

Meilenstein: Schnittstellen

Bitte erklärt anhand eures Architekturdiagramms (voriger Meilenstein), in welcher Form die verteilte Kommunikation zwischen Client und Server stattfinden soll.

Hinweis:

Zu klären sind die Art des Leistungsabrufs vom Client zum Server, ggf. auch vom Server zum Client (Push) und das Datenaustauschformat.

Der Client kann einen Dienst vom Server anfordern. In unserem Fall registriert sich der Client bzw. loggt sich ein, falls er angemeldet ist. Der Server beantwortet die Anforderung, in dem er den Client entweder erfolgreich registriert oder ins Spiel lässt.

Die Authentifizierung beim Server erfolgt über JSON Web Tokens (JWT). Diese dienen zur URL-sicheren Einrichtung für die Darstellung von Claims zwischen zwei Parteien. Hier sind zwei Funktionen von Bedeutung. Die erste Funktion `jwt.sign()` erstellt ein JWT-Token, weist es einem Benutzerobjekt zu und gibt dieses JWT-Token letztlich zurück, sodass man es weitergeben kann. Die Kommunikation kann entweder asynchron oder synchron sein, je nachdem, ob ein Rückruf bereitgestellt wird.

Die zweite asynchrone Funktion `jwt.verify()` kontrolliert das Token des Benutzers, wenn der Zugriff auf eine geschützte Route erfolgt. Das Token wird als Parameter übernommen, den geheimen Schlüssel, den man in der Funktion `jwt.sign()` definiert hat. Über den Rückruf kann man auf geschützte Daten zugreifen und diese senden.

Informationen werden mittels REST-Schnittstelle mit dem Client ausgetauscht. Einige REST-Endpunkte sind zudem ohne Token verfügbar, wie z.B. `/highscore`

Zur Kommunikation mit dem Server verwendet der Client die Library „Unirest-Java“. Unirest ist eine leichte HTTP-Client-Bibliothek. Die gesamte Kommunikation zwischen Client und Server erfolgt synchron. Die Kommunikationspartner synchronisieren beim Senden oder beim Empfangen von Daten immer, also warten, bis die Kommunikation beendet ist; solange hält der Server eine Verbindung offen.

Beim Pull holt sich der Client die nächsten Informationen selbst, z.B. wenn der Spieler sich den Highscore ansehen möchte.