

Dokumentation VideTechTools

VideoTechTools ist eine Sammlung von Simulationen für die Videotechnik-Vorlesung.

Wie lese ich diese Dokumentation am besten?

Für eine ideale Darstellung dieser Markdown-Datei (.md) empfiehlt es sich z.B. [Microsoft Virtual Code](#) zu verwenden. Beim Öffnen dieser Datei erscheint in der Symbolleiste in der oberen rechten Ecke des Fensters ein entsprechender Vorschau-Button. Auf macOS funktioniert auch der Shortcut `shift + cmd + v`.

Installation

Grundlage für die Entwicklung mit React Native ist die Verwendung von [Node.JS](#). Dazu empfiehlt sich die LTS-Version.

Um die einzelnen Bibliotheken zu verwalten kommt der Paket Manager [Yarn](#) zum Einsatz. Dieser lässt sich im Terminal z.B. über folgenden Befehl installieren:

```
npm install --global yarn
```

Ggf. muss der Befehl `sudo` vorangestellt werden.

Sind die Voraussetzungen erfüllt, wird über folgenden Befehl die Anwendung Expo-CLI installiert. Dies kann einige Minuten dauern:

```
yarn add expo-cli
```

Zum Schluss müssen im Haupt-Verzeichnis von VideoTechTools mit folgender Anweisung alle verknüpften Bibliotheken installiert werden:

```
yarn install
```

Ausführen der App mit Expo

Um die App mit Expo auszuführen muss im Haupt-Verzeichnis lediglich folgender Befehl ausgeführt werden:

```
expo start
```

Alternativ funktioniert auch:

```
yarn start
```

Daraufhin öffnet sich im Webbrowser eine Benutzeroberfläche mit einem QR-Code, der mit dem Smartphone gescannt werden kann, sofern die App *Expo Go* darauf installiert ist.

Sollte es hier zur Fehlermeldung `ERROR watch EMFILE` kommen, so muss die Anwendung `Watchman` installiert werden. Hierzu kann mittels `Homebrew` erfolgen: `brew install watchman`

Aufbau der Software

```
VideoTechTools
├── README.md
├── App.js      // Einstiegspunkt der App
├── app.json   // Metadaten etc. der App
├── ...
└── src
    ├── calculations // Mathematische Funktionen/Umrechnungen
    ├── components   // Wiederverwendbare Komponenten
    ├── navigation   // Alles zu Navigationen
    ├── screens      // Aus components zusammengesetzte Seiten
    └── ...
```

Berechnungen

Komponenten

- [CIE-Normfarbtafel](#)
- [Vektorskop](#)
- [Waveformmonitor](#)
- [Signal-Generator](#)

- [Signal-Vorschau](#)
- Allgemeine Komponenten
 - [Einstellungs-Fenster](#)
 - [ScopesCamera](#)
 - [VideoStandardAlertView](#)

Navigation

Seiten

- [CIE-Normfarbtafel](#)
- [Waveformmonitor](#)
- [Vektorskop](#)
- [Komponentensignal](#)
- [Scopes Übersicht](#)

Vorkommende Signal-Arrays

Wie füge ich eine neue Simulation hinzu?

1. **NEUER SCREEN:** Wenn ein neues Themengebiet entsteht empfiehlt es sich zunächst in *screens* einen neuen Ordner auf gleicher Ebene wie *signalAndScopes* anzulegen. Erstelle dann eine neue JavaScript-Datei. Für den Inhalt der Datei empfiehlt es sich an den bestehenden Simulationen zu orientieren.
2. **IN NAVIGATOR VERKNÜPFEN:** Unter *navigation* in *AppNavigation.js* muss zunächst der neue Screen importiert werden. Um eine Header-Bar zu erhalten kann nach dem Vorbild der bestehenden Seiten eine neue *StackWrap*-Funktion angelegt werden. Daraufhin muss diese einen eigenen *Drawer.Screen*-Eintrag in der Funktion **Navigation** erhalten. Um ein neues Icon hinzuzufügen muss nur der *switch*-Ausdruck nach Vorbild der bestehenden einträge erweitert werden.
3. **SCREEN GESTALTEN:** Um die App so variabel wie möglich zu machen sind alle verwendeten Komponenten in *components* untergebracht. Es kann sich also entweder hieraus mittels *Import* im Screen bedient oder neue Komponenten erstellt werden. Mathematische Funktionen kommen im Ordner *calculations* unter.

Wie veröffentliche ich eine neue App-Version in Expo?

Vor veröffentlichung einer neuen Version ermpfiehl es sich in *app.json* zunächst drei Versionsnummern erhöht werden: `expo.version`, `expo.ios.buildNumber` und `expo.android.versionCode`

Verpflichtend ist dies aber nur wenn eine Version für die Veröffentlichung außerhalb Expo mittels Build-Feature vorgesehen wird (siehe Expo Dokumentation).

Dazu bietet Expo eine umfassende Dokumentation. Wichtig ist dabei zu beachten, dass es Release Channel gibt, mit denen man für unterschiedliche App-Versionen einen anderen Link erzeugen kann.

Wird kein Release-Channel angegeben, so wird der "Default"-Channel verwendet.

Im Allgemeinen gilt der Befehl

```
expo publish --release-channel <your-channel>
```

Siehe [Expo Release Channels](#)

Nützliche Tutorials

[React](#)

[React Native](#)

[Durchreichen von Daten](#)

[Three.JS](#)

[React-Three-Fiber](#)

[Expo](#)

Dokumentationen der verwendeten Bibliotheken

[React](#)

[React Native](#)

[Three.JS](#)

[React-Three-Fiber](#)

[Expo](#)