Advanced Acceptance tests

Aurelijus Banelis

About me

Software developer

aurelijus@banelis.lt
aurelijus.banelis.lt

Why we need tests?

Usually: less stress during deployment, lower risk to loose money because of error / failure

Unit vs Acceptance tests

**Many scenarios,
but tested independently**

**Few long and complex
end-to-end tests**

In this presentation "Acceptance tests" means end-to-end test, that can be run on production system

When you use Acceptance tests

1. **Debug**
2. **Speed**
3. **Changes**
4. **Iteraction**

Consider those before choosing right tools or practices

At least from my experience, those fields were most important when actually writing and using acceptance tests

Debug

Makes debugging harder:
So choose tools accordingly

Sync/async

Continue from the middle

Local, production

First thing when you switch from unit to acceptance - it is much harder to debug and find why your test is failing

Debug

Problem: tests are failing

Some validation error.
Exception silently ignored.
Requests via SSL (wireshark is useless)

Cause of the problem (found after debugging):

JavaScript change field name just before POST

$this->log->crit(__METHOD__ . ':' . __LINE__, "Updated");
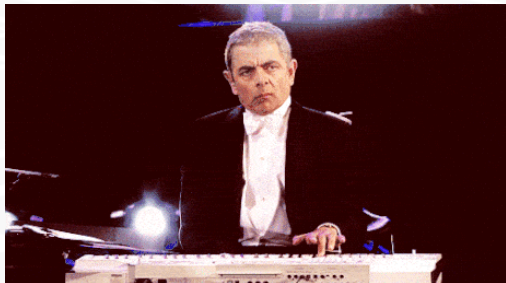
tail -f logfile1.log logfile2.log log/path* | grep CRIT

It could look hacky, but most efficient debugging of acceptance test was: add logging lines into php code and filter output of logs during test execution

Speed

Make tests faster

Chrome vs Phantom vs Firefox

Run on CI in parallel

It was unexpected, that in some cases Chrome (Selenium) driver was faster than Firefox or event PhantomJS. If possible, run your tests in parallel with Continuous Integration tools

Speed

Need to rerun test every time, because:
Unique data needed

Best practice:
Fix in one specific environment/suite/scenario
If successful run whole

vendor/bin/codecept run
--env=testing1-de_DE
cartSuite
PayPal.php::seePayPalDetailsPage

Changes

Solution to sensitive tests:

class="tst-buy js-buy buy"

data-tst-id= "product-123"

DRY, constants, reuse tests

Every small change will not broke acceptance tests, if separate class names will be used for acceptance test, JavaScript and CSS selectors. Do not rely on ids, use custom HTML properties

Changes

JavaScript modal/popup with new field.
Test with primitive/PHP browser.

Search elements by Xpath, CSS, ~~JQuery~~

document.querySelector

class="tsy-modal-new" class="tst-modal-edit"

Acceptance tests frameworks allow to test only one small part. Example shows running codeception test specific environment, suite, file and function - smallest part of acceptance test
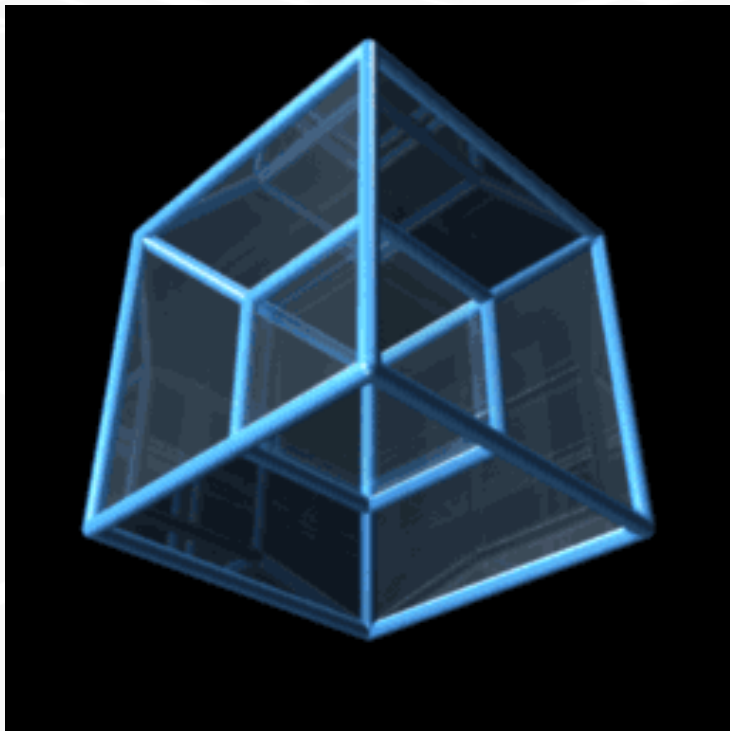
Interaction

Test frameworks simulate only simple interaction:
E.g. click, move, mouse over

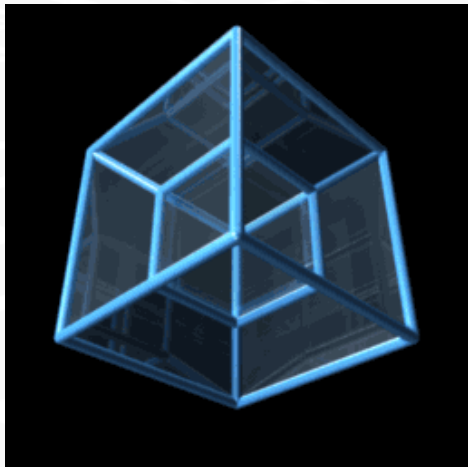Selenium drivers have wait for HTML element to appear functionality:
Wait for Javascript framework
//div[@data-reactid=".0"]

But to simulate more complex user inputs/interaction, you need to:
Call your event functions directly

Interaction

Requirements (complex user input, run on production)
Dragging, touch
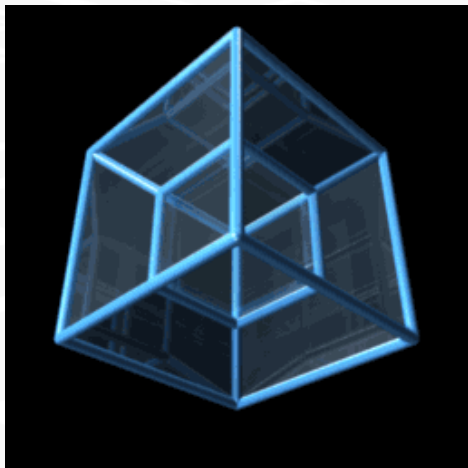## Minified React.js

Problems:
PhantomJs: not all events exits, not all documented, checked even in source
Chrome: touch event invocation not by standards

React.addons.TestUtils.Simulate.*$eventName*(element, event)

Interaction

Dragging, touch
Minified Reac...

PhantomJs not s...
documented, check sou...
Chrome: touch event invoca...
standards

React.min.js does not have TestUtils addon

React.addons.TestUtils.Simulate.*$eventName*(element, event)

Rebuild react locally with one line changed

Summary

Change your software to test it easier

After my journey of using acceptance tests, I realised: You will need to adapt your current code base for acceptance test. Otherwise it could become too hard to maintain them.

Questions

- **Short intro into Acceptance tests**
- **Tips and tricks from practice**
  - **Debug**
  - **Speed**
  - **Changes**
  - **Iteraction**

References

- https://en.wikipedia.org/wiki/Acceptance_testing
- http://codeception.com/
- http://behat.org/
- http://casperjs.org/
- http://www.seleniumhq.org/
- https://devblog.supportbee.com/2014/10/27/setting-up-cucumber-to-run-with-Chrome-on-Linux/
- https://github.com/ariya/phantomjs/tree/master/src/qt/qtwebkit/Source/WebCore/dom
- https://chromium.googlesource.com/chromium/src.git
- https://facebook.github.io/react/docs/test-utils.html

Visuals from wikepedia, devopsreactions.tumblr.com or similar sites

Visuals (borrowed from)

- http://anklestothewall.tumblr.com/post/90539622121/killjoy-cam-woods
- http://allthesupernaturalgifs.tumblr.com/post/31054611364/do-you-have-the-gif-of-zachariah-in-the-bar-with
-