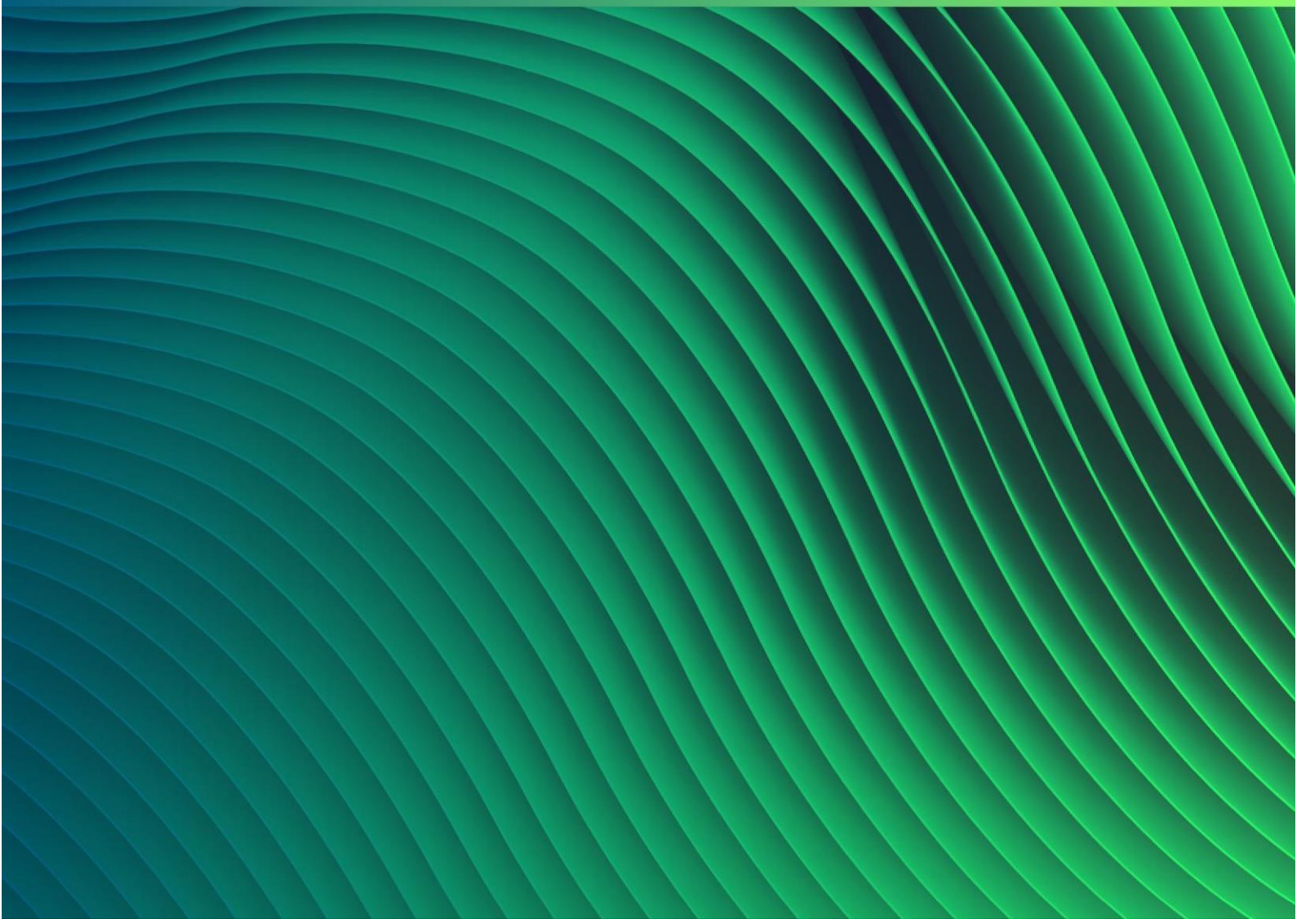
worldpay
from FIS

Salesforce SFCC

Salesforce SFCC Testing guide

Ver q2-23-v23.2.0 June 2023



Contents

Summary

This guide helps you to write test cases for all Worldpay services on the SFCC platform that use the MobileFirst reference application. You can use them to ensure all the main functionality operates correctly.

The guide also includes references to automation test suites. These suites are for functional testing, integration testing and unit testing so you can verify your implementation. Finally, the guide includes instructions for using the *CodeceptJS* testing automation suite for acceptance testing.

Test cases

This cartridge has successfully passed through QA and UAT. The test cases below help you to implement the functional and integration test cases. See *Worldpay_TestCases_Suite.xlsx* at *<Workspce>\link_worldpay\test\Manual_TestSuite.*

Test Data

The test data can be found in Worldpay_TestData.xlsx at <Workspce>\link_worldpay\test\testData

Automation test suit for functional testing

Prerequisites

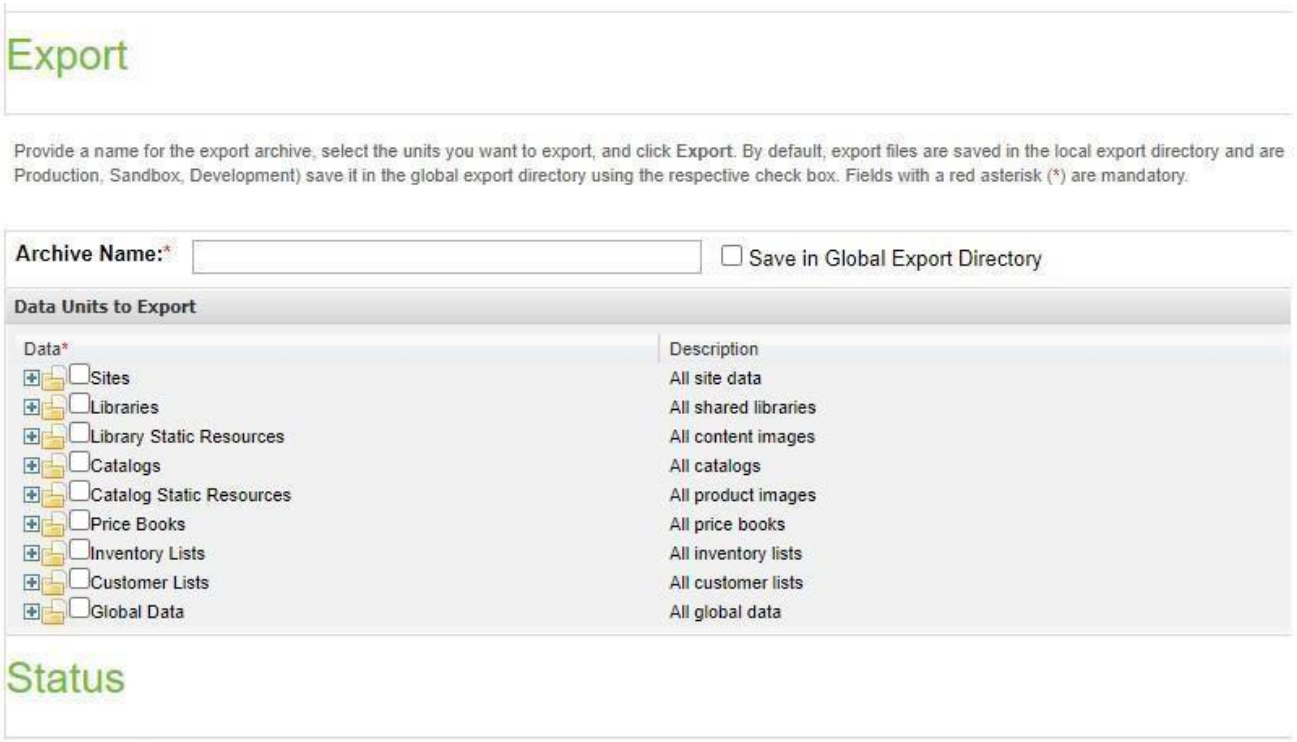
BM Configurations

Redirect credit card payments must be in page format. To do this:

1. Go to the *payment method* section and reset any configurations that are set to *iframe* or *light box* format.

You must do a site export. To do this:

1. Find the xml files you need. They are *customer-list*, *catalogs*, *pricebooks* and *promotions*, which are in *Business Manager* at *Administration/Site Development/Site Import and Export*. See the screenshot below:



2. Select the file you want to export (for example, *Price Books*)
3. Select *All Price Books*, then type a name into the *Archive Name* field and click the **Export** button. A success message appears.
4. To download the file, click the **Download** button.

Local setup

If you want to test *WeChat Pay* (direct mode), do the following:

1. Open the *wechatDirectPaymentMethod.js* file at *test/functional/checkout/guestUser/singleShip*.
2. Make this change at line no.22: (**describe =>describe.skip**).

To setup the *Order Confirmation* page, do the following:

1. Open the configuration file *test/functional/mocks/customers.js*.

2. For automation testing, create one new shopper with the zip (postal) code: 4304, country code: US and telephone number 3333333333 (a 10 digit telephone number)).
3. Save the Visa card number 4917610000000000 for this test shopper.

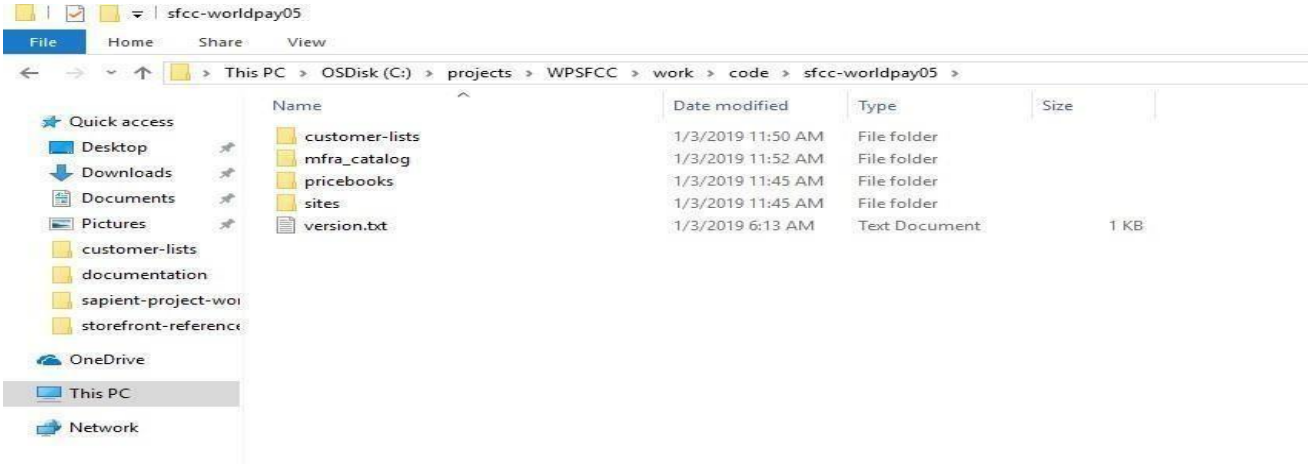
To update the folder path:

1. Update the folder path and file name to the same value as the path given in the file *functional/mocks/testDataMgr/main.js*. For example, the Customer xml path is in the file like this:

```
files: [demoDataDir + '/customer-lists/customer.xml']
```

2. The same path should exist in the site export folder. If your site export folder contains this file path: *C:\projects\WPSFCC\work\code\sfcc_worldpay\customer-lists\{siteid}.xml* then you must rename the file to *customer.xml* and create this structure in your local workspace:

   `<Workspace>\sfcc_worldpay\customer- lists\customer.xml`.
3. Check the xml folder structure corresponds with the path in the file. The site export folder structure should look the screenshot below:



Sometimes there are problems parsing customer-lists. If this is the case do the following:

1. In the *customer-lists/customer* xml file, update the xml string `customer-list` to `customers`.

As an alternative, you can export all the customer files directly from Business Manager and replace the xml in the *customer-lists* folder. To do this:

1. Go to *worldpay_main.js*, *test/mocks/testDataMgr* and input the credentials. They are:

- `loginUserEmail` - The shopper email (see Local setup above)
- `loginPassword` - Your user password
- `newUserPassword` - The new password for when the test user changes the password
- `storefrontPath` - The absolute folder path to the site export
- `ResourcePath` - The absolute folder path to the storefront reference architecture resource folder

- Three variant product IDs (like `variantProduct1`, `variantProduct2`, `variantProduct3`). Choose products that are easily available from the inventory
- Input all the credit card detail scenarios you want to test. For example, `creditCardAmex`, `creditCardDiscover`, `creditCardVisa`, `creditCardError`, `creditCardRefused`, `creditCard3D`. They should include the card name, card type, number, expiry date and CVV number

  NOTE: Please make sure that the login user email is in the *customer list* xml file.

Check that dw.json is not in the root folder. If it is, do the following:

1. Create a root folder that contains the following:

```
{

"hostname": "worldpay03-tech-prtnr-eu04-dw.demandware.net",

"username": "USERNAME_OF_STOREFRONT",

"password": "PASSOWRD_OF_STOREFRONT",

"code-version": "CODE_VERSION"

}
```

Configure and run automation testing

Do the following:

1. Run the command `npm install` in the root folder.
2. Install the *Selenium-standalone* node package globally and use the commands below to start the Selenium server:

```
npm install selenium-standalone@latest -g

selenium-standalone install selenium-standalone

start
```

3. To run functional testing, run the command `selenium-standalone start` on one terminal and the command to run the test, `npm run test:functional` on another terminal. Functional automation testing starts, it runs each file individually in the folder test*functional**checkout**guestUser**singleShip* for guest users and *test**functional**checkout**registeredUser* for registered users.

The test scenarios for payment methods are:

For guest users:

- Direct credit card: *directcreditcardPaymentMethod.js*
- Redirect credit card: *redirectcreditcardPaymentMethod.js*

- Wechat direct: *wechatDirectPaymentMethod.js*
- Paypal: *paypalPaymentMethod.js*
- Refused credit card: *refusedCreditCardPaymentMethod.js* For registered users:
- Login and use saved cards (the user account should have saved cards enabled for this)
- Register the customer and direct credit card: *newCustomerDirectCreditCardPayment.js*
- SEPA payment method: *sepaPaymentMethod.js*
- Alipay payment method: *alipayPaymentMethod.js*
- 3D payment method: *3dPaymentMethod.js*
- Registered customer and saved card: *loginSavedCard.js*
- Mister Cash payment method: *mistercashPaymentMethod.js*
- Error Credit Card: *errorCreditCardPaymentMethod.js*

To run a single scenario or file, change the specs line (line no. 88) in the file *wdio.conf.js* at *test/functional/webdriver*: For example, to test just the save-card functionality, change the line:

```
specs:[ specs

]


To: specs:[

'test/functional/checkout/registeredUser/loginSavedCard.js'

]
```

In the console, you can see a count of the test cases passing, failing and being skipped.

Integration testing

To run integration testing:

1. Open the terminal and run the command `npm run test:integration` to run the test and integration files one by one. You can see the status (success or failure) of each test script in the console.
2. The test scenarios are:

- 3DCreditCardPaymentMethod
- alipayPaymentMethod
- errorCreditCardPaymentMethod
- loginDirectCreditCardPaymentMethod
- loginSavedCardPaymentMethod
- mistercashPaymentMethod
- paypalPaymentMethod
- redirectCreditCardPaymentMethod
- sepaPaymentMethod

- wechatPaymentMethod

Automation test suite (CodeceptJS)

To configure and run automated acceptance testing using *CodeceptJS*, do the following:

1. *CodeceptJS related* dependencies are already available in *package.json* as given below:

```
"@wdio/selenium-standalone-service": "^5.8.0", "codeceptjs": "^2.1.0", "allurecommandline": "^2.9.0"
```

NOTE: To run functional testing on Mocha and Chai, add Chai dependencies and update the version of selenium-standalone dependencies in *package.json*:

```
"wdio-selenium-standalone-service": "0.0.11",
```

Run `npm install` every time you change the version.

2. To run acceptance testing, go to *<Workspace>\Link_worldpay\test\acceptance\Automation_info* and read the *SFCC_WPG_AUTOMATION_ReadMe*.

Unit testing

Open the terminal and run the *command npm run test* to run the test/unit files.

Contact Us

Please contact your Worldpay Relationship Manager or Worldpay Support.

Tel: 0800 096 3997.