

Test Driven Development in JavaScript (using JsTD and Jasmine)

Elliott Sprehn (Google, Inc.)

esprehn@gmail.com

<http://www.elliottsprehn.com/blog/>

<http://www.elliottsprehn.com/cfbugs/>

<http://code.google.com/p/js-test-driver/>

Testing

I love tests!

JavaScript

I love JavaScript!

Testing JavaScript
was hard.

Behavior Driven Development

- Write stories, speak about behavior
 - *"Profiles: it should show only my friends"*
- Practice TDD
- Tests should be readable
- Tests should be fast

Problems

- Modern web applications contain lots of complex JavaScript (DOM, XHR, ...)
- Thousands of lines of JS code not uncommon
- Not enough unit testing
- Integration tests too slow (ex. Selenium)

Goals

- Tests should be blazing fast
 - Run hundreds of tests in under a second.
- Tests should be easy to write
- Tests should run on every save
- Tests should run in many browsers

JavaScript Test Driver (JsTD)

- **Super Fast!**
- Easy configuration
- Can capture any modern browser
- Easy to automate, or use in your IDE
 - Eclipse, IntelliJ, Maven, Ant, Hudson, ...

Example JsTD Config (YAML)

server: http://localhost:4224

load:

- src/*.js
- tests/*.js

exclude:

- src/ignored.js

Example JsTD Test

```
ProfileTest = TestCase("ProfileTest");

ProfileTest.prototype.testFindFriendsUsingOperators = function() {
  var profileService = new ProfileService();
  profileService.setFriends([
    'Ben Nadel',
    'Simon Free',
    'Sean Corfield',
    'Ray Camden'
  ]);
  assertEquals(
    ['Ray Camden', 'Ben Nadel'],
    profileService.findFriends('camden OR Ben'));
};
```

Asynchronous Tests

```
ProfileTest = AsyncTestCase("ProfileTest");

ProfileTest.prototype.testFindFriendsUsingOperators = function(queue) {
  var profileService = new ProfileService();
  queue.call(function(callbacks) {
    // xhr for /friends.json
    profileService.loadFriends(callbacks.noop());
  });
  queue.call(function() {
    assertEquals(
      ['Ray Camden', 'Ben Nadel'],
      profileService.findFriends('camden OR Ben'));
  });
};
```

Goal: Speed and Dependability

- Encapsulate DOM and IO operations
 - DOM Updates, Animations, XHR, Location, ...
- Pass dependencies to constructors
- Create mocks for tests
- Test logic separate from DOM manipulation

Better: Pass your dependencies, Mock them out

```
ProfileTest = TestCase("ProfileTest");

ProfileTest.prototype.testFindFriendsUsingOperators = function() {
  var xhr = new MockXhrService();
  xhr.expect('GET', '/friends.json').respond([
    'Ben Nadel',
    'Simon Free',
    'Sean Corfield',
    'Ray Camden'
  ]);
  var profileService = new ProfileService(xhr);
  profileService.loadFriends();
  // simulate an async return from the xhr
  xhr.flush();
  assertEquals(
    ['Ray Camden', 'Ben Nadel'],
    profileService.findFriends('camden OR Ben'));
};
```

BDD with Jasmine

- Simple consistent syntax
- No special syntax, just JavaScript code
- Flexible matchers and async support
- Integrates with JsTD
 - <https://github.com/ibolmo/jasmine-jstd-adapter>

Example Jasmine Test

```
describe('ProfileService', function() {  
  it('should allow OR operator in friend searches', function() {  
    var xhr = new MockXhrService();  
    xhr.expect('GET', '/friends.json').respond([  
      'Ben Nadel',  
      'Simon Free',  
      'Sean Corfield',  
      'Ray Camden'  
    ]);  
    var profileService = new ProfileService(xhr);  
    profileService.loadFriends();  
    xhr.flush();  
    expect(profileService.findFriends('camden OR Ben'))  
      .toEqual(['Ray Camden', 'Ben Nadel']);  
  });  
});
```

Describe your app, not your test

```
describe('ProfileService', function() {  
  var xhr, profileService;  
  
  beforeEach(function() {  
    xhr = new MockXhrService();  
    profileService = new ProfileService(xhr);  
  });  
  
  it('should allow OR operator in friend searches', function() {  
    // ...  
  });  
  
  describe('photos list', function() {  
    beforeEach(function() {  
      // ...  
    });  
  
    it('should only show photos that belong to friends', function() {  
      // ...  
    });  
  });  
});
```


Questions

