How to Test Enterprise Java Applications

Alex Soto Bueno
Computer Engineer
World Is Changing

The World Is Changing So Fast

HTML

{text/html}

Client

Protocol

Server

Persistence

HGBERNATE

MariaDB
The World Is Changing So Fast

Client

Protocol

Server

Persistence

{text/html}
World Is Changing

The World Is Changing So Fast

Client

Protocol

Server

Persistence

{html}

jQuery

MySQL

Hibernate
World Is Changing

The World Is Changing So Fast

Client

Protocol

Server

Persistence

HTML

{text/html}

{app/json}

jQuery

HIBERNATE

MariaDB

LINUX

TAG
The World Is Changing So Fast

Client

Protocol

Server

Persistence

HTML

{text/html}

{app/json}

{app/xml}

jQuery

hibernate

MariaDB

LINUX

TAG
World Is Changing

The World Is Changing So Fast

Client

Protocol

{text/html}

{app/json}

{app/xml}

Server

Persistence

LINUX

TAG

HIBERNATE

MariaDB

HTML

5

jQuery
World Is Changing

The World Is Changing So Fast

Client
Protocol
Server
Persistence

{text/html}
{app/json}
{app/xml}

HTML
5
jQuery

Hibernate

MariaDB

mongoDB
World Is Changing

Testing Frameworks are Changing Too
Bio

Alex Soto Bueno

Computer Engineer
Diagnostic Grifols
Compiler Tutor
Active Blogger & Speaker
Unit Tests – 80% of code
Overview

Unit Tests - 80% of code

Integration Tests - 15% of code
Overview

Unit Tests – 80% of code

Integration Tests – 15% of code

Acceptance Tests – 5% of code
Unit Tests

Tests small pieces of production code
Unit Tests

Tests small pieces of production code

Confidence to Change
Unit Tests

Tests small pieces of production code

Confidence to Change

Quick Feedback
Unit Tests

Tests small pieces of production code

Confidence to Change

Quick Feedback

Documentation
Unit Testing

FIRST Important Rules

PROHIBIT FITXAR CARTELLS
Unit Tests

Fast
Unit Tests

Fast

Resist Temptation
Unit Tests

Isolation

Mocks
Unit Tests

Repeatable
Unit Tests

Self-Validating
Unit Tests

Self-Validating

Finished after 0.013 seconds

Runs: 3/3  Errors: 0  Failures: 0

com.lordofthejars.nosqlunit.demo.mongodb.BookDaoTest [Runner: JUnit 4] (0.001 s)

testFindAllMethod (0.001 s)
testInsertBookMethod (0.000 s)
testFindAllNoElements (0.000 s)
Unit Tests

Self-Validating
Unit Tests

Timely
Unit Tests

Different Layers – Different Frameworks
JS TestDriver  xUnit approach to Javascript

Apache License 2.0
JS TestDriver xUnit approach to Javascript

Apache License 2.0

code
JS TestDriver xUnit approach to Javascript

Apache License 2.0

code

fixtures
JS TestDriver xUnit approach to Javascript

Apache License 2.0

code

fixtures

configuration
Client Side

**JS TestDriver** xUnit approach to Javascript

Apache License 2.0

- JS code
- HTML fixtures
- YAML configuration

JSTestDriver Server
JS TestDriver xUnit approach to Javascript

Apache License 2.0
Sinon.js mocking library for Javascript

BSD License
Sinon.js mocking library for Javascript

```javascript
sinon.stub(jQuery, "ajax").yieldsTo("success", '{"name":"Alex"}');
```
Server Side

Unit Test Framework for Java

Common Public License
Server Side

Parameterized Tests

Data-driven tests
Parameterized Tests

Data-driven tests

```java
@RunWith(Parameterized.class)
public class DataDrivenTest {}
```
Parameterized Tests

Data-driven tests

@Parameters
public static Collection<Object[]> spreadsheetData() throws IOException {
    InputStream ssheet = new FileInputStream("src/test/resources/data.xls");
    return new SpreadsheetData(ssheet).getData();
}
Server Side

Parameterized Tests

Data-driven tests

```java
private double price;
private double tax;
private double finalPrice;

public DataDrivenTest(double price, double tax, double finalPrice) {
    this.price = a;
    this.tax = b;
    this.finalPrice = finalPrice;
}
```
Parameterized Tests

Data-driven tests

```java
@Test
public void shouldCalculateFinalPrice() {
    double finalPrice = calculateFinalPrice(price,
                                             tax);
    assertThat(finalPrice, is(this.finalPrice));
}
```
JUnit Rules

Modify how tests are run and reported.
JUnit Rules

Modify how tests are run and reported.

@ClassRule
class classRule = new ExternalResource();

@Rule
class rule = new TemporaryFolder();
JUnit Categories

Grouping tests by type
JUnit Categories

Grouping tests by type

```java
public interface SlowTests {}
public interface IntegrationTests extends SlowTests {}
public interface PerformanceTests extends SlowTests {}
```
### JUnit Categories

**Grouping tests by type**

```java
@Category(IntegrationTests.class)
public class AccountIntegrationTest {
    @Test
    public void firstIntegrationMethod() {
    }

    @Test
    public void databaseAccessTest() {
    }
}
```
JUnit Categories

Grouping tests by type

```java
@RunWith(Categories.class)
@IncludeCategory(SlowTests.class)
@SuiteClasses( { AccountTest.class, ClientTest.class })
public class LongRunningTestSuite {}
```
Hamcrest is about expressiveness
Hamcrest is about expressiveness

```java
String officerRank = getRankByDate(officer);
List<String> ranks = ["Captain", "Commander"];
assertEquals(true, ranks.contains(officerRank));
```
Hamcrest is about expressiveness
Hamcrest is about expressiveness

```java
String officerRank = getRankByDate(officer);
assertThat(officerRank,
isOneOf("Captain","Commander"));
assertThat(age, is(10));
assertThat(name, is(not("Alex"));
assertThat(calculated, isNotANumber());
```
Server Side

Simple & Clean Mock API

MIT License
@Mock private OfficerDao officerDao;

... 
MockitoAnnotations.initMocks(this);
...

when(officerDao.findOfficerByAge(22))
.thenReturn(newOfficer);

when(officerDao.findOfficerByAge(greaterThan(50)))
.thenThrow(new OldOfficerException());
RDBMS are Homogeneous
NoSQL are Heterogeneous
Persistence Side (SQL)

Avoid HIT Database
Persistence Side (SQL)

Avoid HIT Database

In-Memory Databases
Persistence Side (SQL)

Avoid **HIT** Database

In-Memory Databases

- Apache Derby
- HyperSQL
- H2

License:
- Apache License 2.0
- BSD License *
- MPL 1.1 & EPL 1.0
Isolation problem with Database
Insert–Delete Problem
Persistence Side (SQL)

Isolation problem with Database
Insert–Delete Problem

GNU LGPL 2.1
Isolation problem with Database
Insert–Delete Problem

Define base data set

GNU LGPL 2.1
Persistence Side (SQL)

Isolation problem with Database
Insert–Delete Problem

Define base data set

@Before
DatabaseOperation.CLEAN_INSERT.execute(connection, dataSet);
Persistence Side (NoSQL)

- MongoDB
- Cassandra
- Neo4j
- CouchDB
- HBase
- DEX

Types:
- Document
- Column
- Graph
Persistence Side (NoSQL)

No Embedded (in-memory) Mode (SERVER – 1153)

No Test Isolation Framework
Persistence Side (NoSQL)

Neo4j
the graph database

GPL v3 & AGPL v3

Embedded (in-memory) Mode
Partially supported
Persistence Side (NoSQL)

DBUnit for NOSQL

Apache License 2.0
Persistence Side (NoSQL)

DBUnit for NOSQL

MongoDB
Neo4J
CouchDB
Cassandra

Apache License 2.0
Integration Tests

Test Collaboration Between Components
Integration Tests

Test Collaboration Between Components

Database

IO system

Special Environment Configuration
Integration Tests

Common Problems in JEE 6

CDI – @EJB, @Inject, @PersistenceContext
Integration Tests

Common Problems in JEE 6

CDI – @EJB, @Inject, @PersistenceContext

Container Lifecycle Management

Create Deployment Archive
Integration Tests

Common Problems in JEE

CDI – @EJB, @Inject, @PersistenceContext

Container Lifecycle Management

Create Deployment Archive

Do NOT Mock
Integration Tests

Arquillian brings tests to runtime

- Manage Lifecycle of Container
- Create/Deploy MicroDeployments
- Enriching Test Cases
- Run Tests Inside/Against Container
- IDE Friendly

Apache License 2.0
Integration Tests

Arquillian brings tests to runtime

Manage Lifecycle of Components

Create/Deploy MicroDeployments

Run Tests Inside/Against Container

Enriching Test Cases

No More Mocks

IDE Friendly

Apache License 2.0
Integration tests
Integration tests
Integration tests
Integration tests
Integration tests
Integration tests
Acceptance Tests

Test Customer Requirements
Acceptance Tests

User Story

everyday language sentence
adds business values

As a <role>,
I want <goal/desire>
so that <benefit>
As an administrator, I want to add new books to a collection, so users can borrow them.
Acceptance Tests

Acceptance Criteria

story completeness
Acceptance Tests

Acceptance Criteria

story completeness

Administrator can add new books.
Administrator can categorise books.
User can borrow books added by administrator.
Acceptance Tests

Concrete Examples

2. Assign J.R.R Tolkien as author.
3. Set Book Description.
4. Add Book to System.
5. Check that Book is shown in screen.
Acceptance Tests
Acceptance Tests

JEE UI is Web

How to access to web elements
Acceptance Tests

JEE UI is Web

How to access to web elements

AT categorised by user stories

How to organize user stories and specs
Acceptance Tests

Thucydides
Acceptance Tests

Thucydides

Selenium 2 Extension

Apache License 2.0
Acceptance Tests

- Thucydides
- Selenium 2 Extension
- Organise tests and stories
Acceptance Tests

Thucydides

Selenium 2 Extension

Organise tests and stories

Record/Report test execution

Apache License 2.0
Acceptance Tests

Thucydides

Selenium 2 Extension

Organise tests and stories

Record/Report test execution

Measure Functional Cover
Acceptance Tests

Choose a user story
Acceptance Tests

Choose a user story

Implement PageObject/s
Acceptance Tests

Choose a user story
Implement PageObject/s
Implement Test Step Library
Acceptance Tests

Choose a user story
Implement PageObject/s
Implement Test Step Library
Implement User Story following A.C
Continuous Integration

Different Kind of Tests Exist
Continuous Integration

Different Kind of Tests Exist

Only Unit tests are run by developers
Continuous Integration

Different Kind of Tests Exist

Only Unit tests are run by developers

Continuous Integration System
Continuous Integration

Provides CI services

Build Jobs are Tasks
Continuous Integration

Build Pipeline Plugin
Continuous Integration

Build Pipeline Plugin

Create chains of build jobs.
Continuous Integration

Build Pipeline Plugin
Create chains of build jobs.
Last Note

No Time No Tests
No Time No Tests
Fix an error during Production Time
Last Note

No Time No Tests

Fix an error during Production Time

Continuous Delivery
Thanks
Questions

Questions?
Questions

Alex Soto Bueno
asotobu@gmail.com
www.lordofthejars.com
@alexsotob
Credits

http://www.flickr.com/photos/55182004@N04/5518854399/sizes/o/in/photostream/
http://www.flickr.com/photos/wildlifewanderer/6133286089/sizes/o/in/photostream/
http://www.flickr.com/photos/luccawithcheese/3180399980/sizes/o/in/photostream/
http://www.flickr.com/photos/wildlifewanderer/6133286089/sizes/o/in/photostream/
http://www.flickr.com/photos/joachim_s_mueller/6913623388/sizes/o/in/photostream/
http://www.flickr.com/photos/dieselpower83/7078627097/sizes/o/in/pool-62802336@N00/
http://www.flickr.com/photos/soyignatius/2476388983/sizes/o/in/photostream/
http://www.flickr.com/photos/flatbushgardener/440319534/
http://www.flickr.com/photos/justinwkern/3597152639/
http://www.flickr.com/photos/tolgasoran/5154570906/
http://www.flickr.com/photos/pennuja/5363515039/
http://www.flickr.com/photos/350org/5069209808/sizes/o/in/photostream/
http://www.flickr.com/photos/lliurealbir/3484066523/sizes/o/in/photostream/
http://www.flickr.com/photos/windsordi/3467758700/sizes/o/in/photostream/
How to Test Enterprise Java Applications

Alex Soto Bueno
Computer Engineer