

Dokumentation: Deployment der Schach-Anwendung

Datum: 01.02.2026

1. Einleitung

Ziel dieser Sitzung war das Deployment (Veröffentlichung) einer Full-Stack Schach-Anwendung auf der Plattform Render.com. Die Anwendung besteht aus einem Java Spring Boot Backend und einem Vue.js Frontend.

2. Architektur & Deployment-Setup

Das Deployment wurde in zwei Services aufgeteilt:

Komponente	Technologie	Hosting-Typ	Konfiguration
Backend	Java / Spring Boot	Web Service (Docker)	Dockerfile
Frontend	Vue.js	Static Site	Node.js Build

3. Problemlösungsprozess (Troubleshooting)

Während des Deployments traten mehrere technische Herausforderungen auf, die schrittweise gelöst wurden:

A. Backend: Docker Build Fehler

Problem: Der Build-Prozess schlug fehl ('Copy failed'), da die Pfade im Dockerfile nicht zur Ordnerstruktur passten.

Lösung: Pfade im Dockerfile korrigiert (z.B. COPY backend/src src statt COPY src src).

B. Frontend: Blueprint & Verbindung

Problem: Die automatische Erkennung des Frontends via render.yaml (Infrastructure as Code) funktionierte nicht zuverlässig.

Lösung: Strategiewechsel zur manuellen Erstellung der 'Static Site' im Dashboard. Dies erwies sich als robuster.

C. Kommunikation: CORS Fehler (Error 403)

Problem: Das Frontend durfte nicht auf das Backend zugreifen ('Cross-Origin Request Blocked').

Lösung: Implementierung einer globalen CorsFilter-Klasse im Backend, die explizit alle Ursprünge freigibt.

D. Datenstruktur: Leeres Schachbrett

Problem: Das Spiel startete, aber das Brett blieb leer. Die Analyse zeigte, dass das Backend leere Objekte sendete.

Lösung: Hinzufügen von expliziten Getter-Methoden in der Java-Klasse Piece.java, da die Lombok-Annotation @Data in der Build-Umgebung nicht wie erwartet griff.

E. Konfiguration: API URL

Problem: 404 Fehler beim Laden des Spiels.

Lösung: Die Environment-Variable VITE_API_URL war falsch gesetzt. Korrektur von ...onrender.com zu ...onrender.com/api.

4. Fazit & Status

Die Anwendung ist nun erfolgreich deployed.

- Backend ist live und erreichbar.
- Frontend lädt korrekt und kommuniziert mit dem Backend.
- Spielmechanik (Erstellen, Ziehen) funktioniert.
- Besonderheit: Aufgrund des Free-Tiers hat der Server eine 'Cold Start'-Zeit von ca. 60 Sekunden.