

Routing

`ng new myproject --routing`

(or just say Yes when asked for routing)

In `app-routing.module.ts`

```
const routes: Routes = [  
  { path: 'custom', component: CustomComponent }  
  //going to /custom will show CustomComponent in <router-outlet>  
]
```

Navigation with RouterLink

~~Goto Custom~~ Causes page to reload

Goto Custom No page reload

<a [routerLink]="['custom', 'level']">Goto Custom Goto /custom/level

Styling Active Link

customCssClass will be added to <a> when link is /abc

Accessing Query and Query Params of URL

```
const routes: Routes = [  
  { path: 'custom/:id', component: CustomComponent }  
]
```

URL = '/custom/123?name=abc'

```
constructor(private route: ActivatedRoute){  
  this.route.queryParams.subscribe((data) => {  
    // data.name == 'abc'  
  })  
  this.route.params.subscribe((data) => {  
    //data.id === '123'  
  })  
}
```

Lifecycle Hooks

Basic Hooks

ngOnInit

Called once after the component is first displayed on the screen and after Angular has set any properties passed down from the parent component

ngOnChanges Called anytime a property of the component is changed

ngOnDestroy Called once when Angular is about to remove the component (eg. when we navigate to different route)

For More: <https://angular.io/guide/lifecycle-hooks>

Making HTTP Requests

In app.module.ts

```
import { HttpClientModule } from '@angular/common/http';
```

Add HttpClientModule to imports Array

```
imports: [... , HttpClientModule]
```

To Use, inject HttpClient into Class

```
constructor(private http: HttpClient){  
  // use http to make requests  
}
```

Making GET Requests

```
const getReq = this.http.get<ResponseType>(url, options); // returns observable  
getReq.subscribe(data => {}); //request will not be made if no one subscribes
```

Options can contain keys for
headers, observe, params, reportProgress, responseType, withCredentials

headers: to add header to request
eg: headers : {"Auth" : "Token132"}

params: To add Query Params
eg: params : {"name" : "abcd"}

responseType: Set type of response received.
Can be 'arraybuffer', 'blob', 'json' or 'text'. Default is json

Making POST Requests

```
const postReq = this.http.post<ResponseType>(url, body, options); // returns observable  
    postReq.subscribe(response => {})
```

Body Contains data to send with request, options same as previous

For more info and methods: <https://angular.io/guide/http>

Deployment

```
npm i -g surge
```

To deploy, open folder in terminal/cmd and run
surge


```
npm i -g now
```

To Deploy, run command: now