

# PROJEKTPROTOKOLL: MONSTER TRADING CARDS GAME (MTCG)

## Überblick

Das Monster Trading Cards Game (MTCG) ist ein serverseitiges Java-Projekt, das eine REST-API für ein Sammelkartenspiel bietet. Benutzer können sich registrieren, Karten sammeln, in Decks organisieren, handeln und Kämpfe gegen andere Benutzer austragen. Die Anwendung nutzt eine PostgreSQL-Datenbank zur Speicherung aller Spiel- und Benutzerdaten. Die Authentifizierung erfolgt über Token, um sichere und kontrollierte Interaktionen zu gewährleisten.

## Komponenten

### Modelle

- **User:** Repräsentiert einen Benutzer mit Attributen wie username, password und token. Verwaltet die Registrierung, das Login und die Authentifizierung über Tokens.
- **Card:** Repräsentiert eine Spielkarte mit Attributen wie name, damage und element. Karten sind entweder Monster- oder Zauberkarten.
- **Deck und Stack:** Ein Deck enthält die besten 4 Karten eines Benutzers für Kämpfe, während der Stack alle derzeitigen Karten eines Benutzers umfasst.
- **Package:**

### Handler

- **UserHandler:** Behandelt Anfragen zur Benutzerregistrierung und gibt entsprechende HTTP-Antworten zurück.
- **SessionHandler:** Verwaltet das Login und die Authentifizierung. Generiert und speichert Token nach erfolgreichem Login.

### Datenbank

- **DatabaseConnector:** Stellt eine Verbindung zur PostgreSQL-Datenbank her und ermöglicht CRUD-Operationen. Nutzt JDBC zur Interaktion mit der Datenbank.

### Server

- **RestServer:** Startet den HTTP-Server und richtet die Endpunkte für die verschiedenen Spielmechaniken ein. Der Server läuft auf Port 10001 und unterstützt mehrere Endpunkte für Benutzerverwaltung, Kämpfe und Handel.

## Design-Entscheidungen

- **Token-basierte Authentifizierung:** Die Entscheidung für eine Token-basierte Authentifizierung wurde getroffen, um die Sicherheit der Benutzer zu erhöhen und sicherzustellen, dass nur authentifizierte Benutzer auf bestimmte Funktionen zugreifen können.
- **Eigene REST-Implementierung:** Anstelle eines Frameworks wurde eine eigene REST-Server-Implementierung gewählt, um eine tiefere Kontrolle über die HTTP-Kommunikation zu haben und die Vorgaben zu erfüllen.
- **Datenbank-Design:** Die Struktur der Datenbank wurde so gewählt, dass sie sowohl die Benutzerverwaltung als auch die Spielmechaniken effizient unterstützt.

## Git-Repository:

[https://github.com/if23b066/SWEN1\\_SemesterProjekt](https://github.com/if23b066/SWEN1_SemesterProjekt)