UI Testing Tools Guidelines Q3 2020

Unit and Integration Tests for Folio UI modules

1. For all new functionality teams have to use RTL and/or Jest to build unit or integration tests.
2. Resources: All UI developers will be utilizing RTL and Jest for creating tests with exception of Stripes Force team and developers who will be building UI end-2-end test suite (see below)
3. Tool Selection:
   a. We are converting to mainstream testing tools adopted by React JS community
   b. Real browser rendering does not required for UI module testing therefor selection of RTL and Jest is appropriate
   c. Onboarding new UI developers will be easier because we’ll be employing tools that are commonly used outside of Folio project
   d. Only RTL/Jest should be used across all UI modules. It will allow developers to switch work from one module to another without learning curve.
4. Transition:
   a. Existing BigTest test cases have to be reimplemented (if applicable) with RTL/Jest according to priorities set by PO.
   b. Teams have to maintain and execute BigTest tests until coverage of 80+% has been achieved with RTL/Jest.
   c. Teams have to maintain 2 separate coverage reports for both BigTest and RTL/Jest
   d. UI module build pipeline has to include additional step for RTL/Jest tests.
   e. BigTest step can be removed from the pipeline only after RTL/Jest step is reporting 80%+ code coverage.

Stripes

1. All Stripes modules should use next release of BigTest.
2. Tool selection:
   a. It is important to test Stripes components rendering in the real browser
   b. BigTest has the best browser support as well as iOS
   c. BigTest project has adequate functionality available to start work now
   d. Allows testing of responsive design features as well as rendering on the iPad
   e. Next version of Big Test requires no UI bundle instrumentation
   f. Tests can be run against any UI bundle as long as URL access is available
3. Transition:
   a. Stripes Force team has to migrate existing BigTest test suite to the new version.
4. Resources:
   a. Only developers working with Stripes will need to learn how to use new version of BigTest.
   b. Frontsite will allocate resources to assist with this transition

Replacement of NightmareJS End-2-End Tests

1. Replace NightmareJS (end-2-end) tests with next release of Big Test
2. Tool selection notes: (see tool selection notes for Stripes)
3. Transition:
   a. Continue to maintain and execute Nightmare JS tests until BigTest test suite provides greater coverage in terms of functionality and test cases.
   b. PO will select test cases from Bug Fest test plan to be automated.
   c. Jenkins pipeline should execute BigTest tests against folio-testing system on the daily basis.
   d. Test suite will be designed to run on any system that deploys platform-complete
4. Resources:
   a. Zak needs to maintain NightmareJS test suite until we're ready to drop it.
   b. Folio project will need to allocate UI developer(s) to implement BigTest test suite.
   c. Frontsite will allocate resources to assist with creation of this test suite.

BigTest Considerations

1. BigTest tool is very promising. However, it is in the Beta phase of v1.0 release.
2. We need to utilize Big Test without overcommitting significant Folio dev resources
3. Approach was chosen to balance the need for browser based end to end tests and exposure to smaller, less mature tooling
1.0 BETA  
Sept 15th
Windows support
diagnostic error messages

1.0  
Nov 5th
Bugfixes
Built-in interactors

1.1  
Q1 2021
iOS agent
Android agent
Node agent

1.2  
Q2 2021
Web capture
Web UI & VSCode Test Runner and debugger