietary and Confidential.

Step 2 : Create Step Class and map Story class steps with Java class with JBehave annotations

Step Class

```
/**
 * E062443 TC401290 Sign up through API
 *
 * @param payloadFile - decide test data property file from data table
 */
@When("user hit the create user api call with update payload data")
public void enterCreateUserAPIwithPayloadData(@Named("payloadFile") String payloadFile) {}

createUser(payloadFile);

public void createUser(String userData) {
    SignUpApiPage signUpApiPage = new SignUpApiPage(webProvider.get(), 10, commonUtils, apiCommonUtils);
    SignUpPage signUpPage = new SignUpPage(webProvider.get(), 10, commonUtils);
    Map<String, String> mapCreateUserAPIData = null;
    // Mapping with property file
    try {
        FileReader reader = new FileReader("src/test/resources/config/" + commonUtils.getEnvironmentName() + "/" + userData + ".properties");
        Properties prop = new Properties();
        prop.load(reader);
        mapCreateUserAPIData = new HashMap<>();
        for (String key : prop.stringPropertyNames()) {
            if (key.equalsIgnoreCase(ALIAS)) {
                String uName = prop.getProperty(key);
                String username = signUpPage.getValidUserId(uName);

                commonUtils.setUser(username, "test@123");
                mapCreateUserAPIData.put(key, username);
            } else {
                mapCreateUserAPIData.put(key, prop.getProperty(key));
            }
        }
    } catch (IOException | NullPointerException e) {
        logger.error("Some issue with file : " + e);
    }

    response = signUpApiPage.postRequestToAPI(env.getProperty("api.createuser"), mapCreateUserAPIData);
    response.prettyPrint();
    commonUtils.waitInSecond(MCCRewriteConstants.TWO);
}
```

Get environment
Name as per
execution
(stage_perf, stage)

Get Payload File
userData =

Mapping keys and values from create user property file
i.e. payload data.

Post End Points from api. Properties.
Keys is api.createuser

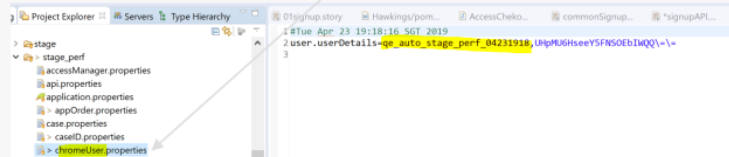
environment	payloadFile	code
stage_perf	createUser	SUCCESS_00

Step 3 : Create Test Data table where we can pass the required test data for API call Post or to validate expected get request

Data Table file

1	environment	payloadFile	code	requestBody	requestStatus	saUserName	saPassword	username	password	preferenceGroupCode
2	stage_perf	createUser	SUCCESS_002	ORDER(S) SUCCESSFULLY PLACED	false	smoke_admin01	test@123	NA	NA	com.mastercard.global
3										

"NA" means , test case required new user and user will take from property file as per environment



Step 4 : API Call with payload data , Endpoint along with or without Cookies

Page Class

```
/**
 * E062443
 * Used in - Sign up TC401290
 * Calling API with POST request with payload as Map
 * @param endpoint API URL which need to accessed
 * @param requestParams payload Map
 * @return response
 */
```

Example of POST

```
public Response postRequestToAPI(String endpoint, Map<String, String> requestParams){
    return apiCommonUtils.postRequestToAPIJson(endpoint, requestParams);
}
```

```
// @signinPage.verifyFirstUserTour();
```

Example of GET

```
SignInApiPage signinApi = new SignInApiPage(webProvider.get(), 10, commonUtils, apiCommonUtils);
response = signinApi.getRequestToAPI(env.getProperty("api.verifyfirstusertour"));
```

API Common Utils Class

```
/**
 * E062443 TC401290 - Sign up Post without Cookies
 *
 * @param endpoint
 * @param requestParams
 * @return
 */
```

POST Without Cookies

Content type JSON Format

```
public Response postRequestToAPIJson(String endpoint, Map<String, String> requestParams) {
    return given().log().all().contentType(APPLICATION_JSON).body(requestParams).when().post(endpoint).then().extract().response();
}
```

```
public Response postRequestToAPIJsonWithCookies(Set<Cookie> cookies, String endpoint, Map<String, String> requestParams) {
    List<Cookie> restAssuredCookies = new ArrayList();
    for (org.openqa.selenium.Cookie cookie : cookies) {
        restAssuredCookies.add(new io.restassured.http.Cookie.Builder(cookie.getName(), cookie.getValue()).build());
    }
    return given().log().all().contentType(APPLICATION_JSON).cookies(new io.restassured.http.Cookies(restAssuredCookies))
        .formParams(requestParams).when().post(endpoint).then().extract().response();
}
```

POST With Cookies,

Whenever we are going to login the application ,cookies required to validate

Step 5 : Validate actual and expected get response

```
Response resp = this.postRequestToAPI(endpoint, "{\\ids\\":[\" + orderId + \"],\\reason\\\":\"test\\\"}");
resp.prettyPrint();
if(action.equalsIgnoreCase("Decline") && resp.asString().contains("SUCCESS_005"))
{
    isTrue = true;
}
else if(action.equalsIgnoreCase("Approve") && resp.asString().contains("SUCCESS_004"))
{
    isTrue = true;
}
```

Validation from response

Pretty-print the response body if possible and return it as string. Mainly useful for debug purposes when writing tests. Pretty printing is possible for content-types JSON, XML and HTML.

Few Common methods

postRequestToAPIJson(String endpoint, Map<String, String> requestParams)

postRequestToAPIJsonWithCookies(Set<Cookie> cookies, String endpoint, Map<String, String> requestParams)

postRequestToAPI(Set<Cookie> cookies, String endpoint, Map<String, String> requestParams)

getServiceResponseBody(Set<Cookie> cookies, String endpoint)

getServiceResponseBody(String endpoint)



APRIL 24, 2019

7

© 2019 Mastercard. Proprietary and Confidential.

Step 6 : Payload or test data from property file

Property File

```
1 addressLine1=
2 addressLine2=
3 addressLine3=
4 agreementId=1.18
5 alias=ge_auto
6 city=
7 consentLangTxtId=1.18
8 countryCode=
9 departmentName=
10 email=girija.mohanty@mastercard.com
11 firstName=girija
12 ica=3108086
13 institutionId=100106
14 lastName=mohanty
15 organizationName=HSBC Bank Nevada N.A.
16 password=test@123
17 phoneNumber=454353
18 postalCode=
19 comments=test
20 securityAnswer1=pet
21 securityAnswer2=color
22 securityQuestion1=9
23 securityQuestion2=10
24 state=
25 title=unspecified
26 userCountryCode=
27 userTypeCode=ISR
```

Update Consent Language and agreement ID value as per build



APRIL 24, 2019

8

© 2019 Mastercard. Proprietary and Confidential.