

Selenium

Remote Control

Unit Testing

- Tests modules (usually classes)
- Verifies the public interface (API)
- A test should test just one module
- Great for re-factoring - you are reassured that your code still works

Selenium Versions

- Three different versions (that matter)
- IDE: Firefox extension
- Core: JavaScript library
- RC: Java Proxy (can run in chrome://)

	Core	IDE	RC
Browsers	All	Firefox	Many
Requires server access	Yes	No	No
Multi-domain	No	Yes	Yes
HTTPS	Yes	Yes	No

Use the right one!

Test-Driven Development

- Dogmatic:
 - Tests are written before the code
 - Only code is written for which tests already exist
- Pragmatic:
 - Test are written before or with the code
 - Only most frequent cases being tested
 - All bug-fixes are being tested

Do it! Do it!

Selenium Remote Control

- Requirements:
 - Java Server
 - Properly installed Firefox
 - Selenium Protocol handling (e.g. pear Testing_Selenium)

- Looks easy:

```
$selenium = new Testing_Selenium("*chrome", "http://liip.ch");  
$selenium->start();  
$selenium->open("/");
```

```
$page_title = $selenium->getTitle();  
if ($page_title == 'Liip AG') { echo "YALLAH!\n"; }
```

SimpleTest

- “Classic” Unit testing
- HTTP Requests

- It's easy:

```
class TestValidations extends UnitTestCase {  
    function testValidate() {  
        $this->assertIdentical(true, hlprs_validation::validate('email', 'ebi@liip.ch');  
    }  
}
```

Uhm, ya, simple!

Selenium RC & SimpleTest

- Nice duo:
 - Control the browser, while simpletest is doing the assert work
 - Get nice reports

- Not much left to do:

```
function testLiipTitle() {  
    $this->selenium->open("/");  
    $this->assertEqual('Liip AG', $this->selenium->getTitle());  
}
```

NICE!

Common Stuff

Commands:

- getTitle
- getPath
- getText
- isTextPresent
- isVisible
- click

Locators:

- identifier
- id
- name
- dom
- xpath
- link
- css

Suffixes:

- AndWait
- Not (deprecated with simpletest)
- WaitFor

Not all Core functions in RC

Bamboo

- Blame the build breaker
- Get nice statistics
- Locate errors, when needed (e.g. backend) components fail

Builds on my machine!

Social

- It's NOT a waste of time
 - It's not much extra work when you do it right away
 - Manual testing is reduced to a minimum
 - You know a bug really is fixed (and will remain fixed)
 - But: requires some initial effort
- Get the team on board
 - For the local.ch frontend a bug is only fixed when a test exists
 - Show the value (“I found this bug because of Selenium”)

When will you start?

Links

- <http://openqa.org>
- <http://simpletest.org>
- <http://release.openqa.org/selenium-core/0.8.0/reference.html>
- http://developer.local.ch/php/selenium_class

Thanks to Patrice Neff

He made me believe in Unit Testing and lend me some slides :)

Visit <http://blog.patrice.ch>

Time for questions