Testing the Data Access Layer

Dan Geabunea
SENIOR SOFTWARE DEVELOPER

@romaniancoder www.romaniancoder.com

Overview

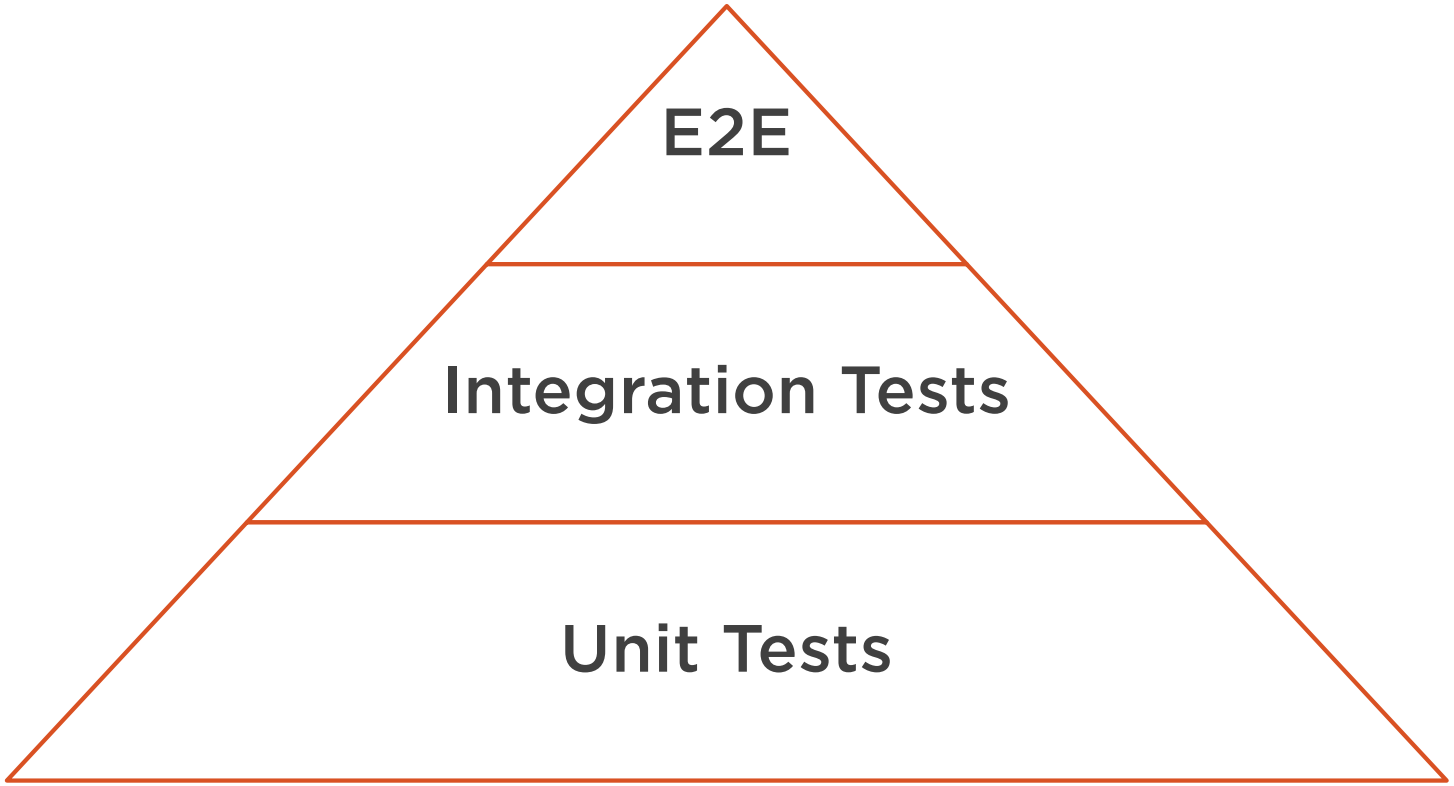
Integration testing for the DAL

Creating Mongo integration tests in Spring applications

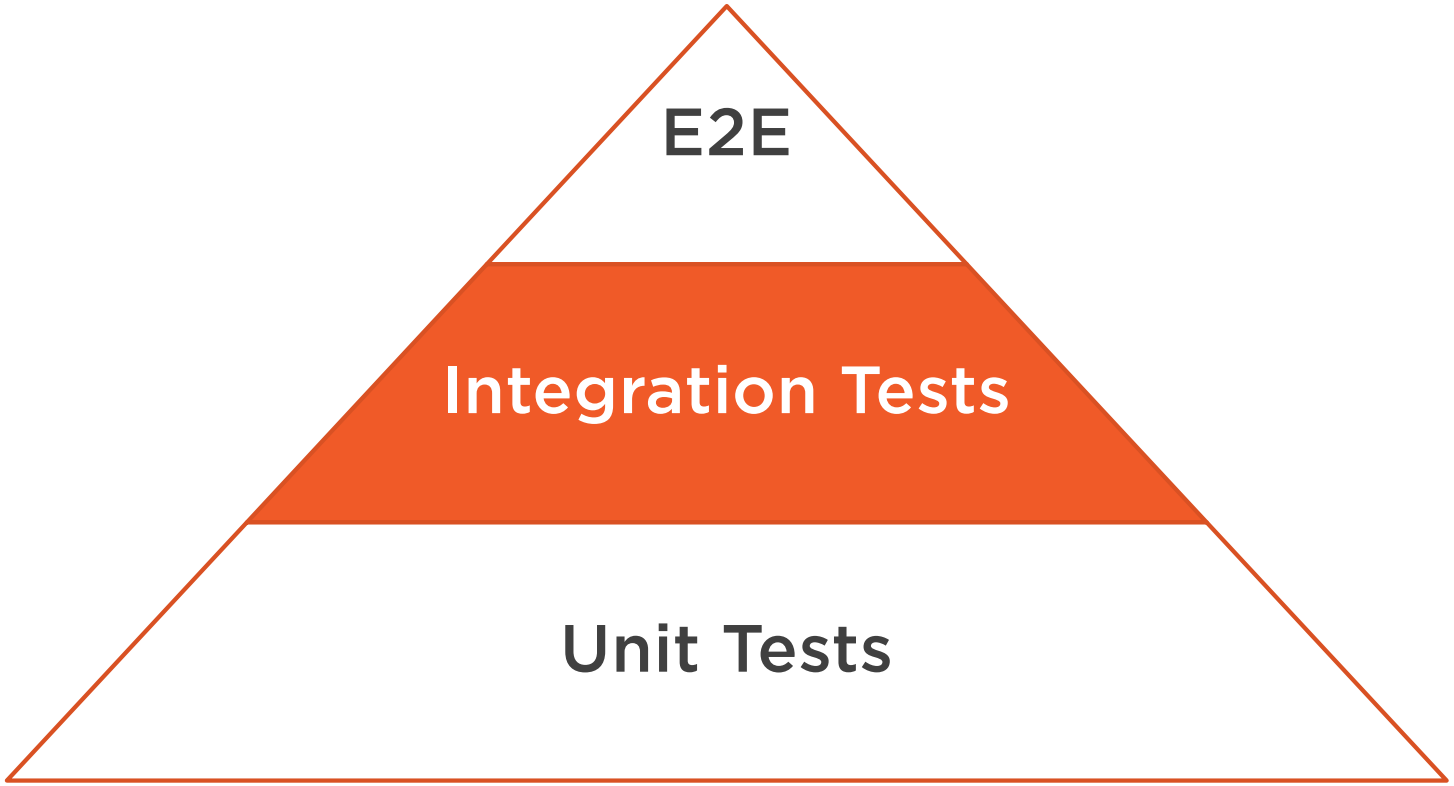
Demo: Implementing Mongo integration tests in Spring applications

Integration Testing for the DAL

Testing Pyramid



Testing Pyramid



Integration Testing

The type of testing where individual pieces of software are combined and executed as a group.

“Why would I want to test the data access layer?”

You

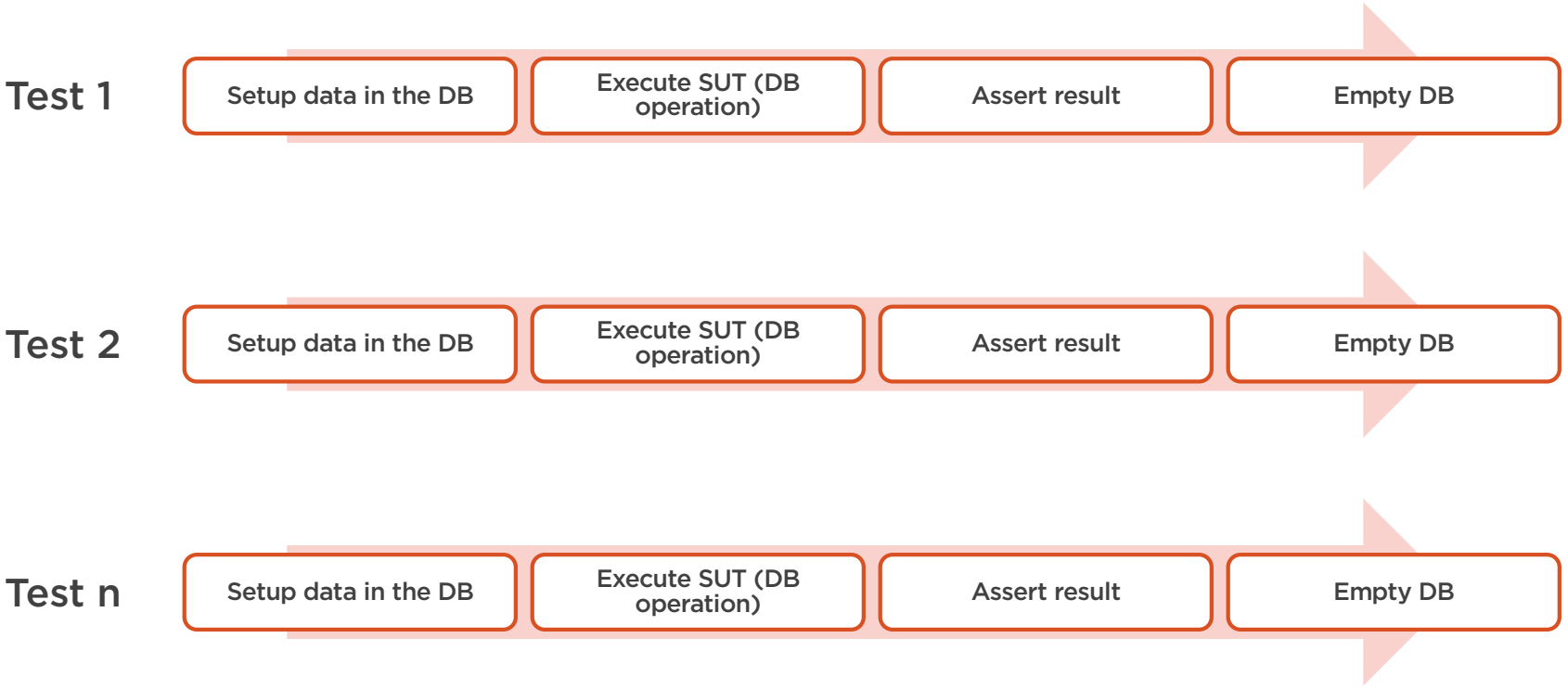
Reasons to Test the DAL

Data is at the core of most enterprise software applications

Graceful handling of edge cases

Easy to accomplish in Spring

Data Access Test Flow



Tests should be predictable; if the data in the DB is volatile, your tests will be flaky

“So, do we have to test every piece of code that calls the DB?”

You

What to Test

AirportService.java

```
List<Airport> res = mongoTemplate
      .findAll(Airport.class)
```

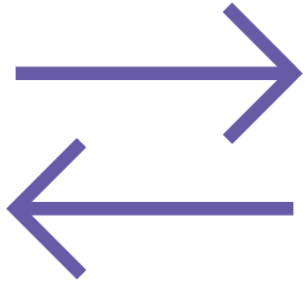
AirportRepository.java

```
@Query("{'country' : ?0}")
List<Airport> byCountry(String c)
```

What to Test

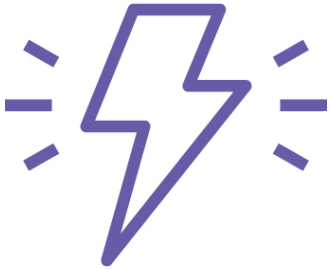
Complex queries

Watch out for strings, complex criteria and projections



Object <-> Document

Mappings, especially custom ones



Lifecycle events

If you have logic like cascading

Creating Mongo Integration Tests in Spring Applications

Embedded vs. Standalone

Embedded Mongo Database

Standalone Mongo Database

Embedded vs. Standalone

Embedded Mongo Database	Standalone Mongo Database
No dependencies needed	Need a dedicated Mongo database
Integrates with CI/CD pipelines easily	Increased set-up complexity for CI/CD
Faster tests	Slower tests
It’s not an official Mongo product	You discover how your application behaves in production like environments
It’s not quite “integration” testing	

“So, what should I choose? Running tests with an embedded DB or with a standalone DB?”

You

Start with an embedded DB
and progress to a
standalone DB before
shipping to production

@DataMongoTest

Annotation that can be used for a MongoDB test that focuses **only on MongoDB components**. Using this annotation will disable full auto-configuration and instead apply only configuration relevant to MongoDB tests.

Running Tests Using an Embedded DB

pom.xml

```xml
<dependency>
    <groupId>de.flapdoodle.embed</groupId>
    <artifactId>de.flapdoodle.embed.mongo</artifactId>
    <scope>test</scope>
</dependency>
```

AircraftTests.java

```java
@DataMongoTest
@ExtendWith(SpringExtension.class)
@Category("integration")
class DatabaseIntegrationTests {
    @Autowired MongoTemplate mongoTemplate;

    // to be continued
}
```

AircraftTests.java

```
class DatabaseIntegrationTests {

    …

    @BeforeEach public void beforeEach() {
        this.mongoTemplate.insertAll(aircraft);
    }


    @AfterEach public void afterEach() {
        this.mongoTemplate.dropCollection(Aircraft.class);
    }

    …

}
```

AircraftTests.java

```
class DatabaseIntegrationTests {

    …

    @Test
    public void findByMinAircraftNbSeatsShouldWork() {
        List<Aircraft> result = mongoTemplate.findAll(Aircraft.class);
        assertEquals(3, result.size());
    }

    …

}
```

What Happened?

Running Tests Using a Standalone DB

TestConfiguration.java

```java
@TestConfiguration
public class TestConfiguration {
    @Bean public MongoDbFactory mongoDbFactory() throws Exception {
        return new SimpleMongoClientDbFactory("<test-db-mongouri>");
    }

    @Bean public MongoTemplate mongoTemplate(MongoDbFactory factory) {
        return new MongoTemplate(factory);
    }
}
```

AircraftTests.java

```java
@DataMongoTest
@ExtendWith(SpringExtension.class)
@Category("integration")
@Import(TestConfiguration.class)
class DatabaseIntegrationTests {
    @Autowired MongoTemplate mongoTemplate;   // points to provided DB

    // the rest is the same
}
```

What Happened?

Demo

Demo: Implementing Mongo integration tests in Spring applications

Summary

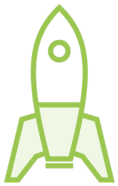
Database integration tests should be done for complex queries, lifecycle events and mappings

Keep tests stable, always populate and clear the test database

Start with an embedded database and progress to standalone

Course Summary

Skills



Get started with Spring and Mongo

Document references & lifecycle events

Spring Mongo annotations & CRUD operations

Data migrations

Repositories

Integration testing

You now have all the skills to start creating enterprise applications with Spring and MongoDB

Where to Go from Here

Nuri Halperin

Introduction to MongoDB

Nuri Halperin

MongoDB Administration

Code Samples

https://github.com/dangeabunea/pluralsight-spring-mongodb

Dan Geabunea

Let's get in touch

- @romaniancoder
- https://ro.linkedin.com/in/dangeabunea
- www.romaniancoder.com

Other Pluralsight course

- SOLID Software Design Principles in Java