Spring Boot on a Diet

A LEANER SPRING TESTING COOKBOOK

MIKAËL FRANCOEUR

Contents

Spring TestContext

Test Slices

Recipes!

Method Validation, Method Security, Caching, Aspects, Kafka

Spring TestContext

- Starts an ApplicationContext for each test class
- Contexts are cached
- To reuse a context:
  - @SpringBootTest, @WebMvcTest, @DataJpaTest, @MockBean, @SpyBean, @ContextConfiguration, @Import, @ImportAutoConfiguration, @TestPropertySource, @ActiveProfiles, @EnableXXX...

Test Slices

- @WebMvcTest, @DataJpaTest, @DataJdbcTest, @JsonTest, @RestClientTest...
- Control
  - Component scan
  - AutoConfiguration

Rolling your own

- @SpringBootTest can be costly
- Start just what you need
- More flexibility
- Less coupling with unrelated concerns

Recipe Template

1. Import relevant parts of the application
   - @SpringBootTest(classes = XXX.class)
   - @Configuration
   - @ComponentScan (gray box)
2. Import relevant AutoConfigurations
   - @ImportAutoConfiguration

Diet Recipes

Method Security

- @PreAuthorize, @PostAuthorize, @PreFilter, @PostFilter, @Secured, @RolesAllowed
- Ingredients
  - @SpringBootTest
  - @Configuration
  - @ComponentScan

Method Validation

- JSR 303
- @Valid, @NotNull, @Pattern, @NotEmpty, @Size, @Email ...

Ingredients

- @SpringBootTest(classes = XXX.class)
- @ImportAutoConfiguration(ValidationAutoConfiguration.class)
- Validation helper

Aspects

- @Before, @After, @Around, @AfterThrowing...
- AspectJ language
- Ingredients
  - SpringBootTest(classes = XXX)
  - @ImportAutoConfiguration(AopAutoConfiguration.class)
  - Helper class with annotations
  - @MockitoBean

Caching

- @Cacheable
- Ingredients
  - @SpringBootTest
  - @Configuration
  - @ImportAutoConfiguration(CacheAutoConfiguration.class)
  - Mock!

Kafka!

- Ingredients
  - @SpringBootTest(classes = MyKafkaConsumer.class)
  - @ImportAutoConfiguration(KafkaAutoConfiguration.class)
  - @EmbeddedKafka

Other Considerations

- Have at least one full @SpringBootTest
- Code smell: @EnableXXX directly on a test
- Use SRP-y configuration classes

Conclusion

- Smaller tests are faster
- Smaller tests are more flexible
- Smaller tests are easier to troubleshoot

Questions

https://github.com/LeMikaelF/testing-spring-boot-confoo-2025