



Pr0ject Zer0

Inhaltsverzeichnis

- Vorstellung des Projekts
- Datenbank
- Backend
- Frontend
- Tests
- Live-Demo
 - Was funktioniert?
 - Was nicht?
- Mögliche Erweiterungen

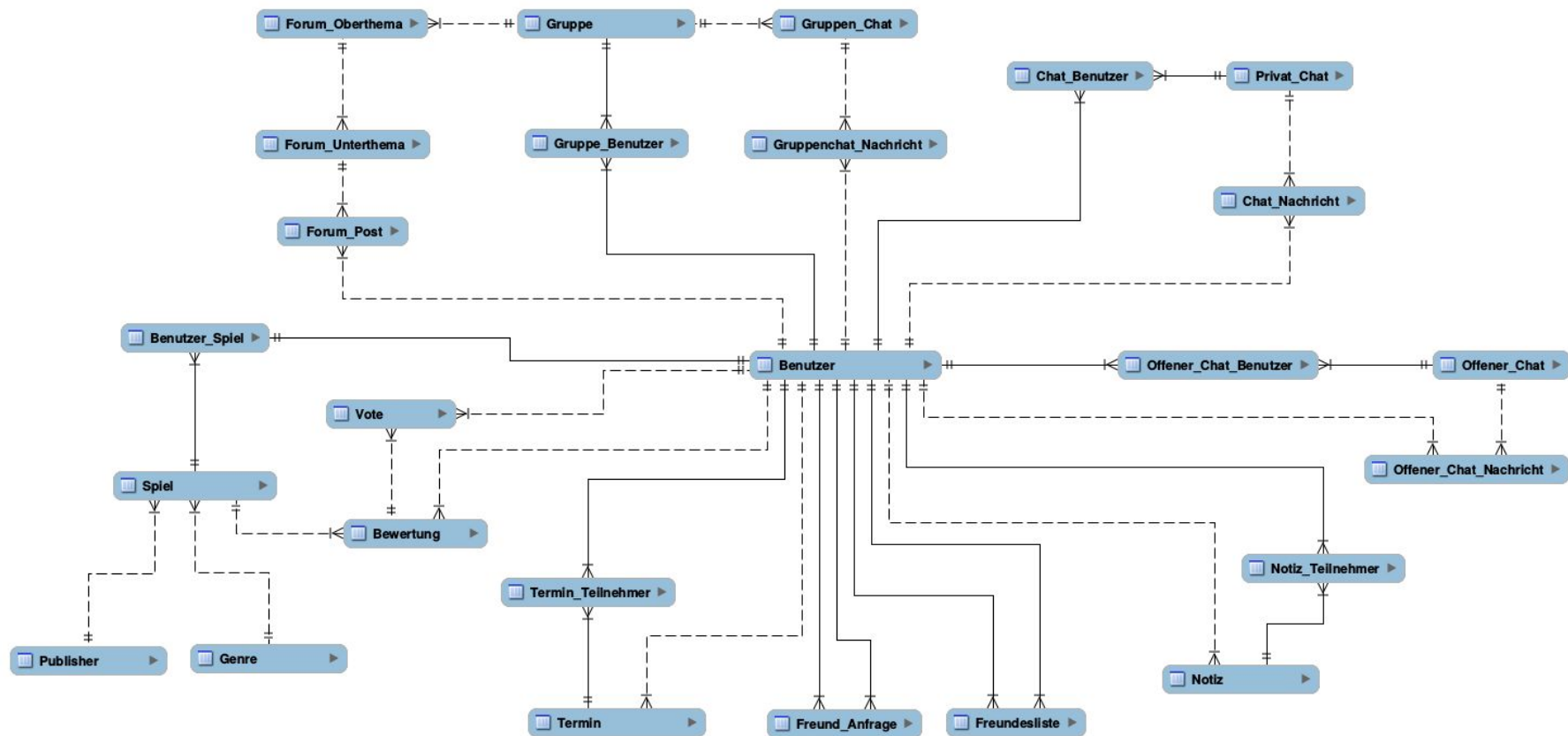
Was ist Pr0ject Zer0?

Community-Plattform für Computerspiele:

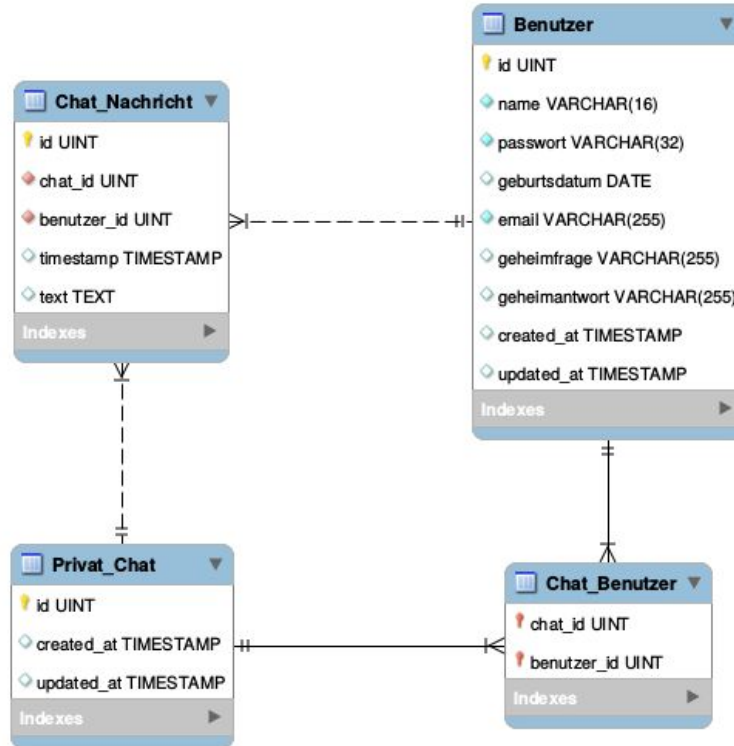
- Spielebibliothek
- Freundesliste
- Chats
- Gruppen
- Termine
- Notizen

Datenbank

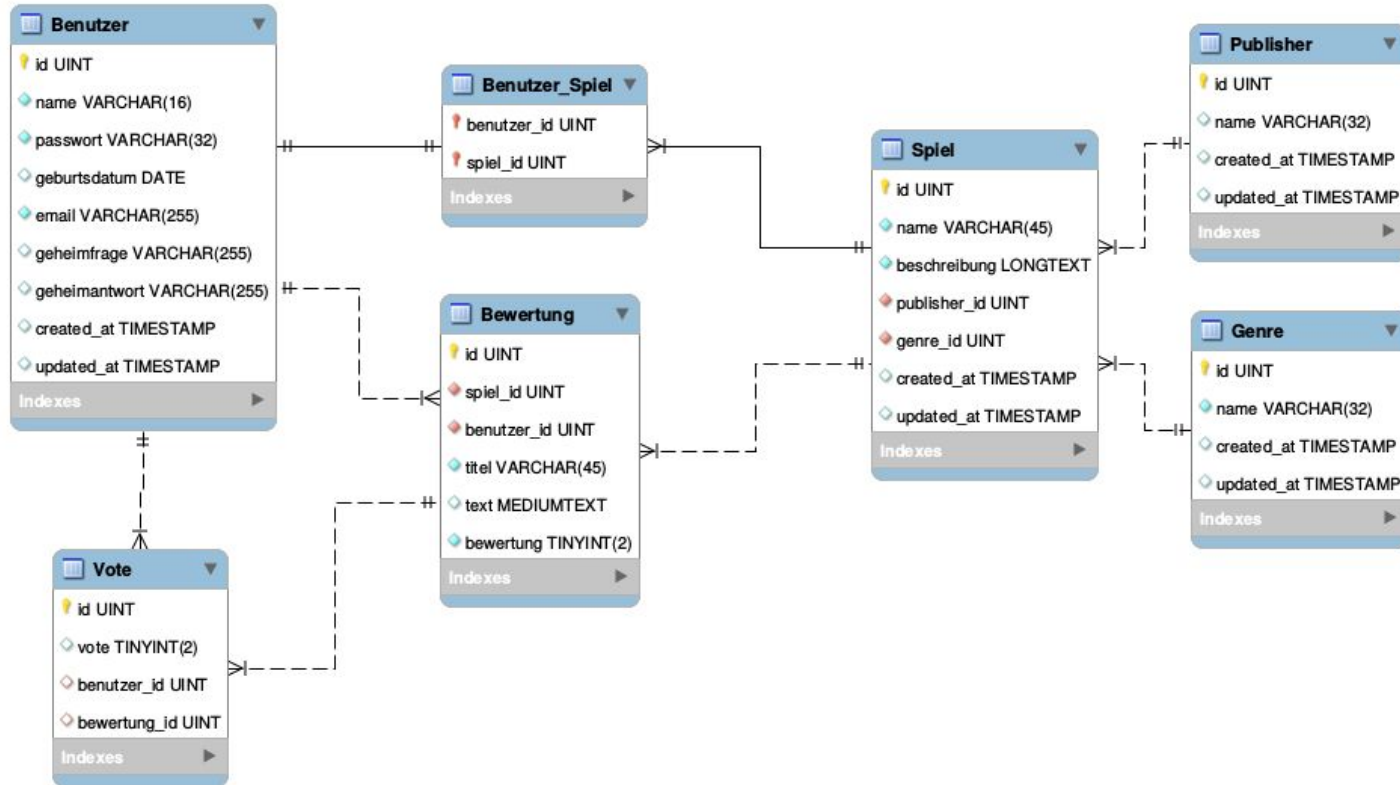
ER-Modell Übersicht



Chats



Spielebibliothek



Backend

Backend



- Laravel 5.4 (PHP Framework)
- RESTful API umsetzung
 - 43 GET Methoden
 - 16 POST Methoden
 - 7 DELETE Methoden
 - 5 PUT Methoden
- Authentifizierung
 - JSON Web Token (JWT)
- Broadcasting
 - Pusher
- Deployment
 - Laravel Forge

JWT JSON WEB TOKEN

- Dient der sicheren Übertragung von Informationen als JSON Object
- JWT ist nur ein String (header.payload.signature)
 - Header: enthält Token-Typ, Signatur- und Verschlüsselungsalgorithmus
 - Payload: enthält die Claims (Key/Value-Paare)
 - Signatur: stellt sicher dass die Nachricht vom angegebenen Absender stammt.
- Server Authentifiziert Benutzer durch den String

JWT JSON WEB TOKEN

Encoded

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiaWF0IjoiYWRtaW4iOnRydWV9.TJVA95OrM7E2cBab30RMHrHDcEfxjYZgeFONFh7HgQ
```

Decoded

HEADER:

```
{
  "alg": "HS256",
  "typ": "JWT"
}
```

PAYLOAD:

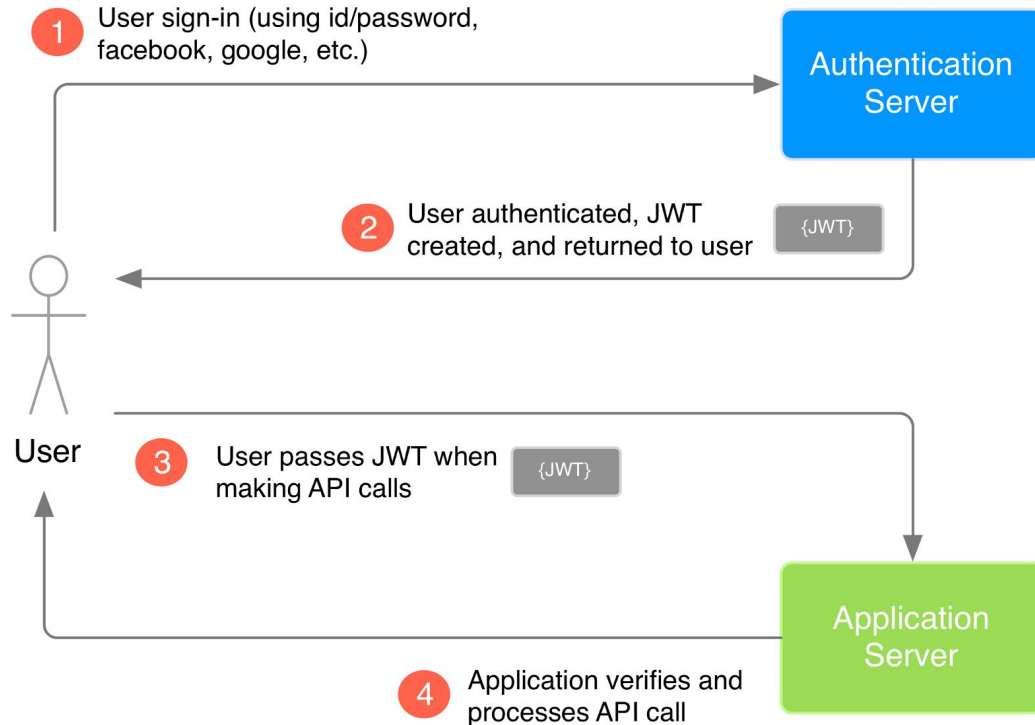
```
{
  "sub": "1234567890",
  "name": "John Doe",
  "admin": true
}
```

VERIFY SIGNATURE

```
HMACSHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
  
) ☐ secret base64 encoded
```

JWT JSON WEB TOKEN

Funktion:





- Wird für den Chat verwendet.
- Stellt eine real-time Schicht zwischen Server und Client dar.
- Für die Persistente Verbindung wird WebSocket verwendet.
- Wird über die Broadcasting API von Laravel verwendet.
- Ermöglicht mehrere hunderte offene Verbindungen Ressourcensparend zu verwalten.

FORGE Deployment

Bereitstellung von unbegrenzten PHP Anwendungen auf:

- DigitalOcean, Amazon AWS, Linode

Funktionen:

- Einfach Bedienbar
- Kostenlose SSL Zertifikate
- Push To Deploy
- Load Balancing
- Sicherheit

Frontend

Features Checkliste

	Desktop	Web	Mobil
Registrieren			
Login			
Bibliothek			
Freundesliste			
Gruppen			
Chats/Chaträume			
Terminplaner			
Notizen			

Webanwendung - Technologien



Browser: Chrome - Version 63.0.3239.132



IDE: Webstorm 2017.3



Angular5



HTML5 & CSS3



Bootstrap - Version 4



PusherJS



Font-Awesome

Desktopanwendung - Technologien



Java



IDE: Eclipse



Pusher-Websocket-Java



JavaFX™

Oberfläche: JavaFx



GUI-Editor: Scene Builder



FontawesomeFX



Jfoenix

Mobileanwendung - Technologien



Ionic 3.19.1



Atom 1.23.1



Chrome - Version 56.0.2924.87



PusherJS



HTML5 & CSS3



Angular5

Desktopanwendung - Besonderheiten

- Aufteilung der Klassen in verschiedene (aufeinander aufbauende) Kategorien
 - API: Hauptsächlich Aufbau der Anfragen an den Server, Verarbeitung der vom Server erhaltenen JSON-Objekte
 - Model: Hilfsklassen mit verschiedenen Funktionen, z.B. werden mit Hilfe dieser Klassen Instanzen von Notizen oder Terminen dargestellt
 - View: Klassen die in erster Linie zur Kontrolle der Oberfläche und Einbindung der API-Klassen zuständig sind

Tests

Webanwendung - Tests

Verschiedene Tests, die mit Karma realisiert wurden, zur Überprüfung der API-Abfragen sowie erzeugung der Komponenten und Services

Desktopanwendung - Tests

56 JUnit-Testfälle für die API-Klassen, da diese Klassen die Grundlage für die Anwendung sind.

Mögliche Erweiterungen

- Bewertungsfunktion
- Geheimfrage zur Rücksetzung des Passworts
- Profilbilder