Grails Testing Plugin

testing in grails is no fun

- Slow (just you wait)
  - Iterative test driven development is hard
- MockFor and StubFor are evil
  - metaClass isn't so nice either
- IDE support is missing
  - I miss the green bar and clickable stack traces

Piragua CONSULTING

Enter: Testing Plugin (aka Grails 1.1)

Plugin

- Grails 1.0.4, 1.0.3 (and below?)

```
grails install-plugin testing
```

Grails Core 1.1

- Built-in support – no need to install plugin

Piragua CONSULTING

the new world

- Defer to writing unit tests instead of integration tests
- Extend `GrailsUnitTestCase` (or subclass) instead of `GroovyTestCase`
- Run right from your IDE

Piragua CONSULTING

constraints testing

```
class AuthorTests extends grails.test.GrailsUnitTestCase {

    Author author

    void setUp() {
        super.setUp()

        // easier in testing-plugin 0.4 (mockForConstraintsTest(Author))
        registerMetaClass(Author)
        grails.test.MockUtils.prepareForConstraintsTests(Author)

        author = new Author(firstName: 'Michael', lastName: 'Scott')
        assertTrue author.validate()
    }

    void testLastNameConstraint() {
        author.lastName = ''
        assertFalse author.validate()
        assertEquals "Error should be present",
            "blank",  //name of constraint
            author.errors["lastName"] //map/property access
    }
```

Piragua CONSULTING

dynamic methods

- mockDomain(Book)
- mockDomain(Book, listOfBooks)

- .list()
- .save()
- .get()
- .findBy() / findAllBy()
- validate()

use mockFor() instead
(or integration test)

Piragua CONSULTING

```
class SearchService {
    def findBook(String title) {
        log.debug "Searching for book with title: ${title}"
        return Book.findByTitle(title)
    }
}
```

Testing:
- -mockLogging
- -mockDomain

```
class SearchServiceTests extends grails.test.GrailsUnitTestCase {
    void testFindBook() {
        mockLogging(SearchService)
        SearchService service = new SearchService()

        Book tripwire = new Book(title:'Tripwire')
        Book stateOfFear = new Book(title:'State of Fear')
        mockDomain(Book, [tripwire, stateOfFear])

        assertEquals tripwire, service.findBook(tripwire.title)
    }
}
```

Piragua CONSULTING

a couple bugs…

- .save() **always** adds an object to the list (even if you're saving an existing item (GRAILS-3553)
- .validate() for unique constraints fails on update to existing item (GRAILS-3552)
- .delete() doesn't remove an item from the list
- log.isDebugEnabled() and log.isTraceEnabled() aren't implemented (GRAILS-3539)
- .list() doesn't support parameters (e.g.max:10)

Piragua CONSULTING

controller testing

- Extend grails.test.ControllerUnitTestCase
- mockFlash
- mockRequest
- mockResponse
- mockSession
- renderArgs
- redirectArgs

Piragua CONSULTING

```
def show = {
        def book = Book.get( params.id )
        if(!book) {
            flash.message = "Book not found with id ${params.id}"
            redirect(action:list)
        }
        else { return [ book : book ] }
    }
```

Testing with:
- redirect with **redirectArgs**
- flash with **mockFlash**

```
void testShow_bookNotFound() {
        mockDomain(Book)
        controller.params.id = 999 // no object with id 999 exists
        controller.show()

        assertEquals('flash message',
                "Book not found with id ${controller.params.id}",
                mockFlash.message)
        assertEquals('redirect action',
                controller.list, redirectArgs.action)
    }
```

Piragua CONSULTING

```
def show = {
        def book = Book.get( params.id )
        if(!book) {
            flash.message = "Book not found with id ${params.id}"
            redirect(action:list)
        }
        else { return [ book : book ] }
    }

void testShow() {
        Book expectedBook = new Book()
        mockDomain(Book, [expectedBook])
        controller.params.id = 1
        Map model = controller.show()
        assertEquals(expectedBook, model.book)
    }
```

Testing with:
-model returned as map

Piragua CONSULTING

```
def save = {
        def book = new Book(params)
        if(!book.hasErrors() && book.save()) {
            flash.message = "Book ${book.id} created"
            redirect(action:show,id:book.id)
        } else {
            render(view:'create',model:[book:book])
        }
    }

void testSave_withErrors() {
        mockDomain(Book)
        // mock out save method for non-happy path
        Book.metaClass.save = {-> return false}

        controller.save()
        assertEquals('render view', 'create', renderArgs.view)
        assertNotNull('book', renderArgs.model.book)
    }
```

Testing with:
- -render with **renderArgs**
- -overriding .save()

Piragua CONSULTING

```
class SearchController {
    def searchService
    def query = {
        return [results:searchService.findBook(params.title)]
    }
}
```

Testing with
-mockFor()

```
void testQuery(){
        def searchServiceControl = mockFor(SearchService)
        Book expectedBook = new Book(title:'A Book Title')
        searchServiceControl.demand.findBook(1) {title ->
            assert expectedBook.title == title
            return expectedBook
        }
        controller.searchService = searchServiceControl.createMock()

        controller.params.title = expectedBook.title
        Map model = controller.query()

        assertEquals(expectedBook, model.results)
        searchServiceControl.verify() // optional
}
```

Piragua CONSULTING

taglib

```
class BookTagLib {
    def formatTitle = {attrs ->
        out << "<u>${attrs.title}</u>"
    }
}

//class BookTagLibTests extends grails.test.TagLibUnitTestCase
void testFormatTitle(){
    String title = 'To Kill a Mocking Bird'
    def resultBuffer = tagLib.formatTitle([title:title])
    assertEquals "<u>${title}</u>", resultBuffer.toString()
}
```

* Workaround required to do this with version 0.3 of testing plugin; version 0.4 works

Piragua CONSULTING

and more…

- Controller test case has support for XML requests; JSON is probably coming, too
- UrlMappingTesting – 0.4 of the plugin
- Filter testing?

- Samples:
  - http://github.com/mjhugo/bookstore/tree/master
  - http://github.com/mjhugo/graina/tree/master/hubbub/test/unit

Piragua CONSULTING

One more thing…

Piragua CONSULTING