This is sample app, that created with following Angular 4 setup just to perform CRUD operations. Like Adding/Update or Editing details/ Deleting the details.

\_ \_ \_\_\_\_ \_ \_\_\_

/ \ \_ \_\_ \_\_ \_ \_ \_| | \_\_ \_ \_ \_\_ / \_\_\_| | |\_ \_|

/ △ \ | '\_ \ / \_` | | | | |/ \_` | '\_\_| | | | | | |

/ \_\_\_ \| | | | (\_| | |\_| | | (\_| | | | |\_\_\_| |\_\_\_ | |

/\_/ \\_\\_| |\_|\\_\_, |\\_\_,\_|\_|\\_\_,\_|\_| \\_\_\_\_|\_\_\_\_\_|\_\_\_|

|\_\_\_/

@angular/cli: 1.0.0

node: 9.3.0

os: win32 ia32

@angular/common: 4.4.6

@angular/compiler: 4.4.6

@angular/core: 4.4.6

@angular/forms: 4.4.6

@angular/http: 4.4.6

@angular/platform-browser: 4.4.6

@angular/platform-browser-dynamic: 4.4.6

@angular/router: 4.4.6

@angular/cli: 1.0.0

@angular/compiler-cli: 4.4.6

**To setup and run the app:**

Both fake server and angular app should run separate cmd windows

To access data customer.sample.json I've used "json-server" to make FAKE API call, that you can get to know about this from below opensource link

https://github.com/typicode/json-server

**To run the fake api server:**

json-server -p 3001 customers-sample.json

* Currently, am using 3001 as my dynamic port.

**To run Angular app :**

ng serve --open or ng serve - this will listen at default port 4200

**Endpoints for Lising/adding/editing/deleting customer records:**

* Get: http://localhost:3001/custDetails
* Post: http://localhost:3001/custDetails/
* Put: http://localhost:3001/custDetails/1212 -> [note 1212 is particular id of the record on which we are going to update or edit.]
* Delete : ttp://localhost:3001/custDetails/1212 [similar as PUT method, deleting by record by id ]

**Code Walkthrough :**

*getAllCustomerList() function from home/home.component.ts*

will call http://localhost:3001/custDetails as follows

getAllCustomerList = function() {

this.http.get("http://localhost:3001/custDetails").subscribe(

(res: Response) => {

console.log("customer data", this.custDetails)

this.custDetails = res.json();

}

)

}

Then am populating data as below , with \*ngfor

\*ngFor = "let cust of custDetails"

<td>{{cust.id}}</td>

<td>{{cust.name.first}} {{cust.name.last}}</td>

getAllCustomerList() will be called inside at page initialization ngOnInit() to load data.

**routerLink = '/addCust'** from home.component.html will be the responsible for communicating loading corresponding component view . Here this will load **addCustomer.component** and **addCustomer.component.html** page.   
  
using function constructor pattern, the requested data will be wrapped in object as below

addNewCustomerRecord = function(cst) {

this.addNewCustomer = {

"id":parseInt(cst.id),

"name": {

"first": cst.first,

},

"birthday":cst.birthday,

"customerLifetimeValue":parseInt(cst.customerLifetimeValue)

}

this.http.post("http://localhost:3001/custDetails/", this.addNewCustomer).subscribe((res:Response) => {

console.log("res",Response)

this.isRecordInserted = true;

})

}

**addNewCustomerRecord()** from view will trigger the event with (ngSubmit) , that takes form data in the form of ‘.value’ notation. In **addCustomer.component**, *this.addNewCustomer* object will hold the data and passed on to post method. It write new details in our **customer-sample.json** file.

Both PUT(update) and DELETE(delete) method working similar. (I,e) calling the PUT/DELETE method by passing id of object as parameter in endpoints. Please take a look below.

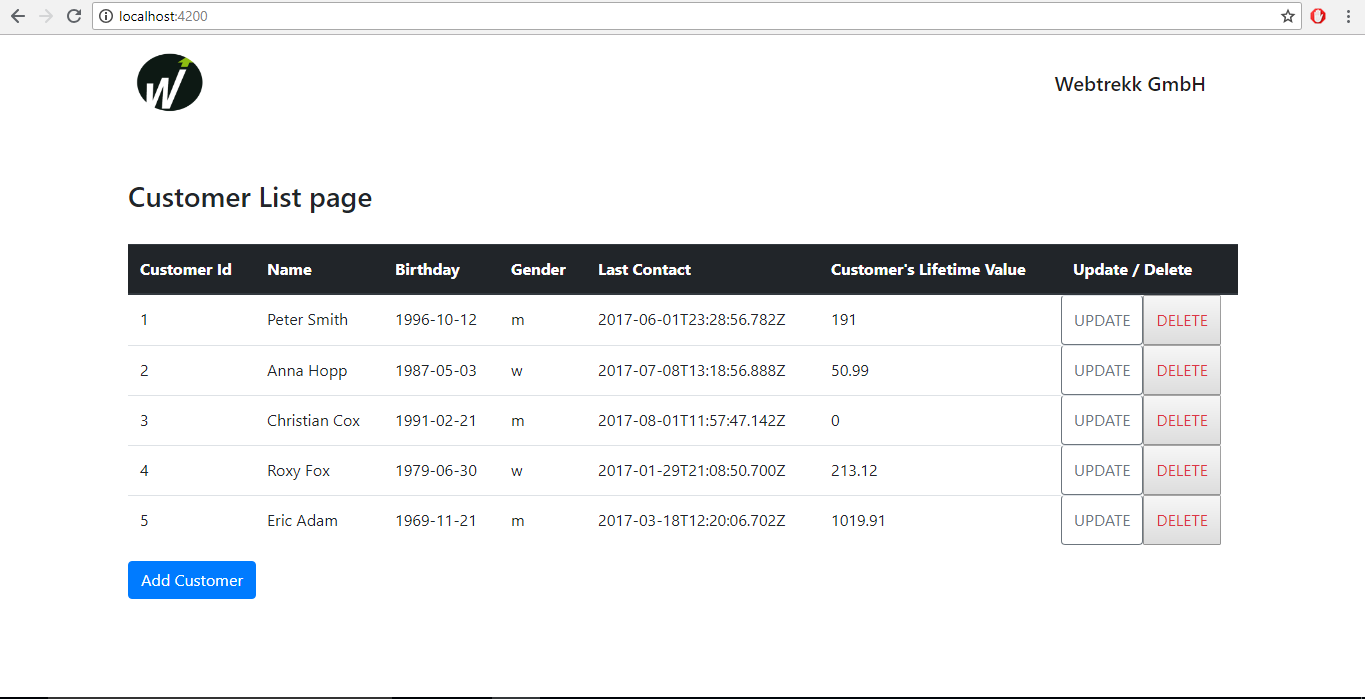
PUT - const url = `${"http://localhost:3001/custDetails"}/$**{this.id}`;**

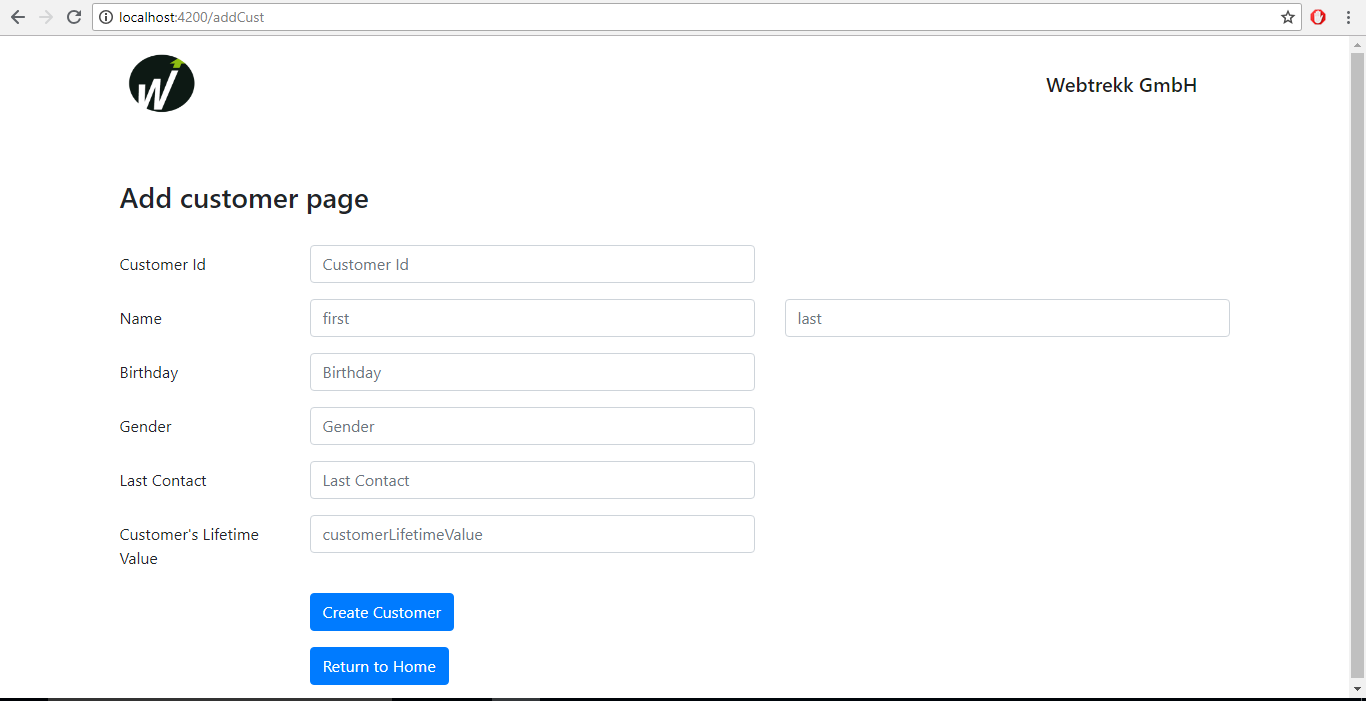
**[routerLink] = "['/editCustomerDetails', cust.id]"** this rout link in home view construct with route params that has cust.id which parameter of **the editCustDetails(cst){}** function, here the function barely created with pure Object with necessary properties.

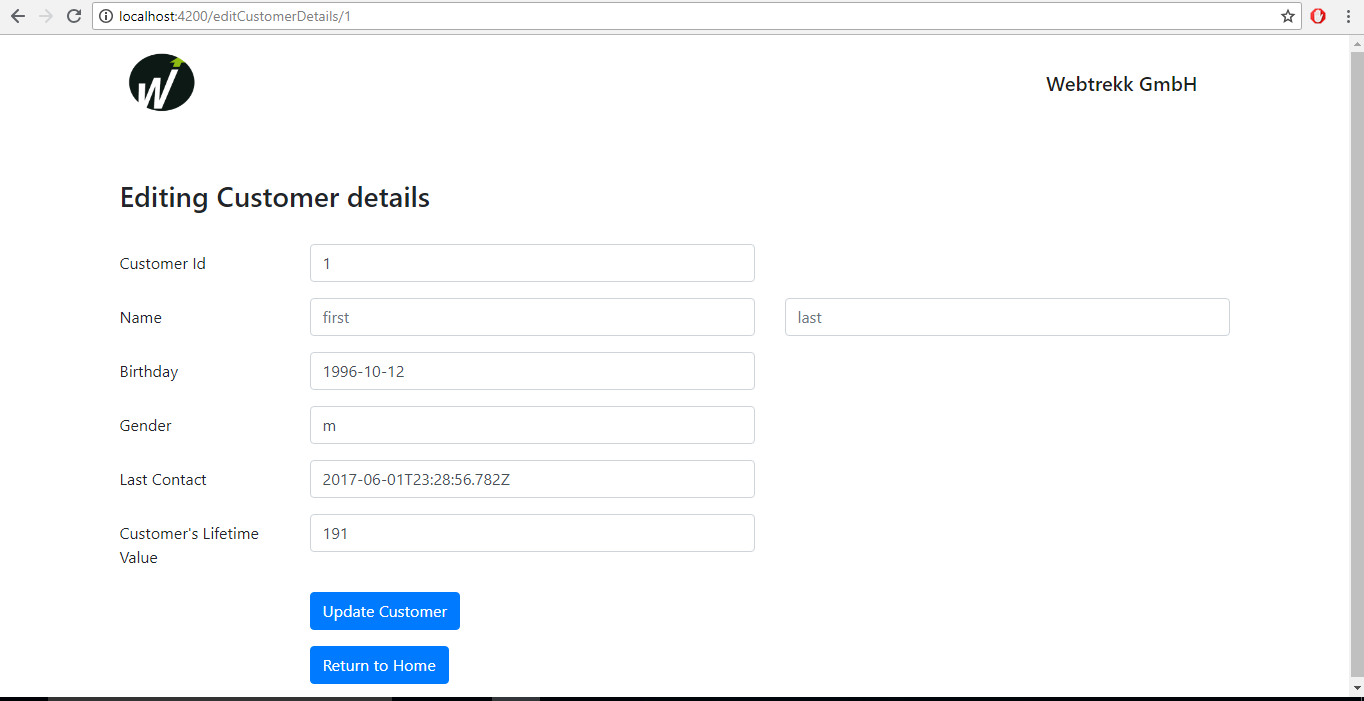
**Common things to note:**

To make a Http calls, I’ve used angular 4 Http, Response, Headers from angular/http along with promises from rxjs.

How our Front end /UI look like ?







**Road map:**

* Hosting the current static repo in gihub.io.
* Setting up the same app with Node + express + Angular 4 - under development

**Challenge:**

* Hosting dynamic app Node + Express + Angular 4 in Heroku.

Thank you