NativeScript & Angular 2

Von der Web-App zur nativen Smartphone-App

Frederik von Berg
W11K / theCodeCampus

Überblick

<1> Über mich

<2> Ausgangslage

<3> Was ist Angular?

<4> Was ist NativeScript?

<5> Was muss man bei Apps beachten?

<6> Portierung

Über uns

Frederik von Berg

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

W11K GmbH - the Web Engineers

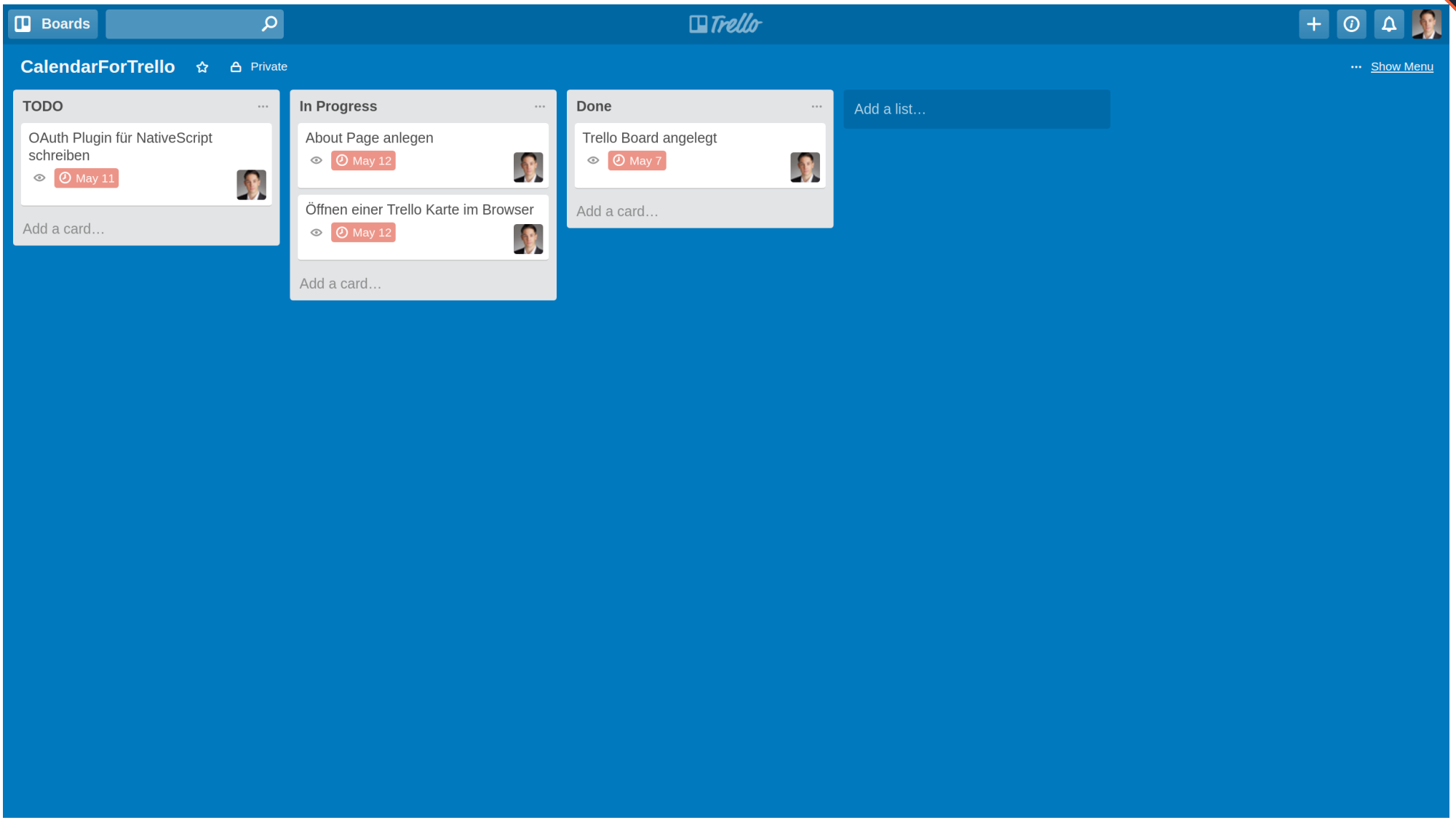
<> Gegründet 2000
Entwicklung / Consulting
Web / Java
Esslingen / Siegburg </>

theCodeCampus.de - Weiter. Entwickeln.

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

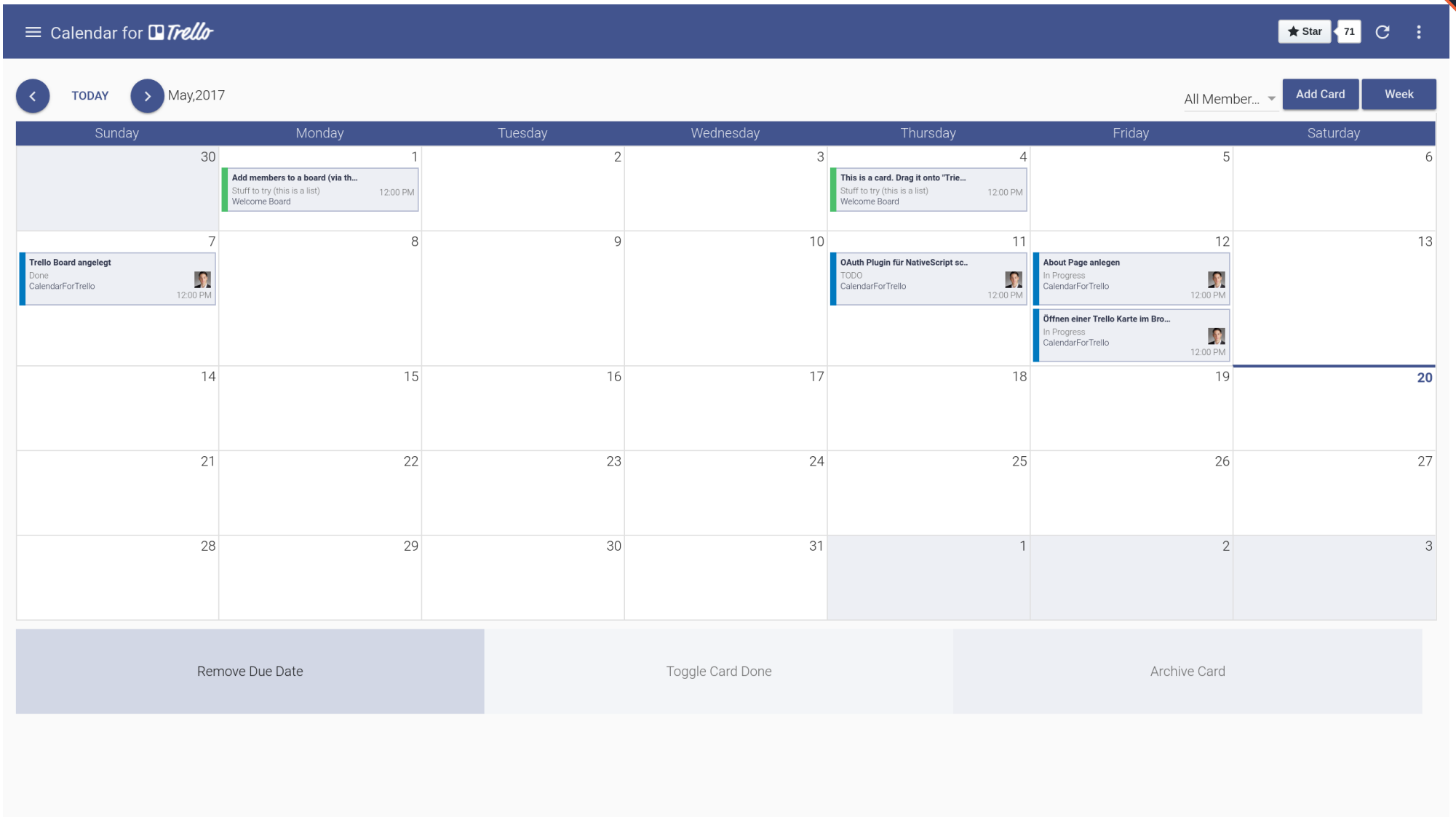
Trello

Screenshot



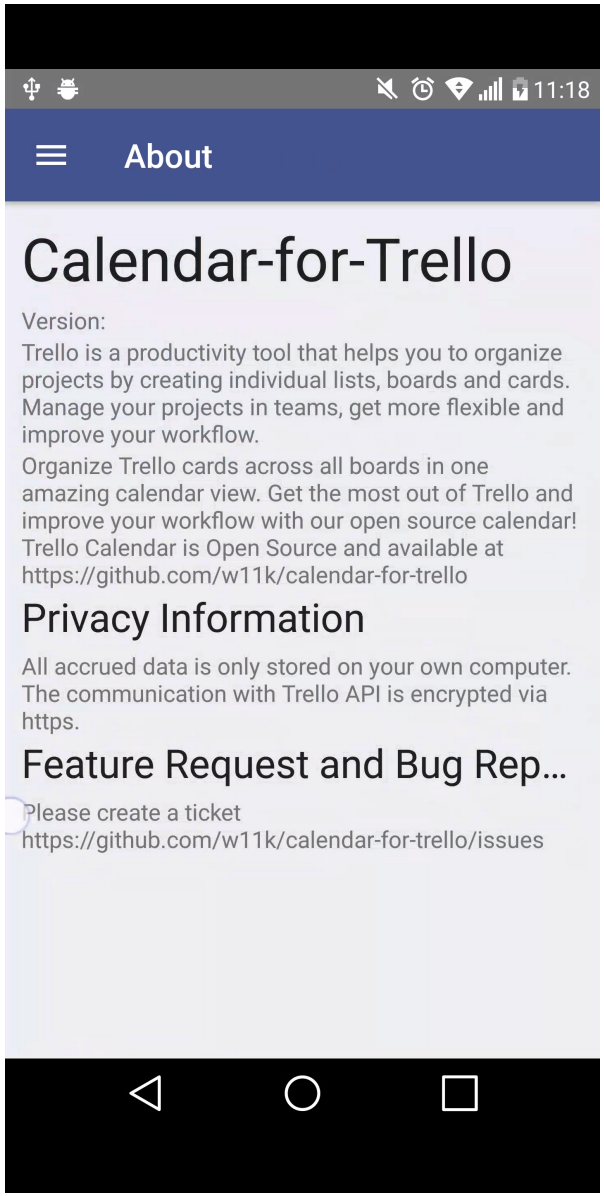
Calendar for Trello

Screenshot



Native App

Screenshot



https://calendar-for-trello.com/

Ausgangslage

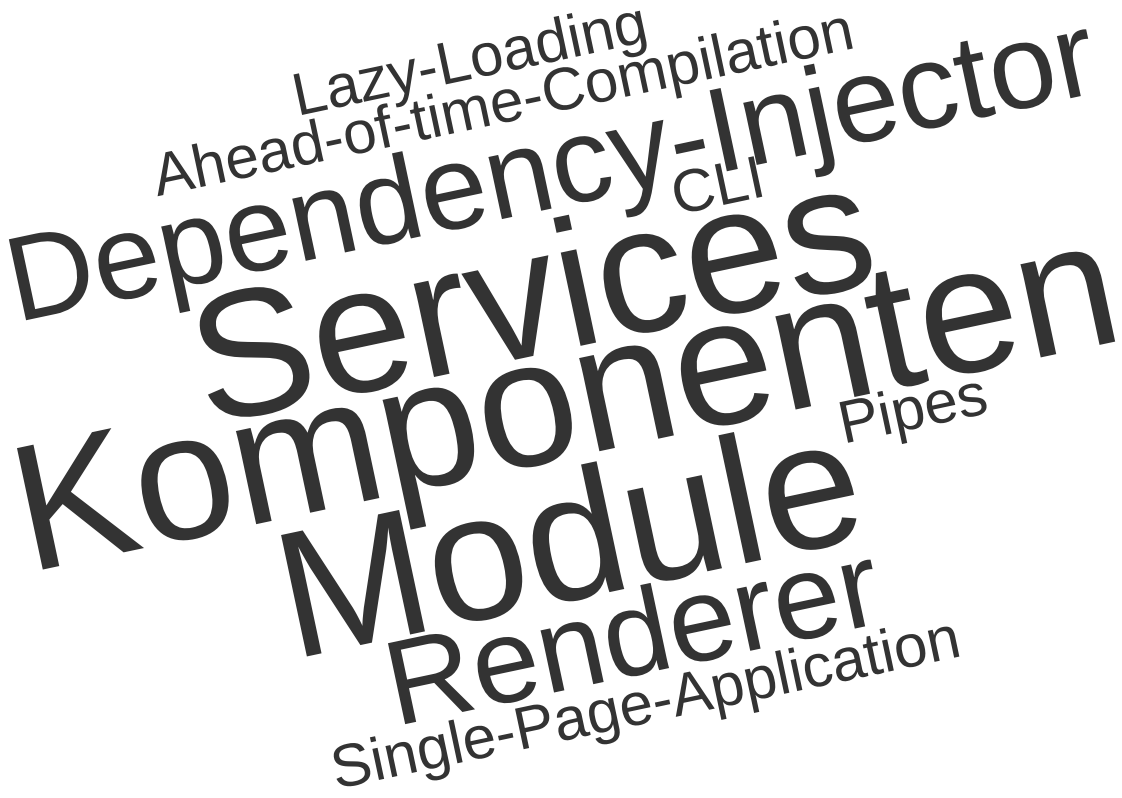
<> Calendar for Trello - Angular Webapp

<> Portierung mit wenig Aufwand

<> keine Hybrid App

Was ist Angular?

Was ist Angular?



Komponenten

<> **Grundlegendes Konzept in Angular 2**

<> **Gesamtes UI ist aus Baum von Komponenten aufgebaut**
- Wiederverwendbare Elemente
- Grungerüst der Seite

<> **Komponenten bestehen aus**
- Template / View (HTML)
- Klasse
- Decorator an Klasse

Komponenten

Code

Typescript

```
1  import { Component } from "@angular/core";
2
3  @Component({
4      // css selector to find the component's usage
5      selector: "music-app",
6      template: `
7      <h1>Music App</h1>
8      `
9  })
10 export class MusicAppComponent {}
```

Html

```
1 <body>
2 <music-app></music-app>
3 </body>
```

Services

<> **Sehr allgemeines Konzept**
- Irgendwelche Funktionalität kapseln
- Nicht UI spezifisch
- Implementierungsdetails verstecken

<> **Einsatzmöglichkeiten**
- Höhere Abstraktion schaffen (REST)
- Integration anderer Bibliotheken (WebSocket)
- Datenhaltung, Datenzugriff, Caching, Logik, ...

<> **Können Abhängigkeit zu anderen Services haben**

Services

Code

<> Services sollten in Angular 2 als Klassen implementiert werden

```
1  import {Artist} from "./artist.model";
2
3  export class ArtistService {
4      getAllArtists(): Artist[] {
5      /* return demo data or data from local storage
6      * or data loaded from server or ... */
7      return [];
8      }
9  }
```

Module

<> **Angular-spezifischer Code muss organisiert werden**

- Gleiche Grundgedanken und Aufbau wie bei ES6 Modulen
- Kapselung & Wiederverwendbarkeit
- Leichtes Einbinden von Bibliotheken
- Genaue Steuerung was wo verwendet wird

<> **Ein Modul ist eine annotierte Klasse**

Module

Code

<> **Angular-Module bestehen u.a. aus**

- Imports: Welche anderen Module werden intern verwendet
- Declarations: Was intern bekannt ist (quasi lokale Variablen)
- Exports: Was wird nach außen bekannt gemacht (für andere Imports -> Module)

```
@NgModule({
    declarations: [ AppComponent, MusicAppComponent ],
    imports: [ BrowserModule, FormsModule, HttpModule ],
    exports: [ AppComponent ],
    providers: [ ArtistService ]
})
export class AppModule { }
```

Renderer

<> **DOM Entkoppelung**

<> **Renderer wird beauftragt, um DOM zu ändern**

<> **Typen:**

- Angular Renderer
- Angular Universal
- NativeScript

Was ist NativeScript?

Was ist NativeScript?

- <> Open Source
- <> JavaScript, XML(HTML), CSS
- <> TypeScript -> Angular2
- <> Layouts
- <> Generiert Native Elemente
- <> Android 4.2, iOS 8

Was ist NativeScript? - Funktionsweise

<> 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

<> Node Plugins können genutzt werden

<> Unit Testbar

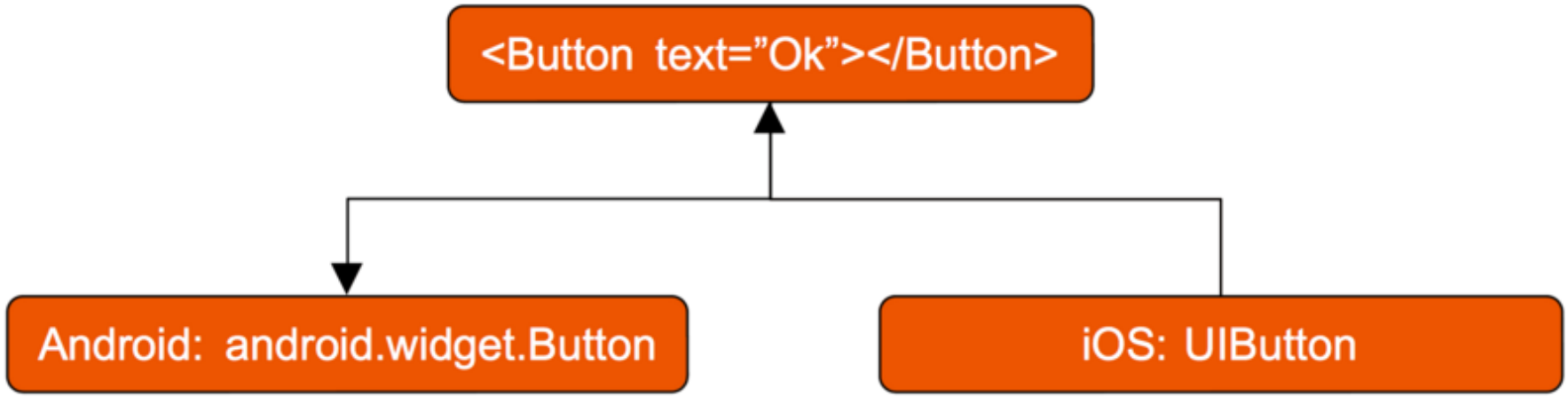
<> UI Testframework verfügbar

Was ist NativeScript? - Funktionsweise

Code

```
if (application.android) {
    let intent =
        new android.content.Intent(
                android.content.Intent.ACTION_VIEW,
                android.net.Uri.parse(this.card.url)
            );
    application.android.currentContext.startActivity(
                android.content.Intent.createChooser(intent,
                "Open Card...")
            );
} else if (application.ios) {
    dialogs.alert("Sorry, this is not implemented yet.")
    .then(function() {
        console.log("Dialog closed!");
    });
}
```

Was ist NativeScript? - UI-Komponenten



Was ist NativeScript? - UI-Komponenten

Was ist NativeScript? - Layout

- <> AbsoluteLayout
- <> DockLayout
- <> GridLayout
- <> StackLayout
- <> WrapLayout
- <> FlexBoxLayout

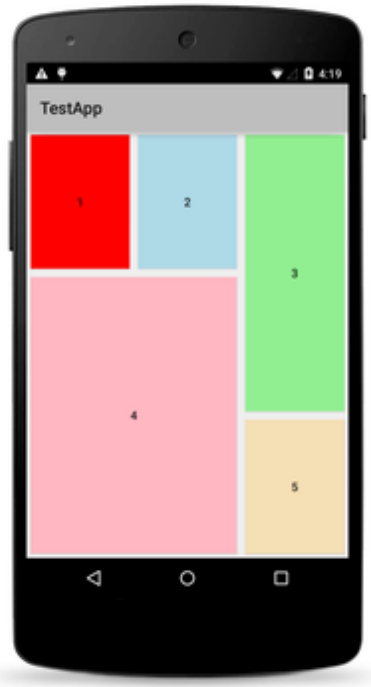


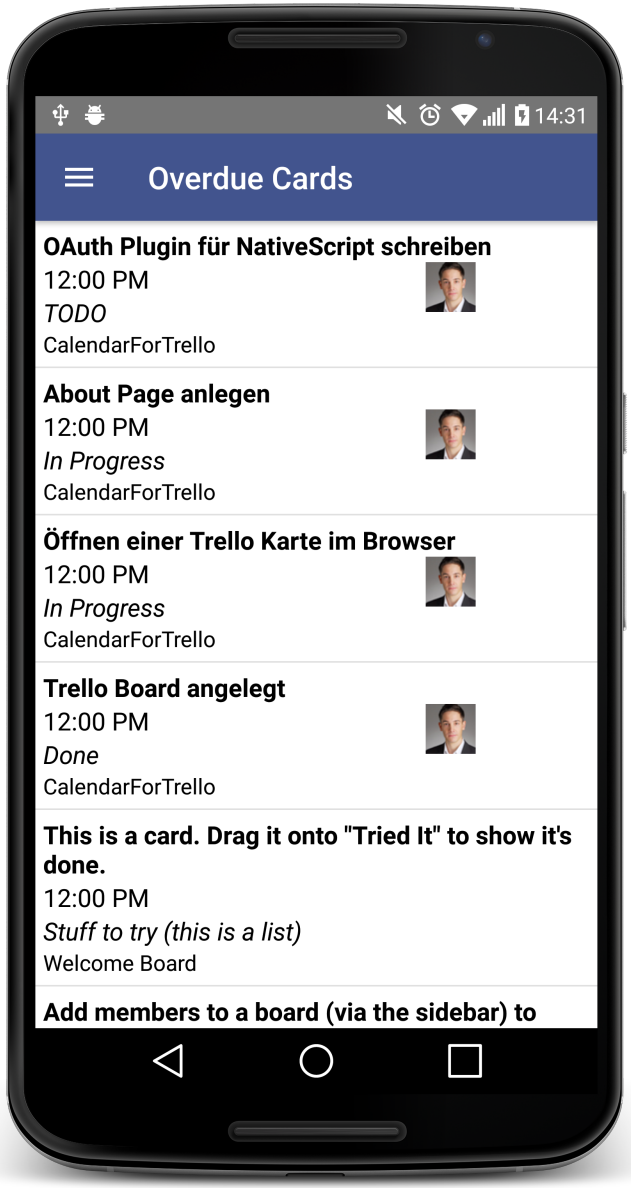
Abb: NativeScript Docs

Was ist NativeScript? - Layout

Code

```
<MainActionBar title="Overdue Cards"></MainActionBar>
<StackLayout class="page">
  <ListView [items]="cards" *ngIf="cards.length > 0"
                class="list-group">
    <ng-template let-card="item">
        <calendar-card [card]="card" class="list-group-item">
        </calendar-card>
    </ng-template>
  </ListView>
</StackLayout>
```

Was ist NativeScript? - Layout



Was ist NativeScript? - Styling

<> CSS Styling (Sass & Less möglich)

<> Ganze App: `app.css`

<> Plattformspezifisch: `app.android.css` | `app.ios.css`

<> ein Stylesheet pro Template (Angular spezifisch)

<> Inline-Styling

<> Theme seit 2.4
- definierte CSS-Attribute (`h1, body, text-primary, m-t-5`)
- Themebuilder

Was ist NativeScript? - Tooling

<> **Benötigt:**

- Node, npm ... IDE der Wahl
- Android SDK + Emulator
- XCode für iOS
- JS, HTML, CSS - Kentnisse ... learning by doing ;-)

<> **Liefert:**

- CLI -> `npm install -g nativescript`
- Visual Studio Code Plugin
- Hot Reloading / Livesync
- Debugging (Chrome oder Visual Studio Code)

Was ist NativeScript? - Tooling

<> **Projektsetup:**

- `tns create FirstApp --ng`
- `cd FirstApp/`
- `tns run android`

Vorteile einer App

Was muss man bei Apps beachten?

Was muss man bei Apps beachten?

<> **Web App / Website:**

- Informativ
- Benötigt eine Datenverbindung
- einfache Updates
- Responsive

<> **Mobile app:**

- Interaktiv - Task driven
- ohne Datenverbindung möglich
- Benachrichtigungen
- Branding / Kundenbindung
- Performance
- Native Oberfläche, gewohnte Benutzerführung

Mobile App auf Web App Basis

Eine Codebasis?

Mobile App auf Web App Basis - Allgemein

<> **UI und plattformunabhängiger Code teilen (Services, Models, Pipes, ...)**
- NPM Abhängikeiten müssen gleich sein
- `window` darf nicht referenziert sein
- platformspezifischer Code kann evtl. nicht geteilt werden (z.B. "Localstorage")

<> **Komponenten haben unterschiedliche Templates**

<> **Authentifiezierung muss evtl. umgestellt werden**

Mobile App auf Web App Basis - Allgemein

<> **Möglichkeiten:**

- Projektstart: Angular + NativeScript Seed/Starter Projekt
- Komplex: geteilte NPM Pakete
- Pragmatisch: Symlinks

⚠ **Geteilte Resourcen müssen immer unterhalb des /app Verzeichnis in NativeScript liegen, sonst werden diese nicht mit in die App gepackt und können nicht genutzt werden.**

Mobile App auf Web App Basis - Seed Projekte

<> **Verfügbarkeit:**
- NathanWalker/angular-seed-advanced
- jlooper/angular-starter

<> **Aufbau:**
- alles in einem Verzeichnis
- geteilter Code in "shared" Verzeichnis
- Templates werden ausgetauscht (*.tns.html)

<> **Funktionsweise:**
- Build kopiert shared files in NativeScript
- Build ersetzt die Importpfade entsprechend

<> **Für neue Projekte geeignet**

<> **Annahme: die UI wird auseinanderlaufen**

<> **viel "Magic" im Spiel**

Mobile App auf Web App Basis - NPM Paket

<> geteilen Code in npm Pakete packen

<> kapseln in Module

<> Paket in NativeScript Projekt nutzen

<> Herausforderung:
- Pflege
- Build + Auslieferung

<> Geeignet: falls die WebApp schon fertig ist

<> Projekte müssen nicht am gleichen Ort liegen

Mobile App auf Web App Basis - Shared Folder

<> Pragmatisch

<> Dependencies des Clients auf NativeScript anpassen

<> geteilten Code in "shared"-Ordner legen

<> "shared"-Ordner per Symlink in NativeScript "app"-Ordner einbinden

<> Funktioniert super für einen Prototyp

<> erstmal keine Buildanpassung nötig

Na, noch Fragen?

frederikvonberg

github.com/fvonberg

info@thecodecampus.de
@thecodecampus

www.w11k.de
www.thecodecampus.de

Quellen

Webseiten:

- https://www.angular.io/
- https://www.nativescript.org/
- https://developer.android.com/index.html
- https://developer.apple.com/

Logos und Bilder:

- Font Awesome by Dave Gandy - http://fontawesome.io