Windmill

Automated Testing for Web Applications

Demo!

Requirements

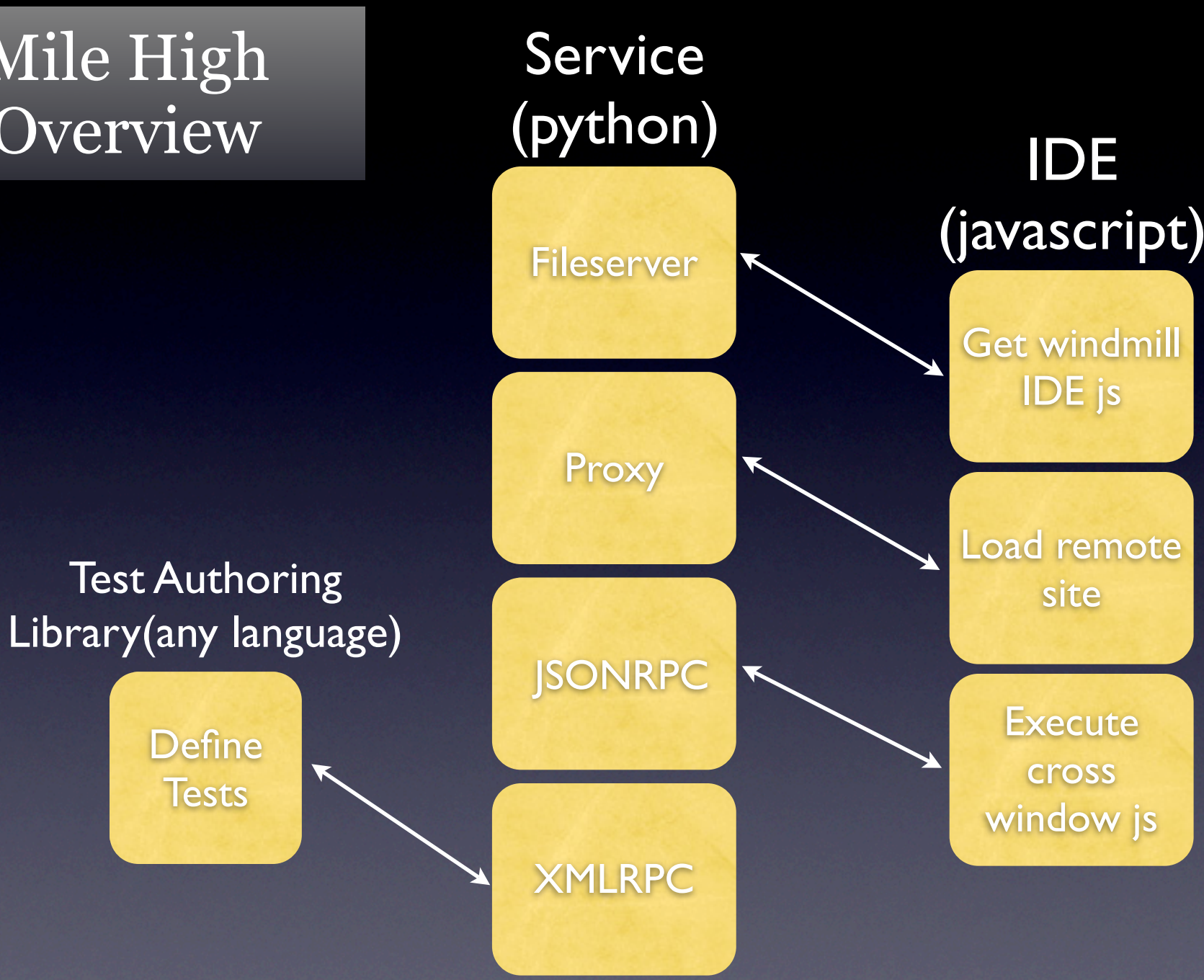
- Quickly build regression tests
- Run single test on all target browsers
- Easily debug tests
- Easily fit in to continuous integration

Other Tools

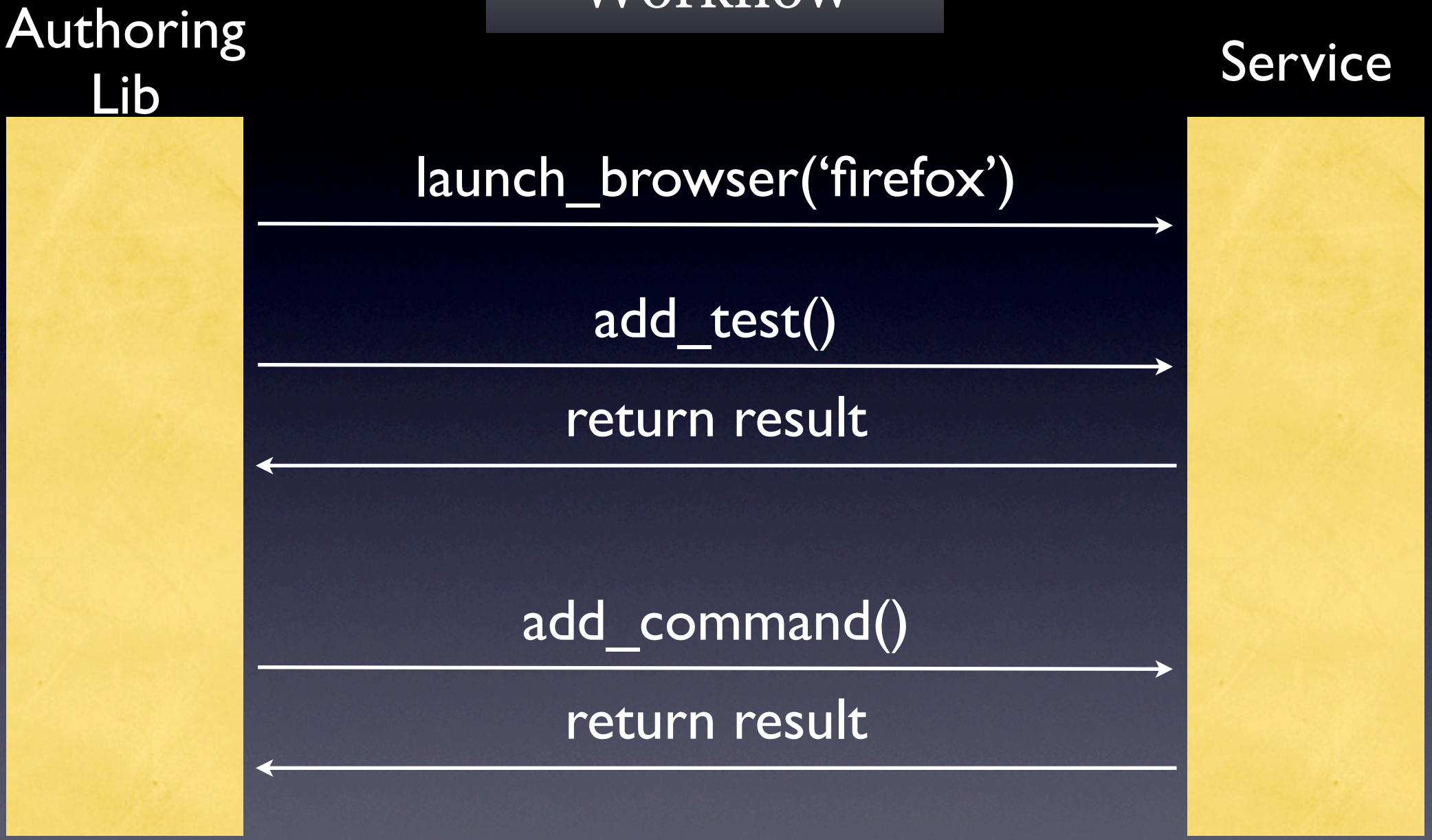
- Selenium, Sahi
  Main alternative, Java, Debugging, Integration
- Watir
  Trusted space, different project for each browser
- WebTest
  Watched demo, didn't appear to fulfill needs

Architecture

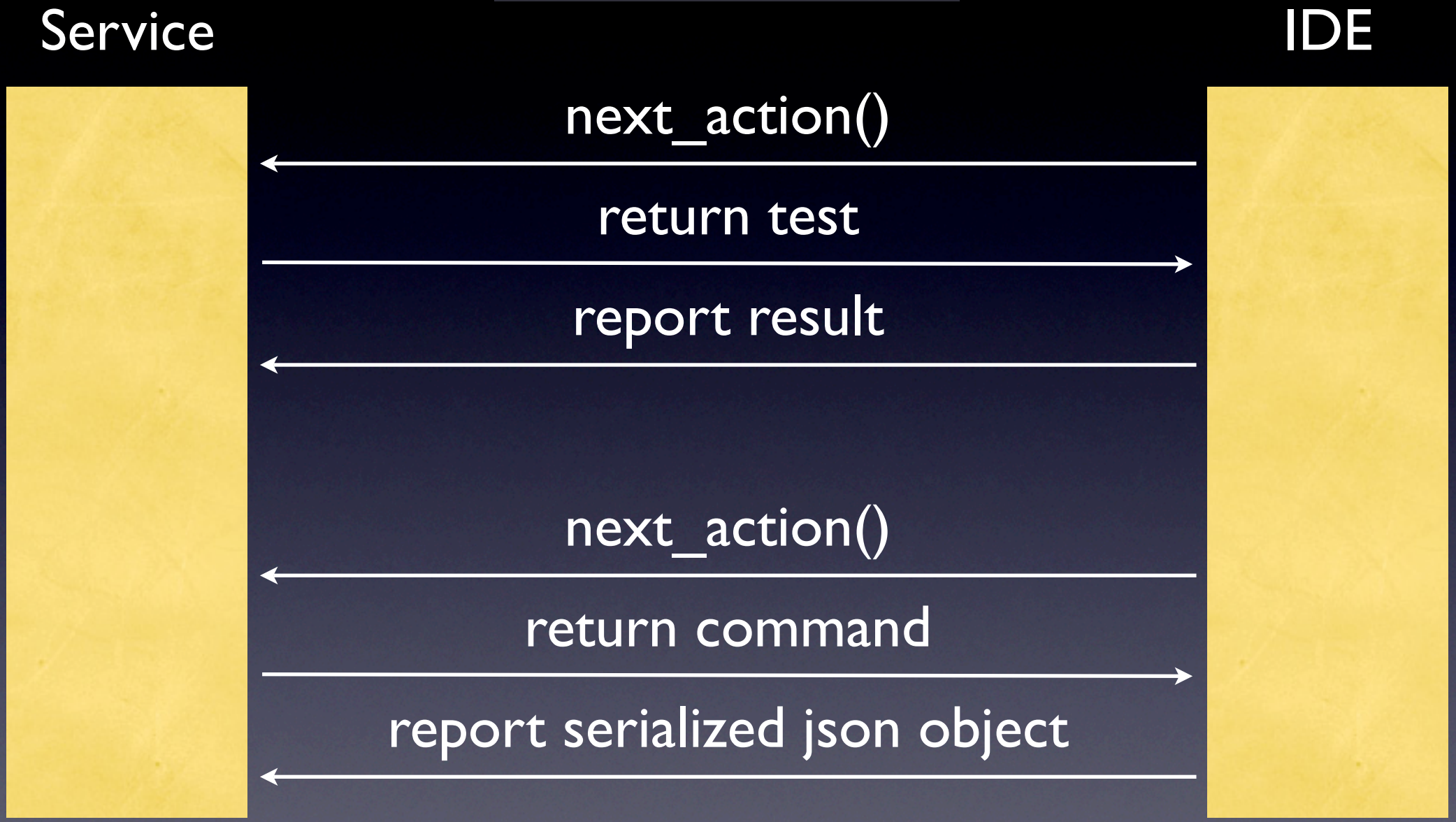
Mile High Overview



XMLRPC Workflow



JSONRPC Workflow



Supporting continuous integration

Service	IDE
Browser Launching and configuration	Build UI and start running tests
Test Input	Test running, performance, debugging
Results Tracking and output	Stop on Failure and interactive debugging
Dynamic debugging information requests	Arbitrary remote javascript execution
Dynamic commands	Command execution

Now the fun stuff

What can I automate?

OOTB support for:

open, click(s), check, radio, wait, type, select, wait, assert,
mouse events, keyboard events

Advanced:

Drag and Drop, Window Management...

DOM Element Access:

id, link, name, jsid, classname, tagname, xpath

Extending the Controller API:

Extensions are simple to write in JavaScript
Extensions are first class citizens

Features

Windmill IDE

- Cross browser test recording
- Multiple layers of support for performance information and metrics
- DOM explorer, Assertion builder
- Javascript shell. (on the list to be enhanced)

Tests

- Creation
  IDE: Recorder/Editor
  Manual test writing
- Formats
  Python, JavaScript, JSON
- Dependencies
  Directory heirarchy
  Setup/TearDown
  Suite Modules

Customizable Reporting

Details:
http://windmill.osafoundation.org/trac/wiki/BookChapter-5-RunningTests

Tests Cont.

- Python
  Server side logic
  One action at a time
  Best for linear linear tests
- JavaScript
  Flexible, dynamic, more control
  Pure JavaScript, helper functionality
  Access to JUM functions
  Answer to conditionals and loops

```
import functest
from windmill.authoring import WindmillTestClient
from windmill.authoring import setup_module, teardown_module


def test_login():
    client = WindmillTestClient(__name__)
    client.type(text=functest.registry['username'], id=u'email')
    client.type(text=functest.registry['password'], id=u'pass')
    client.click(id=u'doquicklogin')
    client.waits.forPageLoad(timeout=u'40000')
```

```
windmill.jsTest.require('shared/test_login.js');

var test_bazThing = function () {
  var nodeId = getSomeDomNodeDynamically();
  windmill.jsTest.actions.click({ id: nodeId });
  jum.assertEquals(nodeId, 'test');
}
```

Service Features

- Extensible logging system (stdlib logging module)
- 3 run modes.
  - "headless" command line. stdout.
  - "shell" mode. ipython or stdlib python shell.
  - wx GUI. PyCrust shell (currently broken, Jacob)
- Browser launching and configuration.
- Server level pre and post hooks for test results parsing, validation, etc.

Status

Flash

- Currently not implemented in Windmill
- Adding support would involve:
  Building a layer between JS and Flash (ExternalInterface hooks)
  Addition of Windmill JS controller methods
- I don't know flash/flex
- Feel free to build this layer :)

http://kb.adobe.com/selfservice/viewContent.do?externalId=tn_15683

Use Cases

- Developer Sanity Check
  Using the test hierarchy developers could run selected, tests/suites, or the entire test repo against the app they are working on to avoid regressions.
- Continuous Integration
  Using build bot, tbox, or a simplified home brew, dedicated machines constantly run test suites against stage app instances. Failures reported to a list, email, etc.
- Push QA Cycle
  While QA is doing manual validation of new features, specific test runs can be fired off on dedicated boxes, or VM's on QA engineer machines. QA sign off requires manual feature validation and automated regression test success.

Demo!

Development

- Proxy domain forwarding improvement
  Social Networks, iFrames
- Simplify install/service stability (Linux, Win)
- XPath implementation in IE
- Documentation, (Slide Quickstart Guide)
- 0.5 Release for Slide Adoption (use src until then)
- Moving the project to GetWindmill.com

Getting Started

- Get Windmill From Source
  http://windmill.osafoundation.org/trac/wiki/BookChapter-1-Install
- Record/Write Some Tests
  http://windmill.osafoundation.org/trac/wiki/BookChapter-3-UsingWindmillIDE#RecordingTests
  http://windmill.osafoundation.org/trac/wiki/BookChapter-4-WritingTests
- Play Your Tests
  http://windmill.osafoundation.org/trac/wiki/BookChapter-5-RunningTests
- Log Those Bugs
  http://windmill.osafoundation.org/trac/newticket
- Thanks for your patience.

Q & A

Let me know your thoughts, needs, ideas:
Email, One-on-One, Meetings?

http://windmill.osafoundation.org