Testing of Spring

Mattias Severson

Mattias

mattias.severson@jayway.com

http://blog.jayway.com/

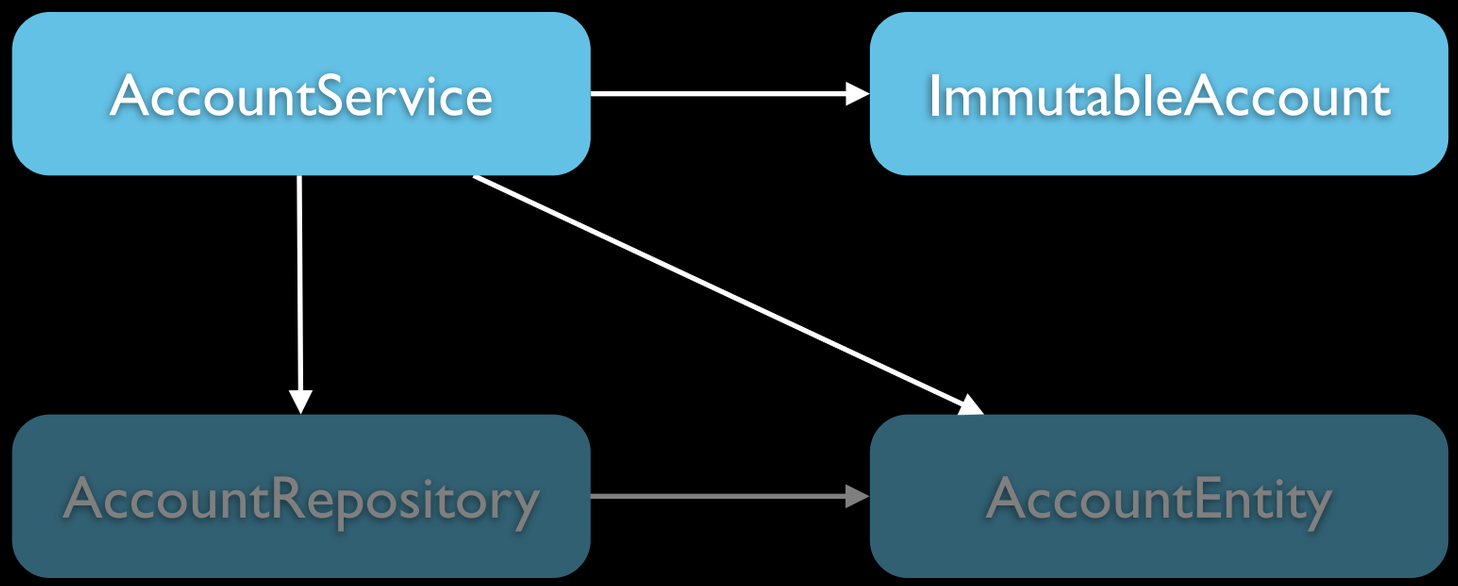
Agenda

- Basic Spring Testing
- Embedded Database
- Transactions
- Profiles
- Controller Test

Bank App

JAYWAY

JAYWAY


AccountService
ImmutableAccount
AccountRepository
AccountEntity

JAYWAY

Basics

jUnit test

```
public class ExampleTest {

    Example example;

    @Before
    public void setUp() {
        example = new ExampleImpl();
    }

    @Test
    public void testDoSomething() {
        example.doSomething();
        // verify ...
    }
}
```

JAYWAY

@Autowired

```
public class ExampleTest {

    @Autowired
    Example example;

    @Test
    public void testDoSomething() {
        example.doSomething();
        // verify ...
    }
}
```

JAYWAY

@ContextConfiguration

```
public class ExampleTest {

    @Autowired
    Example example;

    @Test
    public void testDoSomething() {
        example.doSomething();
        // verify ...
    }
}
```

JAYWAY

@ContextConfiguration

```
@ContextConfiguration("classes=TestConfig.class")
public class ExampleTest {

    @Autowired
    Example example;

    @Test
    public void testDoSomething() {
        example.doSomething();
        // verify ...
    }
}
```

JAYWAY

@ContextConfiguration

```
@ContextConfiguration("/application-context.xml")
public class ExampleTest {

    @Autowired
    Example example;

    @Test
    public void testDoSomething() {
        example.doSomething();
        // verify ...
    }
}
```

JAYWAY

SpringJUnit4ClassRunner

```
@ContextConfiguration("/application-context.xml")
public class ExampleTest {

    @Autowired
    Example example;

    @Test
    public void testDoSomething() {
        example.doSomething();
        // verify ...
    }
}
```

JAYWAY

@ContextConfiguration

- Caches ApplicationContext
  - *unique* context configuration
    - within the same *test suite*

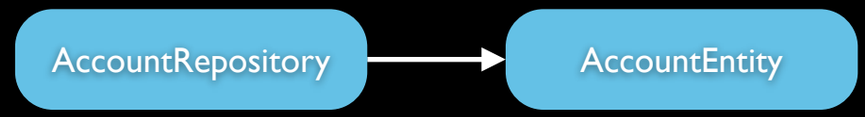
- All tests execute in the same JVM

@ContextConfiguration

- @Before / @After
  - Mockito.reset(mockObject)
  - EasyMock.reset(mockObject)

- @DirtiesContext

JAYWAY



Embedded DB

JAYWAY

XML Config

```
<jdbc:embedded-database id="dataSource" type="HSQL">
    <jdbc:script location="classpath:db-schema.sql"/>
    <jdbc:script location="classpath:db-test-data.sql"/>
</jdbc:embedded-database>
```

Demo

JAYWAY

Java Config

```
@Configuration
public class EmbeddedDbConfig {

    @Bean
    public DataSource dataSource() {
        return new EmbeddedDatabaseBuilder()
            .setType(EmbeddedDatabaseType.HSQL)
            .addScript("classpath:db-schema.sql")
            .addScript("classpath:db-test-data.sql")
            .build();
    }
}
```

JAYWAY

Transactions

Tx Test

```
@RunWith(SpringJUnit4ClassRunner.class)
@ContextConfiguration("/application-context.xml")
public class ExampleTest {

    @Test
    public void testDoSomething() {
        // call DB
    }

}
```

JAYWAY

Tx Annotations

@TransactionConfiguration

@BeforeTransaction

@AfterTransaction

@Rollback

Demo

JAYWAY

Spring Profiles

XML Profiles

```
<beans ...>


        <bean id="dataSource">
            <!-- Test data source -->
        </bean>


        <bean id="dataSource">
            <!-- Production data source -->
        </bean>


</beans>
```

JAYWAY

Java Config Profile

```
@Configuration
public class EmbeddedDbConfig {

    @Bean
    public DataSource dataSource() {
        return new EmbeddedDatabaseBuilder().
            // ...
    }
}
```

JAYWAY

Component Profile

```
@Component
public class SomeClass implements SomeInterface {

}
```

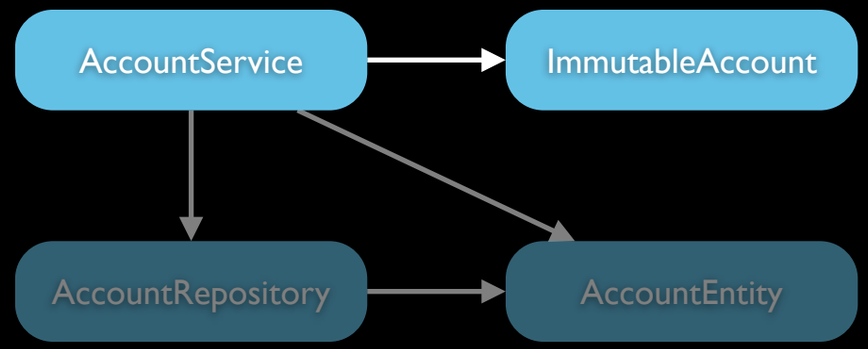
Tests and Profiles

```
@RunWith(SpringJUnit4ClassRunner.class)
@ContextConfiguration("/application-context.xml")
public class SomeTest {

    @Autowired
    SomeClass someClass;

    @Test
    public void testSomething() { ... }
}
```

JAYWAY


AccountService
ImmutableAccount
AccountRepository
AccountEntity

Demo

JAYWAY

web.xml

```
<web-app ...>

    <listener>
        <listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>
    </listener>

</web-app>
```

ApplicationContext

```
AnnotationConfigApplicationContext ctx =
    new AnnotationConfigApplicationContext();
ctx.getEnvironment().setActiveProfiles("someProfile");
ctx.register(SomeConfig.class);
ctx.scan("com.jayway.demo");
ctx.refresh();
```

Env Property

System.getProperty("spring.profiles.active");

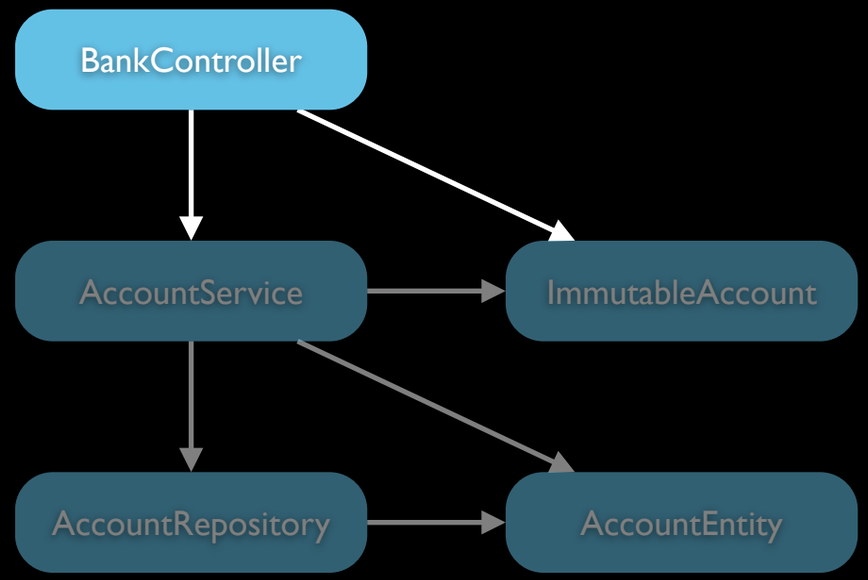
mvn -Dspring.profiles.active=testProfile

Default profiles

```
<beans profile="default">
    <!-- Default beans -->
</beans>
```

```
ctx.getEnvironment().setDefaultProfiles("...");
```

```
System.getProperty("spring.profiles.default");
```



Test Controller

JAYWAY

Demo

JAYWAY

spring-test-mvc

XML config

```
MockMvc mockMvc = MockMvcBuilders
    .xmlConfigSetup("classpath:appContext.xml")
        .activateProfiles(...)
        .configureWebAppRootDir(warDir, false)
        .build();
```

Java config

```
MockMvc mockMvc = MockMvcBuilders
    .annotationConfigSetup(WebConfig.class)
        .activateProfiles(...)
        .configureWebAppRootDir(warDir, false)
        .build();
```

Manual config

```
MockMvc mockMvc = MockMvcBuilders
    .standaloneSetup(new BankController())
        .setMessageConverters(...)
        .setValidator(...)
        .setConversionService(...)
        .addInterceptors(...)
        .setViewResolver(...)
        .setLocaleResolver(...)
        .build();
```

Test

```
mockMvc.perform(get("/")
    .accept(MediaType.APPLICATION_JSON))
        .andExpect(response().status().isOk())
.andExpect(response().contentType(MediaType))
.andExpect(response().content().xpath(String).exists())
.andExpect(response().redirectedUrl(String))
.andExpect(model().hasAttributes(String...));
```

Demo

JAYWAY

Integration tests

jetty-maven-plugin

```
<executions>
    <execution>
        <id>start-jetty</id>
        <phase>pre-integration-test</phase>
        <goals>
            <goal>run</goal>
        </goals>
    </execution>
    <execution>
        <id>stop-jetty</id>
        <phase>post-integration-test</phase>
        <goals>
            <goal>stop</goal>
        </goals>
    </execution>
</executions>
```

maven-failsafe-plugin

IT*.java

*IT.java

*ITCase.java

REST Assured

```
@Test
public void shouldGetSingleAccount() {
    expect().
            statusCode(HttpStatus.SC_OK).
            contentType(ContentType.JSON).
            body("accountNumber", is(1)).
            body("balance", is(100)).
    when().
            get("/account/1");
}
```

JAYWAY

Conclusions

- Basic Spring Testing
- Embedded database
- Transactions
- Profiles
- Controller Test

Questions?

mattias.severson@jayway.com

http://blog.jayway.com/

JAYWAY