End to End Testing of the Primo GUI

Matthias Einbrodt
Free University of Bozen-Bolzano

What?

A methodology used to test whether the flow of an application is performing as designed from start to finish.

e.g. Login in, requesting a book, performing a search ...

Why?

- Many customizations of our Primo GUI (Javascript and CSS)
- Will new Primo functionalities break them and do they break new Primo functionalities?
- Time savings in testing

How?

- Configure testing framework and runner
- Write test cases consisting of:
  - commands to be executed against the browser (e.g. open/resize window, click a button wait until element is visible/hidden etc.)
  - verifications/assertions (e.g. does element have a certain class, is element hidden/visible etc.)
- Feed them to a test runner and go
- Analyze reports generated by test runner and/or refine tests

Implementation

- Nightwatch.js (http://nightwatchjs.org/)
  - node.js powered
  - based on Selenium WebDriver API which in turn uses build in automation capabilities of each browser
  - Good Github Ratings (~3200 Stars)
  - Actively developed (last commit 9 days ago)
  - Active user-driven support (last post on Google Group on the 04.09.2015)
  - Extendable (custom assertions, commands, reports)
  - Easily to setup and run

Implementation

- First locally executed via command line on computer
- We wanted periodic runs, better human-readable reports, e-mail notifications after failed tests = Jenkins (Continuous Integration Tool)

How to setup Nightwatch and Jenkins

http://juristr.com/blog/2014/02/nightwatch-test-automation/

Reflections

- Seeing all tests passing is extremely reassuring
- Sometimes false errors due to interaction with external systems (CAS) and ?
- We still test some areas manually
- Even small changes in GUI can break tests
- Using Cross Browser Testing services would be the real deal (possible with Jenkins and Nightwatch)

Thank you

matthias.einbrodt@unibz.it