

# React Native Tutorial

Dieses Tutorial zeigt dir, wie du die Entwicklungsumgebung einer React-App aufsetzt und wie du eine einfache App erstellst. Für dieses Tutorial solltest du bereits ein Basiswissen über HTML, CSS und JavaScript mitbringen. Sollten dir diese Technologien ganz neu sein, dann wirf zuerst einen kurzen Blick in die Tutorials auf [W3Schools](https://www.w3schools.com)<sup>1</sup>.

React Native ist nicht zu verwechseln mit React.js. React.js ist ein Framework für Webanwendungen. React Native baut auf diesem auf, verwendet aber anstelle von HTML und CSS eigene Komponenten. Mit diesen lassen sich Apps entwickeln, die das *Look and Feel* von nativen Apps von iOS und Android haben. Somit lassen sich mit nur einer Code-Basis Apps für beide Betriebssysteme entwickeln.

## Hello World

### 1. Installiere node.js

Lade die letzte stabile Version von [node.js](https://nodejs.org/)<sup>2</sup> herunter und installiere sie. Dadurch wird npm, ein JavaScript-Package-Manager, auf deinem Rechner installiert.

Download for macOS (x64)

14.15.5 LTS

Recommended For Most Users

15.9.0 Current

Latest Features

[Other Downloads](#) | [Changelog](#) | [API Docs](#)

[Other Downloads](#) | [Changelog](#) | [API Docs](#)

⚠ Solltest du Node.js bereits installiert haben, geh sicher, dass du zumindest Version 12 verwendest. Dies kannst du im Terminal mit `node -v` überprüfen.

### 2. Installiere Expo<sup>3</sup>

Mit Expo kannst du ganz einfach eine React Native App erstellen und ausführen. Im Terminal führe folgenden Befehl aus:

```
npm i -g expo-cli
```

⚠ Auf MacBooks kann es notwendig sein, dass du Expo nur mit Administrationsrechten installieren kannst. Dafür füge `sudo` am Anfang des Befehls ein.

Es werden einige Warnhinweise bei der Installation angezeigt. Mach dir diesbezüglich aber keine Sorgen.

Lade dir zudem die App Expo Go für [Android](#)<sup>4</sup> oder [iOS](#)<sup>5</sup> herunter. Damit kannst du die

---

<sup>1</sup> <https://www.w3schools.com>

<sup>2</sup> <https://nodejs.org/>

<sup>3</sup> <https://expo.io>

<sup>4</sup> <https://play.google.com/store/apps/details?id=host.exp.exponent&hl=de&gl=US>

<sup>5</sup> <https://apps.apple.com/de/app/expo-client/id982107779>

entwickelte App direkt auf deinem Smartphone ausprobieren und auch Code-Beispiele im Weg direkt ausprobieren. Dafür ist allerdings ein gratis Expo-Account notwendig.

### 3. Erstelle deine erste React Native App

Im Terminal gib folgenden Befehl ein, wobei `testApp` der Name der zu erstellenden App ist.

```
expo init testApp
```

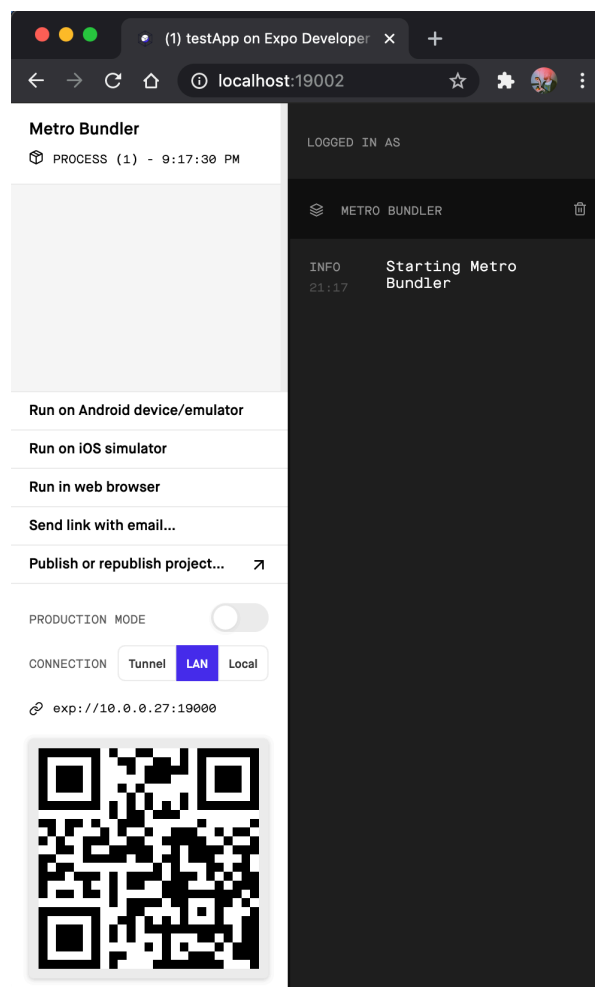
Dieser Befehl erstellt deine erste App, im unten abgebildeten Kontext-Menü wähle mit den Pfeiltasten `blank` zum Erstellen einer leeren App.

```
[ (base) Refugees-Welcome:React-Tutorial Bernhard$ expo init testApp
? Choose a template: > - Use arrow-keys. Return to submit.
----- Managed workflow -----
> blank a minimal app as clean as an empty canvas
  blank (TypeScript) same as blank but with TypeScript configuration
  tabs (TypeScript)  several example screens and tabs using react-navigation and TypeScript
----- Bare workflow -----
  minimal bare and minimal, just the essentials to get you started
  minimal (TypeScript) same as minimal but with TypeScript configuration
```

### 4. Starte deine erste React Native App

Wechsle mit `cd testApp` in den neu erstellten Ordner und starte die App mit `npm start`.

Jetzt sollte sich automatisch ein Browserfenster öffnen und folgenden Startbildschirm anzeigen:



Sollte sich nicht automatisch ein Fenster öffnen, dann gib folgende Adresse in deinen Browser ein:

`http://localhost:19002/`

⚠ Sollte sich das oben abgebildete Fenster weiterhin nicht öffnen, kannst du auch folgenden Befehl probieren: `expo start --web`

Wenn du die Expo Go App bereits auf deinem Smartphone installiert hast, brauchst du nur mit der Kamera-App den QR-Code scannen und kannst die App direkt auf dem Handy ausprobieren. Du kannst die App aber auch am Rechner emulieren, indem du *Run on Android device/emulator* oder *Run on iOS* wählst. Hierfür benötigst du die zugehörigen Emulatoren, die du installieren kannst, indem du den Tutorials für Android Studio<sup>6</sup> oder Xcode mit Command Line Tools für iOS<sup>7</sup> folgst. Hierbei wähle am besten gleich die Mobilgeräte, auf denen wir die App bewerten werden:

**iOS:** iPhone 8, iOS 14, 4.7" 1334x750 326 ppi (Xcode 12.3)

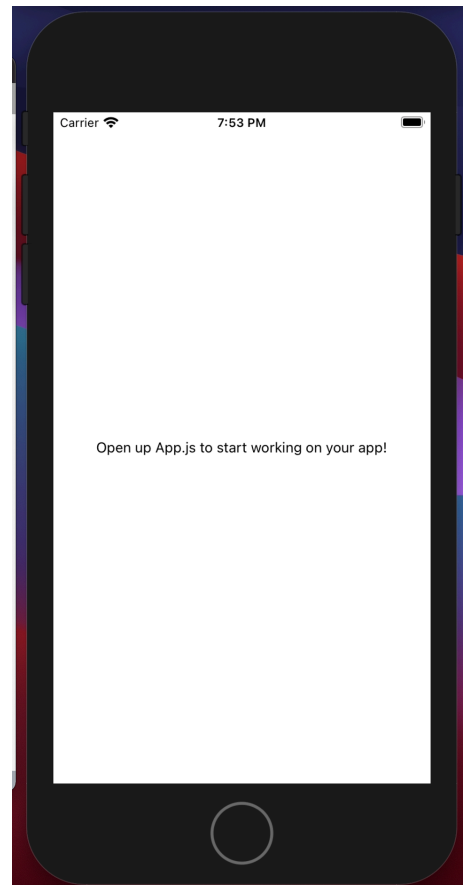
**Android:** Pixel 2, API-Level 27-30

⚠ Solltest du auf einem Android-Gerät emulieren wollen, dann öffne am besten den Emulator bevor du *Run on Android device/emulator* drückst. Dies kann zu Problemen führen.

Die App sollte nun im Emulator wie rechts abgebildet aussehen.

## 5. Öffne das Projekt in einer IDE

Starte die IDE (integrierte Entwicklungsumgebung) deiner Wahl und öffne die erstellte App darin. Im Grunde kannst du jede IDE verwenden, die du möchtest. Empfehlen kann ich allerdings **Android Studio**<sup>8</sup> (gratis), **Visual Studio Code**<sup>9</sup> (gratis) oder JetBrains' **WebStorm**<sup>10</sup> (Mit gratis Studierendenlizenz).



<sup>6</sup> <https://nabendu82.medium.com/android-emulator-setup-for-expo-on-mac-3511d3ef67e5>

<sup>7</sup> <https://medium.com/dev-genius/creating-apps-in-react-native-with-an-ios-simulator-39bab189bbee>

<sup>8</sup> <https://developer.android.com/studio?hl=de>

<sup>9</sup> <https://code.visualstudio.com>

<sup>10</sup> <https://www.jetbrains.com/de-de/webstorm/>

## 6. Hello World

Öffne die Datei App.js. Dort musst du nur Zeile 8 in `<Text>Hello World!</Text>` ändern. Diese sollte nun so aussehen wie hier abgebildet. Drücke auf speichern, die emulierte App aktualisiert sich von selbst und schon hast du deine erste Hello World!-App! Gratulation!

```
1  import { StatusBar } from 'expo-status-bar';
2    import React from 'react';
3  import { StyleSheet, Text, View } from 'react-native';
4
5  export default function App() {
6    return (
7      <View style={styles.container}>
8        <Text>Hello World!</Text>
9        <StatusBar style="auto" />
10     </View>
11   );
12 }
13
14 const styles = StyleSheet.create({
15   container: {
16     flex: 1,
17     backgroundColor: '#fff',
18     alignItems: 'center',
19     justifyContent: 'center',
20   },
21 });
```

## 7. Nächste Schritte

Bis hierher solltest du es selbst schaffen. Solltest du Probleme und Fragen haben, schreib ins Forum auf moodle! Sollte etwas dennoch ungelöst bleiben, können beim Tutorial noch Fragen geklärt werden. Ansonsten werden wir dort gemeinsam unsere erste kleine App bauen und eine API implementieren.

# Debugging

Für Chrome<sup>11</sup> und Firefox<sup>12</sup> gibt es eine Erweiterung namens React Developer Tools:



## React Developer Tools

Angeboten von: Facebook

★★★★★ 1.311 | [Entwicklertools](#) | 👤 2.000.000+ Nutzer

Im iOS Simulator kannst du mit Command + D das Expo Go Menü öffnen und dort mit „Debug Remote JS“ in den Debug Modus wechseln. Danach kannst du in Chrome im Entwicklermenü debuggen. Nach der Session solltest du den Modus aber wieder beenden, da er sehr viele Ressourcen frisst.

## Nützliche Links

### Installation

- Setting up the development environment: <https://reactnative.dev/docs/environment-setup>

### Tutorials

- YouTube-Tutorial für Anfänger: <https://www.youtube.com/watch?v=0-S5a0eXPoc>
- React Native The Basics Tutorial: <https://reactnative.dev/docs/tutorial>
- React-Tutorial auf W3Schools: <https://www.w3schools.com/react/default.asp>
- Introducing JSX: <https://reactjs.org/docs/introducing-jsx.html>

### Documentation

- React Native: <https://reactnative.dev/docs/getting-started>
- Basiskomponenten: <https://reactnative.dev/docs/components-and-apis>
- API: <https://reactnative.dev/docs/accessibilityinfo>

### Tools

- Tool zum Ausprobieren von React Native-Apps direkt im Browser: <https://snack.expo.io>
- Developing mobile apps with React Native in WebStorm: <https://blog.jetbrains.com/webstorm/2016/12/developing-mobile-apps-with-react-native-in-webstorm/>

## Erstelle Applikation

Wenn du deine App fertig entwickelt hast und abgeben möchtest, geh folgendermaßen vor:

Öffne in deinem Projekt die Datei app.json. Hier kannst du Namen, Icon und andere Parameter deiner App ändern.

### ANDROID

Öffne den Terminal im Root-Directory deiner App und führe folgenden Befehl aus:

```
expo build:android -t apk
```

---

<sup>11</sup> <https://chrome.google.com/webstore/detail/react-developer-tools/fmkadmapgofadopljbjfkapdkoienihi>

<sup>12</sup> <https://addons.mozilla.org/de/firefox/addon/react-devtools/>

Du wirst gefragt werden, ob Expo einen Keystore für dich erstellen soll, wähle:

1) Let Expo handle the process!

Dies kann nun mehrere Minuten lang dauern. Am Ende bekommst du einen Link, mit dem du die fertige App downloaden kannst. Um zu überprüfen, ob alles richtig funktioniert, kannst du das File einfach per drag'n'drop in den Simulator ziehen.

Diese .apk-Datei sollte Teil deiner Abgabe sein. Wir überprüfen dann im Simulator, ob alles funktioniert und schauen uns auch den Code an.

## IOS

Öffne den Terminal im Root-Directory deiner App und führe folgenden Befehl aus:

```
expo build:ios
```



Eventuell bekommst du folgende Warnung:

```
expo build:ios currently only supports managed workflow apps.  
Diese kannst du mit y quittieren.
```

Dann wirst du nach dem Build-Type gefragt, hier wähle Simulator:

```
? Choose the build type you would like: > - Use arrow-keys. Return to submit.  
archive  
> simulator - Run the build on a simulator
```

Dies kann nun mehrere Minuten lang dauern. Am Ende bekommst du einen Link, mit dem du die fertige App downloaden kannst.

Dieses .tar-File musst du nun entpacken und schon hast du deine fertige App. Um zu überprüfen, ob alles richtig funktioniert, kannst du das File einfach per drag'n'drop in den Simulator ziehen. Dann wird am Homebildschirm die App erstellt, die du dann testen kannst.

Diese Datei sollte dann Teil deiner Abgabe sein. Wir überprüfen dann im Simulator, ob alles funktioniert und schauen uns auch den Code an.

## Fragen?

Schreib ins Forum auf moodle!

## Probleme?

Da ich diesen Leitfaden neu zusammengestellt habe freue ich mich über Feedback und auch über die Info, sollte etwas nicht wie geschrieben funktioniert haben. Schreib ins Forum auf moodle!

von Bernhard Jordan, [bernhard.jordan@univie.ac.at](mailto:bernhard.jordan@univie.ac.at), 2021