

# Top 10 Spring Boot Hacks

Pragmatismus statt Magie

10. Juli 2025 – Java Forum Stuttgart

# Julius Mischok

---

Als Engineering Manager löse ich mit Software-Teams komplexe Projektherausforderungen.

Diplom Mathematiker (univ.)

Spring Boot Experte

Geschäftsführer Mischok GmbH



# Meine ersten Schritte mit Spring Boot.

Oder: Der Tag, als ich begann, all meinen Code zu löschen...

Mehr Funktion + Weniger Code = Weniger Ärger!

# Projekttagbuch des Teams Aurora





# Liebes Tagebuch...

---

Heute war ein schwieriger Tag. Wir mussten eine neue Testklasse hinzufügen. Wie immer in diesem Fall schlug nun plötzlich ein komplett anderer Test fehl und wir mussten unsere Testdaten einen ganzen Nachmittag lang aufräumen.



# Tipp 1

---

@Transactional in Tests

@Transactional rollt nach jedem Test zurück –  
unabhängig vom Testergebnis!

[https://gitlab.mischok.de/open/persons/-/blob/strategy/  
testscripts/src/test/java/academy/mischok/persons/repository/  
PersonRepositoryTest.java?ref\\_type=heads](https://gitlab.mischok.de/open/persons/-/blob/strategy/testscripts/src/test/java/academy/mischok/persons/repository/PersonRepositoryTest.java?ref_type=heads)



# Warnung

---

Durch den Transaktionskontext verhalten sich Tests anders als die Laufzeitumgebung. Beispiel: `LazyInitializationException`

## Tipp 2

---

Flyway und @Sql kombinieren

Wird NICHT  
zurückgerollt

Flyway: Initialisierung Schema und Grundkonfiguration

@Sql: Spezifisches Szenario/Bewegungsdaten

Wird zurückgerollt



# Warnung

---

Testdatenerzeugung in Skripten erfordert beim Testfehlschlag  
Recherchearbeit.

## Tipp 3

---

Testcontainers

Testcontainers sind einfach einzubinden und stellen die später verwendete Umgebung nach.

```
@SpringBootTest
@Testcontainers
public class PostgresIntegrationTest {

    @Container
    static PostgreSQLContainer<?> postgres = new PostgreSQLContainer<>("postgres:16.2")
        .withDatabaseName("testdb")
        .withUsername("testuser")
        .withPassword("testpass");

    @DynamicPropertySource
    static void configureProperties(DynamicPropertyRegistry registry) {
        registry.add("spring.datasource.url", postgres::getJdbcUrl);
        registry.add("spring.datasource.username", postgres::getUsername);
        registry.add("spring.datasource.password", postgres::getPassword);
    }

    ...
}
```

# Warnung

---

Auf allen Entwicklungsmaschinen muss Docker installiert sein...

# Liebes Tagebuch...

---

Heute war ein schlimmer Tag. Wir mussten unsere Pipeline für das Staging-Deployment viermal starten, da die Tests gegen die externe API immer wieder fehlschlugen. Die Testversion war leider wieder mal instabil...



## Tipp 4

---

Mockito

Ein Test darf sein Verhalten nur bei Änderung des zu Grunde liegenden Codes ändern.

Mockito simuliert externe Abhängigkeiten innerhalb des eigenen Projekts.

@MockitoBean und @MockitoSpyBean faken oder überwachen die Schnittstellen-Beans.

[https://gitlab.mischok.de/open/cityservice/-/blob/feature/mockito/src/test/java/de/mischok/academy/cityservice/web/CityServiceTest.java?ref\\_type=heads](https://gitlab.mischok.de/open/cityservice/-/blob/feature/mockito/src/test/java/de/mischok/academy/cityservice/web/CityServiceTest.java?ref_type=heads)



# Warnung

---

Die definierten Stubs erfordern Wartung und fortlaufende Anpassung.  
Optimalerweise sichert ein Integrationstest sie ab.

## Tipp 5

---

WireMock

WireMock baut simuliert die externe Schnittstelle  
außerhalb des eigenen Projekts.

```
https://gitlab.mischok.de/open/cityservice/-/blob/feature/  
wiremock/src/test/java/de/mischok/academy/cityservice/web/  
CityServiceTest.java?ref\_type=head
```



# Warnung

---

Auch WireMock Stubs erfordern Wartung und fortlaufenden Abgleich.  
Es gibt viele verschiedene Wege zur Konfiguration von WireMock in Spring  
Boot Projekten!

# Liebes Tagebuch...

---

Heute war kein schöner Tag.  
Jemand hat unsere Staging  
Datenbank geleert. Die  
Zugangsdaten liegen nur in Git,  
nach Umgebungen gruppiert. Es  
muss also jemand aus der  
Entwicklungsabteilung gewesen  
sein...



## Tipp 6

---

# Config Management

# Quizfrage

---

Wie viele Möglichkeiten gibt es in Spring Boot um Konfigurationsdaten in die Anwendung zu bekommen?

Spring Boot hat 15 Möglichkeiten,  
Konfigurationsparameter zu setzen und zu überschreiben.

[https://docs.spring.io/spring-boot/reference/  
features/external-config.html](https://docs.spring.io/spring-boot/reference/features/external-config.html)



# Warnung

---

Parameter ohne sinnvolle Defaults führt zu quälender Fehlersuche.

## Tipp 7

---

Profiles

Profile strukturieren sinnvolle Konfigurationen und  
Defaults für Tests und Betrieb.

Profile aktivieren über Konfigurationsdateien oder an Tests

Beans in Abhängigkeit von Profilen laden

# Warnung

---

Profilwechsel in Tests kann durch Merged-Context zu unerwartetem Verhalten führen, da nicht alle Beans neu aufgebaut werden.

# Liebes Tagebuch...

---

Heute war ein fürchterlicher Tag. Ein Angreifer ist in unser Livesystem eingedrungen. Seltsam, unsere individuelle Login-Lösung lief eigentlich seit Jahren stabil...



## Tipp 8

---

Spring Security und @WithMockUser

Verschlüsselung, Datumsarithmetik und Security bitte  
niemals selbst implementieren!

Spring Security liefert eine leicht bedienbare Grundkonfiguration für die Absicherung von Anwendungen.

```
@Test
@WithMockUser(username = "mail@example.com", roles = "FINANCEVIEW")
public void callPage_onlyOneDate_authorizedUserView() throws Exception {
    // do your tests here...
}
```

# Warnung

---

Tests mit `@WithMockUser` oder generell innerhalb der Spring Security Abstraktion sind hilfreich, sollten aber durch einen Integrationstests abgesichert werden.

# Liebes Tagebuch...

---

Heute war ein guter Tag! Wir haben endlich unsere Datenzugriffe so umgestellt, dass wir alle Tabellen paginiert anfragen können! Ein halbes Jahr Arbeit, jetzt können wir die Früchte ernten!



# Tipp 9

---

## Spring Data JPA

Spring Data JPA ist das Schweizer Taschenmesser für den Zugriff auf relationale Datenbanken! Leichtgewichtig für einfache Queries, schlagkräftig auch bei hoher Komplexität.



# Warnung

---

Unbedacht eingesetzt kann JPA massive Performanceprobleme verursachen.

# Tipp 10

---

@Cacheable

Mit der Annotation `@Cacheable` versehene Methoden verwenden einen eingebauten Caching-Mechanismus.

[https://gitlab.mischok.de/open/company-database/-/blob/master/src/test/java/de/mischok/academy/companydatabase/service/CachingTest.java?ref\\_type=heads](https://gitlab.mischok.de/open/company-database/-/blob/master/src/test/java/de/mischok/academy/companydatabase/service/CachingTest.java?ref_type=heads)



# Warnung

---

Wie `@Transactional` funktioniert `@Cacheable` nur an public Methoden bei Aufruf durch eine andere Bean.

# Warnung

---

Wer setzt den Cache zurück? Und wann?

# Liebes Tagebuch...

---

Heute war der beste Tag unserer Team-Geschichte! Mit den Spring Boot Hacks, die wir beim Java Forum Stuttgart gelernt haben, konnten wir viel Code und Komplexität loswerden!

Auf in die Zukunft!



Team Aurora steht heute da, wo viele hinwollen – mit wartbarer Codebasis, sicherer Architektur und einem Team, das sich auf das Wesentliche konzentrieren kann.

Sei wie Team Aurora!

# Vielen Dank!

---

Kontakt:

[julius.mischok@mischok.de](mailto:julius.mischok@mischok.de)

<https://www.linkedin.com/in/julius-mischok/>



Icons: <https://icons8.com/>



**mischok**  
better. software\_