

JS goes mobile: Eine Übersicht

Frederik von Berg
w11k / theCodeCampus

- <1> Über mich
- <2> Historie
- <3> Mobile Native JavaScript
- <4> Vergleich mit Nativ & Hybrid
- <5> Frameworks
- <6> Setup & Codebeispiel

Frederik von Berg

- <> Software Developer
Seit 2013 bei w11k GmbH
Scala und Web-Entwicklung </>

w11k GmbH - the Web Engineers

- <> Gegründet 2000
Entwicklung / Consulting
Web / Java
Esslingen / Siegburg </>

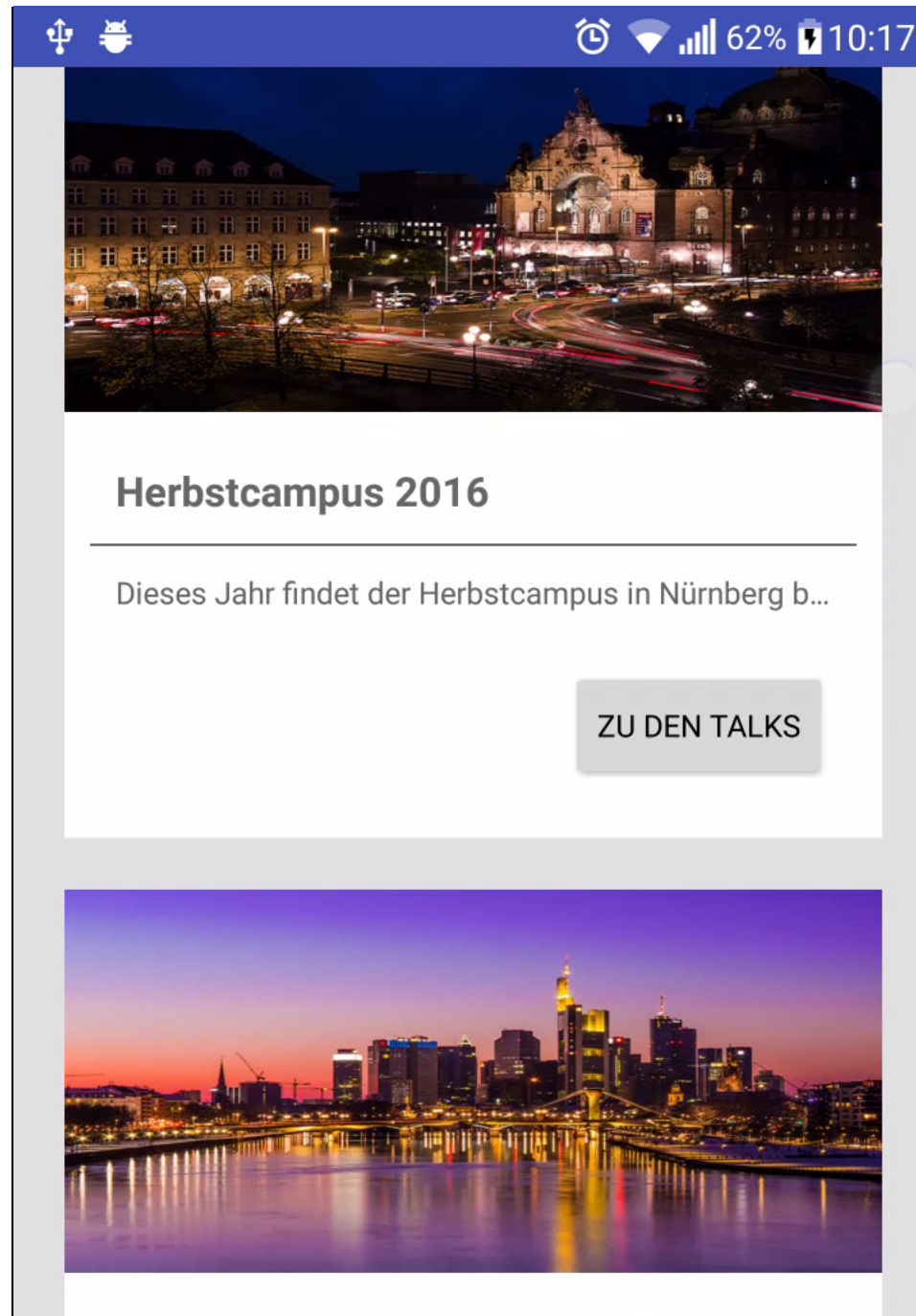
thecodecampus.de - Weiter. Entwickeln.

- <> Schulungen (seit 2007)
Projekt-Kickoffs
Unterstützung im Projekt </>

JS goes mobile

Ein Beispiel

Native Apps geschrieben in JavaScript?



IT Tage 2016

IT-Management- und Entwickler-Konferenz



NativeScript + TypeScript + Angular 2 [Code auf Github](#)

Native Apps

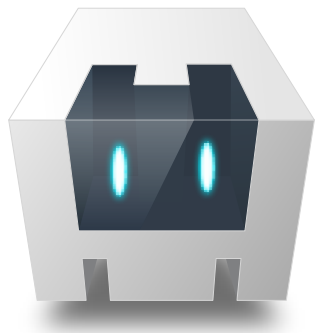
Vor gar nicht allzu langer Zeit ...



Hybrid Apps

Halt, ich schreibe doch keinen doppelten Code!

Halt, doppelter Code? - Hybrid



APACHE
CORDOVA™

[←](#) Add a Set

Title

Some Title

Another

Foobar

SAVE

Der Neuling - Mobile Native JavaScript

Native Apps in JavaScript



<> Native Anwendungen geschrieben in
JavaScript

<> kein WebView ⚡



Nein, nicht schon wieder ein Framework!

Was brauch ich als Entwickler?



<> Einen Rechner mit NodeJS

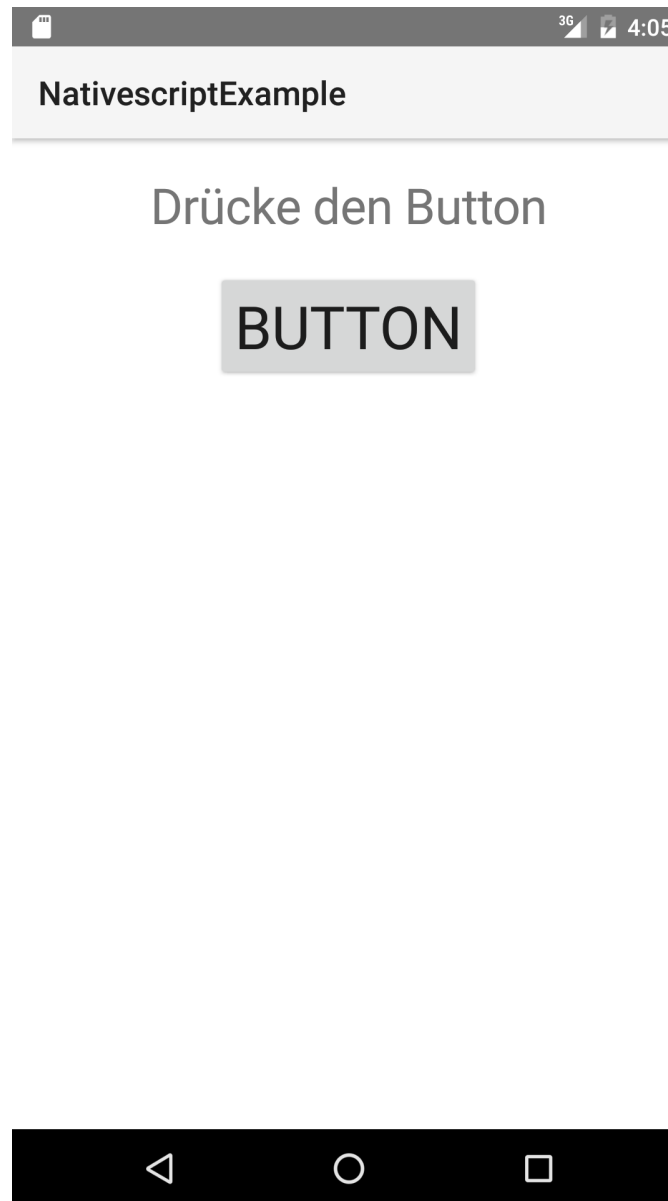
<> Kenntnisse

- JavaScript / TypeScript
- HTML & CSS
- Als Plus: Android / iOS Kenntnisse

Ein kleines Beispiel ...

NativeScript + Angular2 + TypeScript

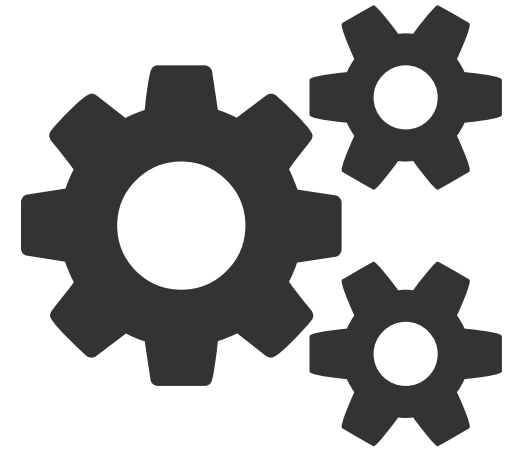
```
1 import {Component} from "@angular/core";
2
3 @Component({
4   selector: "my-app",
5   template: `
6     <StackLayout>
7       <Label text="Drücke den Button" class="title"></Label>
8       <Button text="Button" (tap)="onTap()"></Button>
9     </StackLayout>
10  `
11 })
12 export class AppComponent {
13   public onTap() {
14     console.log("Ich wurde geklickt");
15   }
16 }
```

Was ist *'Mobile Native JavaScript'* ?

Was ist 'Mobile Native JavaScript'?

- <> JS Engine generiert native Elemente
- <> Schnittmenge von nativen Komponenten
 - Label
 - Button
 - Switch, Slider, Image ...
- <> Voller Zugriff auf native Funktionalitäten (z.B. Kamera)
- <> Plattformspezifische Benutzerführung
- <> Plattformspezifische Anpassungen möglich



Vergleich zu nativen Apps

Vergleich zu nativen Apps

Vorteile MNJS:

- <> keine spezifischen Plattformkenntnisse nötig
- <> Wissen aus der Webentwicklung
- <> Ein Framework, mehrer Plattformen
- <> Wiederverwendbarkeit von Code

Vorteil Nativ:

- <> Schneller Startupzeiten
- <> höhere Performance möglich
- <> Besseres Tooling
- <> Größere Community

Vergleich zu hybrid Apps

Vorteile MNJS:

- <> Plattformspezifische Darstellung
- <> Native Bedienkonzepte
- <> Flüssige Bedienung
- <> Erlaubt direkten nativen Zugriff (z.B. Kamera, Android/iOS Features)

Vorteil Hybrid:

- <> Liefert (oft) noch eine Web-/Desktop-App
- <> Entwicklung im Browser
- <> Größere Community

Welche Frameworks gibt es?

React Native

- <> Facebook
- <> Open Source
- <> "learn once, write anywhere"
- <> JavaScript (JSX)
- <> CSS artiges Styling
- <> Layouts per Flexbox
- <> Native Elemente, aber kein natives Styling
- <> Debugging in Chrome möglich



Frameworks - React Native: Codebeispiel

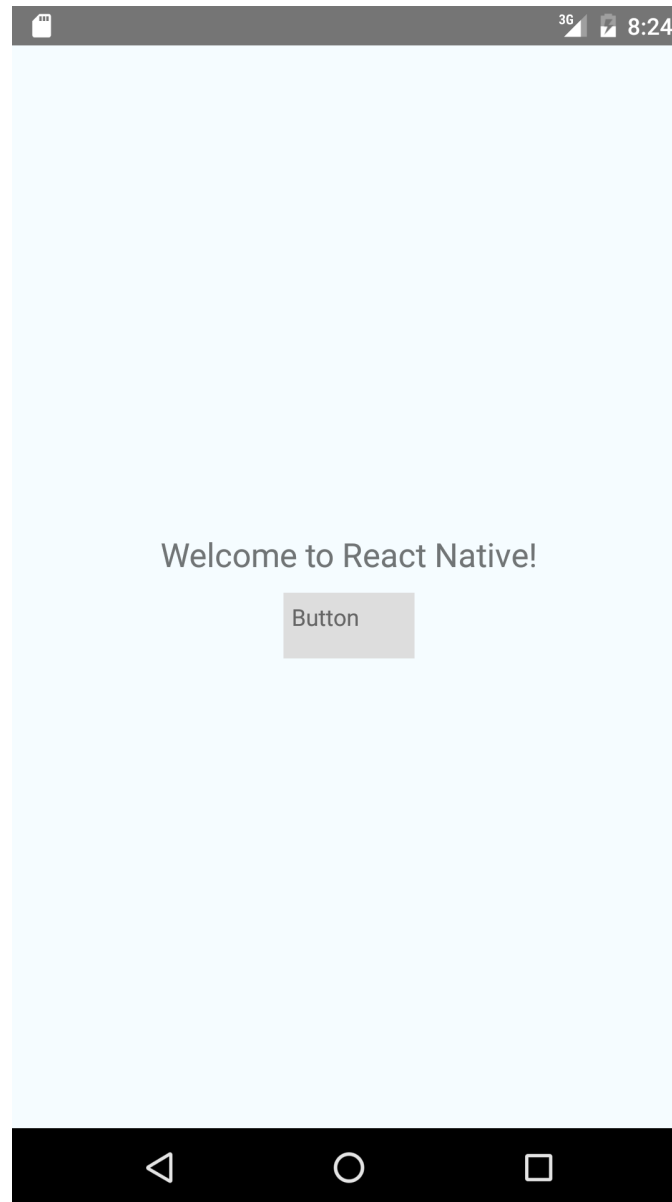
Code

React Native

```
1 import React, { Component } from 'react';
2 import { AppRegistry, StyleSheet, Text, TouchableNativeFeedback, View } from 'react-native';
3
4 class ReactNativeExample extends Component {
5   render() {
6     return (
7       <View style={styles.container}>
8         <Text style={styles.welcome}>
9           Welcome to React Native!
10        </Text>
11        <TouchableNativeFeedback onPress={this._onPressButton}>
12          <View style={{width: 80, height: 40, backgroundColor: '#DDDDDD' }}>
13            <Text style={{margin: 5}}>Button</Text>
14          </View>
15        </TouchableNativeFeedback>
16      </View>
17    );
18  }
19  _onPressButton() {
20    console.log("Ich wurde geklickt");
21  }
22 }
23
24 const styles = StyleSheet.create({
25   /* hier folgen noch mehr Styles */
26 });
27
28 AppRegistry.registerComponent('ReactNativeExample', () => ReactNativeExample);
29
```

Frameworks - React Native: Screenshot

Screenshot

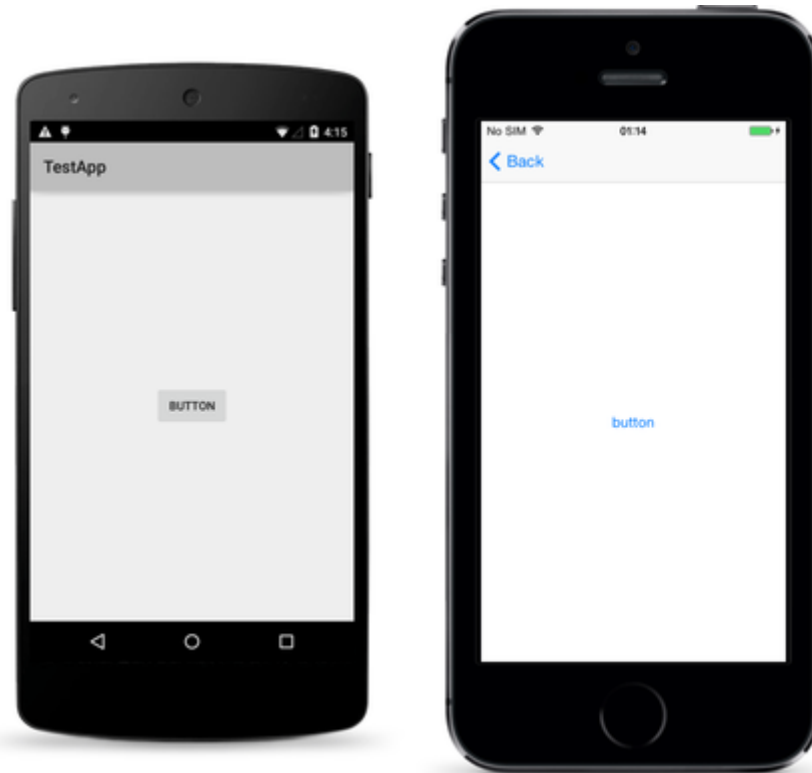


NativeScript

- <> Telerik
- <> Open Source
- <> JavaScript, XML(HTML), CSS
- <> TypeScript -> Angular2
- <> Bekannte Layouts
- <> Natives Styling als Standard
- <> Debugging über VSCode Plugin



NativeScript - Natives Styling



Ein einfaches Setup

NativeScript - Setup

Voraussetzung:

<> Android SDK/ X-Code & Node installieren

<> `npm install -g nativescript`

Projekt anlegen:

<> Normales Setup: `tns create Demo`

<> Angular 2 & Typscript: `tns create Demo --ng`

Anwendung starten:

<> `cd Demo`

<> Android hinzufügen: `tns platform add android`

<> App starten: `tns run android`

<> Livereload: `tns livesync android --watch`

Live Coding

Live Coding - Quiz

<> Layouts

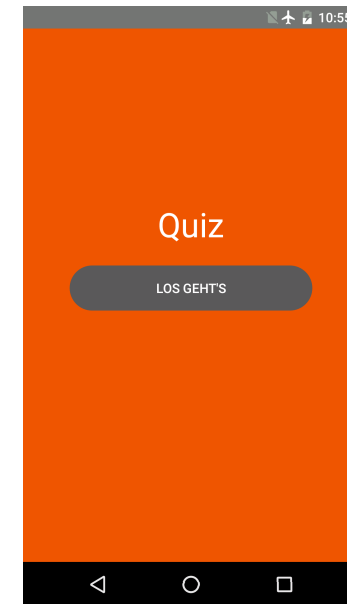
<> Styling

<> Aufbau

<> Animationen

<> Kameramodul

<> Navigation



Na, noch Fragen?

 frederikvonberg

 github.com/fvonberg

info@thecodecampus.de
@thecodecampus

www.w11k.de
www.thecodecampus.de

Webseiten:

<https://www.nativescript.org/>

<https://facebook.github.io/react-native/>

<https://developer.android.com/index.html>

<https://cordova.apache.org/>

<https://crosswalk-project.org/>

<http://phonegap.com/>

<https://developer.apple.com/>

Logos und Bilder:

[Font Awesome by Dave Gandy -](#)

<http://fontawesome.io>

[Nativescript Android Button](#)

[Nativescript iOS Button](#)