

# Protocol - Media Ratings Platform (MRP)

**GitHub:** <https://github.com/m-sarikaya/mrp-media-ratings-platform>

---

## 1. Architektur-Entscheidungen

Architektur: 3-Schichten-Architektur

- **Handler:** HTTP-Routing, Request/Response
- **Service:** Business-Logik (Validierung, Regeln)
- **Repository:** Datenbankzugriff (SQL)

Entscheidungen

- **Token-Auth mit Bearer Header:** zentraler Filter (AuthFilter) prüft Token für alle Endpoints außer Register/Login.
- **Kein ORM:** Zugriff über PreparedStatement → SQL-Injection Schutz.
- **Mehrere Genres pro Medium:** eigene Tabelle media\_genres (1:n).
- **Rating-Regeln auf DB-Ebene:** Unique Constraints (ein Rating pro User/Medium, ein Like pro User/Rating).
- **Data Transfer Objects (DTOs):** klare Trennung zwischen Request-Body und Domain (UserProfileUpdate, RatingInput, UserProfileResponse).

## 2) Tests und Teststrategie (Unit Tests)

**Ziel:** Business-Regeln isoliert testen (ohne echte Datenbank).

**Ansatz:** In-Memory Fake-Repositories für Unit-Tests.

Anzahl Tests: 40 Unit-Tests.

Getestete Bereiche:

### - UserService (8 Tests)

Registrierung (Username/Password Pflicht, Duplicate), Login (falsches Passwort), Token-Setzen, Profil-Update, Profil-Statistiken.

- **MediaService (8 Tests)**

Pflichtfelder, MediaType-Validierung, Genre-Validierung, Ownership-Checks.

- **RatingService (12 Tests)**

Sterne-Validierung, 1-Rating-Regel, Owner-Checks, Like-Regeln.

- **FavoriteService (6 Tests)**

Favorit hinzufügen/entfernen, Duplikate, Favoritenliste.

- **RecommendationService (6 Tests)**

Genre- und Content-Empfehlungen, leere Ergebnisse, Fehlerbehandlung.

### 3) Probleme & Lösungen

#### **Problem 1:** Java LocalDateTime in JSON

- Fehler beim Serialisieren von createdAt.
- Lösung: Jackson Modul jackson-datatype-jsr310 eingebunden und WRITE\_DATES\_AS\_TIMESTAMPS deaktiviert → ISO-Format.

#### **Problem 2:** Update/PUT Rating fehlerhaft

- Falsches Pfad-Parsing für Rating-ID → „socket hang up“.
- Lösung: ID korrekt aus letztem Pfadsegment extrahieren, Fehler sauber mit 400 behandeln.

#### **Problem 3:** Genres als Liste

- Anfangs nur genre als String → nicht spezifikationskonform.
- Lösung: List<String> genres + Tabelle media\_genres.

### 4) SOLID-Prinzipien (mind. 2 Beispiele)

#### **Single Responsibility Principle (SRP)**

- Jede Handler-Klasse übernimmt nur eine Aufgabe (z. B. MediaHandler, RatingHandler, FavoritesHandler).
- Business-Logik liegt ausschließlich im Service-Layer.
- Dependency Inversion Principle (DIP)

- Services arbeiten gegen Repositories, nicht gegen Datenbank direkt.
- Repositories können durch Fakes ersetzt werden (Unit-Tests), ohne Business-Logik zu ändern.

## 5. Zeitaufwand

Bereich	Zeit
Grundgerüst Server + DB	8 Stunden
Auth + User-Profile	5 Stunden
Media-CRUD + Genres	6 Stunden
Recommendations	4 Stunden
Favorites + Leaderboard	5 Stunden
Ratings + Likes + Confirm	8 Stunden
Tests (40 Unit-Tests)	3 Stunden
Postman + README + Protokoll	4 Stunden
Gesamt	43 Stunden