

Frontend Coding Guidelines - Visual Board GmbH

Dokumentenmanagement

Erstellungsdatum: 01.07.2020

Autoren: Samuel Salomon (SSA)

Dateiname: OEG-LWI-SSA_Frontend_Coding_Guidelines_0.2_Draft_4

Änderungsverzeichnis

Version	Datum	Autor	Beschreibung
0.1	01.07.2020	SSA	Dokument erstellt
0.2	06.07.2020	SSA	Coding Guidelines definiert

Inhaltsverzeichnis

1	Ordnerstruktur	3
1.1	api	3
1.2	assets	3
1.3	components	3
1.4	mixins	3
1.5	router	3
1.6	views	3
2	Code Style	3
3	Logging	3
4	Validierung	3
5	Variablenbenennung	4

1 Ordnerstruktur

Alle Projektrelevanten Dateien befinden sich im im src Ordner des Projektes. Die verschiedenen Unterordner werden folgenden näher erklärt. In dem Hauptverzeichnis befinden sich bereits zwei Dateien (App.vue, main.js).

1.1 api

Falls API Anfragen nötig sind werden die dafür erforderlichen Dateien in diesem Ordner abgelegt.

1.2 assets

In diesem Ordner werden Daten wie Bilder, HTML Seiten oder ähnliches gespeichert die Später in die js Dateien importiert werden.

1.3 components

Beinhaltet alle Komponenten des Projektes die später in den Hauptansichten (views) verwendet werden. Komponenten sollten in weitere Unterordner strukturiert werden.

1.4 mixins

Die Mixins sind Teile des Javascript-Codes, die in verschiedenen Komponenten wiederverwendet werden. Demnach befinden sich in diesem Ordner Methoden die in jeder Komponente aus Vue.js wiederverwendet werden können.

1.5 router

Routen werden verwendet um Komponenten auf spezifische Pfade der Web Applikation zu mappen. JS Dateien mit Routen definiert werden in diesem Ordner abgelegt.

1.6 views

Geroutete Komponenten werden als Views definiert. Um das Projekt übersichtlicher zu gestalten werden diese getrennt von den anderen Komponenten in diesem Verzeichnis abgelegt.

2 Code Style

Vue JS besitzt eigene Code Style Empfehlungen auf die hier verwiesen werden. [Vue.JS Style Guide](#)

3 Logging

Abweichung zu Konstanten-Style → Klein- und Kurzschreibung

```
private final static Logger log = new Logger(...class);
```

4 Validierung

Für die Validierung sind folgende 2 Konstrukte zulässig:

boolean isValid / hasValid...(Email)

void validate... / check...(Email) **throws** ...Exception

5 Variablenbenennung

Für die Validierung sind folgende 2 Konstrukte zulässig:

Regel

Programmieren immer auf Englisch

auch bei Variablennamen

ganze Wörter

camelCase

Beispiel

spielerName → nickname

pId → playerId

playername → playerName

Ausnahme: "Exception e" in einem Catch-Block