

Eine Dokumentation für den beginn der Arbeit am schubwerk FRONTEND.. sollte nach der Fertigstellung ins Github README.md formatiert werden..

FRONTEND

VUE-CLI | BOOTSTRAP | BOOTSTRAP-VUE | AMCHARTS

„Recherchiere 5 Minuten vor deiner Aufgabe ob es eine einfachere oder neuere Variante gibt um dein Vorhaben umzusetzen.“

EINSTELLUNGEN

VISUAL STUDIO CODE

Das ist der **Standard Editor** für die Arbeit an schubwerk Frontend.

Install Extensions: **Vetur**, **Prettier**, **ESLint**

Zusätzliche **Einstellungen am Editor**:

Code -> Preferences -> Settings: Suche nach ESLint und dann Edit in settings.json

```
{
  "vetur.validation.template": false,
  "eslint.validate": [
    {
      "language": "vue",
      "autoFix": true
    },
    {
      "language": "html",
      "autoFix": true
    },
    {

```

```
    "language": "javascript",
    "autoFix": true
  },
],
"editor.formatOnSave": true,
"editor.codeActionsOnSave": {
  "source.fixAll.eslint": true
},
"eslint.codeAction.disableRuleComment": {}
}
```

FRONTEND - DARSTELLUNG

BOOTSTRAP | BOOTSTRAP-VUE | AMCHARTS |
NUMERAL

VUE

COMPONENTS

<STYLE>

Stylings die nur eine Komponente entsprechen können in dieser eingefügt werden. Wenn es allgemeinere Styles sind, die mehrere Komponenten betreffen, sollen diese global definiert werden! Für diese globalen Styles muss erstmal geprüft werden in wie weit das Bootstrap-Theme per BootstrapVariablen angepasst werden kann oder welche eigenen Styling-Klassen dafür erstellt werden müssen. Auf dieses Thema wird im Abschnitt BOOTSTRAP genauer eingegangen!

BOOTSTRAP

ZWECK

BENUTZUNG / STRUKTUR

....

Eigene Styling-Komponenten müssen immer sinnvoll im Abhängigkeit zum Bootstrap Theme erstellt werden. Zum Beispiel Klassen Erstellung mit For-Loops über die Theme-Colors!

BEISPIELE

TEXT

AMCHARTS

ZWECK

Mit AMCHARTS bietet die Möglichkeit verschiedene Diagramme zu visualisieren! Diese Bibliothek wird verwendet wenn Diagramme erstellt werden müssen!

BENUTZUNG / STRUKTUR

AMCHARTS ist per **Node-Package** bereitgestellt! Es wird bisher **kein Vue-Plugin** von AMCHARTS bereitgestellt! Jede **Komponente** die AMCHARTS benutzen möchte, muss die Bibliothek im script Tag **importieren**. Vue-cli oder Webpack kümmert sich darum, dass im kompilierten Kode, die Bibliotheken nicht redundant enthalten sind!

BEISPIELE

IMPORTIEREN

```
<script>
import * as am4core from "@amcharts/amcharts4/core";
import * as am4charts from "@amcharts/amcharts4/charts";
import am4themes_animated from "@amcharts/amcharts4/themes/
animated";
```

NUMERAL

ZWECK

Für die verschiedenen Darstellungen von **Nummern, Währungen, Bytes, Prozenten, Zeiten, Exponential** soll die Javascript Bibliothek <http://numeraljs.com/> verwendet werden.

PAKET

Es gibt speziell für Vue ein Paket Namens **vue-numeral-filter**, dass die Benutzung mit Vue erleichtert. Dieses Paket wurde installiert und stellt direkt **numeraljs** bereit

BEISPIEL

Tabelle mit verschiedenen Beispielen! Weitere Informationen auf:

<https://www.npmjs.com/package/vue-numeral-filter>

<http://numeraljs.com/>

| Type | Numeral | Alias | Result |
|------------|-----------------------------------|---------------------------|--------|
| Bytes | {{ 10485760 numeral('0b') }} | {{ 10485760 bytes }} | 10 MB |
| Percentage | {{ 0.5567 numeral('0.[00]%') }} | {{ 0.5567 percentage }} | 55.67% |

| | | | |
|---------------------|---|--------------------------------|-------------|
| Thousands Separator | {{ 561739482 numeral('0,0') }} | {{ 561739482 separator }} | 561,739,482 |
| Ordinal | {{ 20 numeral('Oo') }} | {{ 20 ordinal }} | 20th |
| Abbreviate | {{ 1000000 numeral('0.0a') }} | {{ 1000000 abbreviate }} | 1.0m |
| Exponential | {{ 123987.202 numeral('0.[00]e+0') }} | {{ 123987.202 exponential }} | 1.24e+5 |
| Currency | {{ 200.42 numeral('\$0,0.00') }} | {{ 200.42 currency }} | £200.42 |

FRONTEND - DATEN

VUE-CLI | STORAGE | STATEMANAGEMENT