Entwicklung einer Spring Boot basierten Mocking-Lösung mit WireMock zur Simulierung externer Systeme

Projektbeschreibung

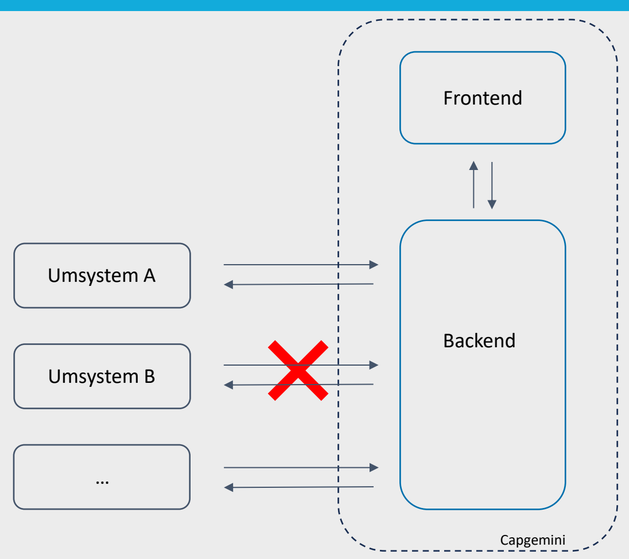
Bundesverwaltungsamt

- Der Kunde des Projekts ist das BVA
- Es liegt eine Geheimhaltungsvereinbarung vor
- Das Projektteam kümmert sich um die Pflege und Weiterentwicklung eines komplexen Softwaresystems mit zahlreichen Umsystemen
- Meine Hauptaufgabe im Rahmen des Teilprojekts war es, diese Umsysteme zu simulieren, um bessere Testbedingungen zu schaffen

Problemstellung

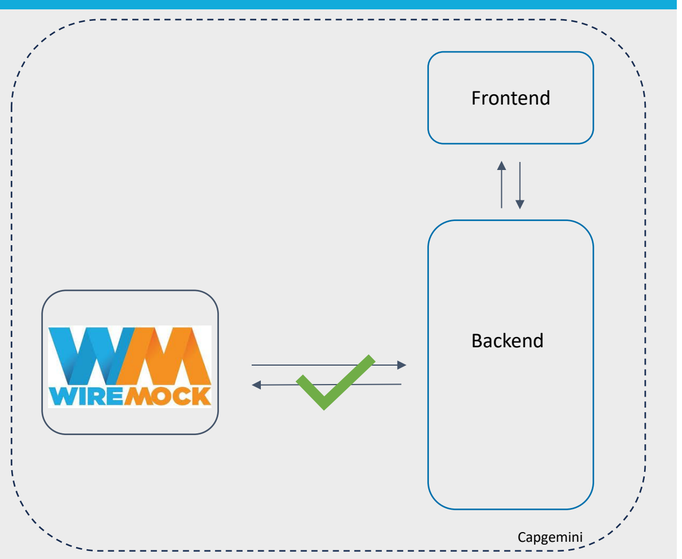
Herausforderung

- Die Software steht in Kommunikation mit zahlreichen Umsystemen, von denen einige nicht in der Verantwortung von Capgemini liegen.
- Diese Umsysteme sind nicht immer verfügbar, was das testen erschwert
- Ohne eine Simulation können Fachtests nicht unabhängig durchgeführt werden



Lösung

- WireMock als Mocking-Tool
- Simuliert externe Schnittstellen und liefert vorher definierte oder aufgenommene Antworten
- Erlaubt es, Testfälle unabhängig von externen Systemen durchzuführen
- Einfach integrierbar und erweiterbar über eine API



Umsetzung: Record & Playback

1. Voraussetzungen

- Alle relevanten Systeme müssen laufen
- Die Schnittstellen in der Hauptanwendung müssen auf WireMock umgestellt sein
- WireMock agiert als Proxy und nimmt Anfragen anstelle des Umsystems entgegen

2. Recording starten

- POST-Request an WireMock senden, um das Recording zu starten

```
POST /__admin/recordings/start
```

- Request-Body (Beispiel):

```
{
  "targetBaseUrl": "http://test.endpoint:1234",
  "filters": {
    "urlPathPattern": "/test-path/.*",
    "method": "ANY"
  },
  "persist": true
}
```

- WireMock beginnt, alle eingehenden Anfragen und Antworten aufzuzeichnen

3. Fachlichen Test durchführen

- Anwendung nutzen und relevante Testfälle durchspielen
- WireMock leitet Anfragen an das Umsystem weiter und speichert die Antworten
- Die aufgenommenen Daten bilden die Grundlage für Stubs

4. Recording stoppen

- POST-Request an WireMock senden, um das Recording zu beenden

```
POST /__admin/recordings/stop
```

- Die gespeicherten Mappings können nun verwendet werden
- WireMock antwortet jetzt direkt mit den gespeicherten Antworten

5. Stub anpassen

- Gegebenenfalls können Parameter abgepasst werden
- Nutzung von „bodyPatterns“ für flexiblere Requests
- Dynamische Antworten können eingerichtet werden, sodass sie sich an den eingehenden Requests anpassen

Testing

- Bevor das System ausgeliefert wurde, wurde es auf kundennaher Umgebung getestet
- WireMock wurde auf einem Tomcat-Server installiert und gestartet
- Die konfigurierten Stubs und Mappings wurden in dieser Umgebung getestet
- Über verschiedene Testanfragen wurde überprüft, ob WireMock korrekt antwortet
- Es wurde sichergestellt, dass das simulierte System wie das echte Umsystem reagiert
- Vor der finalen Auslieferung wurden alle relevanten Szenarien durchgespielt, um Fehler frühzeitig zu erkennen

Dokumentation

- Es wurde eine umfassende Dokumentation für den Kunden erstellt:

1. SOAP UI Collection
   - Enthält alle relevanten API-Calls
   - Ermöglicht es dem Kunden, die verschiedenen Endpunkte direkt zu testen
2. Detaillierte Anleitung auf Confluence
   - Erklärung der Funktionsweise von WireMock und der erstellten Stubs
   - Hinweise zur Anpassung und Erweiterung der Stubs
3. Release Letter
   - Beschreibung der gelieferten Komponenten und deren Funktion
   - Hinweise zur Konfiguration auf der Zielumgebung

Fazit & Erkenntnisse

- Allgemeine Erkenntnisse:
  - Durch WireMock konnten externe Systeme erfolgreich simuliert werden
  - Testbarkeit wurde verbessert
  - Die Lösung ermöglicht unabhängige und reproduzierbare Testszenarien
  - Fachliche Tests können nun ohne direkt Abhängigkeiten durchgeführt werden
- Persönliche Erkenntnisse:
  - Tiefergehendes Verständnis für API-Mocking gewonnen
  - Praktische Erfahrung im Arbeiten mit REST-APIs, WireMock und SOAP UI gesammelt
  - Eigenständiges Lösen von Herausforderungen in einer realen Umgebung gelernt
  - Die Bedeutung einer detaillierten Dokumentation für Kunden erkannt

h_da hochschule darmstadt informatik
member of EUT+ EUROPEAN UNIVERSITY OF TECHNOLOGY
Capgemini

Praxisprojekt 2
Informatik Dual
2025

Gruppe A