

## Ionic 5 – Start, Seite mit Loader-Icon anlegen

1. Projekt im Server starten und im Browser anzeigen lassen
2. Entwicklermodus öffnen für Smartphone Ansicht
3. Erste Änderung und Text ändern
4. Neue Seite anlegen mit einem Icon „loader“
5. Navigation umstellen, damit man mit dem „loader“ beginnt

### 1.) Projekt im Server starten und im Browser anzeigen lassen

Öffne den Ordner wieder in Visual Studio-Code und starte zuerst das neue Projekt indem man in dieses hinein wechselt mit

- `cd erstesApp`

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

```
PS D:\ionic_start> cd erstesApp
PS D:\ionic_start\erstesApp> |
```

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

- `ionic serve`

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

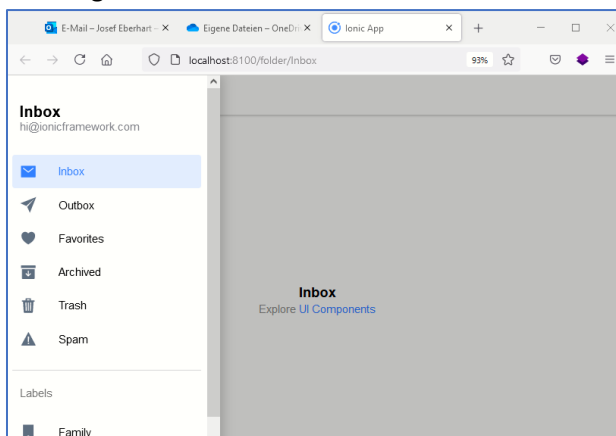
oder Kurzversion:

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

### 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:



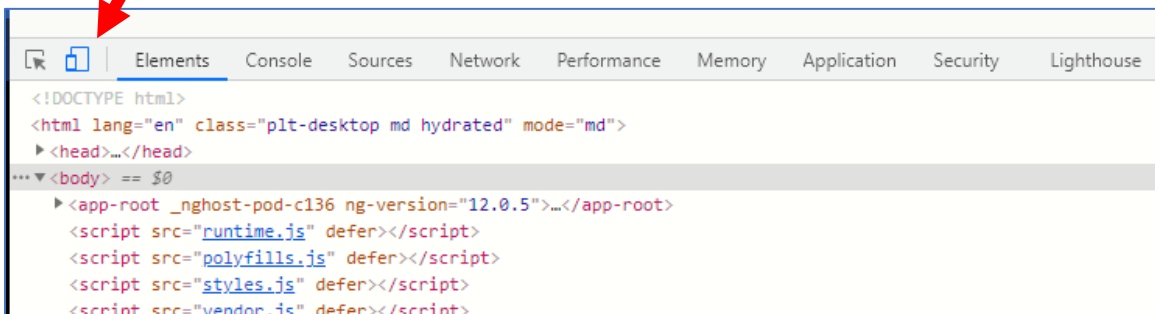
Wir haben ein Projekt mit Side-Menü erstellt. Daher erhalten wir auch das. Sieh dir das Menü an.

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

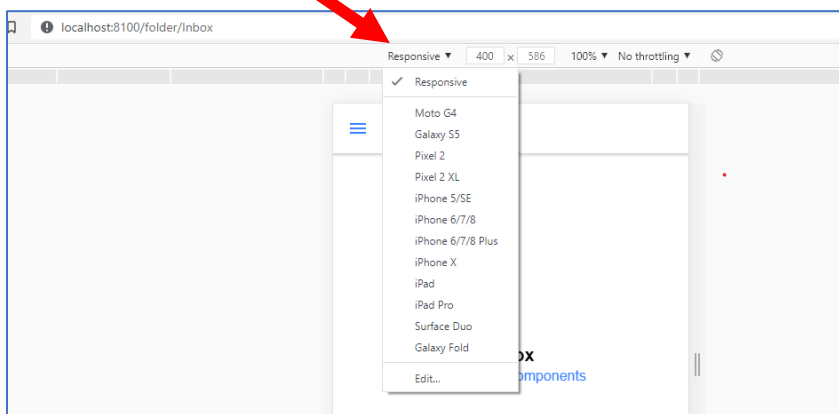
<http://localhost:8100>

## 2.) App im Smartphone-Modus betrachten

Öffne mit der F12-Taste den Entwicklermodus und suche in der Menüleiste links das Symbol für die Handy-Darstellung.



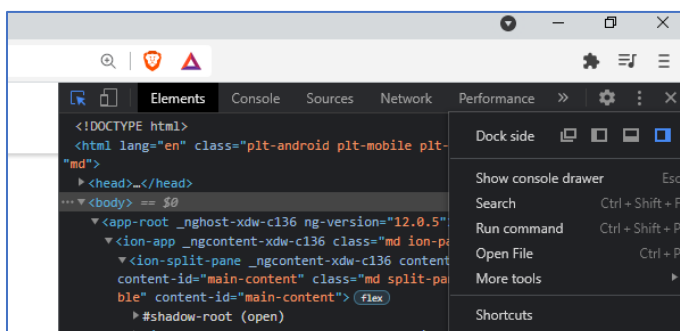
Danach kann man verschiedene Arten auswählen:



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“



### 3.) Erste Änderung – Text ändern

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

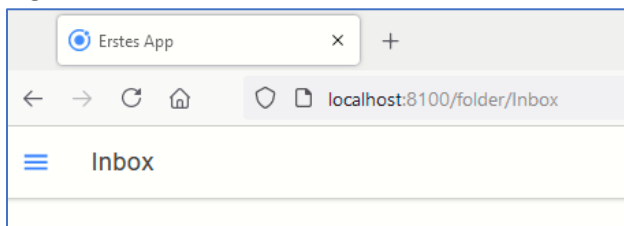
```
index.html •
erstesApp > src > index.html > html > head > title
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="utf-8" />
6   <title>Erstes App</title>
7
8   <base href="/" />
9
```

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

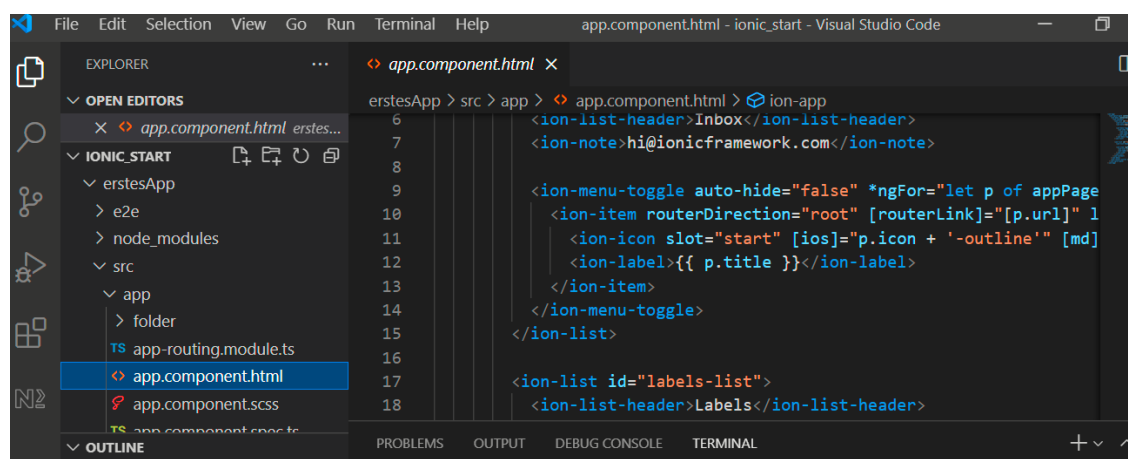
Der <body> besteht nämlich nur aus der Umleitung:

```
21
22 <body>
23   <app-root></app-root>
24 </body>
```

Ergebnis:



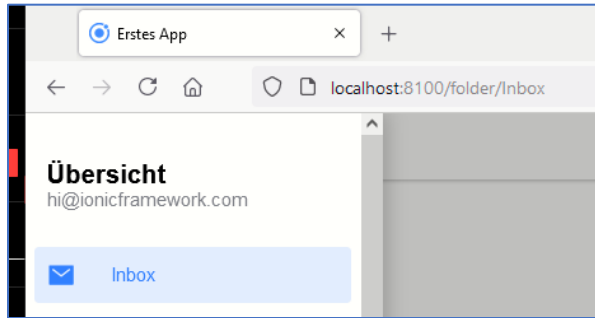
- Öffne in src / app / app.component.html

A screenshot of the Visual Studio Code editor. The Explorer sidebar on the left shows the project structure with 'app.component.html' selected. The main editor window shows the content of 'app.component.html' with line numbers 6 through 18. The code includes an ion-app container with an ion-list containing an ion-item with an ion-label and an ion-note.

**Ändere hier header von „Inbox“ auf:**

```
5 <ion-list id="inbox-list">
6   <ion-list-header>Übersicht</ion-list-header>
7   <ion-note>hi@ionicframework.com</ion-note>
```

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.

### 4.)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
```

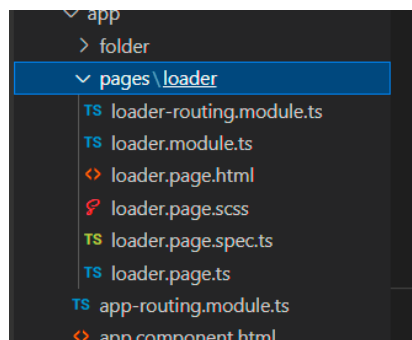
Diese Seite soll nämlich ein automatisches Loader-Symbol enthalten. Dies soll ein sich drehendes Icon sein.

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

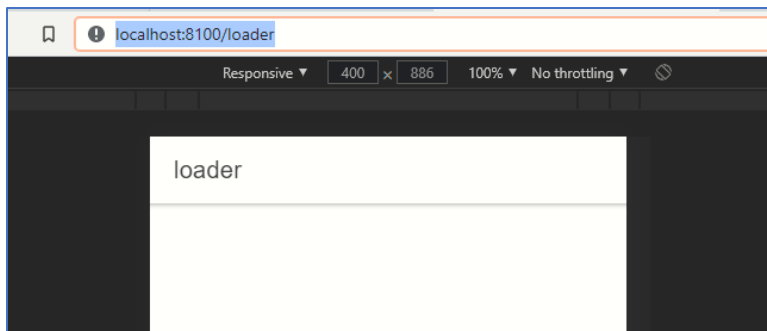
Kurzschreibweise: Statt dem „generate“ genügt ein kurzes „g“

**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:

```
TS loader.page.ts
TS app-routing.module.ts 1
<> app.component.html
13 },
14 {
15   path: 'loader',
16   loadChildren: () => import('./pages/loader/loader.module').then( m => m.LoaderPageM
17 }
18 ];
```

- **Öffne die Site“ loader.page.html“**

Ändere den Text auf:

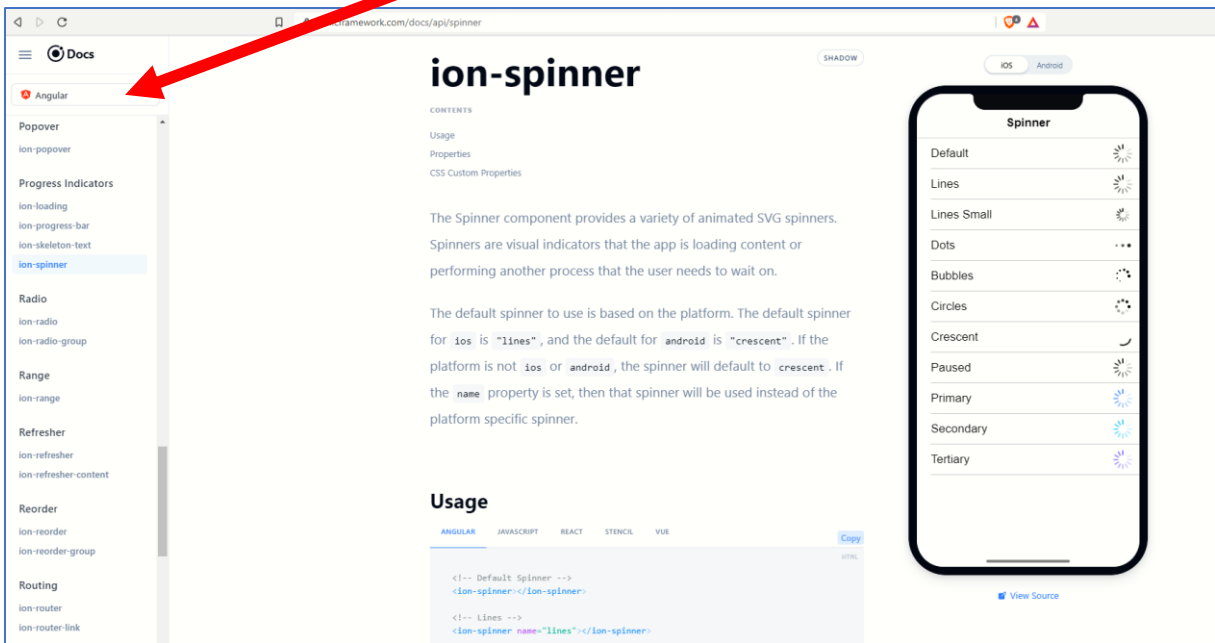
```
<> loader.page.html ●
erstesApp > src > app > pages > loader > <> loader.page.html > ion-content
1 <ion-header>
2   <ion-toolbar>
3     <ion-title>gleich geht's weiter</ion-title>
4   </ion-toolbar>
5 </ion-header>
6
7 <ion-content>
8 |
9 </ion-content>
10
```

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.

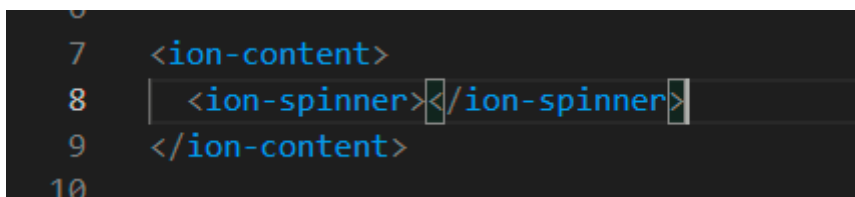
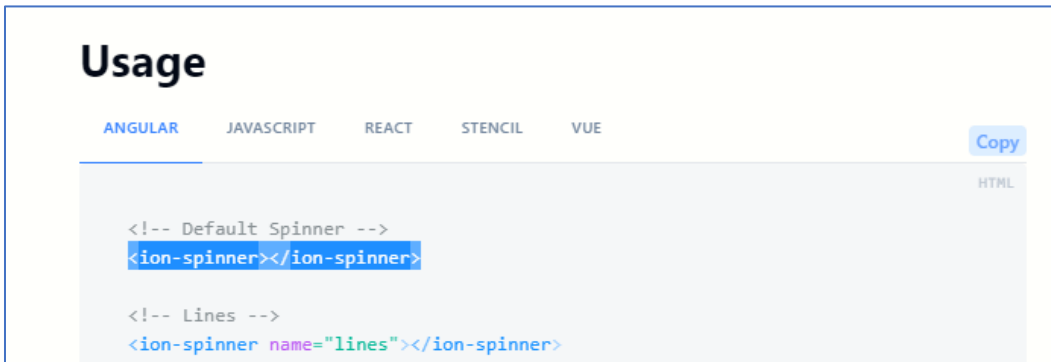
<https://ionicframework.com/docs/components>

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

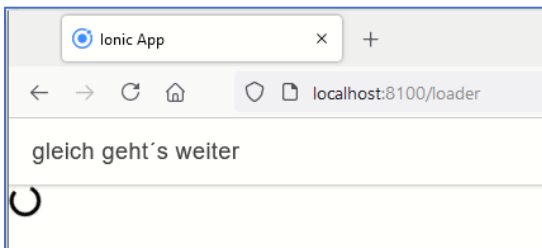
Beachte, dass du im Bereich „Angular“ – mit dem wir ja als Basis im Hintergrund arbeiten, bist.



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



Ergebnis:

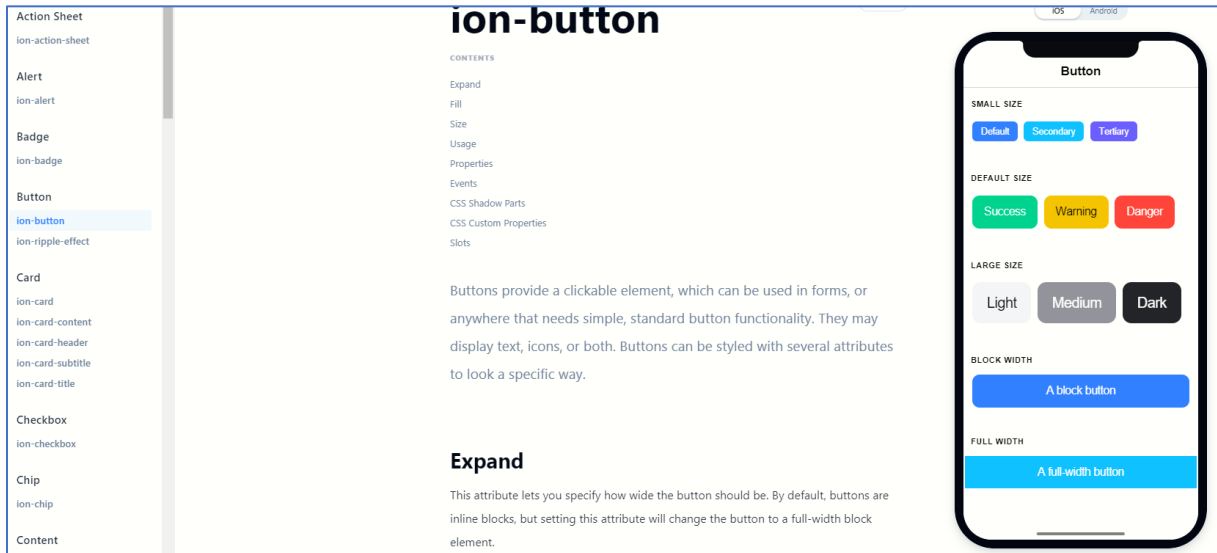


Nun soll noch einiges damit geübt werden:

- 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“

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

## Style ändern – um den Loader in die Mitte zu bringen

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

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 wirklich in der Mitte ist (wegen des div-Containers) benötigt man eine height: 100%;

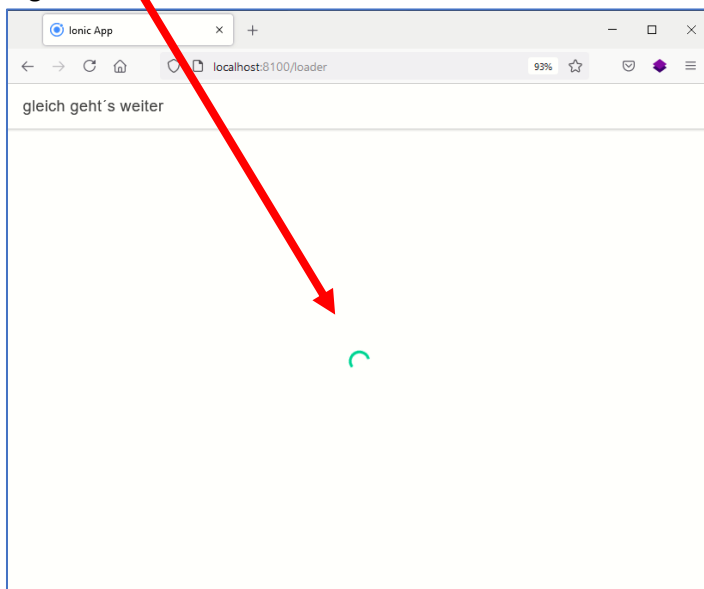
Für die Klasse einen Punkt davor. Darin dann der CSS-Befehl:

```
loader.page.html loader.page.scss
erstesApp > src > app > pages > loader > loader.page.scss > ...
1  .flex-center {
2    display: flex;
3    justify-content: center;
4    align-items: center;
5    height: 100%;
6  }
```

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

```
7 <ion-content>
8   <div class="flex-center">
9     <ion-spinner color="success"></ion-spinner>
10  </div>
11 </ion-content>
```

Ergebnis:





## 5.) 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      },
```

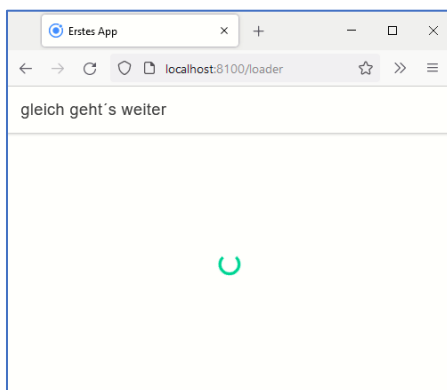
In Zeile 7 steht das „redirectTo“ zu „folder“. Dies erfolgt, wenn der Pfad in der URL leer ist, da dort in Zeile 6 angeführt ist „path: “. Wenn also die URL ohne eine Unterseite eingegeben wird, was ja für das normale Aufrufen einer Website üblich ist, soll die Weiterleitung (redirect) erfolgen, hier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