

How to test your Grails application

gr8conf.us 2014

Original Title

How to test the *\$%#* out of your rails app

whoami

Colin Harrington
Principal Consultant
Object Partners Inc

@ColinHarrington
colin.harrington@objectpartners.com

Basics

<http://grails.org/doc/latest/guide/testing.html>

grails test-app

```
grails test-app
```

```
-----  
Running Unit Tests...  
Running test FooTests...FAILURE  
Unit Tests Completed in 464ms ...  
-----
```

```
Tests failed: 0 errors, 1 failures
```

Targeting

```
grails test-app SimpleController
```

```
grails test-app *Controller
```

```
grails test-app some.org.*Controller
```

```
grails test-app some.org.*
```

```
grails test-app some.org.**.*
```

```
grails test-app SimpleController.testLogin
```

```
grails test-app some.org.* SimpleController.testLogin BookControll
```

Phase:Type

```
grails test-app unit:
```

```
grails test-app :spock
```

grails test-app *phase:type*

```
grails test-app functional:spock
```

Phases

- unit
- integration
- functional
- custom

Support to create your own Phases!!

Framework

- Services
- Controllers
- Filters
- Domains
- Constraints
- Command objects
- Codecs
- Configs
- GSP
- UrlMappings
- WebFlow

Test mixins

```
rails.test.mixin.TestFor
```

```
rails.test.mixin.Mock
```

```
rails.test.mixin.TestMixin
```

```
rails.test.mixin.support.GrailsUnitTestMixin
```

```
rails.test.mixin.domain.DomainClassUnitTestMixin
```

```
rails.test.mixin.services.ServiceUnitTestMixin
```

```
rails.test.mixin.web.ControllerUnitTestMixin
```

```
rails.test.mixin.web.FiltersUnitTestMixin
```

```
rails.test.mixin.web.GroovyPageUnitTestMixin
```

```
rails.test.mixin.web.UrlMappingsUnitTestMixin
```

```
rails.test.mixin.hibernate.HibernateTestMixin
```

GrailsUnitTestCase

@Deprecated

Example Controller Test

```
import grails.test.mixin.TestFor
import spock.lang.Specification

@TestFor(SimpleController)
class SimpleControllerSpec extends Specification {

    void "test hello"() {
        when:
            controller.hello()

        then:
            response.text == 'hello'
    }
}
```

model

UrlMappings

```
static mappings = {  
    "/actionOne"(controller: "simple", action: "action1")  
    "/actionTwo"(controller: "simple", action: "action2")  
}
```

```
import com.demo.SimpleController  
import grails.test.mixin.TestFor  
import spock.lang.Specification  
  
@TestFor(UrlMappings)  
@Mock(SimpleController)  
class UrlMappingsSpec extends Specification {  
  
    void "test forward mappings"() {  
        expect:  
        assertForwardUrlMapping("/actionOne", controller: 'simple'  
        assertForwardUrlMapping("/actionTwo", controller: 'simple'  
    }  
}
```

@Mock

Real in-memory GORM database (unit tests)

@Mock(Book)

Magic









no SQL support
(GORM API not GORM Hibernate)

Spock Data Driven

http://spock-framework.readthedocs.org/en/latest/data_driven_testing.html

```
class Math extends Specification {  
    def "maximum of two numbers"(int a, int b, int c) {  
        expect:  
        Math.max(a, b) == c  
  
        where:  
        a | b | c  
        1 | 3 | 3  
        7 | 4 | 4  
        0 | 0 | 0  
    }  
}
```

Hierarchy

- ▼  **Object** (*java.lang*)
 - ▼  **GrailsUnitTestMixin** (*grails.test.mixin.support*)
 - ▼  **ControllerUnitTestMixin** (*grails.test.mixin.web*)
 -  **FiltersUnitTestMixin** (*grails.test.mixin.web*)
 -  **WebFlowUnitTestMixin** (*grails.test.mixin.webflow*)
 -  **GroovyPageUnitTestMixin** (*grails.test.mixin.web*)
 -  **FiltersUnitTestMixin** (*grails.test.mixin.web*)
 -  **UrlMappingsUnitTestMixin** (*grails.test.mixin.web*)

GrailsUnitTestMixin

```
static GrailsWebApplicationContext applicationContext  
static GrailsWebApplicationContext mainContext  
static GrailsApplication grailsApplication  
static ConfigObject config  
static MessageSource messageSource  
Map validationErrorsMap = new IdentityHashMap()  
Set loadedCodecs = []
```












ControllerUnitTestMixin

- webRequest
 - request
 - response
 - servletContext
 - groovyPages
 - views
 - session
 - params
 - view
 - model
 - flash
-
- sets up codecs































GroovyPageUnitTestMixin

▼ GroovyPageUnitTestMixin.groovy

▼ GroovyPageUnitTestMixin

- m  bindGrailsWebRequest(): void ControllerUnitTestMixin
- m  clearPageScope(): void
- m  mockTagLib(Class): Object
- m  render(Map): String
- m  applyTemplate(String, Map): String
- m  applyTemplate(StringWriter, Object, Object): void
- m  renderTemplateToStringWriter(StringWriter, Template, Object): Object
- m  assertOutputEquals(Object, Object, Object, Closure): void
- m  assertOutputMatches(Object, Object, Object, Closure): void
- v  pageScope: GroovyPageBinding

UrlMappingsUnitTestMethodMixin

```
▼  UrlMappingsUnitTestMethodMixin.groovy
  ▼   UrlMappingsUnitTestMethodMixin
    m   mockUrlMappings(Class): UrlMappingsHolder
    m   getUrlMappingsHolder(): UrlMappingsHolder
    m   mapURI(String): Object
    m   assertController(Object, Object): void
    m   assertAction(Object, Object, Object): void
    m   assertView(Object, Object, Object): void
    m   assertUrlMapping(Map, Object): void
    m   assertUrlMapping(Map, Object, Closure): void
    m   assertForwardUrlMapping(Object, Object): void
    m   assertForwardUrlMapping(Object, Object, Object): void
    m   assertReverseUrlMapping(Map, Object): void
    m   assertReverseUrlMapping(Map, Object, Closure): void
    m   getControllerClass(Object): GrailsControllerClass
    v  assertionKeys: Object =
```

Perspectives

Unit Testing

No IO in Unit tests!

Unit tests just test your unit!
Mocks/Spies/Stubs

Unit being tested:
Class, Hierarchy, MethodLevel.

Integration Testing

integration of Spring components

Real Database

Spring Container bootstrapped

Transaction wrapper

No HTTP connector.

Functional Testing

Test the whole Application

From the outside
All the layers are there.

As a user would

Browser, API, etc.

Stubs

```
service.metaClass.myMethod = {  
    //does nothing  
    //we don't care if it was called or not  
}
```


Spies

Spy on the input ..

```
service.metaClass.someMethod = { input ->  
    assert input == "expected"  
}
```

Mocks

Mocks

Spies

Stubs

Mocks

- **GrailsMock** and **mockFor**
- in the GrailsUnitTestMethod
- Groovy's **MockFor** (Capital M)
- Spock's Interactions Mock()

MetaClass Magic

Additional Perspectives

- Smoke tests
- performance testing
- security (pen) testing
- 3rd party
- Prod testing/monitoring.
- Continuous Integration
- Occasional Integration
- Cluster testing

Tools

IDE

Jenkins

GROM

Quick feedback

Script to open test-reports.

Plugins

Plugin Portal

<https://grails.org/plugins/>

Test build scope

```
test ':fixtures:1.3'
```

Build Scope

```
BuildConfig.groovy  
  
plugins {  
    test ':fixtures:1.3'  
}
```

Data Generator/helper

- [Fixtures](#)
- [Build Test Data](#)
- [Seed Me](#)
- [DBMonster](#)
- [DbUnit-Operator](#)
- [Sham](#)
- [DB Stuff](#)

Code Coverage/Quality

- Coveralls (coveralls.io)
- Clover
- Code Coverage
- GMetrics
- CodeNarc

Running tests

- Concurrent Test Plugin
- Auto test / Grails Guard
- Team City
- Grails Partition Tests
- Funky Test Load Plugin
- Random Test Order

Testing Frameworks

- jUnit (built in)
- Spock !!
- **Fitness**
- **Cucumber**
- **EasyB**

Functional Tests

- GEB
- Webdriver
- Selenium RC
- webtest
- functional-test
- rest-client-builder

JavaScript

Jasmine

jsUnit

Karma

CasperJS

Sumatra

Tools

- Remote Control
- Dumpster Plugin
- greenmail
- WeakSSL
- Spock Env Extensions
- BetaMax

Title

Know your tools
Test from different angles
Practice Practice Practice
Build your own tools

Thank you

@ColinHarrington