

Technologien

Google Guice:

Wofür nutzen wir Guice:

Wir nutzen das Guice Framework zur Dependency Injection (DI übernimmt das Auflösen von Abhängigkeiten für Objekte. Mit Hilfe von DI wird die Aufgabe der Objektverwaltung an eine zentrale Komponenten übertragen.) Hierbei bietet Guice die Möglichkeit zur Konstruktor - Injection sowie auch Method u. Field Injection. Guice kann also auch beliebige Methoden nutzen oder Attribute direkt injizieren.

Warum Guice (Vorteile) :

- Die Abhängigkeiten verwaltet Guice nicht in zentralen XML-Dateien, sondern hinterlegt sie im Java-Code.
- „Schlanke Alternative“ zu Spring mit ausschließlicher Nutzung von DI
- Konflikte bei der Arbeit mit gemeinsam genutzten XML-Dateien sind unvermeidbar
- Bearbeiten von Java-Dateien ist einfacher und weniger fehlerträchtig

Retrofit:

- HTTP-Client für Java und Android
- Retrofit wird genutzt um sich mit einem REST-Webservice zu verbinden, indem die API in eine Java-Schnittstellen übersetzt wird
- Von uns genutzt für Kommunikation zwischen Client und Server
- Kommunikation per:
 - **GET** - fordert Daten vom Server an
 - **POST** - übermittelt Daten an den Server
 - **PUT/PATCH** - ändern bestehende Daten auf dem Server
 - **DELETE** - löscht bestehende Daten auf dem Server

Supabase:

- SQL-Datenbank
- Open Source alternative zu Firebase

Vorteile // Warum von uns genutzt:

- Autorisierung über role level policy
- Automatische API-Generierung der Daten
- Relationales Datenbankmodell
- Nutzerauthentifizierung über Social Provider (Gitlab, Google)

JavaFX:

- Kennt jeder

GitLab

- Nutzen wir für Markdowneditierung