





Spring Test Profiler

Profile & Improve Your Spring Tests for Faster
and More Confident Production Deployments

The Problem

Spring developers waste hours every week waiting for slow test suites

- ✓ Context switching during long test runs
- ✓ Delayed feedback on code changes
- ✓ Slower CI/CD pipelines affecting deployments
- ✓ Reduced confidence in making frequent commits



The Hidden Solution

Spring TestContext Cache

Your tests are like web pages, contexts are like cached resources

✓ 3 test classes sharing 1 context = Fast

✗ 3 test classes creating 3 contexts = Slow

Cache keys affected by: `@TestPropertySource`,
`@MockBean`, `@Profile`



Context Sharing Example

When contexts get reused vs recreated

Shared Context (Good)

```
@SpringBootTest
class UserServiceTest { }

@SpringBootTest
class OrderServiceTest { }
```

Separate Contexts (Bad)

```
@SpringBootTest
class UserServiceTest { }

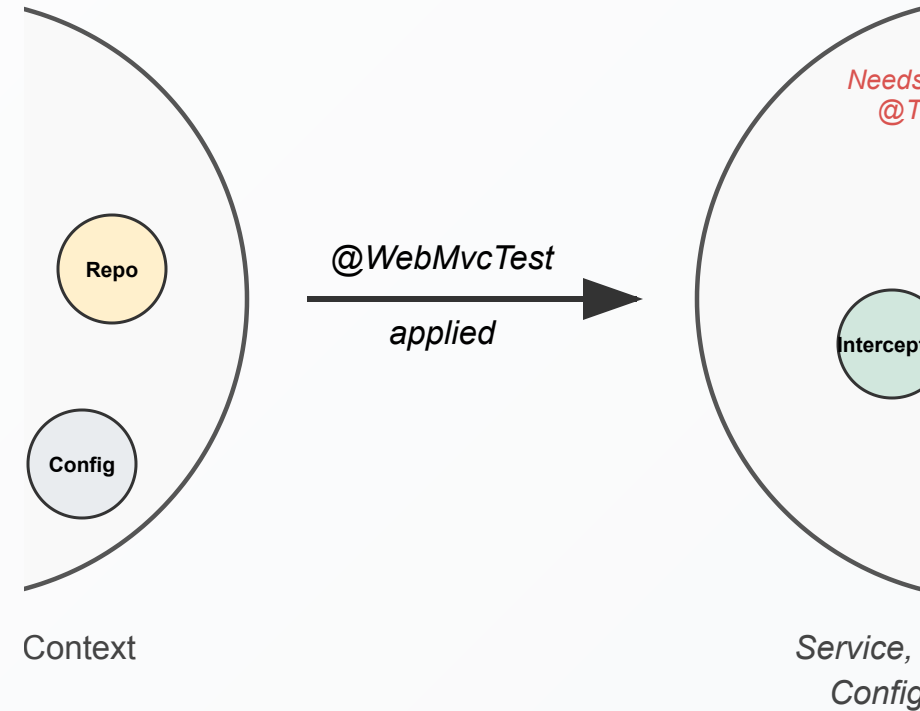
@SpringBootTest
```



| Boot Sliced Context Test with @WebM

Context

Sliced Cont



- Comp: @Component
- Config: @Configuration
- Web: @RestController, Filter, Converter

Spring Test Profiler

The missing observability layer for your Spring tests



✓**Zero Configuration** - Add dependency, run tests

✓**Beautiful Visualizations** - Timeline, statistics, insights

✓**Actionable Insights** - Specific optimization recommendations

✓**Universal Compatibility** - Any Spring Boot version



Live Demo: Setup

Realistic Spring Boot App

REST API with database, mix of test types

Before State: Slow Tests

```
mvn test
# Tests run: 12, Time: 47.2s
```

One-Line Integration



```
<dependency>
  <groupId>digital.pragmatech.testing</groupId>
  <artifactId>spring-test-profiler</artifactId>
  <version>0.0.10</version>
  <scope>test</scope>
```

```
ringBootTest.WebEnvironment.RANDOM
est {
    @ActiveProfiles("integration-test")
    @Import(SomeTestConfiguration.class)
    @ContextConfiguration(initializers
    @SpringBootTest(properties = {"feat
class ShowcaseIT {

    @MockBean
    private OrderService orderService

    @SpyBean
    private CustomerService customers

    @Test
    void shouldInitializeContext(@Aut
        assertThat(applicationContext)
            .isNotNull();
    }
}
```

The Magic: Report Generation

Automatic Activation

```
mvn test
# Spring Test Profiler activated
# Report generated: target/spring-test-profiler/latest.html
```

What You Get

- ✓ Timeline visualization of test execution
- ✓ Context cache statistics and hit/miss ratios
- ✓ Memory usage patterns during test runs
- ✓ Specific optimization recommendations

✓ Tests passed: 2 of 2 tests – 233 ms

| | |
|--------|--|
| 233 ms | /Library/Java/JavaVirtualMachines/zulu-17.jdk/ |
| 228 ms | 2022-05-09 11:24:33.653 DEBUG --- [|
| 228 ms | Instantiating CacheAwareContextLoaderDelegate |
| 5 ms | .DefaultCacheAwareContextLoaderDelegate] |
| 5 ms | 2022-05-09 11:24:33.662 DEBUG --- [|
| | Instantiating BootstrapContext using construct |
| | .support.DefaultBootstrapContext(java.lang.Cla |
| | .CacheAwareContextLoaderDelegate)] |
| | 2022-05-09 11:24:33.687 DEBUG --- [|
| | Instantiating TestContextBootstrapper for test |

```

      _
    _(_)_ _ _ _ _ \ \ \ \
   ' _ | ' _ | | ' _ \ _ ' | \ \ \ \
  |_) | | | | | | | ( _ | | ) ) ) )
 . _ | | | _ | | _ \ _ , | / / / /
 _ |=====|___/=/_/_/_/_/
Boot ::                                (v2.6.7)
```

11:25:51.412 INFO 43090 --- [main] d. ...
ng OrderControllerTest using Java 17 on Philips-Ma
y rieckpil in /Users/rieckpil/Development/git/talk

Optimization in Action

Problem Identified

Report shows: "Test classes creating separate contexts due to different @MockBean configurations"





Quick Fix Applied

```
// Before: Separate @MockBean in each test
// After: Shared @TestConfiguration
@TestConfiguration
static class TestConfig {
    @MockBean PaymentService paymentService;
    @MockBean NotificationService notificationService;
}
```



Results: 47s → 12s (2x faster!)

Advanced Features

- ✓  **Parallel Execution** - Full support for JUnit parallel tests
- ✓  **JSON Export** - CI/CD integration ready
- ✓  **Build Tools** - Maven & Gradle support
- ✓  **Compatibility** - Spring Boot 2.x - 3.x



Spring Test Profiler

Get Started Today

GitHub Repository

github.com/rieckpil/spring-test-profiler

Easy Start

1. ✓ Add one dependency
2. ✓ Run your tests
3. ✓ Get insights immediately

Join the Community

🔹 Developers already optimizing their test suites

★ [Star on GitHub](#)

