Ionic App erstellen - Start

Inhalt:

1) Erstes Projekt erstellen: inkl. CAPACITOR
2) grundlegender Aufbau der App
3) Seiten anzeigen lassen
4) Erste Änderung – Text ändern
5) Neue Seite erstellen
6) Navigation auf den „loader“ als Startseite umstellen

Version 6 ist neu seit ca. Dez. 2021

- Neu: datetime, accordion, breadcrums, select-options, modal neu,
- Migration mit einer Zeile: npm install @ionic/angular@6

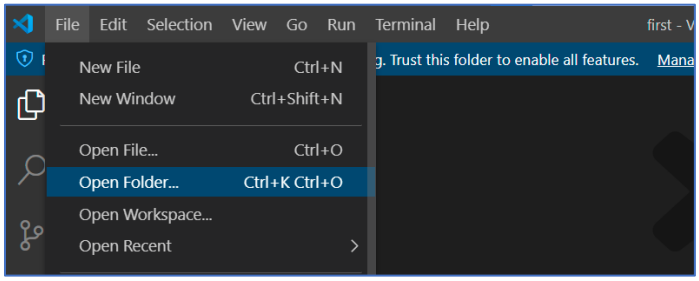
Version 7 ist neu seit ca. August 2023

1)Erstes Projekt erstellen: inkl. CAPACITOR

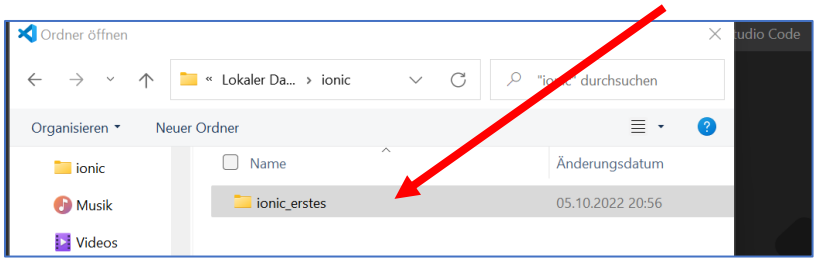
Erstelle auf dem Laufwerk „D:“ einen neuen Ordner namens „ionic_erstes“.

Bitte nicht in einem Unter- Unterordner irgendwo in deinem IM-Verzeichnis. Das kann zu Problemen beim Installieren führen!

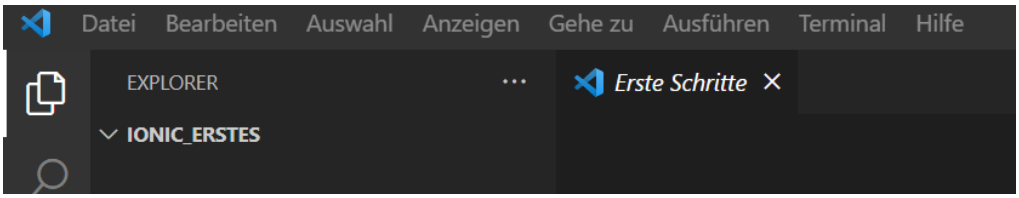
Dann öffne das eben installierte Programm „visual studio code“ und gehe mittels Menüs „File“ und „Open Folder…“ genau zu dem eben erstellten Ordner „ionic_start“.



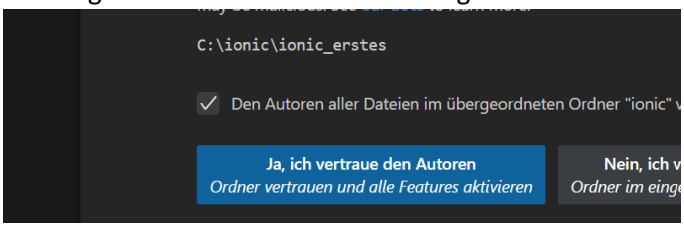
Suche den Folder:



Somit ist dieser Inhalt geöffnet und noch leer. Die Seite rechts „Getting Started“ schließe mit dem „x“.



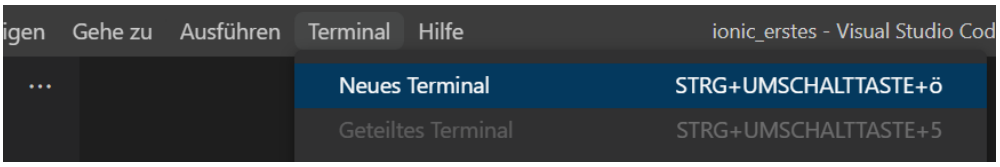
Die Frage nach dem Vertrauen bestätigen:



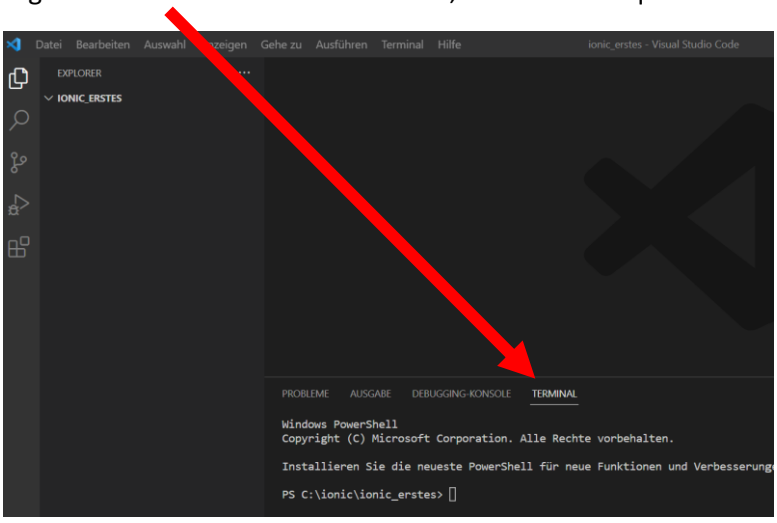
Nun muss man ein erstes App anlegen und starten. Dafür muss man nicht mehr mit der Eingabeaufforderung (CMD) arbeiten, wie wir am Beginn die Programme installiert haben, sondern können es direkt in VS-code durchführen.

- **<u>Dafür starte das integrierte Terminal:</u>**

Unter dem Menüpunkt „Anzeigen“ wähle „Integriertes Terminal“ (bzw. STRG + ö). Damit kommt man direkt in die CMD-Eingabeaufforderung, die sich ganz unten öffnet.



Ergebnis: Neues Fenster rechts unten, indem sich bequem vieles erledigen lässt:



In diesem „Terminal“ befindet man sich direkt im eben angelegten Ordner und kann die gleichen Befehle nutzen, die man sonst in der Eingabeaufforderung (CMD) machen müsste.

<u>INFO:</u>

Das spezielle Projekt muss in VSC angelegt werden und mit ionic start erstellt werden - ist dann die neueste Version.

- **Projekt starten und benennen:**

Tippe ein „ionic start“ und danach den neuen Namen des anzulegenden Projektes, z.B.

```
PS C:\ionic\ionic_erstes> ionic start erstesApp --capacitor
```

Und dahinter die beiden Bindestriche und capacitor

Drücke „enter“ Taste.

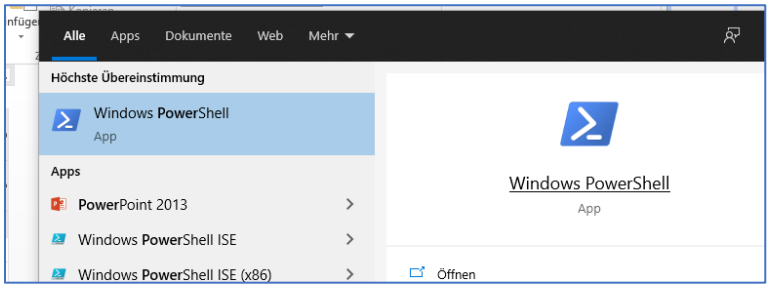
PROBLEM:

Könnte erscheinen: execution policy problems

Bei Problemen die „powershell“ – Änderung durchführen:

```
Installieren Sie die neueste PowerShell für neue Funktionen und Verbesserungen! https://aka.ms/PSWindows

PS C:\ionic\ionic_erstes> ionic start erstesApp --capacitor
ionic : Die Datei "C:\Users\Hallo\AppData\Roaming\npm\ionic.ps1" kann nicht geladen werden, da die Ausführung von Skripts auf
diesem System deaktiviert ist. Weitere Informationen finden Sie unter "about_Execution_Policies"
(https:/go.microsoft.com/fwlink/?LinkID=135170).
In Zeile:1 Zeichen:1
+ ionic start erstesApp --capacitor
+ ~~~~~
    + CategoryInfo          : Sicherheitsfehler: (:) [], PSSecurityException
    + FullyQualifiedErrorId : UnauthorizedAccess
PS C:\ionic\ionic_erstes>
```

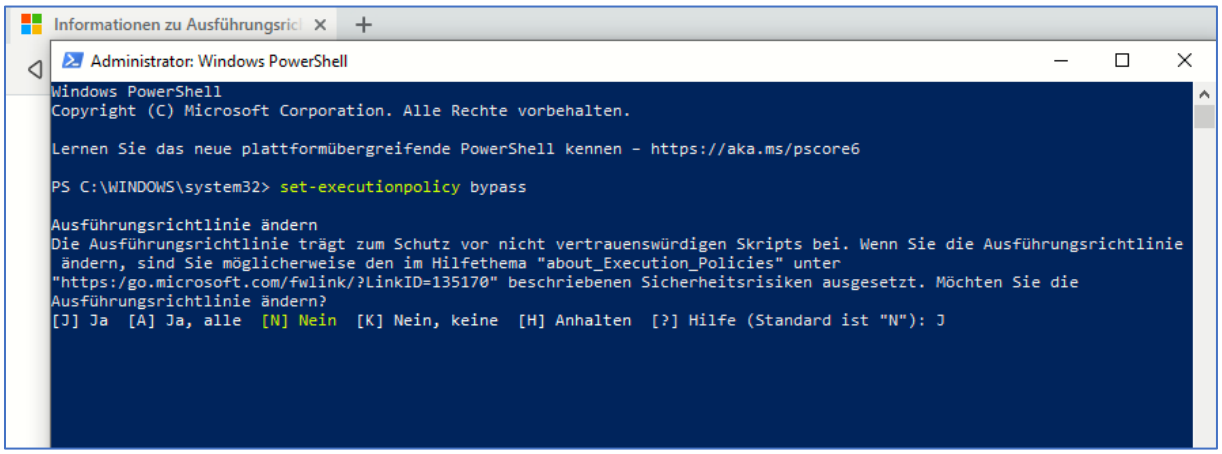
In Suche eingeben: powershell und dann rechter Mausklick als „Administrator ausführen“



Dann gib ein:

set-executionpolicy bypass

Danach ein „j“ für Zustimmung. fertig



Ende „Powershell“ Info.

TIPP:
Manchmal muss man VS-Code schließen und nochmals öffnen und mit dem „ionic start…“ wieder beginnen.

Um das eleganter zu machen, nutzte dafür die Tastenkombination: STRG + c

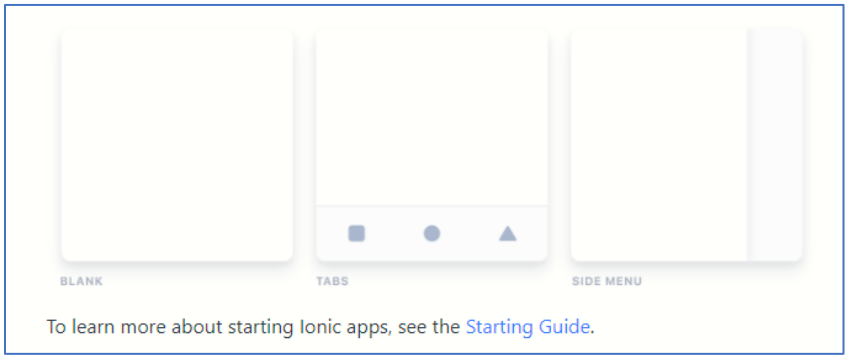
Dann wird gefragt mit welchem Framework gearbeitet werden soll, wir nutzten „Angular“.

```
Pick a framework!

Please select the JavaScript framework to use for your new app. To bypass this
prompt next time, supply a value for the --type option.

? Framework: (Use arrow keys)
> Angular | https://angular.io
  React   | https://reactjs.org
```

Daher mit „enter“ bestätigen, weil es schon blau vor ausgewählt ist.

Danach wird nach dem gewünschten Template gefragt, wobei auf der Website von ionic (https://ionicframework.com) in der Dokumentation nach lesbar ist, dass es drei Varianten gibt, nämlich das

- blank (leer)
- tabs (mit 3 Tabs unten in der Fußzeile) und
- side menu (ein slider von links mit Menüpunkten)



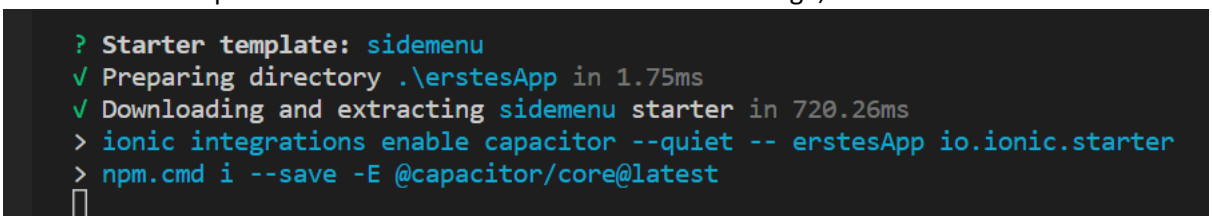
Wähle das „side menu“ mit Hilfe der Pfeiltaste nach unten, damit es blau erscheint:

```
? Starter template:
  tabs         | A starting project with a simple tabbed interface
> sidemenu     | A starting project with a side menu with navigation in the conten
t 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
```

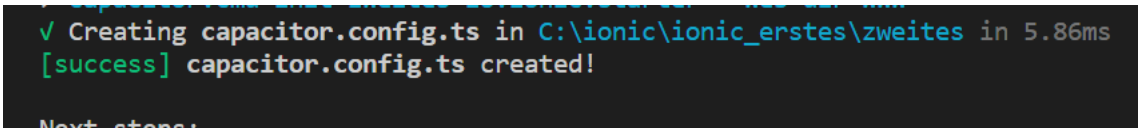
Klicke dann auf „enter“.

Nun wird das Projekt angelegt, was durchaus ein paar Minuten dauern kann.

Nun wird auch Capacitor installiert. Aber hier steht es ziemlich lange, bitte warten!

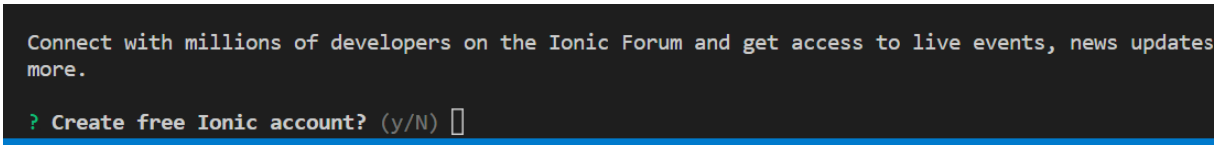


Zwischendurch:

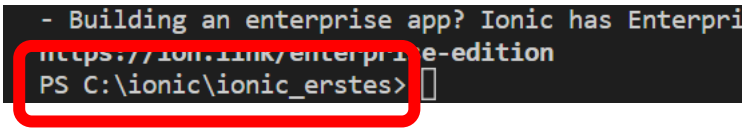


Ein N ist hier ok, muss ja nicht sein: also

- n
- Enter-Taste

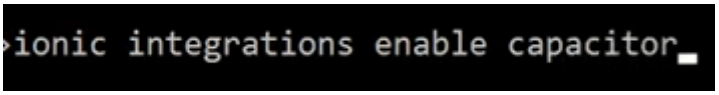


Es ist fertig, wenn wieder der Ordner angezeigt wird:



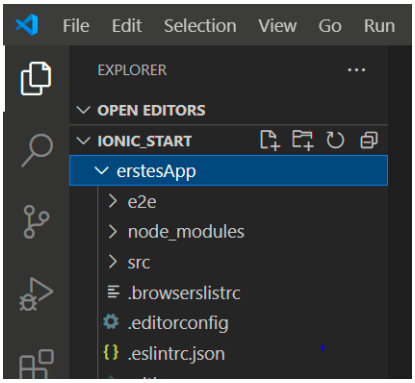
Info:

Man könnte auch danach in ein bestehendes Projekt „capacitor“ integrieren:



2)grundlegender Aufbau der App

Nun wurde das Projekt automatisch angelegt. Den ersten Eindruck kann man sich holen, indem man links auf den Namen des Projektes klickt. Damit öffnet man den umfangreichen Inhalt:



Es befinden sich viele Konfigurationsdateien hier, die für den reibungslosen Ablauf nötig sind.

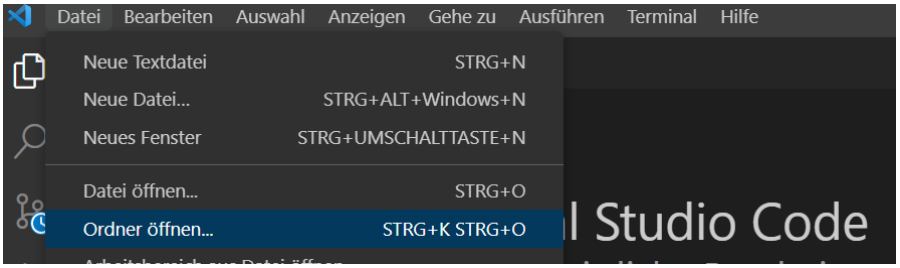
- packager.json für die „dependences“ – da Angular im Hintergrund arbeitet
  Hier sieht man auch die grundlegenden Befehle:

```
"scripts": {
  "ng": "ng",
  "start": "ng serve",
  "build": "ng build",
  "test": "ng test",
```

- **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.
- e2e …end to end – hier kommt das Endprodukt hinein, wenn man einen Test erstellen will, ob alles so funktioniert. Es simuliert einen echten User, der testen kann.
- node_modules beinhaltet viele „libraries“, damit alles funktioniert

3a.) Projekt im Server starten und anzeigen lassen - im Browser ansehen

Öffne den Ordner wieder in Visual Studio-Code aber gleich im gerade angelegten Ordner im Ordner „ionic_start“, also bereits eine Ebene tiefer mit Hilfe von



MAN MUSS in diesem Projekt drinnen sein, um es dann später im Browser ansehen zu können!

Bereits danach kann man den Server im Terminal starten, um das App auch im Browser zu sehen:

- ionic serve
- Abkürzung: ionic s

```
PS D:\ionic_start\erstesApp> ionic serve
```

oder Kurzversion:

```
PS D:\ionic_start\erstesApp> ionic s
```

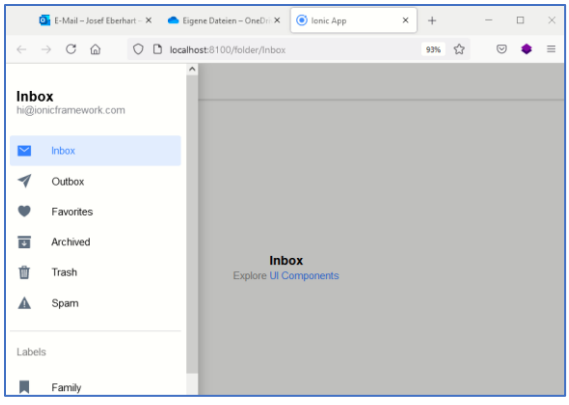
Dauert etwas bis es fertig ist:

```
√ Compiled successfully.
```

Info:

Wenn man in VS-Code speichert, wird der Code sofort kompiliert und man kann dies im integrierten Terminal auch sehen. Das heißt auch, dass die Ansicht neu erstellt wird.

Natürlich muss der Server laufen. Dafür muss vorher im Terminal „ionic serve“ ausgeführt worden sein. Ergebnis:



Wir haben ein Projekt mit Side-Menü erstellt. Daher erhalten wir auch das. Sieh dir das Menü an.

mit **STRG + c** kommt man wieder dazu, weiter Befehle im Terminal eingeben zu können.

```
       Use Ctrl+C to quit this process

[INFO] Browser window opened to http://localhost:8100!

PS C:\ionic\ionic_erstes\erstesApp>
```

Tipp: Mit PFEILTASTE nach oben führt den letzten Befehl noch mal aus:

Info:
Im Browser kann man es auch direkt öffnen, nachdem es am Server einmal gestartet wurde, indem man eingibt:

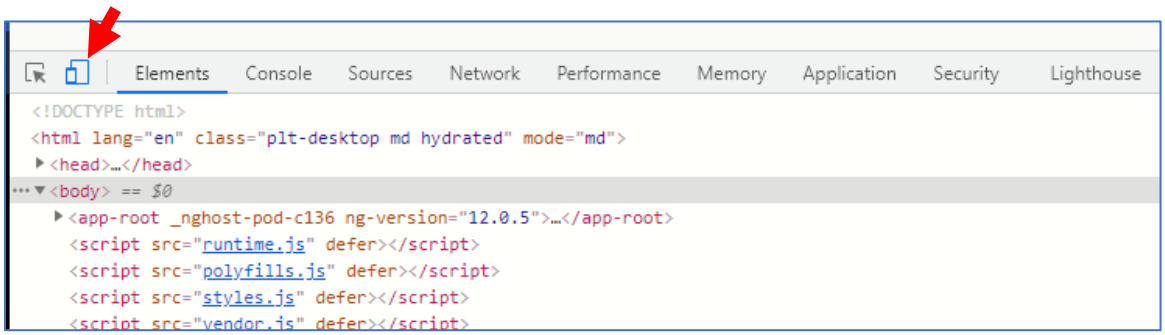
http://localhost:8100

3b.)App im Smartphone-Modus betrachten – auch im Browser

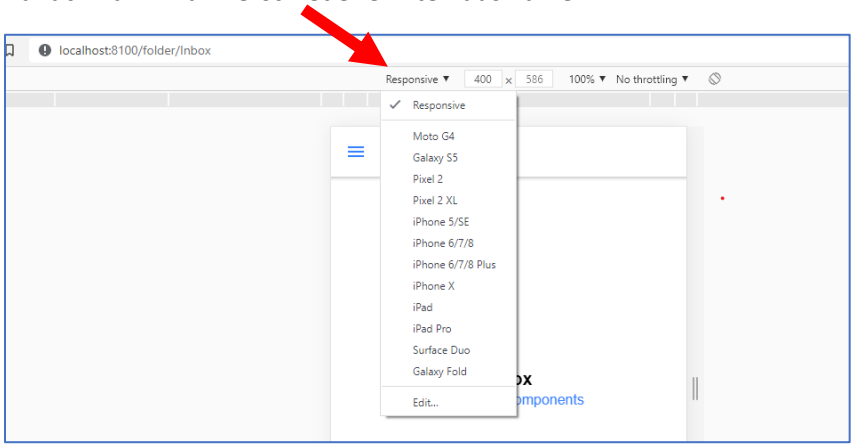
Ist das App durch den oberen Schritt im Browser geöffnet, dann öffne

- **mit der F12-Taste den Entwicklermodus oder**
- **mit STRG + SHIFT + i**

und suche in der Menüleiste links (oder bei anderen Browsern rechts) das Symbol für die Handy-Darstellung.



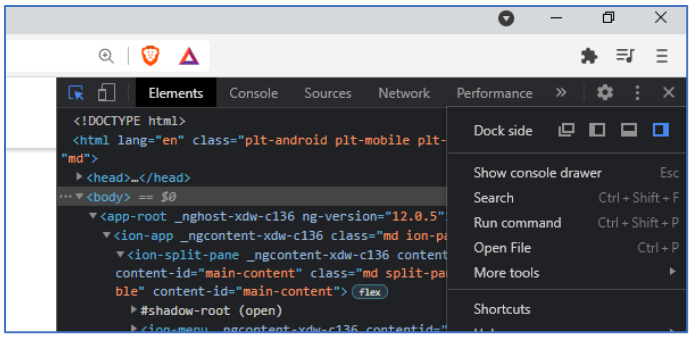
Danach kann man verschiedene Arten auswählen:



Wechselt man die Ansicht, muss man mit Drücken der Taste „F5“ aktualisieren.

Änderungen:

- dark mode – Einstellungen – und dann Theme: Dark
  schließen mit F12 und neu mit F12 öffnen (zum Aktualisieren)
- rechts statt unten – drei Punkte und Symbol „rechts“

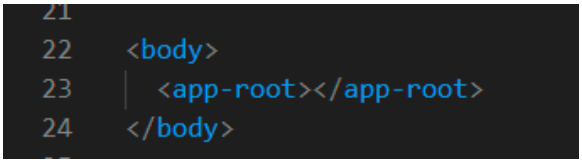


4.)Erste Änderung – Text ändern

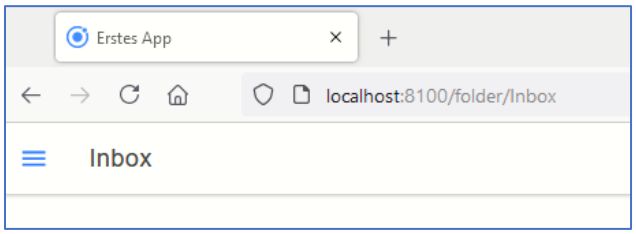
- Öffne „index.html“ und ändere den Titel, der auch im Registerblatt angezeigt wird von „Ionic App“ auf „erstes App“.

Info: In der index.html wird nur viel weitergeleitet, aber nicht viel selbst geschrieben:

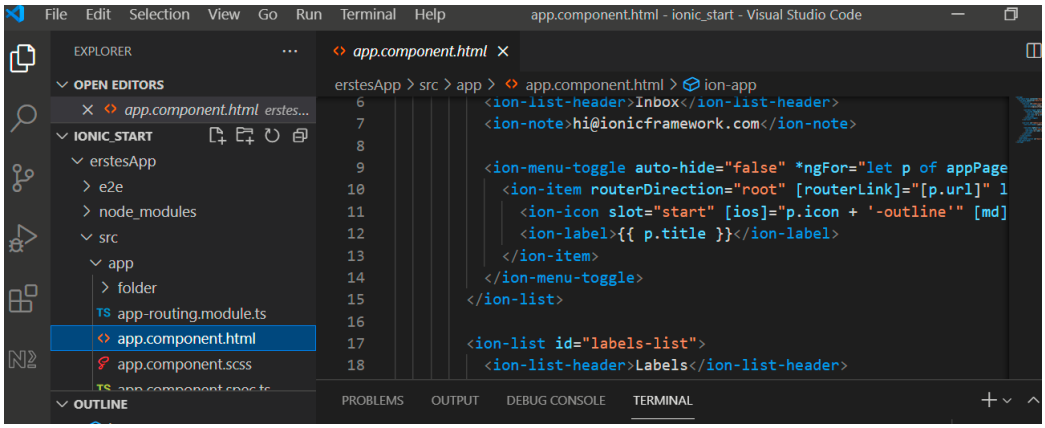
Der <body> besteht nämlich nur aus der Umleitung:



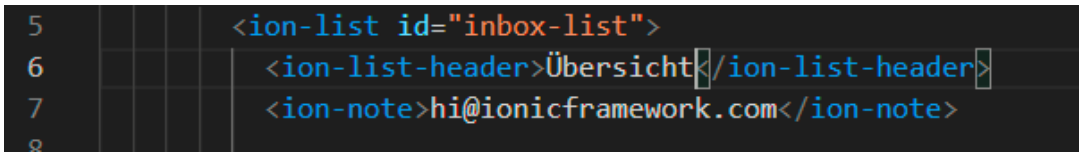
Ergebnis:



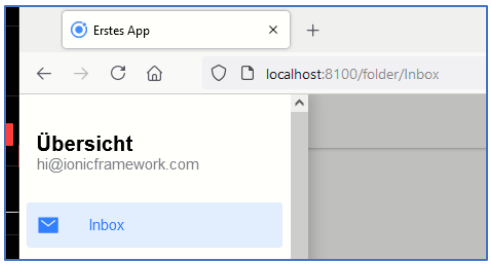
- Öffne in src / app / app.component.html



Ändere hier header von „Inbox“ auf:



Ergebnis:



Info:

Aufbau von Elementen erfolgt mit der Bezeichnung „ion“. Wie hier ersichtlich, startet der „header“ mit „ion-list-header“.

Das Öffnen und Schließen der Tags erfolgt genauso, wie wir es von HTML kennen.

5)Neue Seite erstellen

Dafür gibt es einen Befehl den man im Terminal eingeben kann, der automatisch diese Seite mit allen Abhängigkeiten für das gesamte Ionic-Projekt erstellt.

- ionic generate page „Name“

In unserem Fall soll, wegen der Übersicht, ein eigener ORDNER „pages“ erstellt werden und DARIN dann diese neue Seite – und alle darauffolgenden Seiten.

Daher gib ein:

ionic generate page pages/loader

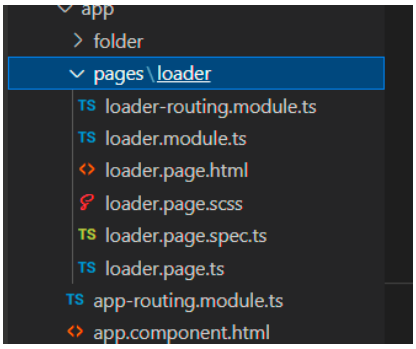
Diese Seite soll nämlich ein automatisches Loader-Symbol enthalten. Dies soll ein sich drehendes Icon sein.

```
PS D:\ionic_start> cd erstesApp
PS D:\ionic_start\erstesApp> ionic generate page pages/loader
```

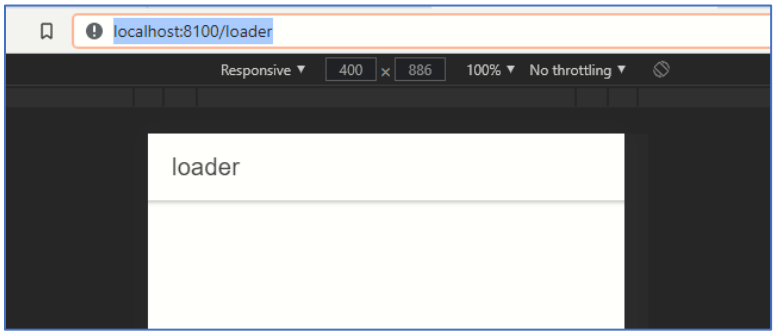
Kurzschreibweise: Statt dem „generate“ genügt ein kurzes „g“

Ergebnis:

Automatisch angelegter Ordner mit den vielen Unterseiten für die Seite „loader“:



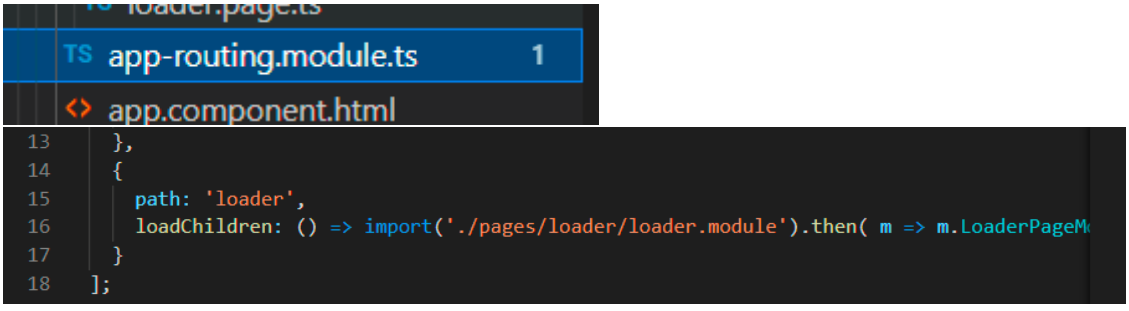
Im Browser noch nur direkt aufrufbar: ändere auf http://localhost:8100/loader



Es gibt hier nur einen Header und einen leeren Inhalt.

INFO:

Beim Erstellen der „loader“-Seite wurde automatisch auch ein passender Pfad dafür angelegt. Das passierte in der Datei:



Info:

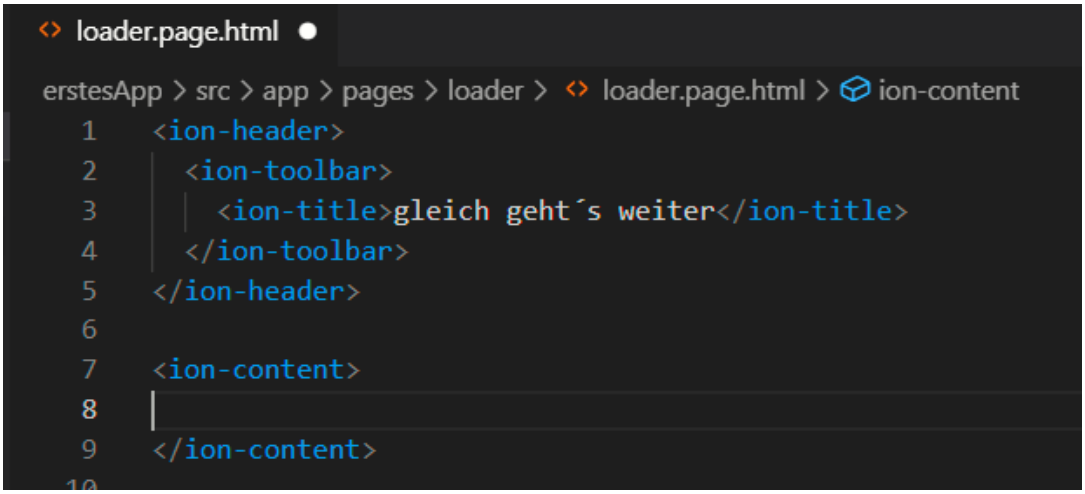
- die HTML-Seite wird für die Ansicht im Browser benötigt
- die TS-Seite wird die Logik enthalten
- die CSS (bzw. scss) nimmt das Design auf

- **<u>Öffne die Site“ loader.page.html“</u>**

Info: Typisch ist die Seitengestaltung mit

- ion-header
- ion-content
- ion-footer

Ändere den Text auf:

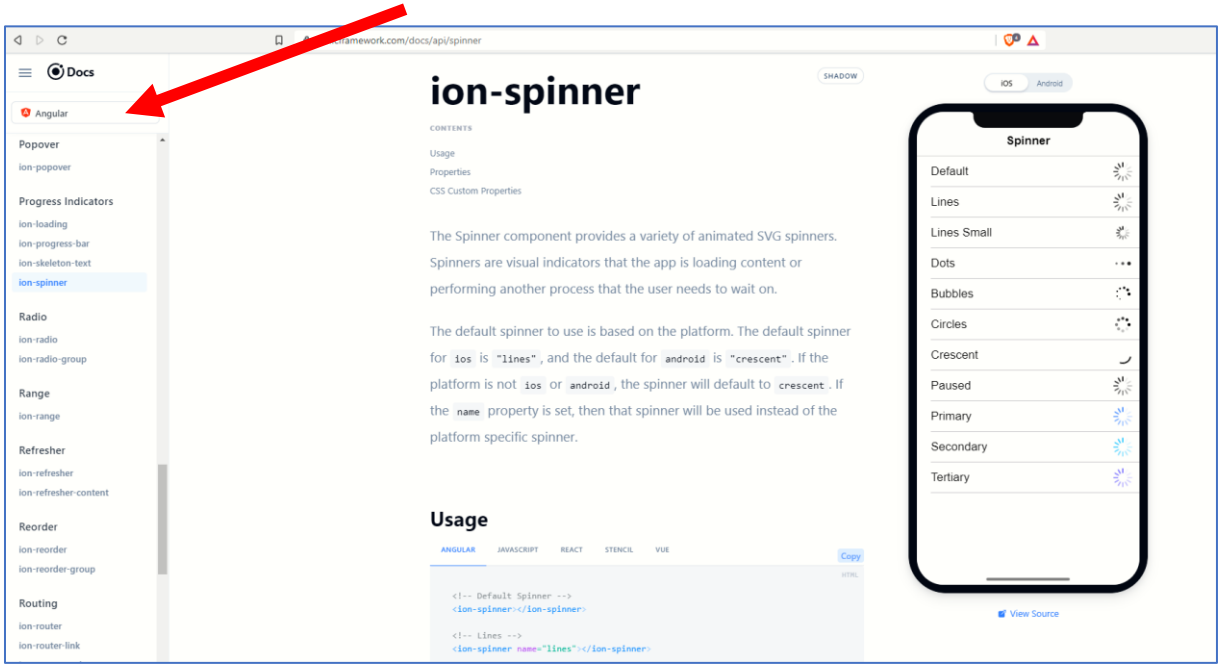


Um ein drehendes Icon einfügen zu können, besuche die Seite des Frameworks, welche die Dokumentation enthält und dabei auch die UI Components anzeigt. Diese User Interface Komponenten dienen dazu, kopiert, erklärt und verwendet zu werden.

https://ionicframework.com/docs/components

Es interessiert hier der „progress indicator“, nämlich der „ion-spinner“.

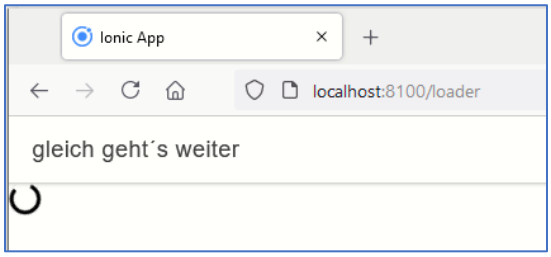
Beachte, dass du im Bereich „Angular“ – mit dem wir ja als Basis im Hintergrund arbeiten, bist.



Nutze den „default“ – kopiere den Code aus dem Bereich „usage“ und füge es in den Code ein:


```
7   <ion-content>
8     <ion-spinner></ion-spinner>
9   </ion-content>
10
```

Ergebnis:

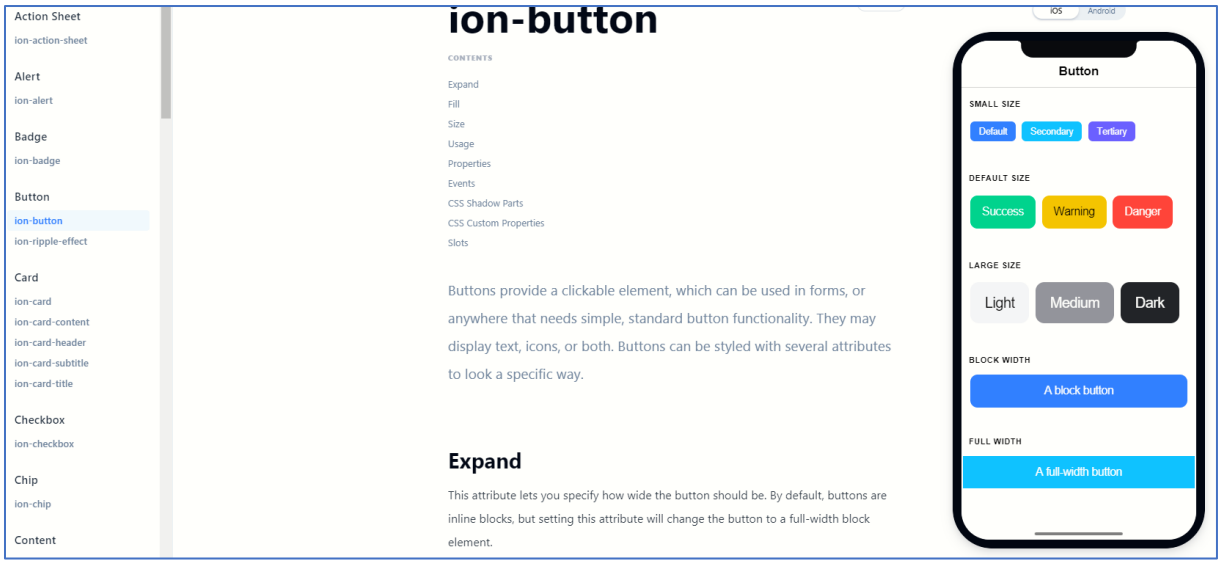


Nun soll noch einiges damit geübt werden:

- Farbe ändern
- In die Mitte stellen mit CSS

Farbe ändern:

wie schon bekannt aus Bootstrap, kann die einfache Namensverwendung hier genutzt werden. Es ist auch hier ersichtlich: dazu findet man den Code bei „Button“



Verwende die Farbe GRÜN „success“

```
<ion-content>
  <ion-spinner color="success"></ion-spinner>
</ion-content>
```

Style ändern – um den Loader in die Mitte zu bringen

Öffne die dazu passende Seite „loader.page.scss“.

Klasse erstellen:

- Zuerst der Punkt
- Name der Klasse ist beliebig, sollte aber sinnvoll sein
- Die geschweiften Klammern auf und zu

Zuerst erstelle in dieser leeren Seite eine Klasse namens (frei wählbar) „flex-center“ – weil das passend ist.

- Der Typ „display: flex“ ist schon dafür ausgelegt, etwas zu zentrieren. Es ist somit nach links und rechts „flexibel“ – also nicht festgelegt.
- Zentrieren horizontal: justify-content: center
- Zentrieren vertikal: align-items: center;
- Damit es vertikal auch in der Mitte ist (wegen des div-Containers) benötigt man eine height: 100%;

Für die Klasse einen Punkt davor. Darin dann der CSS-Befehl:

loader.page.html loader.page.scss

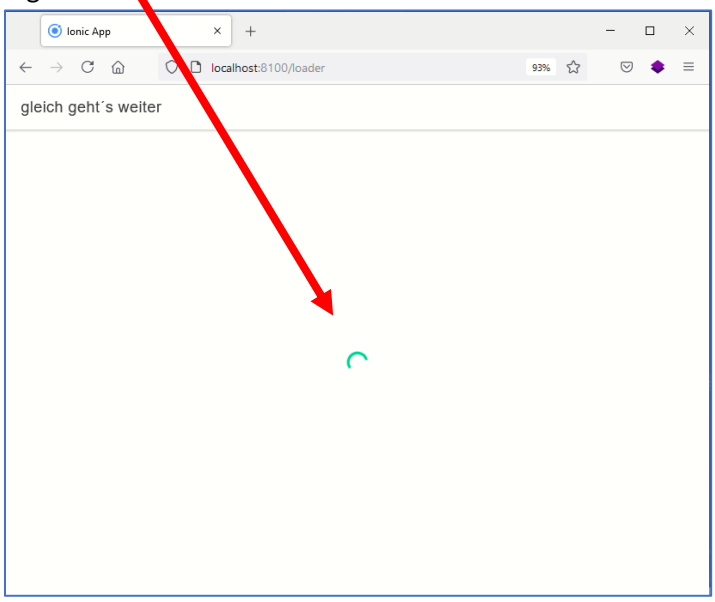
erstesApp > src > app > pages > loader > loader.page.scss > ...

```scss
.flex-center {
  display: flex;
  justify-content: center;
  align-items: center;
  height: 100%;
}
```

Damit dieser Code in der HTML-Seite auch ankommt, muss man dort nun noch diese Klasse „einbauen“.

```html
<ion-content>
  <div class="flex-center">
  <ion-spinner color="success"></ion-spinner>
  </div>
</ion-content>
```

Ergebnis:



6.)Navigation auf den „loader“ als Startseite umstellen

Öffne dafür die Datei „app-routing.module.ts“.

Diese ist für das Routing zuständig.

```
4   const routes: Routes = [
5     {
6       path: '',
7       redirectTo: 'folder/Inbox',
8       pathMatch: 'full'
9     },
```

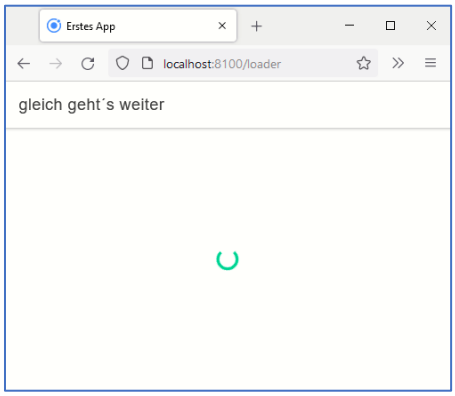
In Zeile 7 steht das „redirectTo“ zu „folder“. Dies erfolgt, wenn der Pfad in der URL leer ist, da dort in Zeile 6 angeführt ist „path: ''. Wenn also die URL ohne eine Unterseite eingegeben wird, was ja für das normale Aufrufen einer Website üblich ist, soll die Weiterleitung (redirect) erfolgen, hier jedoch nun zu unserer Seite „loader“.

Daher ändere im „redirectTo“ den Pfad zu unserer Seite „loader“.

```
6       path: '',
7       redirectTo: 'loader',
8       pathMatch: 'full'
```

Nun sollte beim Aufrufen der Haupt-URL, also ohne einer Site nach den Slash der Loader sichtbar sein:

http://localhost:8100



Wird sofort zum „loader“ weitergeleitet.