



«HONEY BADGER» «Digitaler Begleiter auf den Gornergrat»

Technische Informationen für die Jury



Technische Informationen für die Jury

Aktueller Stand des Sourcecodes

- <https://github.com/rotwurstesser/honeybadger-2019>

Ausgangslage

- Eine ansprechende Applikation, die mit einige Livedaten arbeitet und somit dem User ein ansprechendes Erlebnis bereiten soll
- Wir haben uns darauf fokussiert, nicht das gesamte Design in unserem PoC zu realisieren, sondern nur einzelne Screens
- Wir haben uns am Anfang dazu entschieden Vue, Tailwind und Go einzusetzen, da wir mit diesen Technologien schon einige Erfahrungen gesammelt haben und wir damit auch in einem geringen Zeitraum viel erreichen können
- Die Entscheidung, dass wir eine Webapplikation realisieren, wurde uns durch die Aufgabenstellung abgenommen

Technischer Aufbau

- Frameworks: Tailwind CSS, Vue.js
- Komponenten: Axios, Vue-router, Vuex
- Tailwind wird verwendet um sauber und modular zu stylen, Axios als HTTP Client, Vue für das Rendern der UI Komponenten, Vue-Router für das clientseitige routing und Vuex übernimmt die Aufgabe des sauberen Statehandlings
- Die Abhängigkeiten werden über NPM verwaltet und im Backend existiert keine Paketverwaltung, da dort keine zusätzlichen Pakete verwendet werden

Implementation

- Unsere APIs sind mit dem standard Http-Server von Go geschrieben. Es wurde keine zusätzliche Library dafür verwendet
- Unser Backend arbeitet mit zwei externen Diensten. Für die Wettervorhersage wird die Webseite <https://darksky.net> verwendet und für die Bilder wird der Feed von der Plattform Instagram eingebunden
- Die Website und der Prototype sind Guides welche skalierbar funktionieren, die Konfiguration in Tailwind wurde dem Design entnommen weswegen auch andere Komponente welche im Design nicht direkt definiert sind mit den generierten Klassen umgesetzt werden können und dem Look and Feel entsprechen
- Da die Bilder und Wetterdaten immer aktualisiert werden, bekommt man immer die neusten Infos zu z.B. dem Wetter. Dadurch kann man seine Tour planen (und in zukünftiger Weiterentwicklung dies mit Push notifications verbinden um Warnungen herauszugeben wenn zB bald ein Sturm aufkommt und man am Wandern oder Skifahren ist). Es werden aktuelle Alternativen aufgelistet, wenn die gewählte Aktivität bei dem Wetter nicht zu empfehlen ist

Abgrenzung / Offene Punkte

- Wir haben uns dafür entschieden den Proof of Concept nicht in einer PWA umzusetzen, da wir uns sicher sind dass die Vorteile aus unserer Konzeption (Push notifications wenn schlechtes Wetter ist zum Beispiel) ohne Problem realisierbar wären. Wir haben den Kosten/Nutzen Faktor abgewägt und sind somit zu dem Schluss gekommen, dass eine PWA uns nur unnötige Zeit kostet, die wir besser in andere Dinge stecken können
- Manche Daten in der Applikation sind hartkodiert, da wir nicht genug Zeit haben um alle Daten dynamisch abzubilden