

Mobile & Web Programming

Web- & hybride mobile Apps mit Ionic



kaspar.vongunten@hslu.ch



Programm heute

- Übersicht "Entwicklungsansätze Mobile Apps"
- Mobile Web-Anwendungen
 - Grundlagen: HTML, CSS & JavaScript
- Web-App mit Ionic
 - Grundlagen & Werkzeuge
 - Seiten- und Navigationsstruktur
 - Ionic "How-To"
- Hybride Web-Apps mit Cordova

Viel Stoff heute...



Disclaimer: „Crash Course“

- Viele Technologien heute!
 - Ziel: Sie sollen ein Eindruck von diesen bekommen
 - Sie werden dadurch keine Experten für HTML, CSS, Java-/TypeScript, Ionic und Cordova...
 - ...haben jedoch genügend Infos, um eine eigene einfache hybride App mit ionic und Cordova zu entwickeln (siehe Übung)
- ☞ Wie immer: Fragen gerne direkt stellen!



TypeScript



APACHE
CORDOVA™



<http://en.wikipedia.org/wiki/File:Somethingdifferent.jpg>

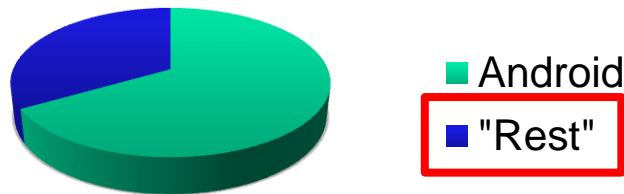
Mobile Web- Anwendungen

Beschreibung MOBPRO-Modul

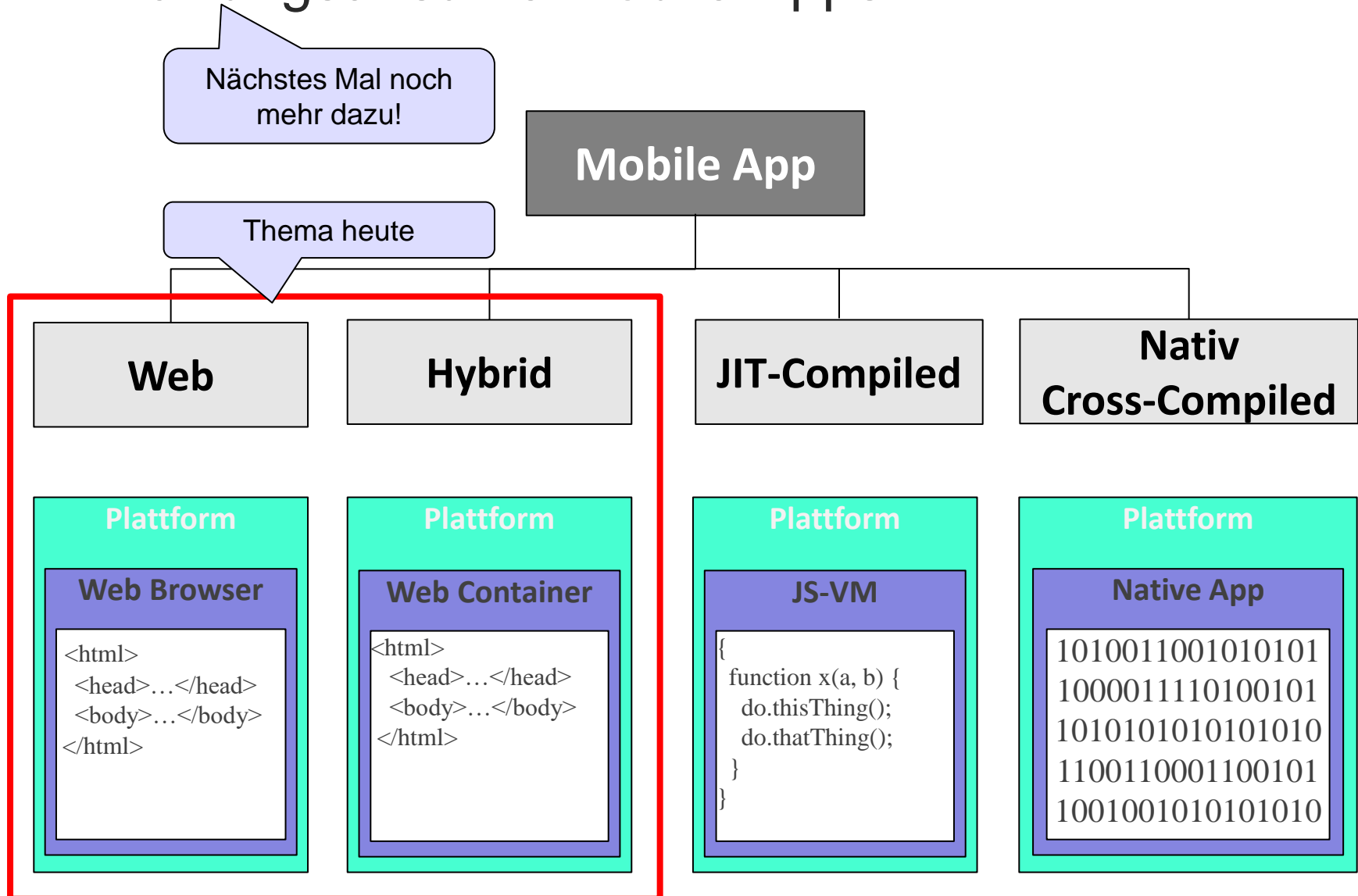
Einführung in **mobile Plattformen** und die App-Programmierung für **Android**, Behandlung von Grundkonzepten der **Android-Programmierung**,
Erstellung von hybriden Apps, Vermittlung und Diskussion von **verschiedenen Entwicklungsansätzen** für mobile Apps (**nativ, cross-platform/hybrid, web**).

■ Inhaltübersicht

- **ca. 2/3:** Android
- **ca. 1/3:** Auslegeordnung & Diskussion Entwicklungsansätze & mobile Plattformen, Vertiefung Web- und hybride Apps (Ionic, Cordova), Vertiefung Ansatz cross-Platform (Native-Script), cross-compile (Xamarin)



Entwicklungsansätze Mobile Apps



Web-App Entwicklung: Optionen

1. Web-App

JavaScript, HTML, CSS mit Responsive Design
Wird im Browser ausgeführt.

2. Hybrid

Webapp in nativen Wrapper «verpackt»
Mit Konnektor-Plugins für native Services
Kann als native App installiert werden.
Bsp: Cordova/Phonegap

Charakteristik mobile **Web-App** & Beispiele

- Eine Web-Anwendung für mehrere/alle Plattformen
- Voraussetzungen pro Plattform: Web-Browser
- Verwenden meist JavaScript-Web-Frameworks, z.B. Ionic, jQuery Mobile, Titanium, etc.

Web-App: Vor- und Nachteile

- + Eine Code-Basis für mehrere/alle Plattformen
- + Kein App-Store, Download oder Installieren notwendig
- + App kann jederzeit veröffentlicht/verändert werden
- + Vorhandene Web-Applikation kann zu mobiler App erweitert werden (Stichwort Responsive bzw. Progressive)
- Läuft im Browser („Rahmenapplikation“)
- Kein Zugriff auf alle Funktionen (Kontakte, ext. HW, ...)
Jedoch: HTML5 kann immer mehr! (Storage, Kamera, Geo, Gyro, ...)
- App nicht im Store, Benutzer müssen sie finden (Marketing!)
Keine installierte App/App-Icon, ggf. Browser-Lesezeichen auf Home-Screen...

Charakteristik **Hybride Apps** & Beispiele

- Eine Web-Anwendung für mehrere/alle Plattformen, verpackt in einen nativen Container
 - “Eingepackte“ Webapp, die in einer WebView läuft und durch einen nativen Container installierbar wird
 - Zugriff auf native Features durch Plugins (können auch selber geschrieben werden)
- Voraussetzungen pro Plattform: Nativer Container, z.B.
 - Apache Cordova (Open Source)
 - PhoneGap (kommerzielles Produkt)
 - (Appcelerator) Titanium



Hybride App: Vor- und Nachteile

- + Eine Code-Basis für alle Plattformen (grundsätzlich)
- + App-Stores: App findbar, gewisse App-Qualität + Sicherheit
- + Zugriff auf viele Geräte-Funktionen durch native APIs
- +/- Nativer Look&Feel? (Je nach Technologie!..)
- Kein garantierter Zugriff auf alle Geräte-Funktionen
- Unterstützung versch. Plattformen kann aufwändig sein, ggf. spezifischer Code / Anpassungen notwendig (durch eigene platform-spezifische Plugin-Implementierungen)

Motivation / Zusammengefasste Vorteile

Warum eine App als Web-Anwendung entwickeln?






- Läuft grundsätzlich auf allen Plattformen
- Einfacher Update-Prozess (für alle Applikationen gleich)
- Developer: Web-Knowhow vs. Native-Knowhow
- Kosten: Entwicklung nur 1x nötig (nicht 1x pro Plattform)
- Leichtgewichtig: Keine Installation nötig (one-time use?)
- Synergien: Ggf. existierende Desktop-Web-Plattform
- Zunehmend mehr native Fähigkeiten mit HTML-5 (Kamera, Persistenz, Location, ...)

Traditionelle Probleme von Webseiten auf mobilen Geräten

- Limitierte Display-Grösse
- Portrait / Landscape Modes
- Interaktion: Touch \neq Maus, Gesten, Gyro, ...
- Unterschiedliche Browser (CSS, JavaScript)
- Kein Zugriff auf HW Ressourcen (Kamera, Location, Storage, Sensoren, Bluetooth, ...) oder SW Ressourcen (Kontakte, ...)

Toolkits

Verschiedene Toolkits adressieren diese Probleme:

- Bootstrap 
 - Sencha Touch  Sencha
 - jQuery(Mobile) 
 - Ionic 
 - Cordova / Phonegap 
 - u.v.a.m. (!!!)
 - ...
- Benutzen wir heute!



<http://en.wikipedia.org/wiki/File:Somethingdifferent.jpg>

Grundlagen: HTML, CSS & JavaScript (Extrem minimale Einführung)

Grundlagen: Struktur einer Web Applikation

Trennung von Struktur, Darstellung & Verhalten

- **HTML:** **Struktur** (DOM = Document Object Modell)
(HyperText Markup Language)
- **CSS:** **Darstellung**
(Cascading Style Sheets)
- **JavaScript:** **Verhalten**



HTML-Seite

- Eine einfaches HTML-Dokument

```
<!DOCTYPE html>
<html>
<head lang="en">
  <meta charset="UTF-8">
  <title>Welcome</title>
  <!--
  <link rel="stylesheet" href="styles.css" type="text/css">
  -->
</head>
<body>
  <div class="title">Hello World</div>
  <p>How are you doing?</p>
</body>
</html>
```

Dazu gleich noch mehr



CSS Styles

- CSS spezifiziert die Darstellung von HTML Elementen
- Spezifiziert mit Property-Namen und Wert:
`font-size: 12pt;`
`text-indent: 1cm;`
- Selektor + Properties = Style Rule

```
body {  
  background: beige;  
  color: darkslategrey;  
  font-size: 12pt;  
}  
  
div.title {  
  font-size: 14pt;  
  font-weight: bold;  
}  
  
p {  
  text-indent: 1cm;  
}
```



Java Script: 4 Verwendungsarten...

```
<!DOCTYPE html>
<html>
  <head lang="en">
    <title>Quack</title>
    <script> // 1. inline script
      var quack = function() {
        var element = document.getElementById("quack");
        var now = new Date();
        element.innerHTML = "The duck quacked at " + now.toLocaleTimeString();
      }
      window.onload = quack; // 3. register html event handler
    </script>
    <script src="myMooScript.js"></script> // 2. external script
  </head>
  <body>
    <p><div id="quack">[quack]</div></p>
    <p><div id="moo">[muh]</div>
    <input id="clickMe" type="button" value="do ext. moo" onclick="moo();" />
    <!-- next line: 4. url with javascript:-protocol -->
    <p><a href="javascript:void(n=prompt('enter a number:', '77'))";
      alert('the number was = '+n)">show number dialog</a></p>
  </body>
</html>
```

Woher kommt wohl
diese Funktion moo()?



...und Effekt im Browser ☺

The duck quacked at 3:34:57 PM

The cow mooed at 3:37:00 PM

do external moo

[show number dialog](#)

The duck quacked at 3:34:57 PM

[muh]

do external moo

[show number dialog](#)

the number was = 1234

☐ Prevent this page from creating additional dialogs

OK

javascript:void(n=prompt('enter a number:','77')); alert('the number was = '+n)

Zwei grosse Probleme bei der Web-Entwicklung

1. Nicht alle Browser rendern CSS-Styles gleich
2. Nicht alle Browser verstehen das gleiche JavaScript

- **Lösung für 1**

- ionic, Bootstrap, u.v.a.m ...

- **Lösung für 2**

- ionic, extjs, dojo, jquery, YUI, u.v.a.m. ...



<http://en.wikipedia.org/wiki/File:Somethingdifferent.jpg>

Mobile App- Entwicklung mit Ionic

Noch ein Disclaimer

1. Ich bin kein Ionic/Angular/Web-Entwickler/Experte
2. Das Thema bietet (endlos) viel Stoff



TypeScript



APACHE
CORDOVA™

👉 Hier gibt's einen kompakten pragmatischen Einblick aus Sicht "Mobile App-Entwicklung" 😊

Crash Course "Ionic"

- Ziel: Praktischer, pragmatischer Einstieg
 - Vor allem anhand einer Demo-App
- Sie sollen danach...
 - Einen Grobüberblick haben
 - Was ist Ionic? Was kann ich damit tun?
 - Eine eigene App entwickeln können, d.h. Sie haben u.a. Basiswissen zu
 - Seiten-Aufbau & Navigation
 - HTTP-Anfragen & Daten anzeigen ("Bindings")
 - Native App mit Cordova

Ionic



TypeScript



"When reviewing any new technology, library, or framework, you'll first ask, "Where does this fit into the stack, and how would it benefit me?" TL;DR: Ionic's ultimate goal is to make it easier to develop native mobile apps with HTML5, also known as Hybrid apps."

<https://blog.ionicframework.com/where-does-the-ionic-framework-fit-in/>

Ionic...

- bietet UI-Komponenten, APIs, usw.
 - Inkl. Projekt Vorlagen (App-Layouts & Navigation)
- Bietet Skripte & CLI (Command Line Interface)
- benutzt u.a. Angular, TypeScript & Cordova

Doku zum Einstieg

- Ionic-Tutorial
 - <https://ionicframework.com/docs/intro/tutorial/>
- Getting Started
 - <https://ionicframework.com/getting-started/>
- Blog-Post von JetBrains zu WebStorm & Ionic
 - Inkl. Hinweise zum Debugging
 - <https://blog.jetbrains.com/webstorm/2017/08/developing-ionic-apps-in-webstorm/>

Installation Ionic & Cordova (mit npm)

```
$ [sudo] npm install -g ionic cordova
```

```
...
```

```
+ cordova@9.0.0
```

```
+ ionic@4.12.0
```

```
added 128 packages, removed 531 packages, updated 2 packages  
and moved 629 packages in 25.996s
```

- Siehe: <https://ionicframework.com/docs/intro/installation/>
 - inkl. Installations-Hinweise für npm (Node Package Manager), siehe <https://nodejs.org/en/>

App-Projekt anlegen

```
$ ionic start myApp sidemenu
```

...

Next Steps:

* Go to your newly created project: `cd ./myApp`

* Get Ionic DevApp for easy device testing: <https://bit.ly/ionic-dev-app>

Blank app

```
$ ionic start myApp blank
```

Tabs



```
$ ionic start myApp tabs
```

Menu

Home	>
Friends	>
Events	>
Close	>

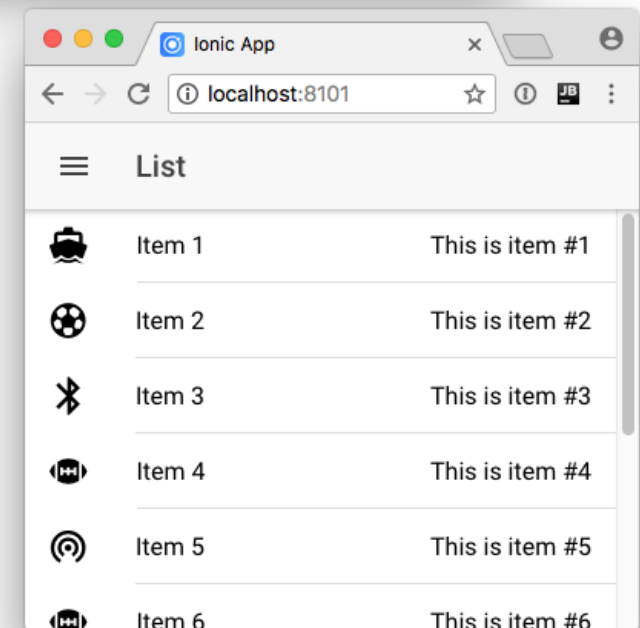
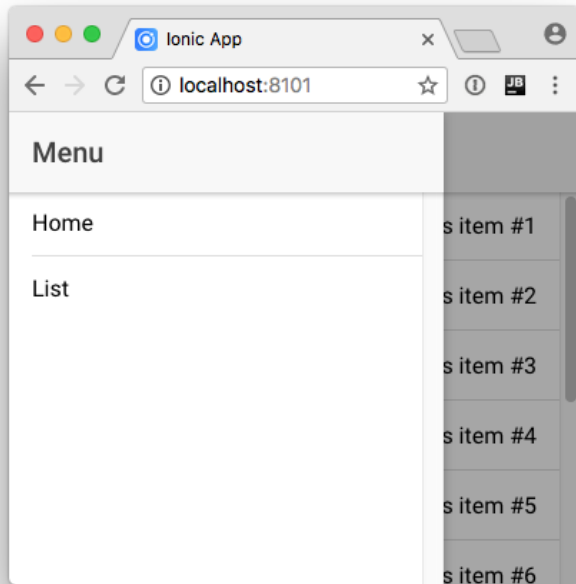
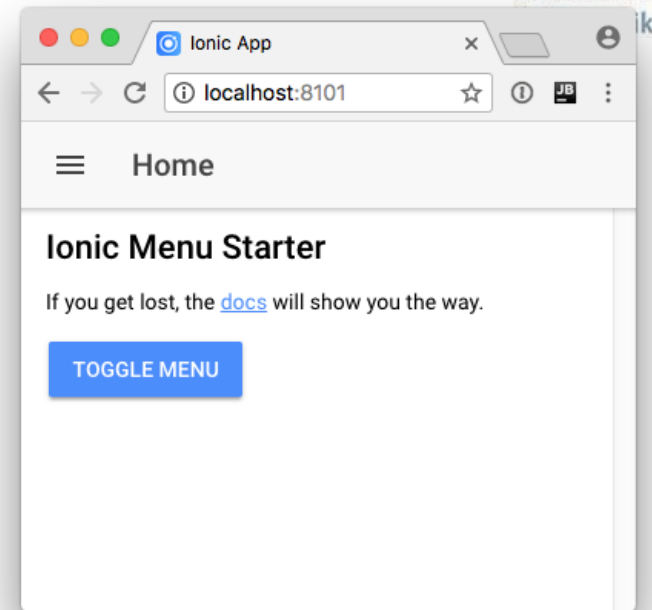
```
$ ionic start myApp sidemenu
```

App starten 😊

```
$ cd ./myApp
```

```
$ ionic serve
```

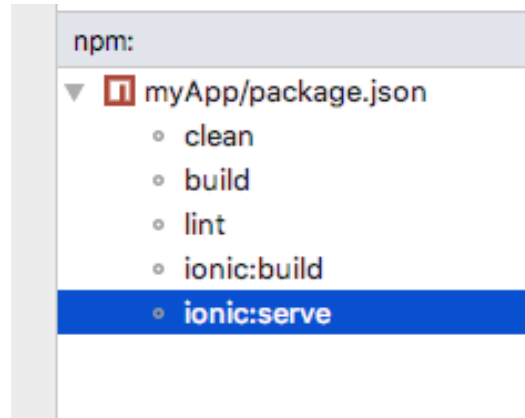
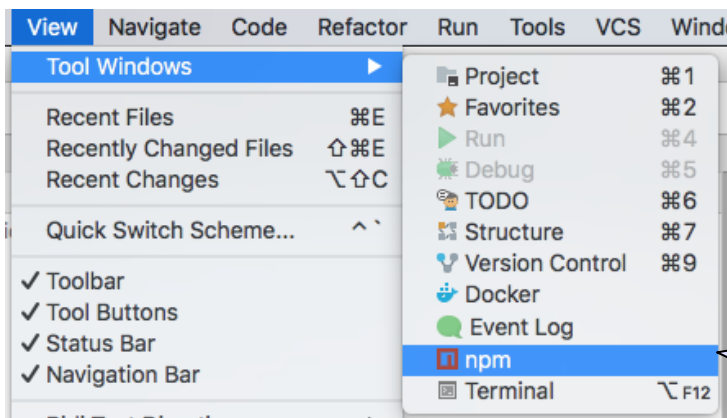
Doku aller Ionic CLI-Kommandi:
<https://ionicframework.com/docs/cli/commands.html>



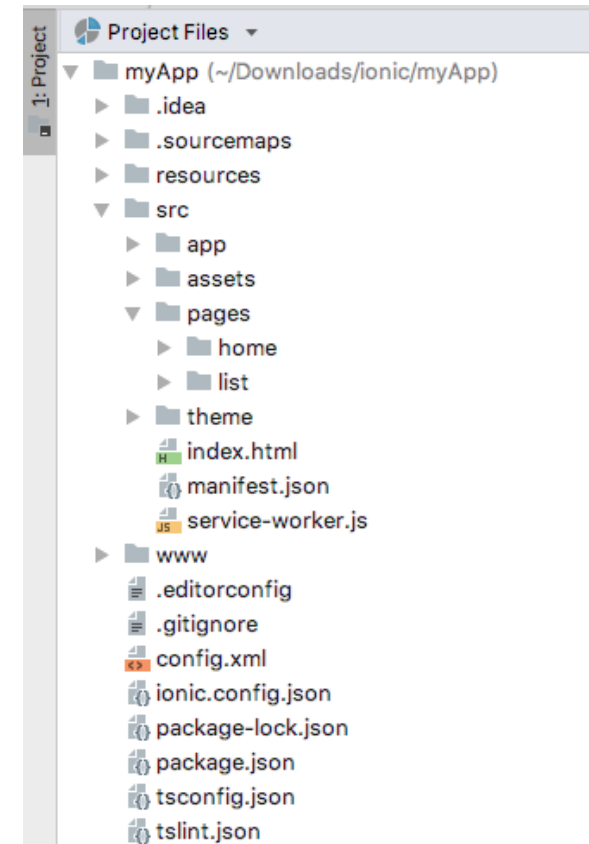
Werkzeug: WebStorm



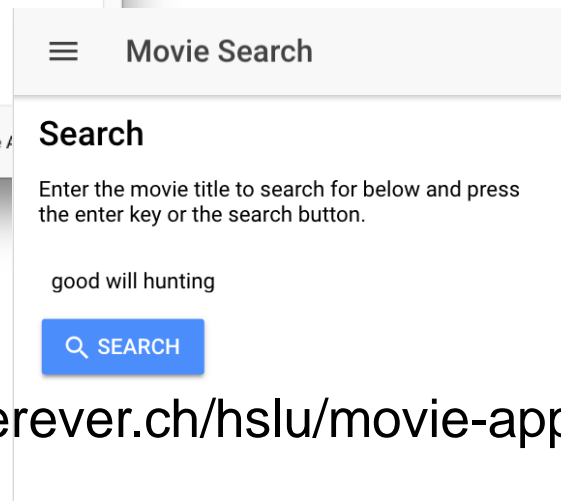
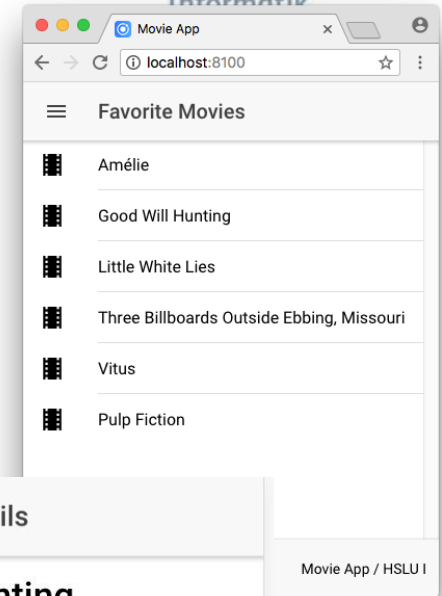
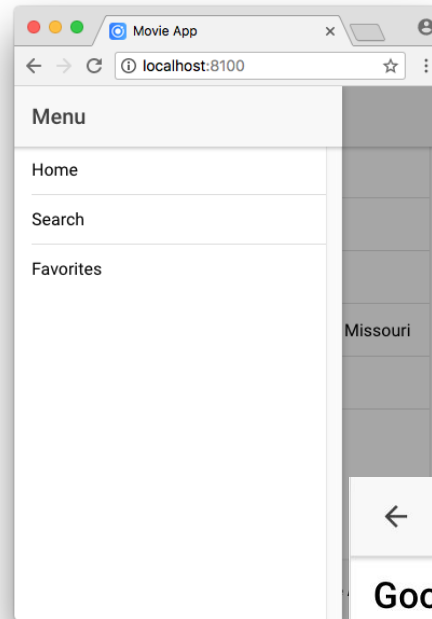
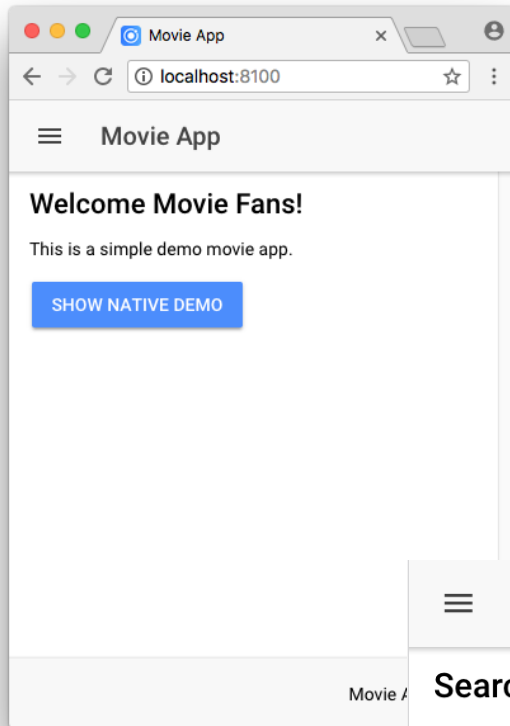
- Professionelle IDE für Web-Entwicklung
 - Freie Lizenz für HSLU-Studenten
 - <https://www.jetbrains.com/student/>
- Ionic-Projekt importieren
 - Menu: File – Open...
- npm run-targets benutzen!
 - z.B. ionic:serve



Alternativ: Rechts-Klick auf package.json > Show npm scripts

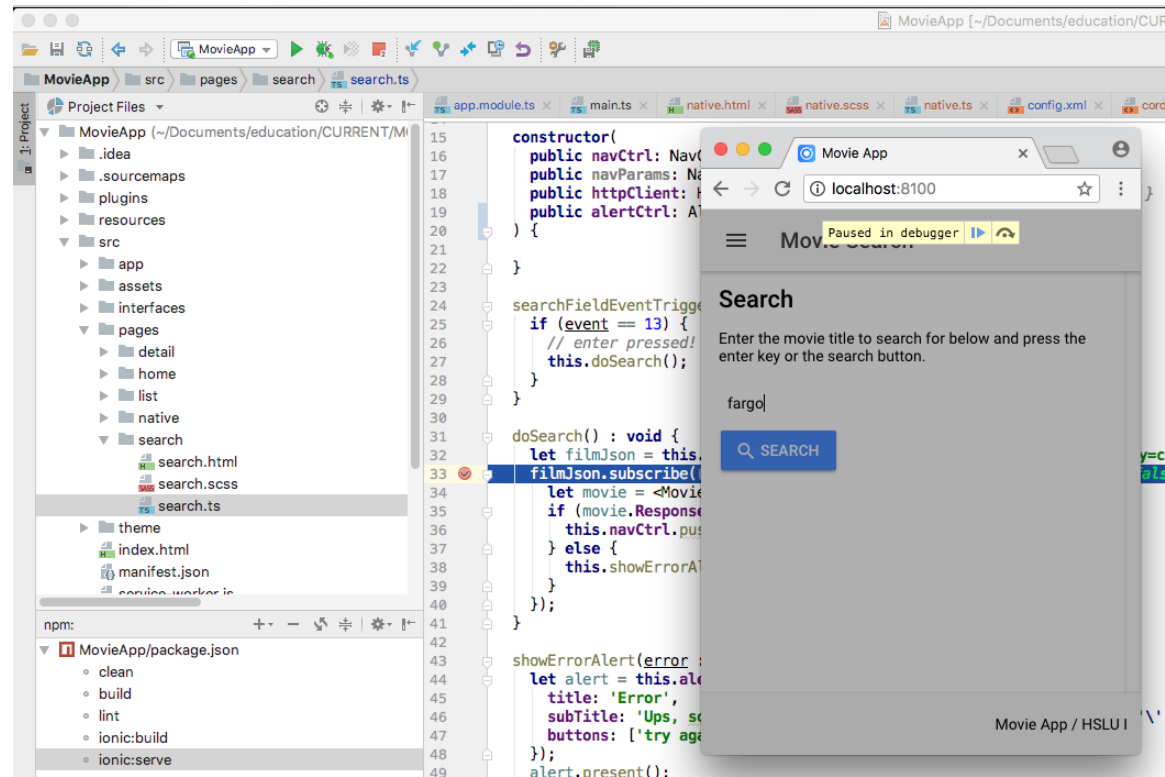
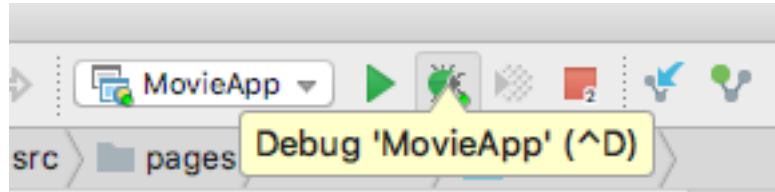


Demo MovieApp (Siehe Übung!)



Online: <http://wherever.ch/hslu/movie-app/>

Demo: Debugging WebStorm @ Chrome



Gemäss <https://blog.jetbrains.com/webstorm/2017/08/developing-ionic-apps-in-webstorm/>


Kurze Demo: Chrome Developer Tools

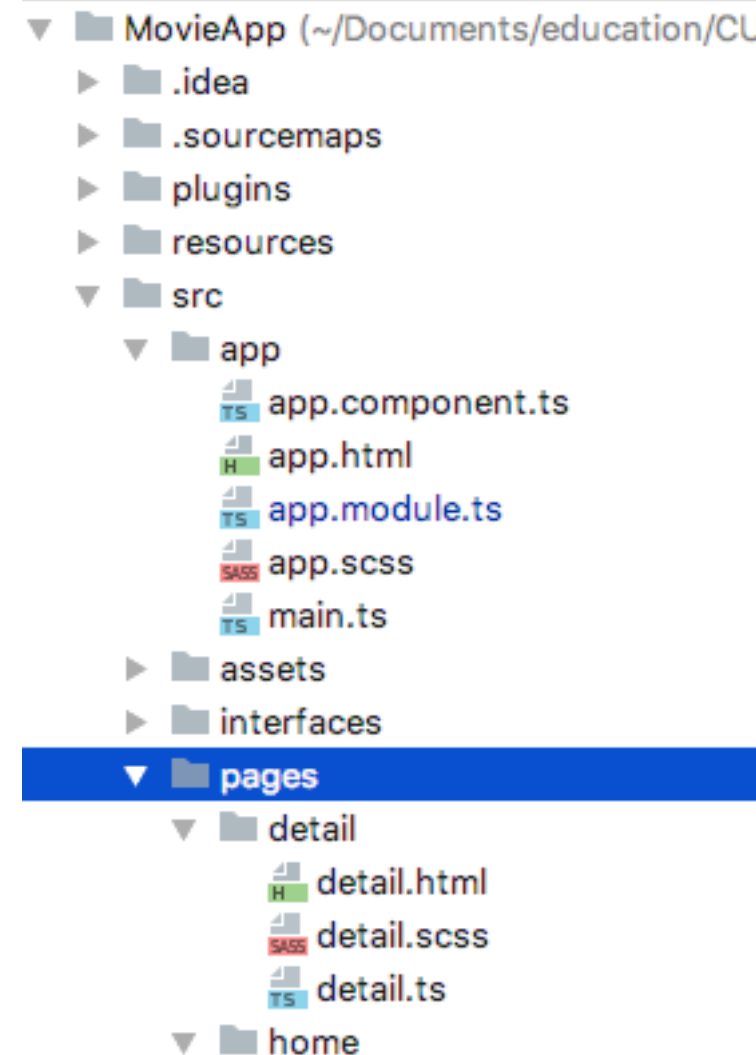
Windows: F12/Ctrl-Shift-I, Mac: Ctrl-Options-I

The screenshot displays a web browser window with the address bar at `localhost:8100`. The page shows a mobile app interface titled "Movie App" with a "Favorite Movies" list containing: Amélie, Good Will Hunting, Little White Lies, Vitus, Pulp Fiction, and Three Billboards Outside Ebbing, Missouri. The browser's developer tools are open on the right, showing the "Elements" panel with the HTML structure of the page, including the `<ion-app>` component. The "Styles" panel shows the default mobile view settings for the `body` element. The "Console" panel at the bottom indicates that Angular is running in development mode.

Ionic: Projektstruktur

- App-Seiten...

- sind *Angular*-Module  *TypeScript* in *TypeScript* (.ts) geschrieben
- liegen in src/app/pages
- müssen registriert werden in src/app/app.module.ts (!)



Angular ???

Kurz: Von Google entwickeltes Web-Framework mit Dependency Injection für komponenten-basierte Entwicklung von WebApps.

Achtung: AngularJS != Angular
(Erste, inkompatible Version)

TypeScript



Typescript ???

Kurz: Von Microsoft entwickelte JS-Dialekt mit statischem Typing, Klassen und Interfaces. Wird nach JS kompiliert (sieht im Browser also anders aus!)

1 Seite = .html + .scss + .ts

```

<ion-header>
  <ion-navbar>
    <button ion-button menuToggle>
      <ion-icon name="menu"></ion-icon>
    </button>
    <ion-title>Movie App</ion-title>
  </ion-navbar>
</ion-header>

<ion-content padding>
  <h3>Welcome Movie Fans!</h3>
  <p>This is a simple demo movie app.</p>
  <button ion-button secondary (click)="pushNativePage()">show native demo</button>
</ion-content>

<ion-footer>
  <ion-toolbar>
    <ion-label align="right">Movie App / HSLU I</ion-label>
  </ion-toolbar>
</ion-footer>

```

```

import { Component } from '@angular/core';
import { NavController } from 'ionic-angular';
import { NativePage } from '../native/native';

```

```

@Component({
  selector: 'page-home',
  templateUrl: 'home.html'
})

```

```

export class HomePage {

  public battery: string;

  constructor(
    public navCtrl: NavController
  ) {

  }

  pushNativePage() {
    this.navCtrl.push(NativePage);
  }
}

```

home

home.html
home.scss
home.ts

```

page-home {
}

```

Auf CSS (sprich Layout & Design von Webseiten) gehen wir hier nicht weiter ein. - Auch nicht auf SCSS und SASS. □

Ionic: Seitenaufbau (html)

- Dreiteilig: Header, Content & Footer

```
<ion-header>
```

```
...
```

```
</ion-header>
```

```
<ion-content>
```

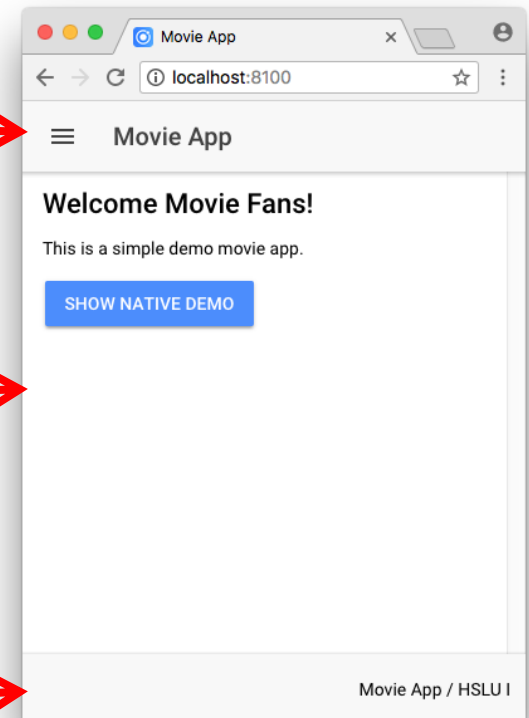
```
...
```

```
</ion-content>
```

```
<ion-footer>
```

```
...
```

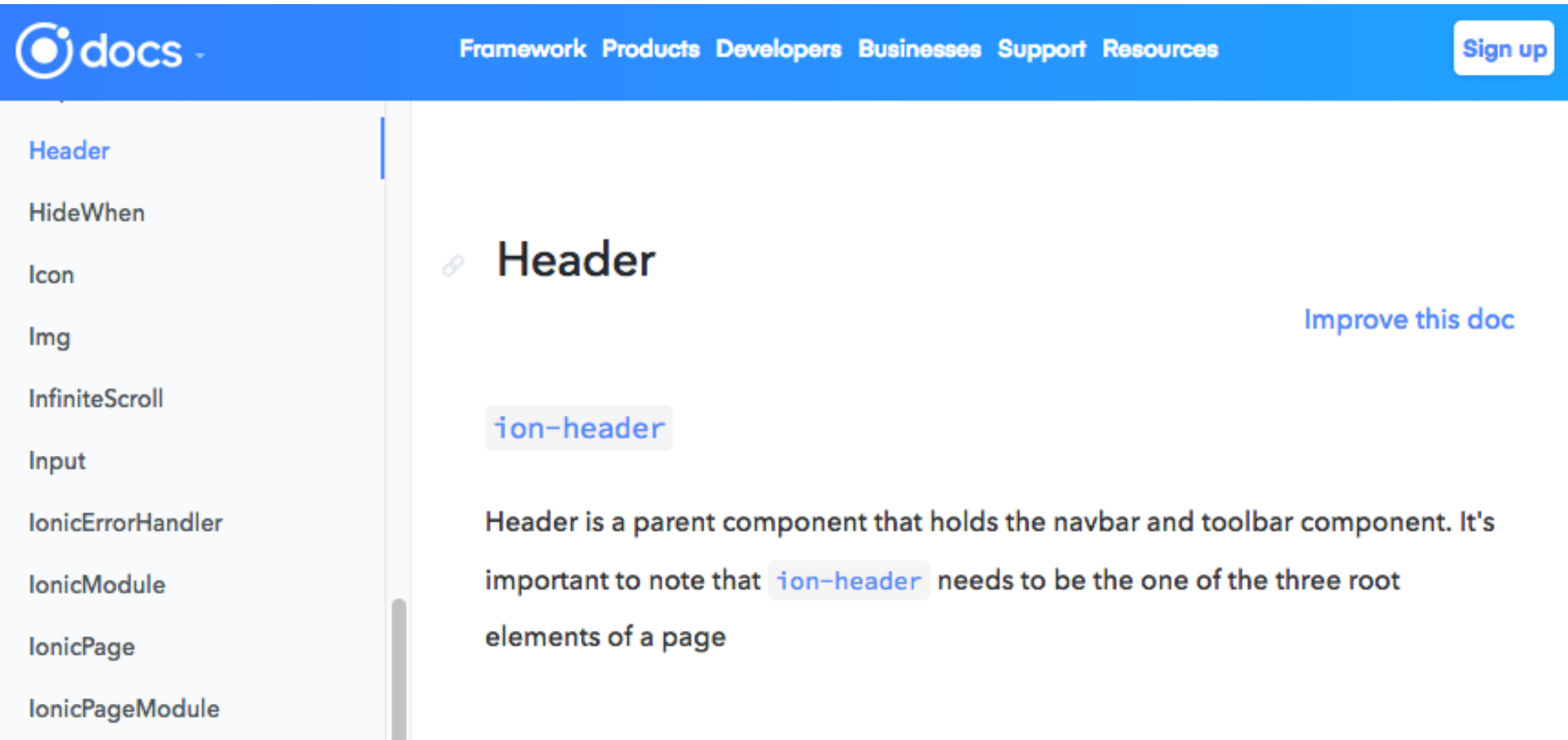
```
</ion-footer>
```



Doku-Beispiel: HTML-Komponente "ion-header"

Generell gute Doku inklusive konkreten Code-Beispielen!

<https://ionicframework.com/docs/api/components/toolbar/Header/>



The screenshot shows the Ionic Framework documentation website. The top navigation bar is blue with the 'docs' logo on the left and links for 'Framework', 'Products', 'Developers', 'Businesses', 'Support', and 'Resources' in the center. A 'Sign up' button is on the right. A left sidebar contains a list of components: 'Header' (highlighted with a blue bar), 'HideWhen', 'Icon', 'Img', 'InfiniteScroll', 'Input', 'IonicErrorHandler', 'IonicModule', 'IonicPage', and 'IonicPageModule'. The main content area has a title 'Header' with a pencil icon, a link 'Improve this doc', and a code example 'ion-header' in a light blue box. Below this, the text states: 'Header is a parent component that holds the navbar and toolbar component. It's important to note that ion-header needs to be the one of the three root elements of a page'.

docs - Framework Products Developers Businesses Support Resources Sign up

Header

HideWhen

Icon

Img

InfiniteScroll

Input

IonicErrorHandler

IonicModule

IonicPage

IonicPageModule

Header

Improve this doc

`ion-header`

Header is a parent component that holds the navbar and toolbar component. It's important to note that `ion-header` needs to be the one of the three root elements of a page



<http://en.wikipedia.org/wiki/File:Somethingdifferent.jpg>

Ionic: "How-Tos"

Zu diesen "How-Tos"

Es folgen konkrete Hinweise zur Lösung der Übung in Form von "How-Tos". D.h. Code-Beispiel-Sequenzen und Hinweise zu...

- http-get
- parse JSON in TypeScript
- data binding
- use html input field
- push another page
- use dynamics lists (aka *ngFor)



TypeScript



"How-To" http-get

```
import { HttpClient } from '@angular/common/http';  
...  
constructor( ... public httpClient: HttpClient ...) {  
...  
    let movieJson = this.httpClient.get('http://...');  
    movieJson.subscribe(data => {  
        // use data  
    })  
}
```

- get ist asynchron: Resultat ist ein *Observable*
- Mit *subscribe* setzen wir einen Callback (*data => {}*) , der ausgeführt wird, wenn die Daten da sind
- Details dazu sprengen hier den Rahmen!

Hinweis zur Verwendung von HttpClientModule

- Verwendung von `http-get` direkt in `Page.ts`, besser wäre Funktionalität auslagern in Service (Provider)
- Damit der Code von letzten Folie läuft, muss u.a. `HttpClientModule` in `src/app/app.module.ts`
 - import werden
 - in `@NgModule` unter `imports`: eingetragen sein

```
@NgModule({  
  declarations: [...],  
  imports: [  
    BrowserModule,  
    HttpClientModule,  
    IonicModule.forRoot(MyApp),  
  ],  
  bootstrap: [IonicApp],  
  entryComponents: [...],  
  providers: [...]  
})  
export class AppModule {}
```

"How-To" parse JSON in TypeScript

```
import { Movie } from '../..//interfaces/Movie'

...

let movieJson = this.httpClient.get('http://...');
movieJson.subscribe(data => {
  let movie: Movie = <Movie>data;
  if (movie.Response == 'True') {
    // use movie
  }
});
```

- Grundidee: JSON-Daten auf passendes TypeScript-Interface **mappen**

Im Interface nur die gewünschten JSON-Properties aufnehmen (die anderen einfach weglassen)

```
export interface Movie {
  Response: string;
  Title: string
  ...
}
```

Erinnere: Gson und Retrofit aus Android-4!

"How-To" do data binding

- page.html:

```
...  
<h1>{{title}}</h1>  
...
```

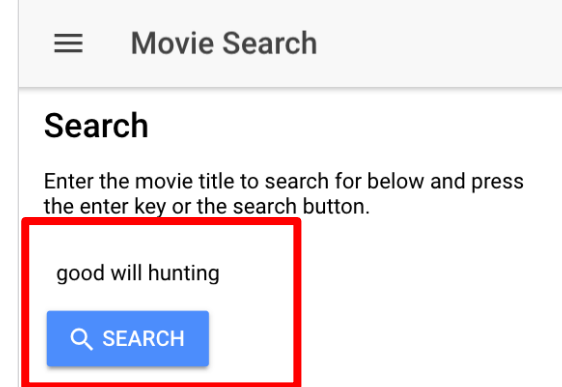
Auto-synchronisiert Wert
im HTML mit Property im
Controller: {{..}}-Notation

- page.ts:

```
...  
export class Page {  
  title: string;  
  ...  
  this.title = ...  
}
```

- Bemerkung: "Basis-Funktionalität" von Angular! 😊

"How-To" use HTML input field



Movie Search

Search

Enter the movie title to search for below and press the enter key or the search button.

good will hunting

SEARCH

■ page.html:

```
...  
<ion-input placeholder="[movie title]"  
  [(ngModel)]="searchQuery"  
  input type="text">  
</ion-input>  
<button ion-button (click)=doSearch()>Search</button>  
...
```

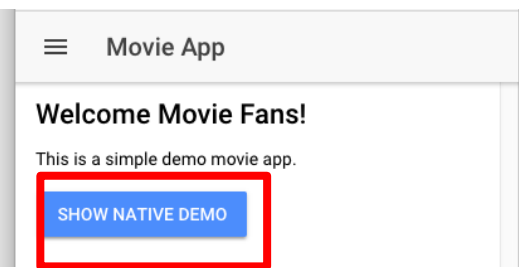
Databinding für Input-Felder (2-way)

■ page.ts:

```
export class Page {  
  searchQuery : string  
  ...  
  doSearch() : void {  
    // use this.searchQuery to access the value
```

- So kommen HTML-Formular-Daten in die dazugehörige Page-Klasse: "Angular-Magic" mit `[(ngModel)]`

"How-To" push another page



- page.html:

```
...  
<button ion-button secondary  
  (click)="pushNativePage()">  
  show native demo  
</button>  
...
```

- page.ts

```
import { NavController } from 'ionic-angular';  
import { NativePage } from '../native/native';  
...  
constructor(... public navCtrl: NavController ...) {  
...  
  pushNativePage() {  
    this.navCtrl.push(NativePage)
```

Eigene
Komponente

«push» = neue Seite
öffnen auf Page-Stack

Siehe <https://ionicframework.com/docs/api/navigation/NavController/>

"How-To" use dynamics lists (aka *ngFor)

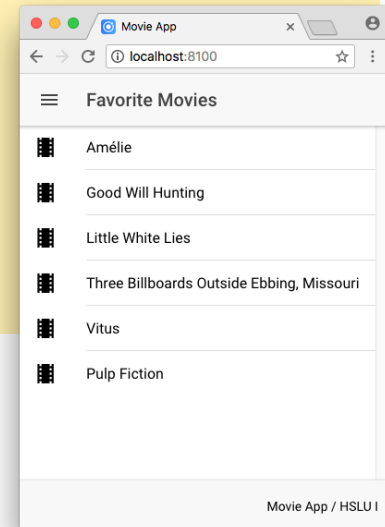
- page.html:

```
...  
<button ion-item *ngFor="let movie of movies"  
  (click)="movieSelected(movie)">  
  <ion-icon name="film" item-start></ion-icon>  
  {{movie.Title}}  
</button>  
...
```

Iteriere über List-
Property von Controller

- page.ts

```
import { Movie } from '../..//interfaces/Movie'  
...  
export class Page {  
  movies: Array<Movie>;  
  ...  
  this.movies = ...  
}
```



- Praktischer Angular-Mechanismus 😊

"How-To" call method on controller on event

- page.html:

```
...  
<button ion-item *ngFor="let movie of movies"  
  (click)="movieSelected(movie)">  
  <ion-icon name="film" item-start></ion-icon>  
  {{movie.Title}}  
</button>  
...
```

- page.ts

```
import { Movie } from '../..//interfaces/Movie'  
...  
export class Page {  
  ...  
  movieSelected(movie: Movie) {  
    this.navCtrl.push(DetailPage, {  
      movie: movie  
    });  
  }  
}
```

Optional: Daten
mitgeben bei Push für
Initialisierung Zielpage
(siehe Docs), hier: die
gewählte Seite

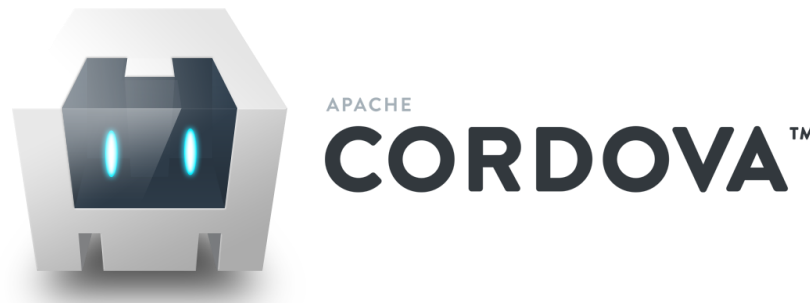


<http://en.wikipedia.org/wiki/File:Somethingdifferent.jpg>

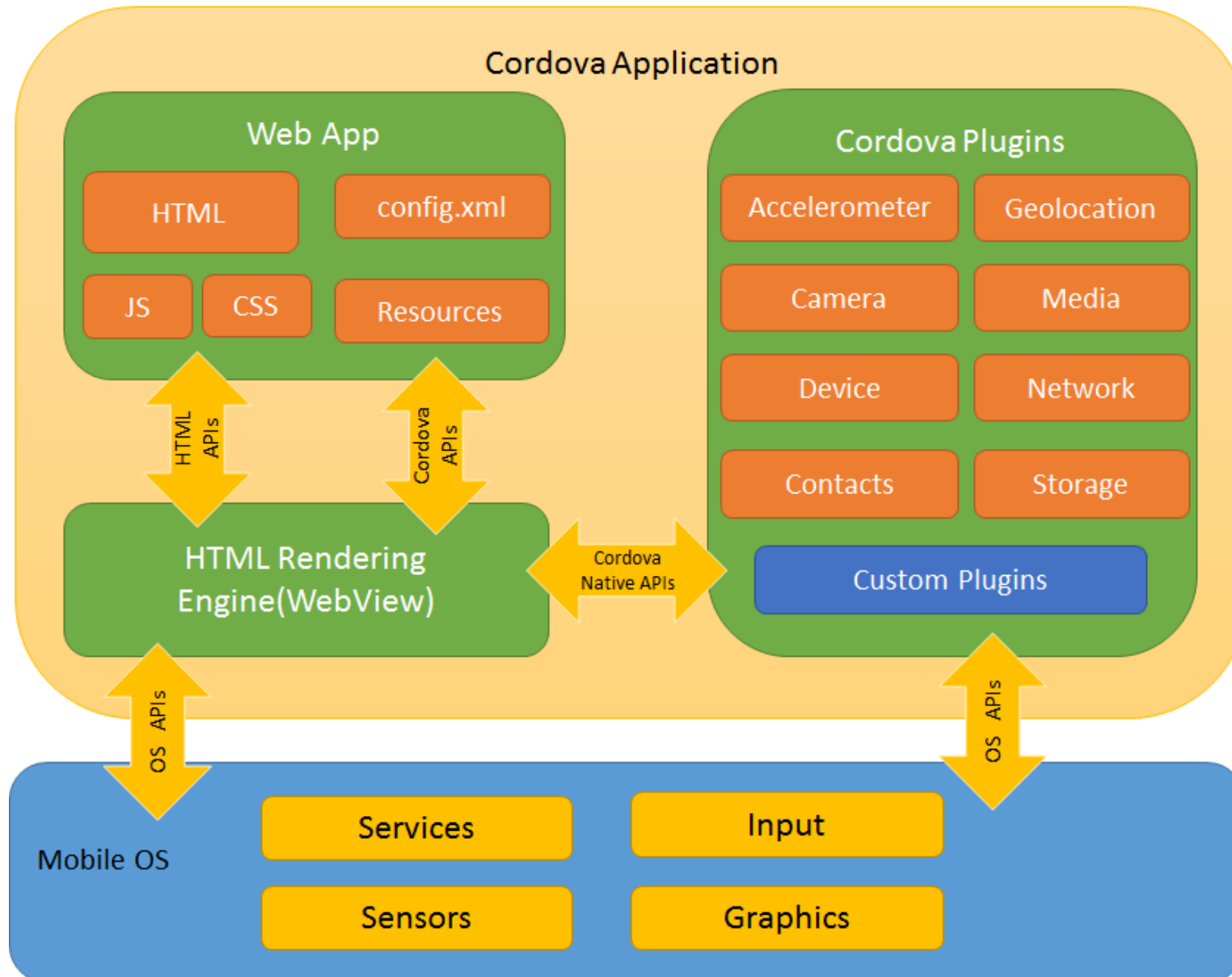
Hybride Web-Apps mit Cordova (= nativer Wrapper)

Cordova: Verpackt Web-App in native App

- Basiert auf HTML5, CSS3 und JavaScript und OS-nativem Code
- Unterstützt mittels Plugins Zugriff auf native APIs
 - Erlauben so z.B. Zugriff auf Ressourcen des Geräts



Übersicht Cordova-Architektur



Cordova vs. Phonegap

- Cordova hiess früher Phonegap
- Adobe kaufte die Firma hinter Phonegap
- Codebase wurde Apache überlassen, aber der Name Phonegap konnte nicht weiter benutzt werden
 - Neuer Name: Apache Cordova
- Phonegap ist nun eine Distribution von Cordova, die Phonegap-Distribution erlaubt eine einfachere Einbindung in Adobe Services
- Sowohl die Phonegap- wie die direkte Cordova Distribution ist freie Software (Apache Lizenz 2.0)

Ionic Native: Cordova@Ionic

- Cordova ist in Ionic integriert, kann direkt zusammen installiert werden
 - Erinnerung: Frage “Cordova Ja/Nein” bei **ionic start**,
 - Alle CLI-Kommandi von Ionic gibt's unter <https://ionicframework.com/docs/cli/commands.html>
- Alles mit Cordova:

cordova build	Build (prepare + compile) an Ionic project for a given platform
cordova compile	Compile native platform code
cordova emulate	Emulate an Ionic project on a simulator or emulator
cordova platform	Manage Cordova platform targets
cordova plugin	Manage Cordova plugins
cordova prepare	Copies assets to Cordova platforms, preparing them for native builds
cordova requirements	Checks and print out all the requirements for platforms
cordova resources	Automatically create icon and splash screen resources
cordova run	Run an Ionic project on a connected device

Ionic App unter Android / iOS starten

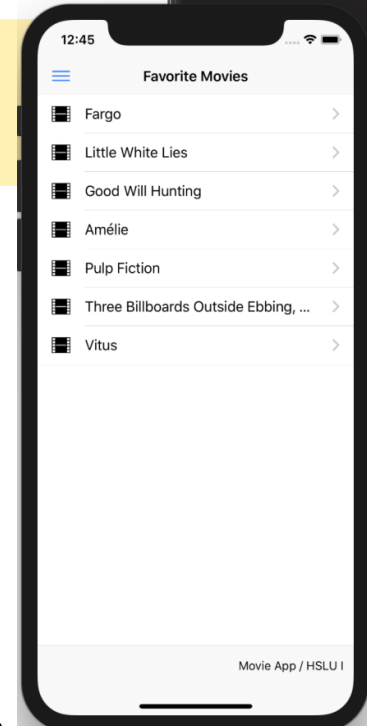
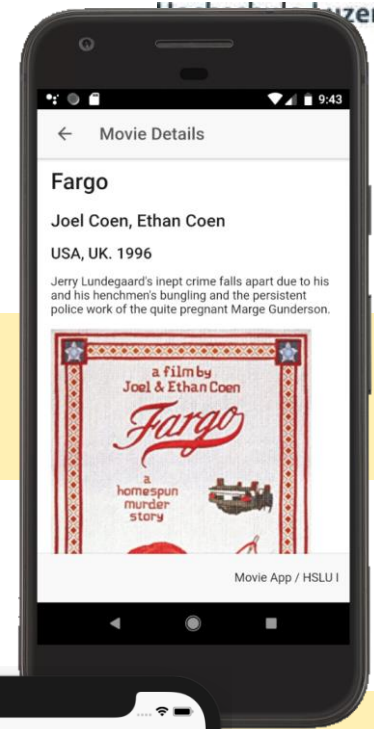
```
$ ionic cordova run android --emulator  
$ ionic cordova run android --device
```

Zuerst Android-Emulator starten! (z.B. aus Android Studio)

```
$ ionic cordova platform add ios  
$ cordova run ios --emulator
```

Direkt aus WebStorm starten, siehe

<https://blog.jetbrains.com/webstorm/2017/08/developing-ionic-apps-in-webstorm/#previewing-the-app>



<https://ionicframework.com/docs/cli/commands/cordova-run>

Cordova: Troubleshooting

- **JAVA_HOME & ANDROID_HOME**

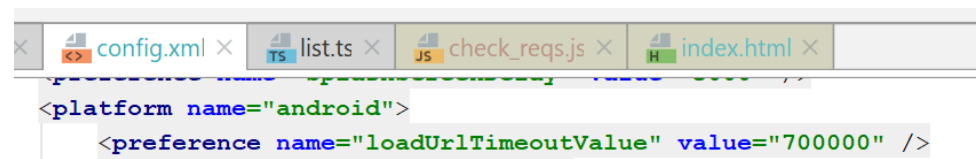
Falls eine entsprechende Fehlermeldung kommt, müssen ggf. die Umgebungsvariablen (Env.-Vars) für Java bzw. Android gesetzt werden:

```
$ JAVA_HOME=/path/to/java/SDK/; export JAVA_HOME
```

```
$ ANDROID_HOME=/path/to/android/SDK/; export ANDROID_HOME
```

Windows

- Bei “Requirements check failed for Java 1.8” (aber JAVA_HOME wird korrekt angezeigt): Prüfen ob ein neueres JDK auf dem PATH liegt und ggf (temporär) entfernen.
- Beim Starten der App Fehler im Emulator: “Application error. The connection to the server was unsuccessful, .../www/index.html” – Grund: Laaaaangsamer Emulator. Lösung: Erhöhe timeout in config.xml im Projekt:



Cordova Plugins

- Plugins bieten die Möglichkeit, mit Cordova auf native Möglichkeiten (APIs, HW, ...) der Plattform zuzugreifen
- Derartige Plugins gibt's zum Beispiel für:
 - Basic Device Info
 - Netzwerk und Batterie Info
 - Beschleunigungssensoren, Kompass, GPS
 - Kamera, Media Playback, Aufnahme
 - File IO, NFC, Bluetooth, ...
 - U.v.a.m...
- Ionic Native Plugin-Liste:
<https://ionicframework.com/docs/native/>

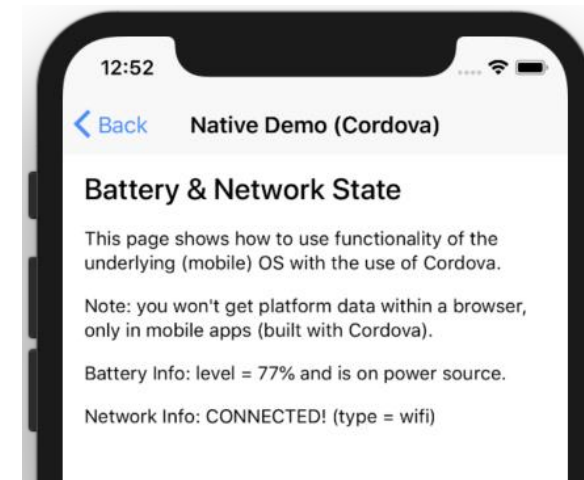
Selber ein Plugin schreiben ist auch möglich! Siehe Cordova-Dokumentation. 😊

Bsp. Plugin: Network

- Siehe Doku zum Plugin Network unter Ionic Native:
<https://ionicframework.com/docs/native/network/>
 - Inkl. Bsp.-Code für Installation & Gebrauch
 - Erlaubt in Ionic Code wie den folgenden:

```
let connSubscrip = this.network.onConnect().subscribe(() => {  
  this.networkInfo = 'CONNECTED!';  
});
```

- Siehe optionaler Teil der Übung!
 - Spielen Sie gerne damit herum 😊



Fazit: Brave New Web World!

- Auch mit Web-Technologien können mobile Applikationen entwickelt werden! 😊
- Browser Unterstützung wird zunehmen (HTML 5...)
- WebApps können dank Cordova als native mobile Apps verteilt werden (d.h. via Store)
- JavaScript-Welt ist dynamisch: Neue Versionen, Sprachen, Frameworks, usw.
- **Persönliches Fazit: Super wenn's läuft – falls nicht: viel Spass beim Debugging!..** 🤔
 - Viele Abhängigkeiten, lange Build-Chains, "Magic", ...

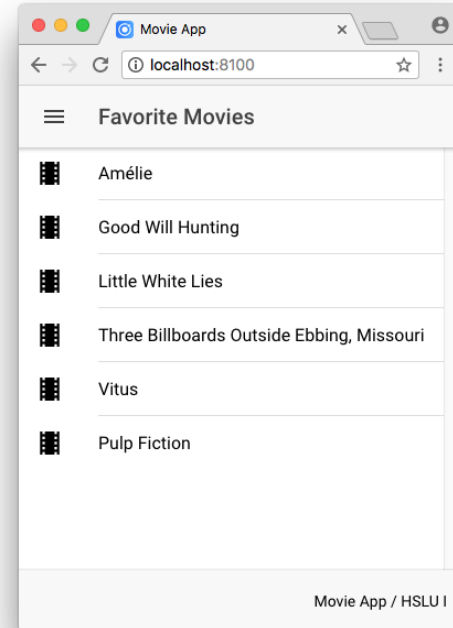


<http://en.wikipedia.org/wiki/File:Somethingdifferent.jpg>

Übung Web-App

Übung Web-App (MobPro: Pflicht!)

- Ionic-Projekt "sidemenu"
 - Seiten, Navigation, usw.
- Film-Suche inkl. Detail-Ansicht
 - HTTP-GET, JSON, "Data Binding", usw.
- Favoriten-Liste
 - Lokale Dateien lesen, *ngFor
- Cordova
 - Native App auf Android / iOS
- [Optional] Cordova Plugin benutzen
 - Batterie- und Netzwerk-Status



Korrektur Webapp-Übungen

- Nächstes mal bereits
ab 18:00 im Foyer
- Reihenfolge gemäss Registrierung
 - siehe aktuelle Liste rechts
 - wer da ist, kann bereits früher zeigen

	Webapp
M.Schuler/L.Brägger	OK
R.Bucher / T.Nguyen	OK
Zelger / Egli	OK
Furrer / Waldi	OK
Künzle/Niederberger	OK
T.Bolzern /P. Keusch	OK
Nikola Strässle & Melina Scodeller	OK
flavio / matej	OK
Jan Leuenberger / Nico Temperli	OK
Mathias / Oli	OK
Y.Luthiger / M. Palkoska	OK
Ha / Ehrler	OK
André, Michael	OK
Rico Scheller/Emanuel Foster	OK
Hauri / Felber	OK
Y. Dürkes / J. Grepper	OK
Thalmann / Sägesser	OK
Ineichen / Nebroj	OK
Gut / Kiser	OK
Zenhäusern / Abgottspon	OK
Althaus/Hasselmann	OK
Tuor / Thevathas	OK
Tran / Gassmann	OK
Wandfluh / Stadelmann	OK
Marty/Steinegger	OK
Rava / Betschart	OK
Kostanjsek / Rüdisüli	OK
F. Thalmann / R. Schraner	OK
O. Arnold / R. Heer	OK
Weber	OK
Vogel / Dörig	OK
Anzahl	31