

# Evernote Projekt Dokumentation

## URLs und Zugangsdaten

- **Anwendungs-URL:** <http://angular.s2110456014.student.kwmhgb.at>
- **Admin-Benutzer:**
  - **E-Mail:** samuel.lange@example.com
  - **Passwort:** password
- **Regulärer Benutzer:**
  - **E-Mail:** john.doe@example.com
  - **Passwort:** password

## Überblick über die Lösung

Das Evernote-Projekt ist in zwei Hauptteile unterteilt:

- **Backend:** Implementiert mit Laravel, um die Datenbankebene und die Geschäftslogik zu verwalten.
- **Frontend:** Implementiert mit Angular, um eine „benutzerfreundliche“ Oberfläche für Interaktionen bereitzustellen.

## Seiten und Navigationspfade

### Seiten:

- **Login-Seite:** Ermöglicht Benutzern das Einloggen in die Anwendung.
- **Listen:** Die Startseite nach dem Login, zeigt eine Übersicht aller Listen. Ohne Login können nur „public“ Listen gesehen werden. Nach Login sieht man die eigenen Liste, „public“ Listen und Listen die mit einem geteilt wurden.
- **Listen-Detailseite:** Zeigt Details einer bestimmten Liste, einschließlich Notizen und deren Todos und geteilten Benutzern. Über diese Ansicht können auch jegliche create update delete zu den Notizen und Todos gemacht werden.
- **Todos:** Zeigt eine Liste von Todos die man selbst erstellt hat, oder die einem zugewiesen wurde. Diese Übersicht kann man mit einem Filter nach Tags eingrenzen.
- **Formularseite:** Zum Erstellen und Bearbeiten von Listen, Notizen und Todos. Dabei wurde immer darauf geachtet, dass die Auswahl an Usern zum Teilen oder der Tags so eingeschränkt ist, dass kein Eingabefehler gemacht werden kann.
- **Home:** Hier sieht man den aktuell eingeloggtten User und wenn dieser Admin ist, kann hier auch create update und delete zu den Tags ausgeführt werden.

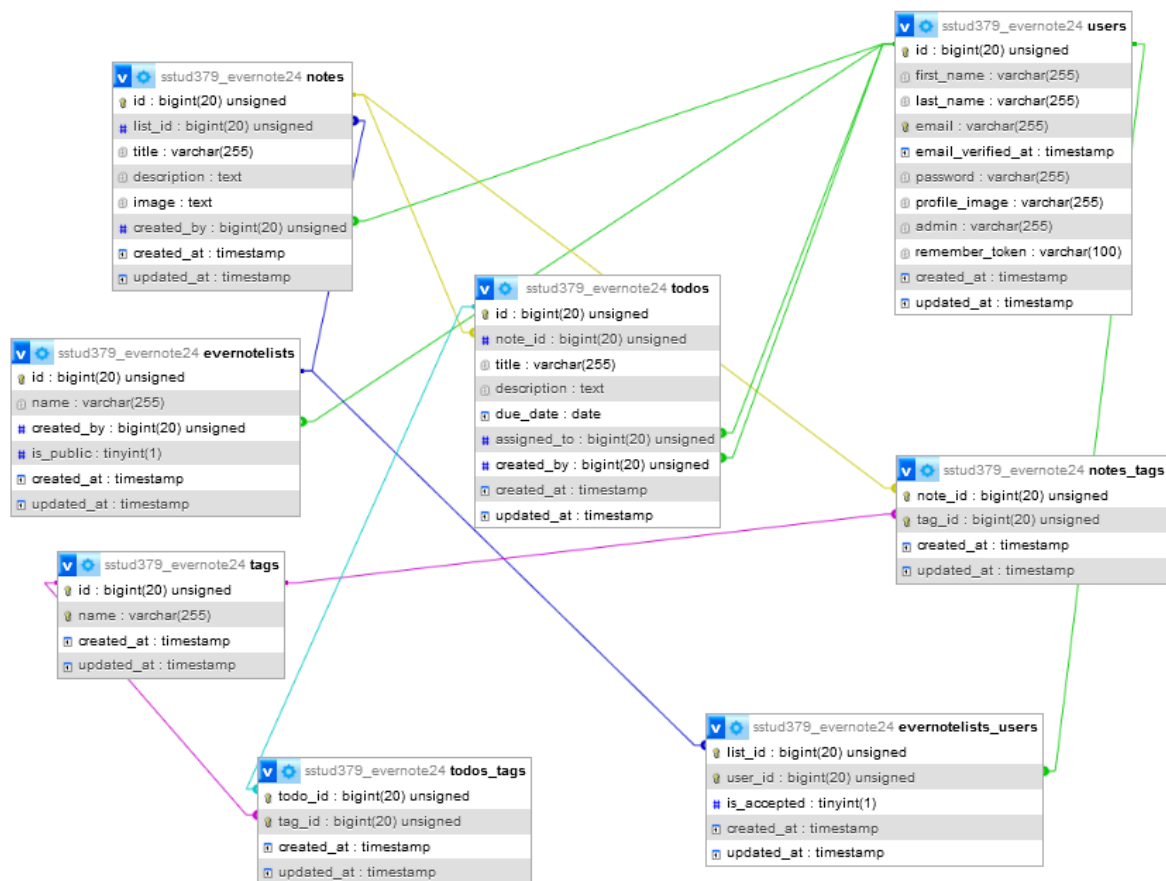
### Navigationspfade:

- `/login:` Leitet zur Login-Seite weiter.
- `/lists:` Leitet zur Listenübersicht weiter

- /lists/:id: Zeigt die Details einer bestimmten Liste an.
- /home: Zeigt den aktuellen User
- /edit/list/:id: Zum Bearbeiten einer bestimmten Liste.
- /edit/list/ Zum Erstellen einer bestimmten Liste.
- /edit/note/:id: Zum Bearbeiten einer bestimmten Note.
- /edit/note/ Zum Erstellen einer bestimmten Note.
- /edit/todos/:id: Zum Bearbeiten eines bestimmten Todos.
- /edit/todos/ Zum Erstellen eines bestimmten Todos.

## Wichtige Entscheidungen und Implementierungsdetails

### Datenbankschema:



- **Users-Tabelle:** Speichert Benutzerinformationen einschließlich Vornamens, Nachname, E-Mail und Profilbild.
- **Lists-Tabelle:** Enthält Listendetails wie Name, erstellt von und öffentlich.
- **Notes-Tabelle:** Enthält Notizdetails wie Titel, Beschreibung, list\_id und erstellt von.
- **Todos-Tabelle:** Enthält Todo-Details einschließlich Titel, Beschreibung, Fälligkeitsdatum, note\_id und zugewiesen an.

- **Tags-Tabelle:** Speichert Tag-Informationen zur Kategorisierung von Notizen und Todos.
- **List\_Users-Tabelle:** Eine Pivot-Tabelle zur Verwaltung von Benutzern, die Listen zugewiesen sind.

## REST-API-Übersicht:

- **Benutzerauthentifizierung:** Endpunkte für Login, Logout und Benutzermanagement.
- **CRUD-Operationen:** Endpunkte zum Erstellen, Lesen, Aktualisieren und Löschen von Listen, Notizen und Todos.
- **Tag-Verwaltung:** Endpunkte zum Zuweisen und Entfernen von Tags von Notizen und Todos.
- **Listenfreigabe:** Endpunkte zum Einladen von Benutzern zu Listen und Verwalten von geteilten Listen.

## Wichtige Überlegungen

### Benutzerauthentifizierung und -autorisierung:

- Implementiert mit Laravel's integrierten Authentifizierungsmechanismen.
- Geschützte Routen, um sicherzustellen, dass nur authentifizierte Benutzer auf bestimmte Endpunkte zugreifen können.

### Datenvalidierung:

- Verwendung von Laravel's Validierungsfunktionen zur Sicherstellung der Datenintegrität.
- Implementierung der Angular-Formularvalidierung für Benutzereingaben.
  - Mittels vordefinierten Selectoptions falscheingabe bei User zum teilen und Zuweisung von Tags

### Benutzeroberfläche und Benutzererfahrung:

- Nutzung von Semantic UI für ein klares und reaktionsschnelles UI-Design.
- Sicherstellung einer intuitiven Navigation und benutzerfreundlichen Interaktionen.
  - Buttons für Update und Delete immer bei dem jeweiligen Objekt
  - Button für Create um einfaches Verständnis wo das Objekt nach dem erstellen auftauchen wird.