Ionic 4 – Framework für „cross-platform apps“

Laufen nicht nur im Web sondern auch auf nativen Geräten in native Apps.

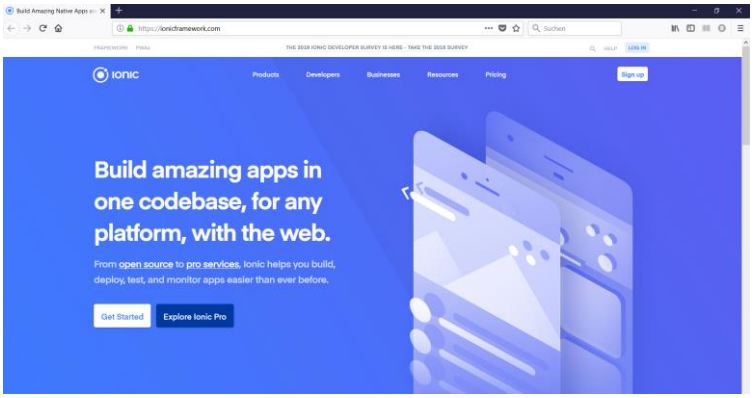
Normalerweise kann man z.B. kein iOS-App erstellen, ohne dabei ein Mac-System bei der Herstellung zu nutzen. Mit Ionic ist das kein Problem, da geht es auch Mac-System.

Ist Open Source. Ist ein Frontend-Framework zum Entwickeln von Hybriden mobilen Apps mit HTML5 und Angular für iOS- und Android-Plattformen. Somit kann man eine Anwendung auf verschiedenen Plattformen nutzen.

Nutzt eine Codebasis, nämlich HTML, JavaScript und CSS.

Benötigt „node.js“.

https://ionicframework.com



Hat eine sehr gute Dokumentation. Leicht zu finden durch Klick auf die Suche und dann wird es schon vorgeschlagen „docs“.

Ionic Native Plugins: bietet viele Plugins wie z.B. Barcodescanner, Kalender, Kamera, Taschenlampe.

Ionic Platform:

- Web Components – damit erzeugt man das, was man am Bildschirm sieht – z.B. <ion-app> oder <ion-button>. Diese Elemente haben bereits ein spezielles Layout, ohne dass man CSS schreiben muss.
- Diese Ionic Native Schnittstellen, die bis Ionic 3 mit „Cordova“ gleichzusetzen waren, nutzen nun die Ionic eigene Platform „Capacitor“, die Cordova ab Ionic 4 ersetzt bieten. Damit sind Zugänge zu den nativen Hardwarekomponenten, z.B. Taschenlampe möglich die mit JavaScript alleine ja nicht möglich wären. Mit „Capacitor“ wird somit ein App erzeugt, das wie ein nativ mobiles App läuft.

  Web app, running as a native mobile app
- Ionic CLI – (comand line interface) + Angular CLI dient für das Projekt Management und den Build Workflow, dient z.B. für den live-reload

Das Ionic-Framework selbst ist mit dem Tool „stencil“ erstellt worden.

https://www.youtube.com/watch?v=r2ga-iXS5i4

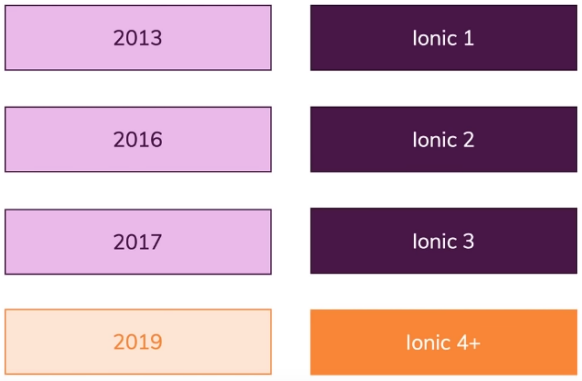
https://ionicframework.com/docs/native/

Ionic erzeugt „cross-kompilierte mobile Apps“

Die Grundidee ist es, native mobile Apps für die unterschiedlichen mobilen Betriebssysteme auf Grundlage EINER Quellcodebasis zu generieren. Dazu werden Cross-Compiler eingesetzt, die den Quellcode ohne weitere Zwischenschicht für unterschiedliche Betriebssysteme cross-kompilieren, sodass eine native mobile App entsteht. Entwicklungswerkzeuge dafür sind Xamarin (iOS), React Native und Apache Cordova mit Ionic.

Ionic und Angular ist eine perfekte Kombination. Man muss aber nicht Angular nutzen. Man kann auch Vanilla JS oder andere Frameworks verwenden (z.B. React), oder sogar ohne Framework arbeiten.

Entwicklung von Ionic:



Ionic 4 verwendet Angular, was einige Vorteile bringt.

- Routing und Navigation
- Einfachere Controller und Dependency Injections

Installation:

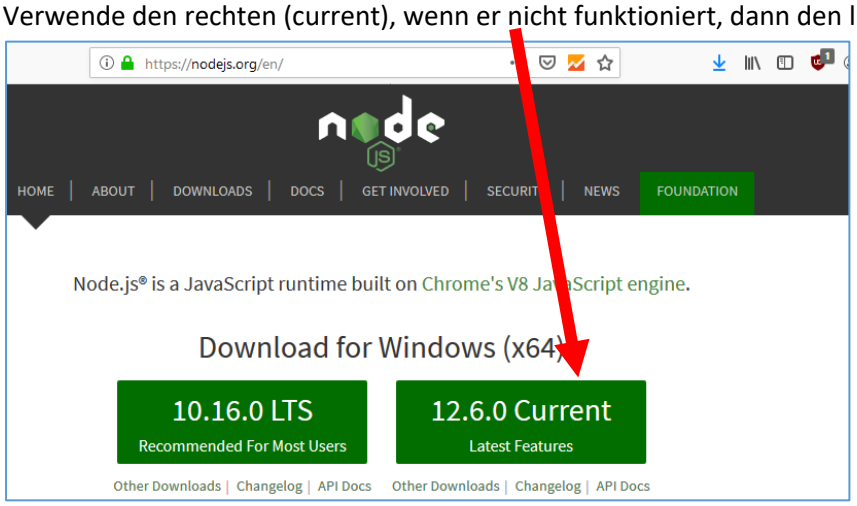
Nötige Programme zum Entwickeln:

Info: https://www.youtube.com/watch?v=gplRGpkOlUY

Node.js mit dem „Node Package Manager“ (NPM) zum Installieren und Nutzen von Ionic

- **http://nodejs.org**
  Wir benötigen node.js aus 2 Gründen:
  1. zur Verwendung des Package-Managers – zum Managen der dependencies
  2. zur korrekten Ausführung von Ionic, welches mit node.js läuft

  Verwende den rechten (current), wenn er nicht funktioniert, dann den linken (LTS).



Eingabeaufforderung öffnen:

Nach der Installation von Node.js öffne die Eingabeaufforderung mit der Suche von Windows und der Eingabe von „Eingabeaufforderung“ oder „cmd“.

Es soll die aktuelle Version von „node“ angezeigt werden.

Daher gib ein

- node –v

```
C:\Users\edi>node -v
v12.6.0
```

Ergebnis: Version 12.6.0

Somit ist auch der Package-Manager installiert worden. Dessen Version sieht man mit

- npm –v

```
C:\Users\edi>npm -v
6.9.0
```

1a)Ionic installieren

Info siehe auch auf https://ionicframework.com/docs/ und dann „Intro und Installation“.

Info: Ionic Apps werden durch die „ionic command line utility (CLI)“ entwickelt, die Capacitor (Cordova) verwendet, um eine native App zu erstellen („build and deploy“).

Ebenfalls in der Eingabeaufforderung. Zum Installieren eines Packages gib ein: npm install und dann den Namen nämlich

- npm install -g ionic

```
C:\Users\edi>npm install -g ionic
```

Das –g erzeugt eine globale Installation.

Kann 1-2 Minuten dauern.

Gib danach nur ein

- ionic

und es öffnen sich die Befehle von der ionic CLI

```
Eingabeaufforderung
 (_) ___   _ __ (_) ___
 | |/ _ \| '_ \| |/ __|
 | | (_) | | | | | (__
 |_|\___/|_| |_|_|\___| CLI 5.2.1


Usage:

  $ ionic <command> [<args>] [--help] [--verbose] [--quiet] [--no-interactive] [--no-color] [--confirm] [options]

Global Commands:

  completion ....................... (experimental) Enables tab-completion for Ionic CLI commands.
  config <subcommand> .............. Manage CLI and project config values (subcommands: get, set, unset)
  docs ............................. Open the Ionic documentation website
  info ............................. Print project, system, and environment information
  init ............................. (beta) Initialize existing projects with Ionic
  login ............................ Log in to Ionic
  logout ........................... Log out of Ionic
  signup ........................... Create an Ionic account
  ssh <subcommand> ................. Commands for configuring SSH keys (subcommands: add, delete, generate, list,
                                     setup, use)
  start ............................ Create a new project

Project Commands:

  You are not in a project directory.
```

Zur Kontrolle der Version gib ein

ionic –v

```
C:\Users\edi>ionic -v
5.2.1
```

1b)Angular installieren

Dort gib den Code ein, um die Angular zu installieren:

npm install –g @angular/CLI

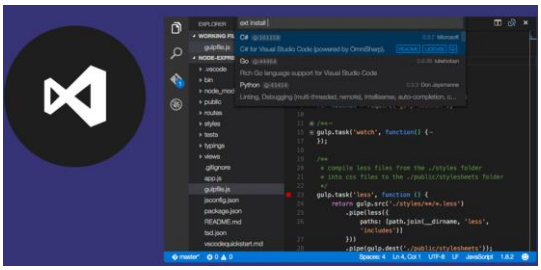
Dann gib ein den Buchstaben S „n“

```
C:\Users\edi>npm install -g @angular/cli
C:\Users\edi\AppData\Roaming\npm\ng -> C:\Users\edi\AppData\Roaming\npm\node_modules\@angular\cli\bin\ng

> @angular/cli@8.1.0 postinstall C:\Users\edi\AppData\Roaming\npm\node_modules\@angular\cli
> node ./bin/postinstall/script.js

? Would you like to share anonymous usage data with the Angular Team at Google under
Google's Privacy Policy at https://policies.google.com/privacy? For more details and
how to change this setting, see http://angular.io/analytics. (y/N) n
```

2)mit einer IDE arbeiten

Auf jeden Fall sinnvoll ist dafür Visual Studio Code

Am Beginn habe ich hier noch mit „brackets.io“ weitergearbeitet, aber sehr bald auch umgestellt und daher die IDE Visual Studio Code installiert.



3)IDE installieren Beispiel Microsoft Visual Studio Code

Der Visual Studio Code ist sehr gut für Ionic-Entwicklungen https://code.visualstudio.com/

(Download für Windows). Man kann dort auf einige Erweiterungen zurückgreifen.

Was kann „Visualstudio“?

Visual Studio Code is a lightweight but powerful source code editor which runs on your desktop and is available for Windows, macOS and Linux. It comes with built-in support for JavaScript, TypeScript and Node.js and has a rich ecosystem of extensions for other languages (such as C++, C#, Java, Python, PHP, Go) and runtimes (such as .NET and Unity).

Was ist eine IDE?

Die integrierte Entwicklungsumgebung, abgekürzt IDE (Integrated Development Environment), sammelt unter einer gemeinsamen Oberfläche die wichtigsten Tools für das Erstellen von Software. Da IDEs dem Programmierer bzw. Software-Entwickler viel Komfort bei der Arbeit bieten und wichtige Arbeitsschritte zusammenfassen, sind sie in vielen Bereichen zu unentbehrlichen Hilfsmitteln geworden. Sie nehmen dem Entwickler häufig wiederkehrende Arbeiten ab und

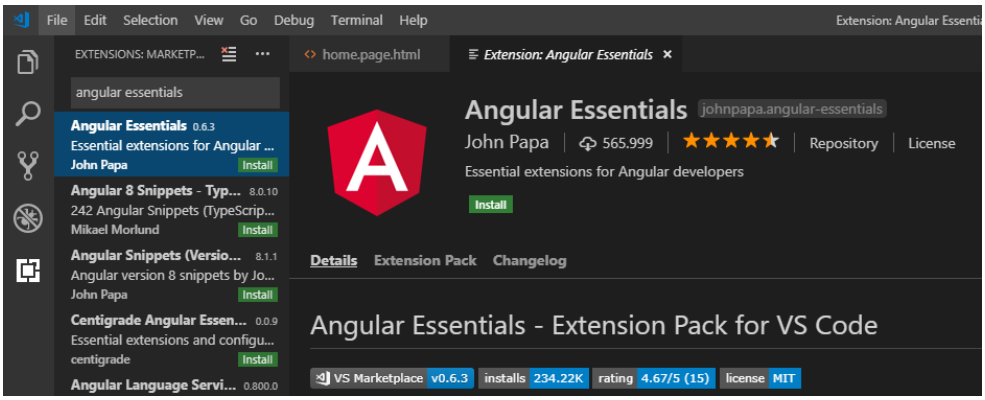
entlasten ihn von formalen Tätigkeiten wie die Verwaltung von Arbeitsergebnissen oder Programmversionen. Dem Softwareentwickler fällt es dank der IDEs wesentlich leichter, sich auf seine eigentliche Arbeit, das Programmieren, zu konzentrieren.

Die wichtigsten Bestandteile einer integrierten Entwicklungsumgebung sind:

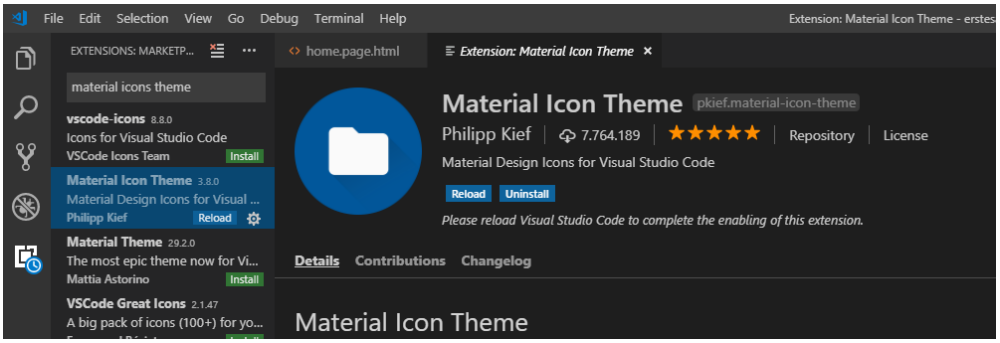
Editor mit Quelltextformatierung und Syntaxhervorhebung, Compiler, Interpreter, Linker, Debugger, Werkzeuge zur Erstellung grafischer Benutzeroberflächen (GUI-Builder), Versionsverwaltungen.

Man sollte auch unter „view“ die Extensions wählen und z.B. folgendes installieren:

- angular essentials



- material icons theme – für die schöner Anzeige in VS-Code selbst



1)Erstes Projekt erstellen:

In der Eingabeaufforderung kann nun einfach das erste Projekt erstellt werden.

```
Eingabeaufforderung
 (_) ___  _ __ (_) ___
 | |/ _ \| '_ \| |/ __|
 | | (_) | | | | | (__
 |_|\___/|_| |_|_|\___| CLI 5.2.1


 Usage:

   $ ionic <command> [<args>] [--help] [--verbose] [--quiet] [--no-interactive] [--no-color] [--confirm] [options]

 Global Commands:

   completion ...................... (experimental) Enables tab-completion for Ionic CLI commands.
   config <subcommand> ............. Manage CLI and project config values (subcommands: get, set, unset)
   docs ............................ Open the Ionic documentation website
   info ............................ Print project, system, and environment information
   init ............................ (beta) Initialize existing projects with Ionic
   login ........................... Log in to Ionic
   logout .......................... Log out of Ionic
   signup .......................... Create an Ionic account
   ssh <subcommand> ................ Commands for configuring SSH keys (subcommands: add, delete, generate, list,
                                     setup, use)
   start ........................... Create a new project

 Project Commands:

   You are not in a project directory.
```

Die Eingabe „ionic“ zeigt das obige Bild. Hier wird in der letzte Zeile angezeigt, dass kein Projekt-Ordner vorhanden ist, in den das neue Projekt abgelegt werden soll.

Befehle sind z.B.:

- dir zeigt den Inhalt eines Ordners an
- mkdir erstellte einen neuen Ordner
- cd wechselt in einen Ordner, z.B. cd desktop (change directory)
- cd .. geht einen Pfad zurück

Lass dir den Desktop anzeigen:

- dir desktop

ionicfErgebnis:

```
 Verzeichnis von C:\Users\edi\desktop

12.06.2019  16:16    <DIR>          .
12.06.2019  16:16    <DIR>          ..
25.04.2019  14:06             1 360 Adobe XD.lnk
11.10.2018  11:46             1 150 CdCoverCreator.lnk
20.02.2019  11:40             2 454 GitHub Desktop.lnk
12.06.2019  08:47             2 395 Microsoft Teams.lnk
25.12.2018  23:54             2 201 Postman.lnk
29.08.2018  07:18             1 396 Visual Studio Code.lnk
               6 Datei(en),         10 956 Bytes
               2 Verzeichnis(se), 73 835 249 664 Bytes frei

C:\Users\edi>
```

Nun soll ein neuer Ordner am Desktop erstellt werden, in dem später das Projekt deponiert werden wird.

- cd desktop

```
C:\Users\edi>cd desktop

C:\Users\edi\Desktop>
```

Dann befindet man sich in dem Ordner „desktop“. Nun soll hier ein neuer Ordner namens „ionic“ angelegt werden:

- mkdir ionic

```
C:\Users\edi\Desktop>mkdir ionic

C:\Users\edi\Desktop>
```

Nun muss man auch in diesen Ordner hineinwechseln mit dem Befehl:

- cd ionic

```
C:\Users\edi\Desktop>cd ionic

C:\Users\edi\Desktop\ionic>
```

Ist man, wie oben sichtbar, im Ordner drinnen (ionic), kann man hier das Projekt neu anlegen.

<u>Um aufzulisten, welche Möglichkeiten</u> gewählt werden können, gib ein

- ionic start --list

```
C:\Users\edi\Desktop\ionic>ionic start --list

Starters for @ionic/angular (--type=angular)

name         | description
-------------------------------------------------------------------------------
tabs         | A starting project with a simple tabbed interface
sidemenu     | A starting project with a side menu with navigation in the content area
blank        | A blank starter project
my-first-app | An example application that builds a camera with gallery
conference   | A kitchen-sink application that shows off all Ionic has to offer
```

Damit werden alle Möglichkeiten angezeigt, auch die schon älteren Versionen wie Ionic 2/3 usw. Bleibe ganz oben bei @ionic/angular.

<u>Erstes Projekt anlegen:</u>

- ionic start

Es wird automatisch nach einem Namen gefragt

```
C:\Users\edi\Desktop\ionic>ionic start

Every great app needs a name!

Please enter the full name of your app. You can change this at any time.
the first argument to ionic start.

? Project name: erstesapp
```

Wähle als Namen z.B. erstesapp (zusammengeschrieben)

Ergebnis:

```
Let's pick the perfect starter template!

Starter templates are ready-to-go Ionic apps that come packed with everything you need to build your app. To bypa
prompt next time, supply template, the second argument to ionic start.

? Starter template:
  tabs         | A starting project with a simple tabbed interface
  sidemenu     | A starting project with a side menu with navigation in the content area
  blank        | A blank starter project
  my-first-app | An example application that builds a camera with gallery
  conference   | A kitchen-sink application that shows off all Ionic has to offer
```

Dann werden 5 verschiedene Modelle für ein neues Projekt angeboten.

<u>Wir wählen „blank“.</u> Man muss es mit der Pfeiltaste auswählen, dass der Pfeil vor „blank“ steht. Die Farbe zeigt den Standort an.

Dann „enter“ und es dauert etwas und dann diese Infos:

```
[INFO] Next Steps:

       - Go to your newly created project: cd .\erstesapp
       - Run ionic serve within the app directory to see your app
       - Build features and components: https://ion.link/scaffolding-docs
       - Get Ionic DevApp for easy device testing: https://ion.link/devapp

C:\Users\edi\Desktop\ionic>
```

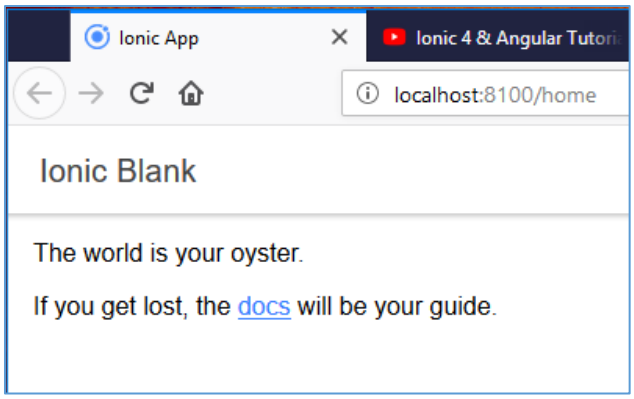
Gehe in das neue Projekt hinein:

```
C:\Users\edi\Desktop\ionic>cd .\erstesapp

C:\Users\edi\Desktop\ionic\erstesapp>
```

Danach gib ein, um den Server zu starten und das Projekt anzuzeigen:

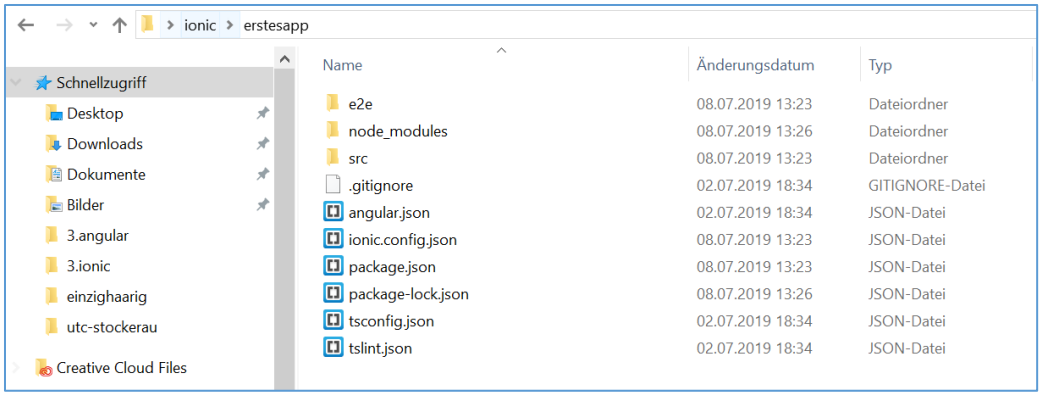
- ionic serve

Ergebnis im Browser:



Ordner am Desktop:

Jetzt, da das App automatisch erstellt wurde kann man es einmal im Windows-Explorer ansehen.



2)Arbeiten in brackets.io

Öffne brackets.io und dann im Menü Datei / Ordner öffnen

Öffne damit den Ordner „ersteapp“ im Ionic-Ordner auf dem Desktop.

- Damit das Ergebnis in einem Browser (am liebsten Google Chrome) auch laufen kann, muss man in der Eingabeaufforderung sich ebenfalls zum Ordner „erstesapp“ hinbewegen. Das erfolgt schrittweise jeweils mit „cd“, z.B. cd desktop, dann befindet man sich darin und gib ein „cd ionic“ usw.
  Ergebnis:

```
C:\Users\joe\Desktop\ionic\erstesapp>
```

- Dann den Befehl „ionic serve“

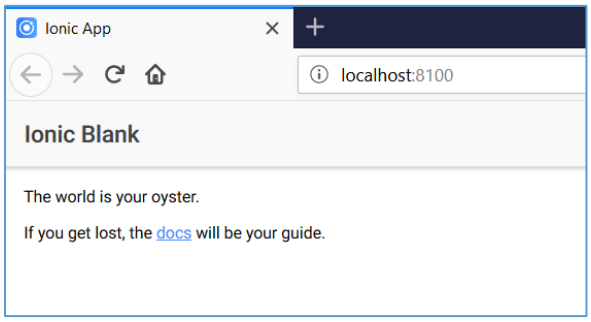
```
C:\Users\joe\Desktop\ionic\erstesapp>ionic serve
```

Mit dem Befehl „**ionic serve**“ startet man den Webserver. Der zeigt dann sofort bei Änderungen im Code auch die Auswirkung im App an.

Dann warten für den Start des Prozesses:

```
C:\Users\joe\Desktop\ionic\erstesapp>ionic serve
Starting app-scripts server: --address 0.0.0.0 --port 8100 --livereload-port 35729 -

Ctrl+C to cancel
[09:22:55]  watch started ...
[09:22:55]  build dev started ...
[09:22:55]  clean started ...
[09:22:55]  clean finished in 10 ms
[09:22:55]  copy started ...
```

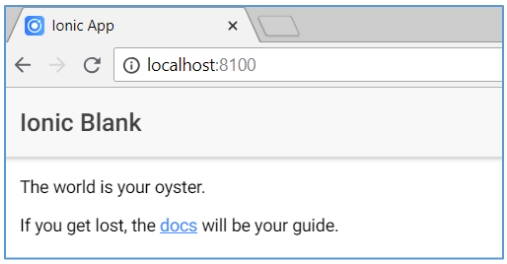
Ergebnis im Browser, der sich automatisch öffnet



Da Google Chrome eine sehr gute Entwicklerumgebung bietet, werden wir dort arbeiten. Daher schließe den Browser und öffne Chrome. Gib dort ein:

- localhost:8100

Ergebnis: wenn nicht erreichbar, muss man nochmals „ionic serve“ in CMD eingeben und neben dem alternativen Browser den Chrome öffnen und „localhost:8100“ eingeben. Den alternativen Browser (z.B. Firefox) schließen.



3)Arbeiten in Visual Studio Code:

Am Beginn muss der Ordner, der für das Projekt in der Eingabeaufforderung erstellt wurde in den linken Übersichtsbereich geholt werden. Dann kann man bequem darauf immer zugreifen.

Ansonsten erfolgt das gleiche Vorgehen wie oben in „brackets.io“.

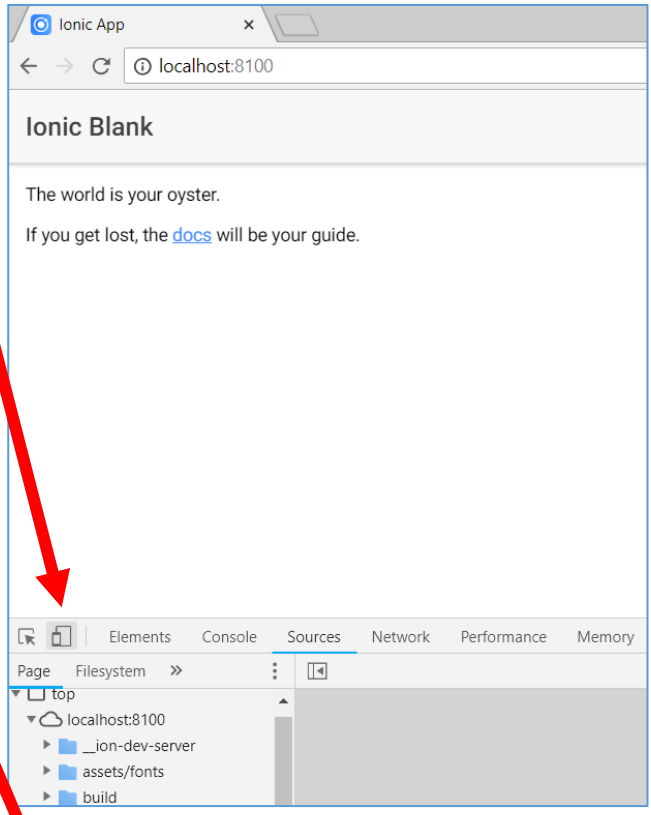
Aber zusätzlich gibt es ein nettes Feature:

Unter dem Menüpunkt „Anzeigen“ wähle „Integriertes Terminal“ (bzw. STRG + ö). Damit kommt man direkt in die CMD-Eingabeaufforderung, die sich ganz unten öffnet. Mit dem Befehl „**ionic serve**“ startet man den Webserver. Der zeigt dann sofort bei Änderungen im Code auch die Auswirkung im App an.

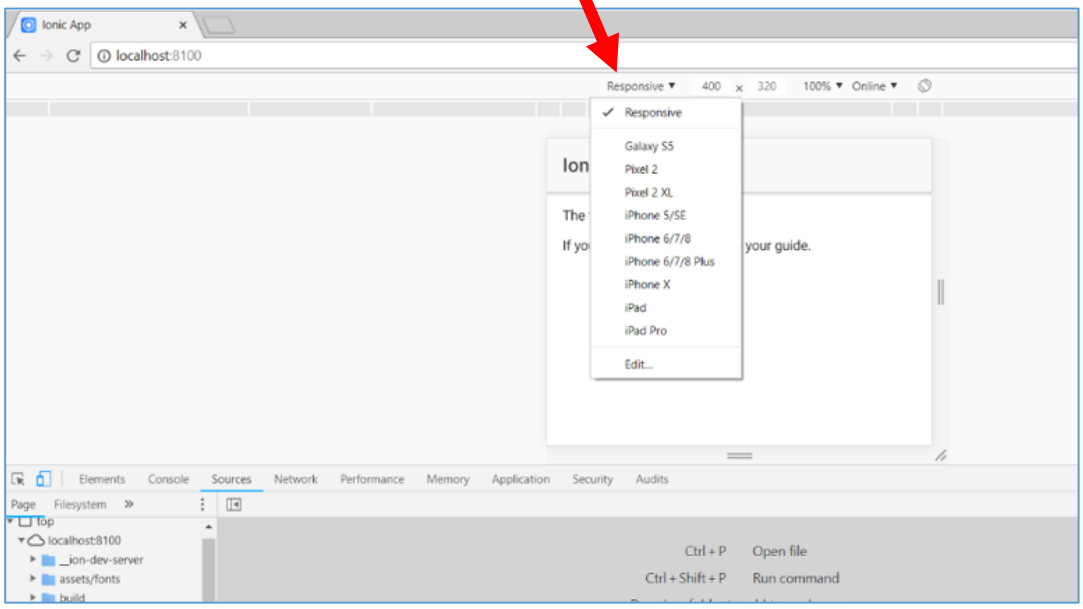
4)Im Chrome die Entwicklerkonsole aufrufen:

Klickt man im Chrome auf F12 ruft man die Entwicklerkonsole auf.

Dann dort links auf die Schaltfläche mit dem Handysymbol. Es können danach unter verschiedenen Sorten von Smartphones die Anzeigen betrachtet werden.



Und somit stehen diese Geräte zur Verfügung durch Klick auf das Dreieck



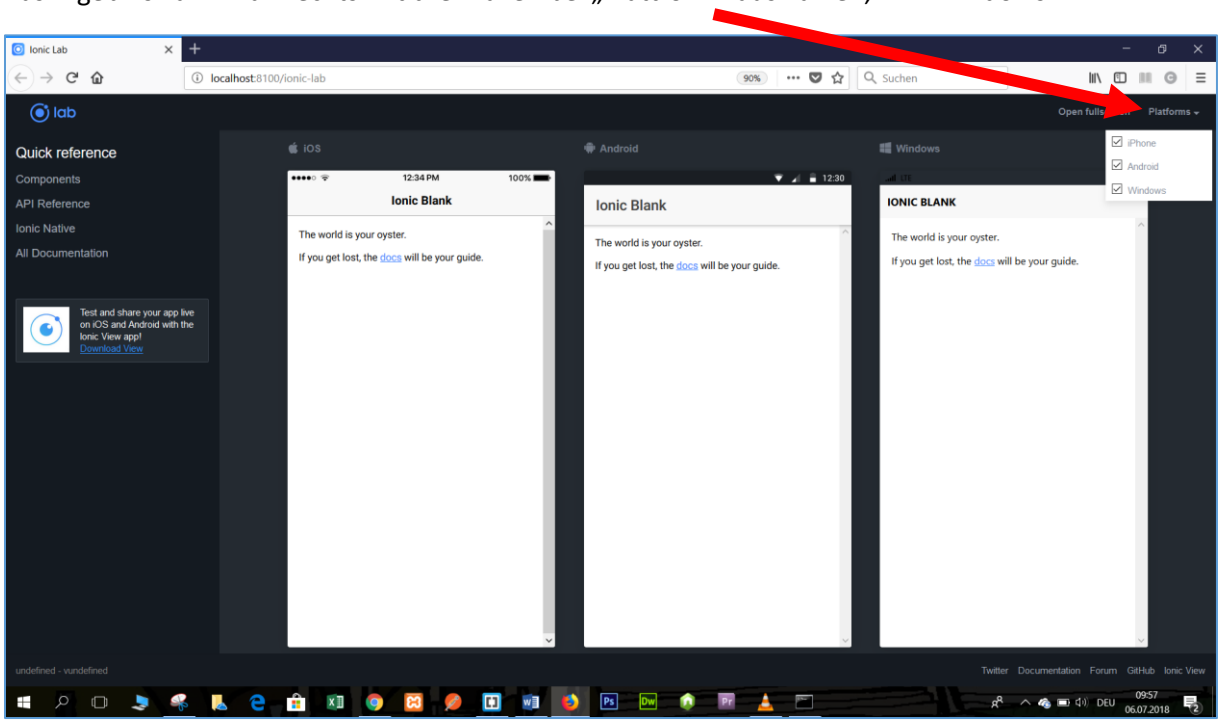
5)Die App übersichtlich in verschiedenen Plattformen (iOS, Android, Windows) ansehen

In der Eingabeaufforderung gib den Befehl dazu ein

- ionic lab

```
C:\Users\joe\Desktop\ionic\erstesapp>ionic lab
```

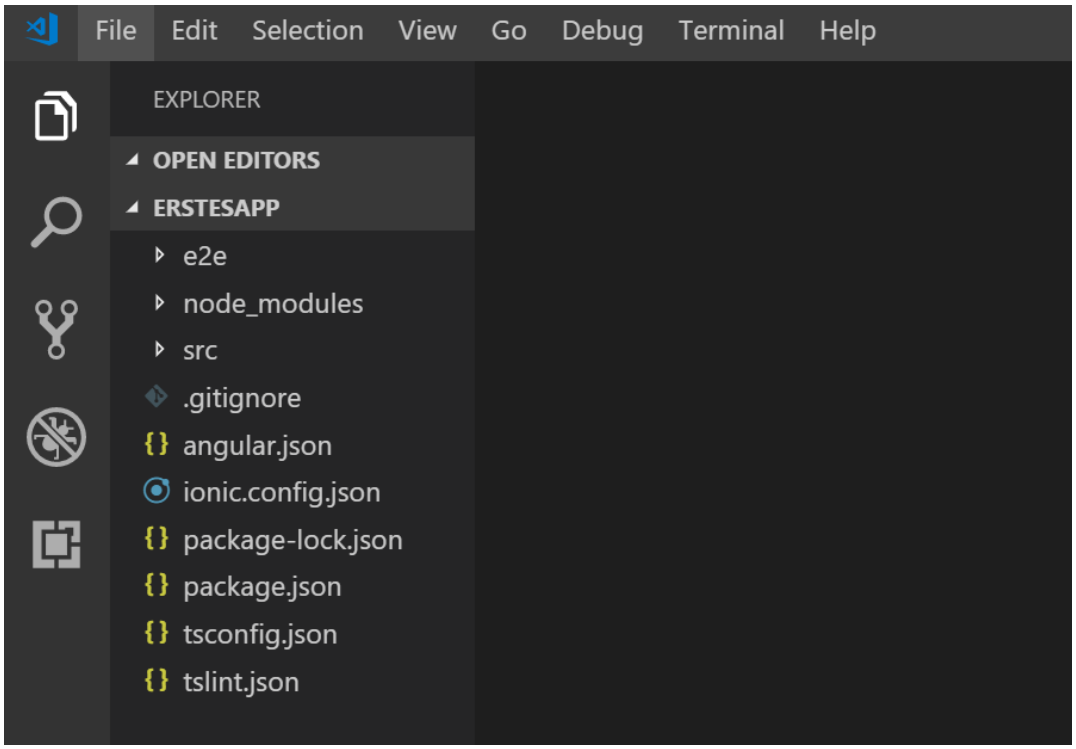
Das Ergebnis kann man rechts mit drei Haken bei „Plattform“ auswählen, inkl. Windows.



Somit sieht man das Aussehen der App, wenn sie am Ende kompiliert wird.

6)Aufbau der App

Starte VS-Code und öffne den Ordner auf dem Desktop „ionic / erstesapp“:



- packager.json für die „dependences“ – da Angular im Hintergrund arbeitet
- Im „src“-Ordner ist die App drinnen. Dabei wird mit TypeScript entwickelt. Am Ende wird es compilled. Das ist somit auf JavaScript-Basis und kann auf einem Webserver geladen werden und funktioniert als Website.
- In src / app / assets kommen die Bilder hinein
- In src / app / pages sind die Unter-Sites zu finden, z.B. home.html und spätere neue Seiten wie z.B. „ueber_uns.html“ usw.