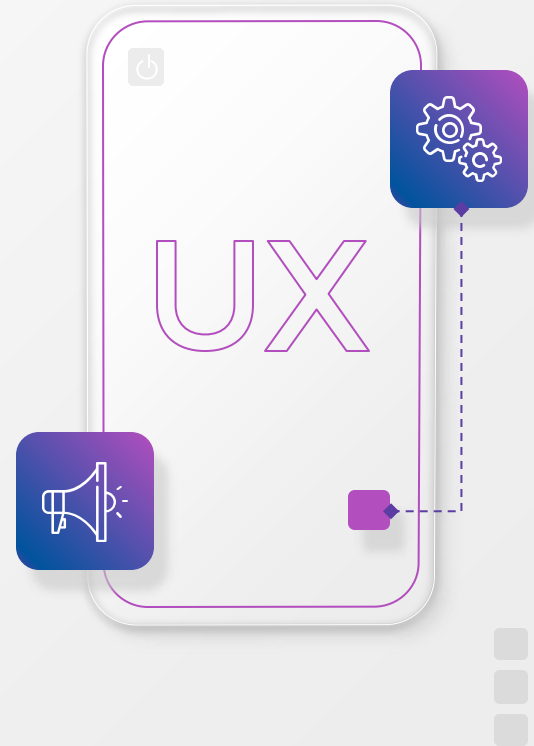


Lernsystem - Webbasierte Anwendungssysteme

Von Merwa Tahanur, Ioana Carmen Moldovan, Dilara Cagla Oguz, Mariam Zadran und Saida Covrk



Winx Club-Mitglieder



**Saida
Covrk
1509852**

**Mariam Zadrán
1518535**

**Ioana Carmen
Moldovan
1514892**

**Dilara Cagla
Oguz
1439735**

**Merwa
Tahanur
1505432**

GLIEDERUNG



01

Systemarchitektur

02

Funktionalitäten

03

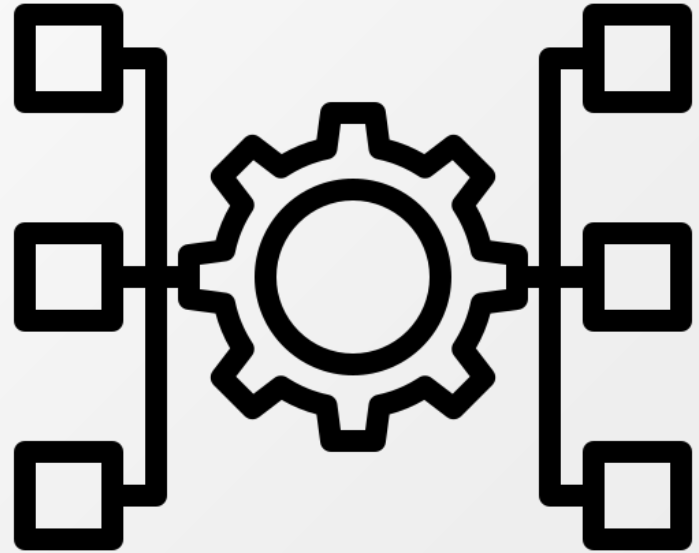
Komplikationen

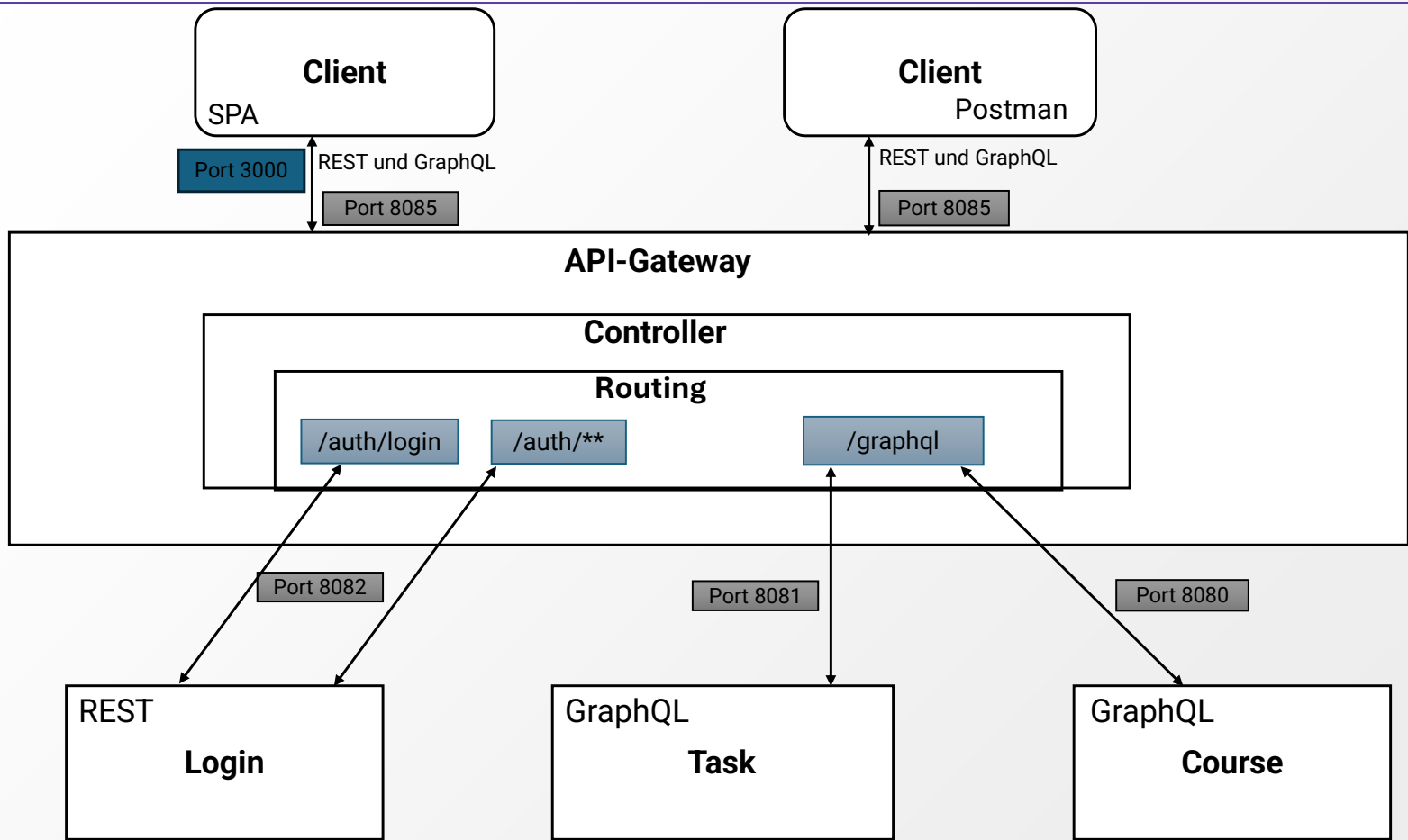
04

Demonstration

01

Systemarchitektur










02

Funktionalitäten

Login Service (REST)

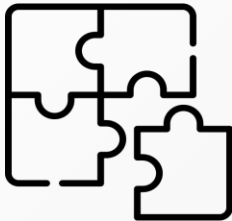
Hauptfunktionen

- | | |
|---|---|
| 1. Benutzeranmeldung → POST /auth/login | Prüft Anmeldedaten  |
| 2. Benutzer erstellen → POST /auth/users | Erstellt neuen Benutzer  |
| 3. Benutzer abrufen → GET /auth/users/{id} | Gibt Daten eines Benutzers zurück  |
| 4. Alle Benutzer abrufen → GET /auth/users | Gibt Liste aller Benutzer zurück  |
| 5. Benutzer aktualisieren → PUT /auth/users/{id} | Aktualisiert bestehende Benutzerdaten  |
| 6. Benutzer löschen → DELETE /auth/users/{id} | Entfernt Benutzer anhand der ID  |

Warum REST für den Login-Service?

Einfache Integration

- Kompatibel mit allen Systemen



Trennung von Client & Server

- Leichtere Wartung



Skalierbar

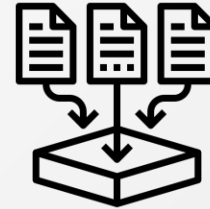
- Keine Sitzungsdaten auf dem Server



Course Service (GraphQL)

Warum GraphQL?

- Alles in einer einzigen Abfrage statt mehrere Requests



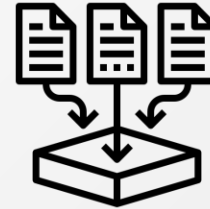
- Genau die Daten abrufen, die benötigt werden



Course Service (GraphQL)

Warum GraphQL?

- Alles in einer einzigen Abfrage statt mehrere Requests



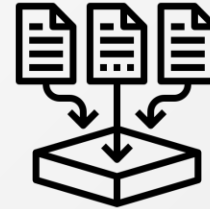
- Genau die Daten abrufen, die benötigt werden



Course Service (GraphQL)

Warum GraphQL?

- Alles in einer einzigen Abfrage statt mehrere Requests



- Genau die Daten abrufen, die benötigt werden







Course Service (GraphQL)

Port: 8080



Endpunkt: /graphql

Hauptfunktionen

1. Kurse verwalten

- Erstellen eines neuen Kurses 
- Abrufen aller Kurse 
- Abrufen eines Kurses nach ID **{id}**
- Abrufen eines Kurses nach Namen **{name}**
- Aktualisieren eines bestehenden Kurses 
- Löschen eines Kurses 

2. Dateien in Kursen verwalten

- Hinzufügen einer Datei in einen Kurs 
- Entfernen einer Datei von einem Kurs 

Task Service (GraphQL)

Port: 8081

Endpunkt: /graphql

Hauptfunktionen:

Erstellen

Abrufen

Bearbeiten

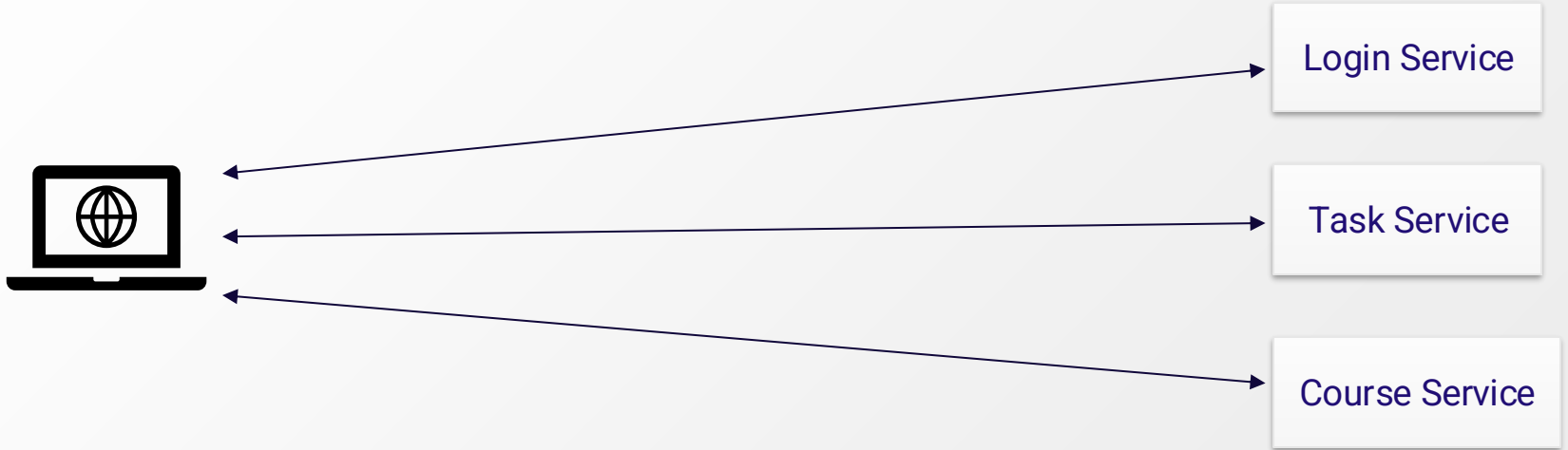
Löschen

To-Do-Liste

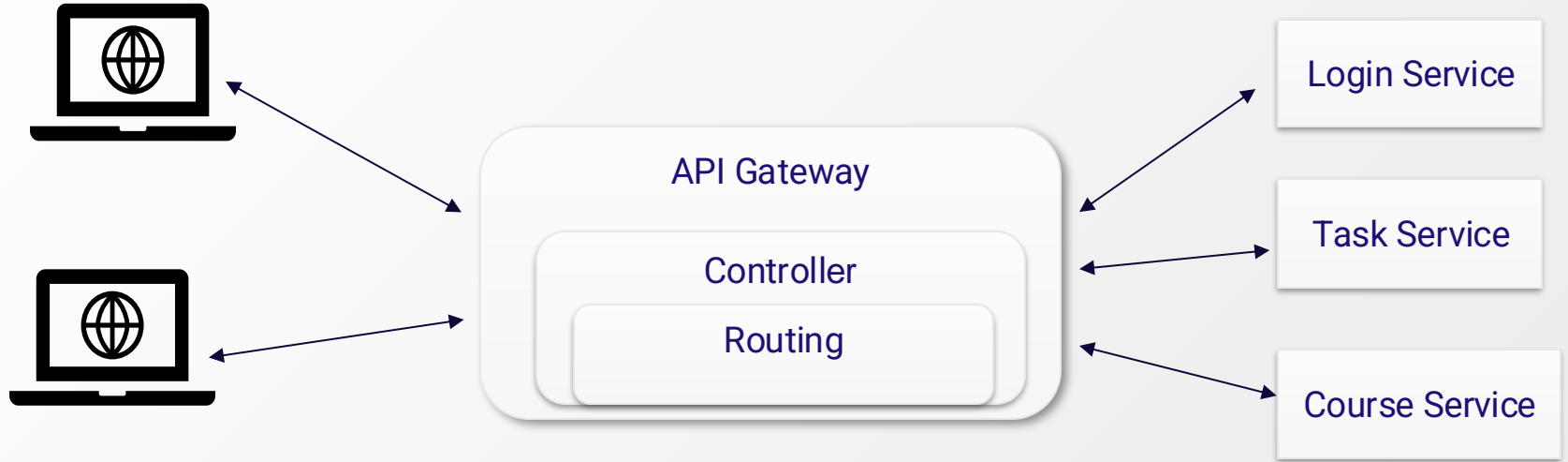
Aufgabe nach UserID

Heute fällig

API-Gateway

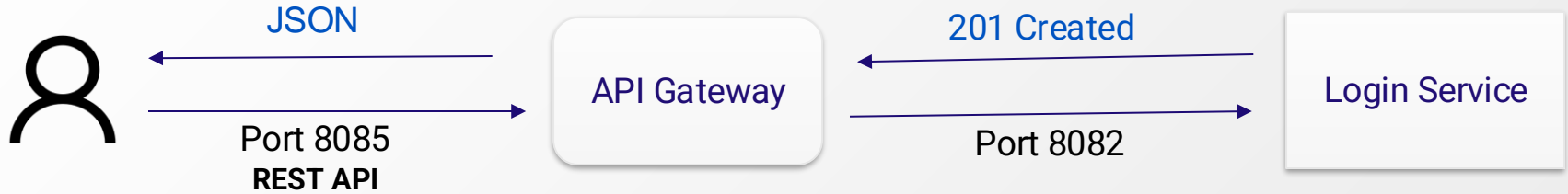


API-Gateway



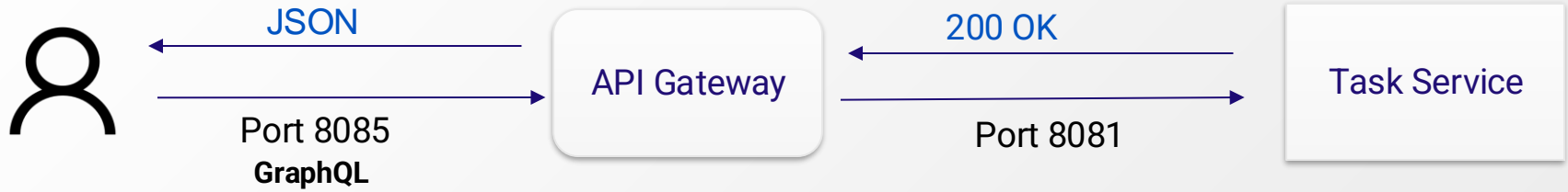
API-Gateway

Request-Response Login Service



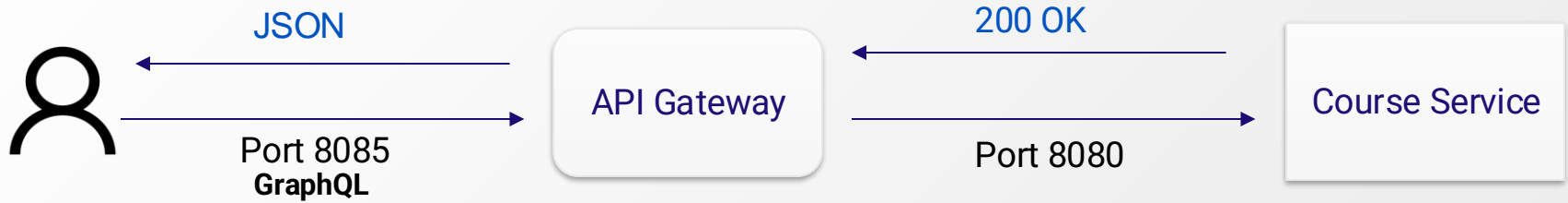
API-Gateway

Request-Response Task Service



API-Gateway

Request-Response Course Service



API-Gateway



Limits



Balancing



Caching

Postman

Über Port: 8085

- Adminsicht 
- CRUD-Operationen aller Services
- Senden von Request 
- Empfangen von Response 



Single Page Application mit React

01

Student View

02

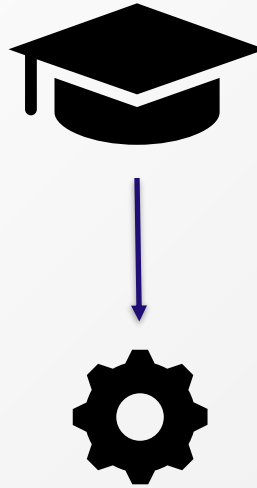
Funktionen

03

Limits

01

Student View



Studentensicht \neq Adminsicht

02

Funktionen

Login

Dashboard

Kurse + Files

To-Do Liste

03

Limits



**Create
Update
Delete**

Operationen

des Login und Course-Service

03

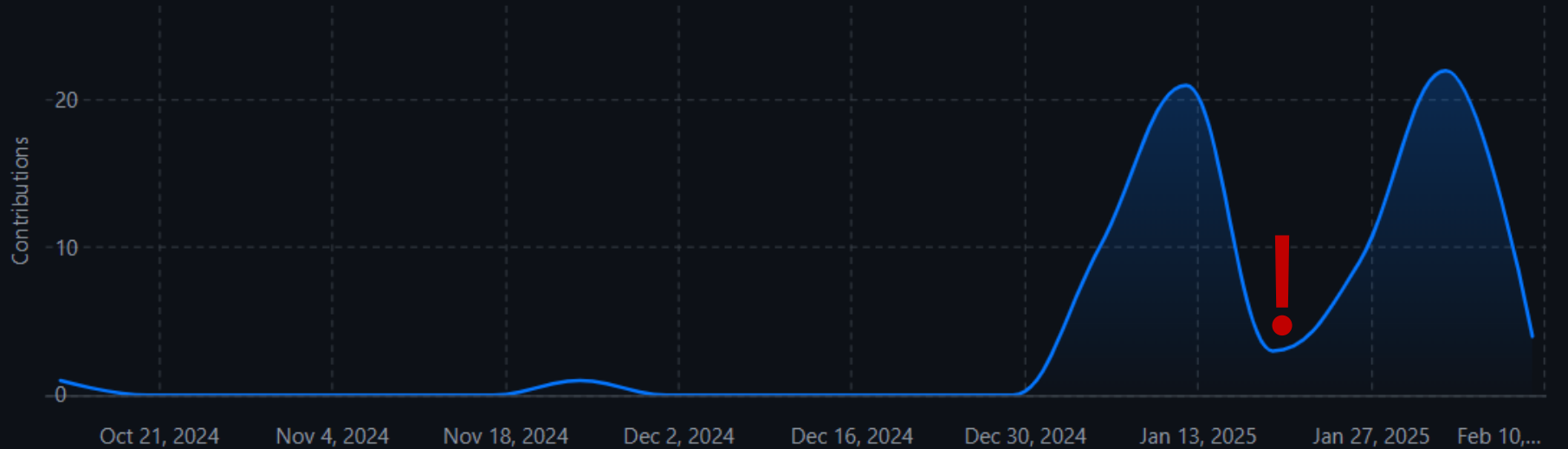
Komplikationen



API-Gateway

Commits over time

Weekly from 13. Okt. 2024 to 9. Feb. 2025



API-Gateway

- **Aufbau** des Gateways unklar → Recherche
- Idee: **3 Controller** (1 Controller pro Service)
- Probleme mit dem **Routing** → der Pfad mit dem Endpunkt musste immer angegeben werden
- Nur **1 Woche** Zeit eingeplant → Zeitdruck
- **Lösung:** Methoden innerhalb eines einzigen Controller



GraphiQL/Swagger UI über API-Gateway

1. **Swagger-Probleme** → Konnte nicht gestartet werden, Tests über API-Gateway & Postman notwendig
 2. **GraphiQL-Problem** → Nur Course Service sichtbar, Task Service fehlte
 3. **GraphiQL-Ausfall** → Vermutlich durch Fehlkonfiguration oder Gateway-Kommunikationsprobleme
 4. **Lösung** → Fokus auf **SPA mit Angular**
-

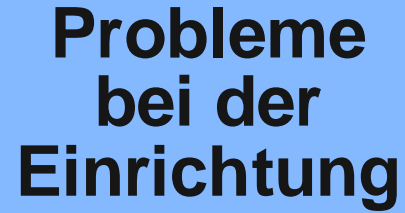
GraphiQL/Swagger UI über API-Gateway

1. **Swagger-Probleme** → Konnte nicht gestartet werden, Tests über API-Gateway & Postman notwendig
 2. **GraphiQL-Problem** → Nur Course Service sichtbar, Task Service fehlte
 3. **GraphiQL-Ausfall** → Vermutlich durch Fehlkonfiguration oder Gateway-Kommunikationsprobleme
 4. **Lösung** → Fokus auf **SPA mit Angular**
-

GraphiQL/Swagger UI über API-Gateway

1. **Swagger-Probleme** → Konnte nicht gestartet werden, Tests über API-Gateway & Postman notwendig
 2. **GraphiQL-Problem** → Nur Course Service sichtbar, Task Service fehlte
 3. **GraphiQL-Ausfall** → Vermutlich durch Fehlkonfiguration oder Gateway-Kommunikationsprobleme
 4. **Lösung** → Fokus auf **SPA mit Angular**
-

SPA mit Angular




**Probleme
bei der
Einrichtung**

Kompliziert

Zeitdruck

SPA mit Angular




Probleme
bei der
Einrichtung

Kompliziert

Zeitdruck

SPA mit Angular



Probleme
bei der
Einrichtung

Kompliziert

Zeitdruck

SPA mit Angular

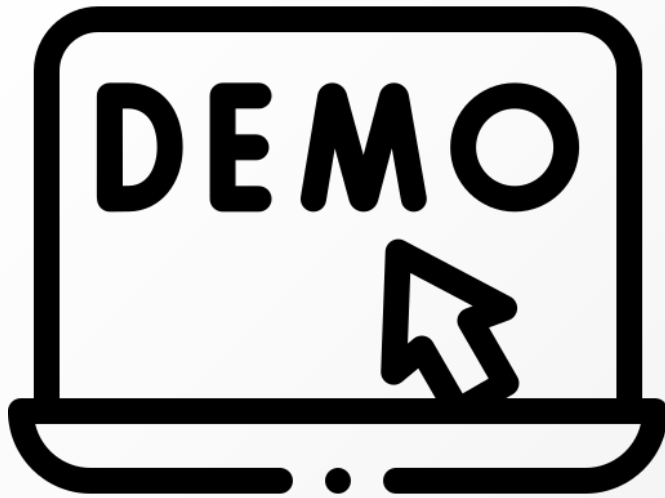
Probleme
bei der
Einrichtung

Kompliziert

Zeitdruck

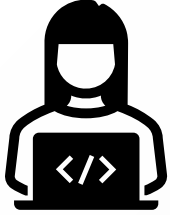


Lösung: Single Page Application mit **React**



04

Demonstration



**Vielen Dank für Ihre
Aufmerksamkeit**

