Sample of Testing Documentation

Introduction

IntelliJ's HTTP Client plugin is used to create, edit, and execute HTTP requests directly from IntelliJ. The treasury team can utilize these functions and test HTTP request/responses from within the IntelliJ code editor. When the application is edited, these features speed up the testing process. The application can be tested in the local, development, and QA environments.

Files

There are three types of files. The .json file holds the environment variables. The http files contain the http requests and their response handlers. And the testing_functions.js file contains the common functions used within the http response handlers.

testing-functions.js

This file contains the common functions that are used in the tests. These functions are exported from this JavaScript file to be imported/called from within the response handler prompts inside the .http files.

- **testStatusNum(client, responseStatus, expectedStatusNum)** - Tests the response status number, does the expected Status Number match the actual Status Number
- **testBody(client, responseBody, expectedBody)** - Tests the string inside the SOAP body tags, does the expected Body match the actual Body
- **testResultIndicator(client, responseBody, expectedResultIndicator)** - Tests the string inside the ResultIndicator tags, does the expected Result Indicator match the actual Result Indicator
- **testReturnCode(client, responseBody, expectedReturnCode)** - Tests the string inside the ReturnCode tags, does the expected Return Code match the actual Return Code
- **testReturnMessage(client, responseBody, expectedReturnMessage)** - Tests the string inside the ReturnMessage tags, does the expected Return Message match the actual Return Message
- **testAddress(client, responseBody, ...)** - Tests the string inside the addressData tags, do the expected IdTypeCode, IdNumber, WithholdFormCode, StationNumber, Name, and Addresses match the actual
- **testWithholdData(client, responseBody, ...)** - Tests that the expected withhold data matches the actual withhold data
- **getDate()** - Returns the current date: YYYYMMDD
- **getXMLNSBody()** - Returns the shared XMLNS Body namespaces
- **formatToResultBody(resultIndicator, resultCode, returnMessage)** - Formats the resultIndicator, resultCode, and returnMessage into the result body
- **formatToAddressDataBody(...)** - Formats the idTypeCode, idNumber, withholdFormCode, stationNumber, name, and addresses into the addressData body
- **findAddrRequestTransactions(responseBody)** - Finds all the AddrRequest transactions in a response body, and returns an array of transactions[i]
- **findCalcRequestTransactions(responseBody)** - Finds all the CalcRequest transactions in a response body, and returns an array of transactions[i]

HTTP Response Handler Format

Response handler scripts are used to test the responses of the individual tests.

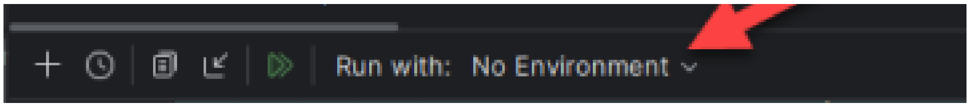
```
> {%
    // Import the required functions
    ...

    import {<function name>} from "./testing-functions"
    ...

    // Call the function/s with client, response.body, and any other necessary parameters ...
    <function name>(client, response.body, <any additional parameters>,...);
    ...
%}
```

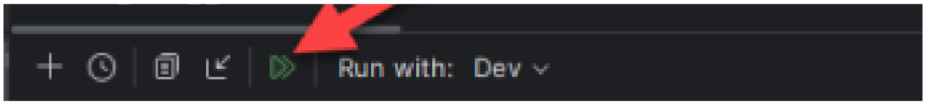
Running the tests

1. Open project in IntelliJ.
2. Create a private env.json file based on the public http-client.env.json file format.
3. In the http-client.private.env.json file, enter the credUsername and credPassword into all environments.
4. Open the http test file you would like to run.
5. In the desired http test file, select the run environment at the top.



Run with: No Environment

9a. To run all requests in the file, click the double start button at the top.

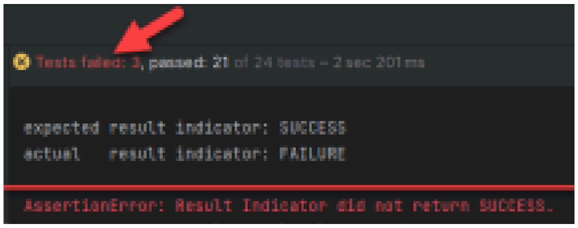


Run with: Dev

9b. To run a single request from within the file, click the single start button next to a request.
10. Test results will get displayed in the “Services” menu.

Failed Tests

If one or more tests fail, a "Tests failed" error will appear with the number of failed tests. For each failed test, there will be an AssertionError with a description of the failed test.



```
Tests failed: 3, passed: 21 of 24 tests – 2 sec 201 ms

expected result indicator: SUCCESS
actual   result indicator: FAILURE

AssertionError: Result Indicator did not return SUCCESS.
```