

08.12.2019 // OAS 4 All

Grundidee des bestehenden Fabaccess Systems

- Frontend (Clickdummy): <https://fabaccess.bian-meyer.de>
 - GitHub: <https://github.com/faaaaabi/fabaccess-admin-interface>
- Backend
 - GitHub: <https://github.com/faaaaabi/fablab-api-gateway>
- Zugriffs-App
 - GitHub: <https://github.com/faaaaabi/fablab-access-app>

Technologieentscheidung

Beruhen zum Großteil auf persönlichen Präferenzen. Daneben:

- Backend:
 - Sprache/Frameworks
 - node
 - Sehr große und aktive Community
 - Sehr große Auswahl an teilweise sehr guten Libraries, Frameworks
 - Niedrige Einstiegshürde
 - TypeScript
 - Mittel der Wahl um im JavaScript Umfeld Typsicherheit zu erhalten
 - Kombiniert die Flexibilität von JavaScript mit der Typsicherheit anderer Sprachen
 - Eignet sich durch Sprachfeatures aus der OOP besonders für größere Projekte
 - Sehr große und aktive Community
 - Express
 - Etabliertes Webframework im node Umfeld
 - Breite erweiterbarkeit durch third party libs
- Frontend
 - Typescript: siehe oben
 - React:
 - Sehr große und aktive Community
 - Breites Spektrum an "Addons" (Third-Party Libs, Komponenten-Libs)
- Zugriffs-App

- JavaScript: Würde gerne TypeScript sein. Sollte zu TypeScript migriert werden
 - ReactNative:
 - Kleine Lernkurve zwischen React und React Native: Sehr viel Code-Sharing zwischen Web und App möglich (nahezu alles bis auf das Markup)
 - Native UI-Elemente (kein WebView), dadurch nativer Look and Feel
 - Cross-Plattform fähig
 - Große und aktive Community
 - Breite Auswahl an Libs und Frameworks
 - Geringe Einstiegshürde (Im Vergleich zur App programmierung in Kotlin, Swift)
-

Version #2

Erstellt: 2024-10-13 01:14:09 CEST von Mario Voigt (Stadtfabrikanten e.V.)

Zuletzt aktualisiert: 2025-02-25 21:22:13 CET von Mario Voigt (Stadtfabrikanten e.V.)