Identity Manager
Web Development with Angular
- Video Series -

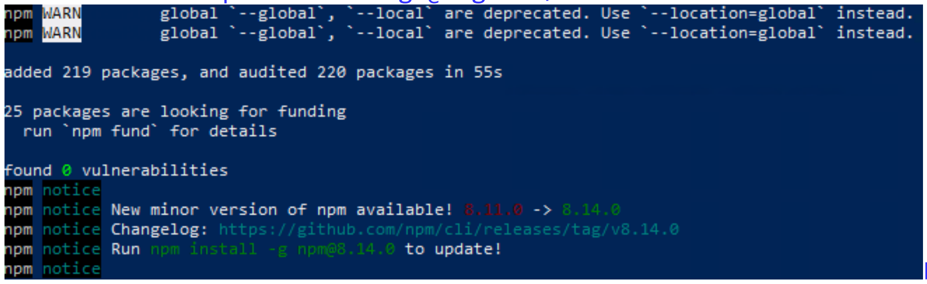
Herwig Abele | Global Enablement Lead IAM
One Identity - Technical Enablement

Table of Contents

Prerequisites and Installation -

MS Windows Workstation 10

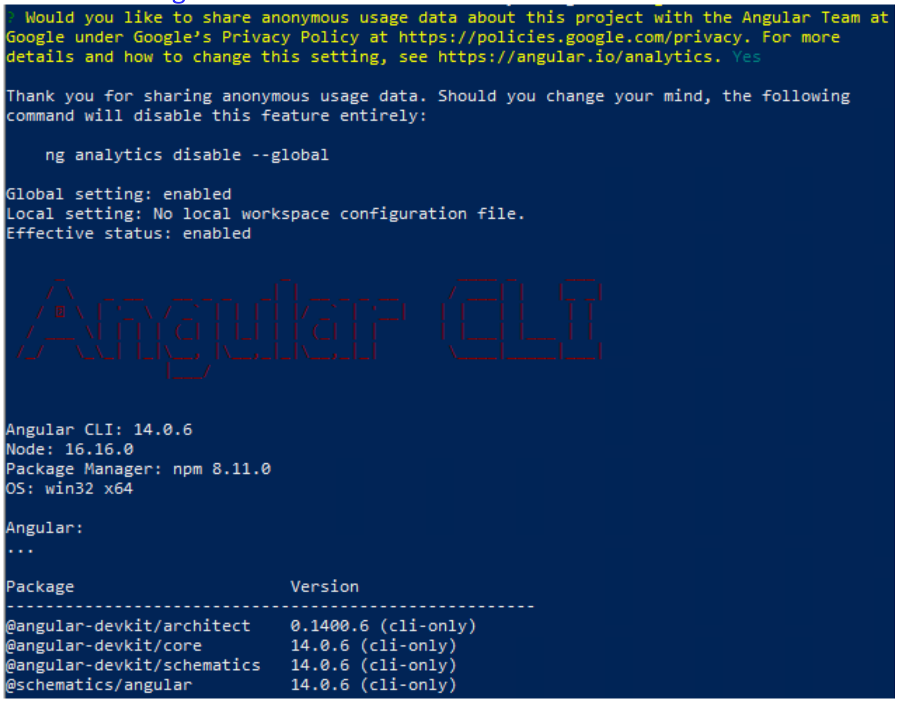
- Visual Studio Code (URL: https://code.visualstudio.com/)
  - `PS> choco install vscode -y`
  - Add plugins
    - Angular Language Service
    - Angular Snippets
    - NX Console
    - HTML CSS Support
    - CSS formatter
- node.js (Angular) (URL: https://nodejs.org/en/)
  - `PS> choco install nodejs-lts -y`
  - `PS> npm -v`
  - `PS> npm install -g @angular/cli`



```
npm WARN        global `--global`, `--local` are deprecated. Use `--location=global` instead.
npm WARN        global `--global`, `--local` are deprecated. Use `--location=global` instead.

added 219 packages, and audited 220 packages in 55s

25 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
npm notice
npm notice New minor version of npm available! 8.11.0 -> 8.14.0
npm notice Changelog: https://github.com/npm/cli/releases/tag/v8.14.0
npm notice Run npm install -g npm@8.14.0 to update!
npm notice
```

P

`PS> ng version`



```
? Would you like to share anonymous usage data about this project 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 https://angular.io/analytics. Yes

Thank you for sharing anonymous usage data. Should you change your mind, the following
command will disable this feature entirely:

    ng analytics disable --global

Global setting: enabled
Local setting: No local workspace configuration file.
Effective status: enabled

Angular CLI: 14.0.6
Node: 16.16.0
Package Manager: npm 8.11.0
OS: win32 x64

Angular:
...

Package                      Version
------------------------------------------------------
@angular-devkit/architect    0.1400.6 (cli-only)
@angular-devkit/core         14.0.6 (cli-only)
@angular-devkit/schematics   14.0.6 (cli-only)
@schematics/angular          14.0.6 (cli-only)
```

Warning: The current version of Node (17.9.0) is not supported by Angular.

- Git

- `PS> choco install git -y`
    - Configure your global username and Password
    - `PS> git config --global user.name “Herwig Abele”`
    - `PS> git config --global user.email “herwig.abele@quest.com”`
- Postman
    - `PS> choco install postman -y`

Linux Mint 20.3

Configure VM-Ware access comunicate with my host

Kernel > 4

```
mkdir /home/[username]/shares
/usr/bin/vmhgfs-fuse .host:/ /home/[username]/shares -o subtype=vmhgfs-fuse,allow_other
```

possible to activate allow other users in /etc/fuse.config

Kernel < 4

```
mkdir /home/[username]/shares
mount -t vmhgfs .host:/ /home/user1/shares
```

Install Angular and development prerequisites

Install node.js and npm

```
sudo apt update
sudo apt install curl build-essential -y
curl -sL https://deb.nodesource.com/setup_18.x | sudo -E bash -
sudo apt install nodejs -y
node --version
npm -v
sudo npm install -g @angular/cli
ng --versiony
```

Install Visual Studio Code

```
sudo apt update
sudo apt -y install wget
sudo apt update
sudo apt install apt-transport-https
curl https://packages.microsoft.com/keys/microsoft.asc | gpg --dearmor > microsoft.gpg
sudo install -o root -g root -m 644 microsoft.gpg /etc/apt/trusted.gpg.d/
sudo sh -c 'echo "deb [arch=amd64] https://packages.microsoft.com/repos/vscode stable
main" > /etc/apt/sources.list.d/vscode.list'
cat  /etc/apt/sources.list.d/vscode.list
deb [arch=amd64] https://packages.microsoft.com/repos/vscode stable main
sudo apt update
sudo apt install code
```

Install Postman

```
sudo rm /etc/apt/preferences.d/nosnap.pref
sudo apt update
sudo apt install snapd -y
sudo snap install postman
```

Install Git

```
sudo apt install git -y
```

Code Development

Create an Angular project

Create a development root if not exist

```
C:\> md %userprofile%\Documents\Angular-Dev
```

```
C:\> cd %userprofile%\Documents\Angular-Dev
```

Protect development root using Git

```
git init
```

Basic Git configuration

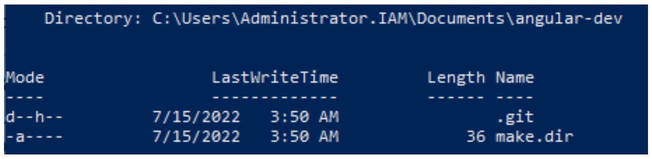
```
git config --local user.name 'Herwig Abele'
git config --local user.email 'Herwig.Abele@OneIdentity.com'
```

Windows (PowerShell) --> show available Git config:

```
dir -Force
```

Linux (Bash):

```
ls -al
```



```
    Directory: C:\Users\Administrator.IAM\Documents\angular-dev


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d--h--         7/15/2022   3:50 AM                .git
-a----         7/15/2022   3:50 AM             36 make.dir
```

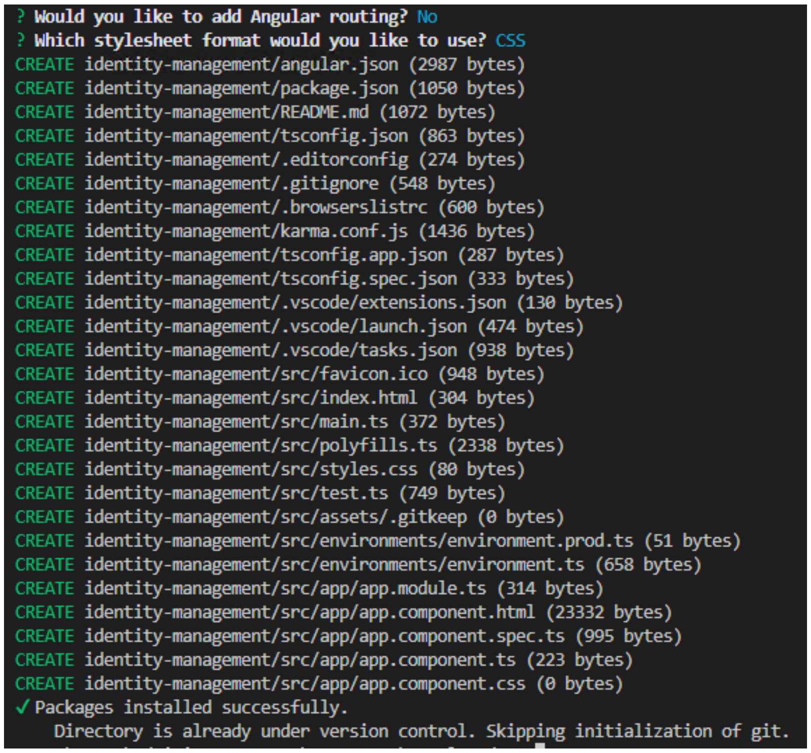
Note: This protects the whole Angular-Dev folder and all of its sub-folders (Angular projects). To protect just one single project a git init needs to happen after the project was created. Therefor the command prompt needs to be located in the project root.

Git:
```
1122248 Initial empty folder for Angular projects -> all protected by one Git repository
1 file changed, 1 insertion(+)
```

Create a new project

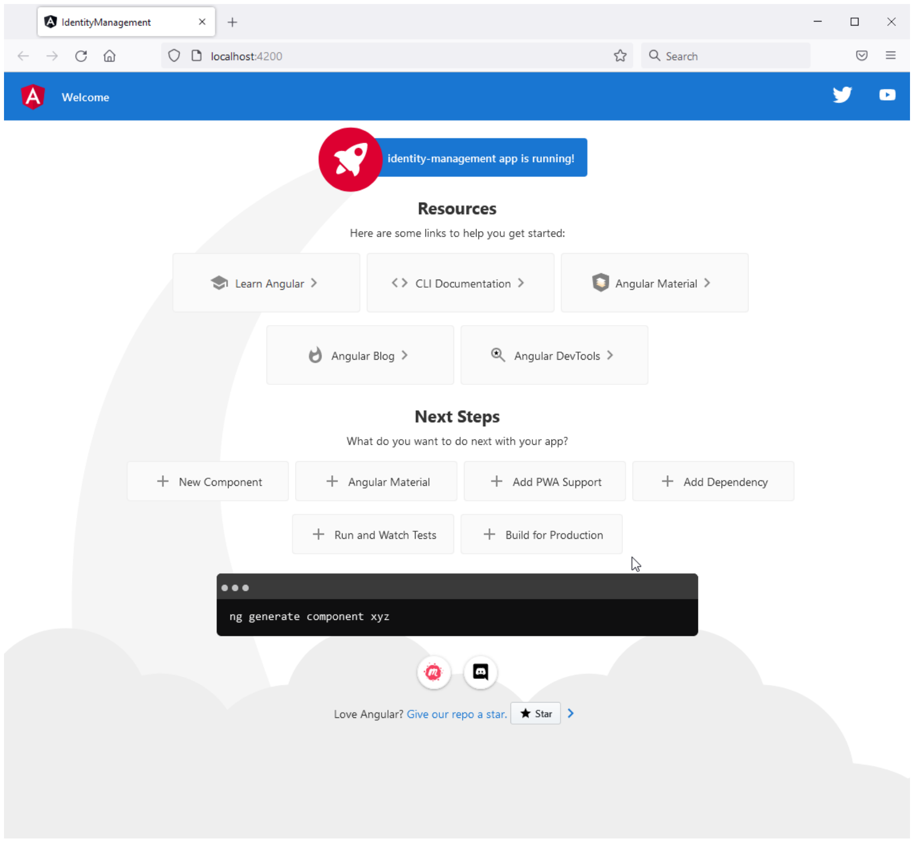
- Open Visual Studio code in `%userprofile%\Documents\Angular-Dev`
  ```
  code .
  ```
- Create a new project
  ```
  ng new identity-management
  ```

Note: We implemented no routing (working with menus) this can be implemented later. We also used the standard CSS style for a layout. This is recommended until you need nothing other else in addition.



```
? Would you like to add Angular routing? No
? Which stylesheet format would you like to use? CSS
CREATE identity-management/angular.json (2987 bytes)
CREATE identity-management/package.json (1050 bytes)
CREATE identity-management/README.md (1072 bytes)
CREATE identity-management/tsconfig.json (863 bytes)
CREATE identity-management/.editorconfig (274 bytes)
CREATE identity-management/.gitignore (548 bytes)
CREATE identity-management/.browserslistrc (600 bytes)
CREATE identity-management/karma.conf.js (1436 bytes)
CREATE identity-management/tsconfig.app.json (287 bytes)
CREATE identity-management/tsconfig.spec.json (333 bytes)
CREATE identity-management/.vscode/extensions.json (130 bytes)
CREATE identity-management/.vscode/launch.json (474 bytes)
CREATE identity-management/.vscode/tasks.json (938 bytes)
CREATE identity-management/src/favicon.ico (948 bytes)
CREATE identity-management/src/index.html (304 bytes)
CREATE identity-management/src/main.ts (372 bytes)
CREATE identity-management/src/polyfills.ts (2338 bytes)
CREATE identity-management/src/styles.css (80 bytes)
CREATE identity-management/src/test.ts (749 bytes)
CREATE identity-management/src/assets/.gitkeep (0 bytes)
CREATE identity-management/src/environments/environment.prod.ts (51 bytes)
CREATE identity-management/src/environments/environment.ts (658 bytes)
CREATE identity-management/src/app/app.module.ts (314 bytes)
CREATE identity-management/src/app/app.component.html (23332 bytes)
CREATE identity-management/src/app/app.component.spec.ts (995 bytes)
CREATE identity-management/src/app/app.component.ts (223 bytes)
CREATE identity-management/src/app/app.component.css (0 bytes)
✓ Packages installed successfully.
    Directory is already under version control. Skipping initialization of git.
```

- Commit the empty project
  `git commit -m “New Angular project”`
- Start the new empty Angular project
  `ng serve --open`



Git:
```
30b5e4a Empty Angular project IDENTITY-MANAGEMENT
28 files changed, 21959 insertions(+)
```

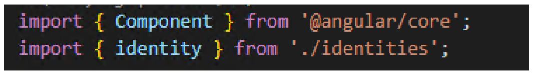
`*** End of videos 1.1-3 ***`

Add and display some records

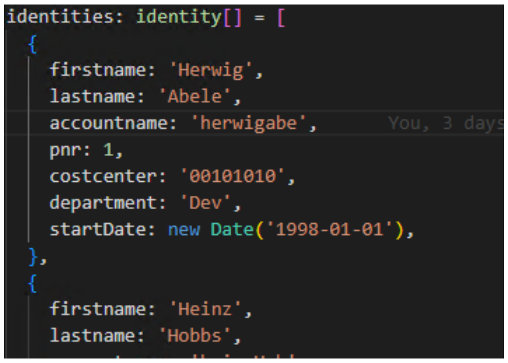
- Delete all from file: ***app.component.html***
- Have a look at the Angular HOT-RELOAD in the served browser window, saving files (recognize the Browser URL: "http://localhost:4200/")
- Insert {{title}} and "Hello World" into the ***app.component.html***
- Misspell the name {{title1}} and start the console Window in your Browser (Ctrl+Shift+J)
- Create a new file for a new data type called ***identities.ts*** to declare the data structure of an identity

Git:
```
9fde8ec (HEAD -> master) Create an interface for an identity - data structure we need
3 files changed, 20 insertions(+), 483 deletions(-)
```

- Link the new ***identities.ts*** in ***app.components.ts*** and create a sample data (Identity array → identities)

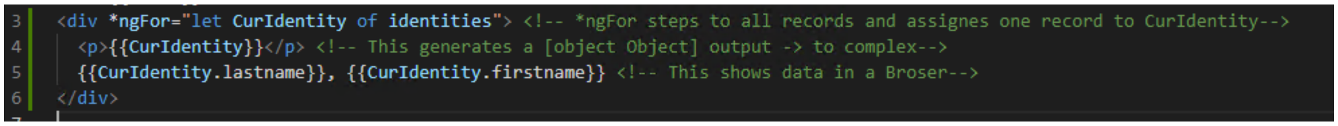


```
import { Component } from '@angular/core';
import { identity } from './identities';
```



```
identities: identity[] = [
  {
    firstname: 'Herwig',
    lastname: 'Abele',
    accountname: 'herwigabe',         You, 3 days
    pnr: 1,
    costcenter: '00101010',
    department: 'Dev',
    startDate: new Date('1998-01-01'),
  },
  {
    firstname: 'Heinz',
    lastname: 'Hobbs',
```

- Show parts of the entries in a Browser (***app.components.html***. If you try to show the whole array you will get [object Object] – warning outlining that the data is to complex for a simple output.
- In ***app.components.html*** define a `*ngfor` loop to show data properties.



```
<div *ngFor="let CurIdentity of identities"> <!-- *ngFor steps to all records and assignes one record to CurIdentity-->
  <p>{{CurIdentity}}</p> <!-- This generates a [object Object] output -> to complex-->
  {{CurIdentity.lastname}}, {{CurIdentity.firstname}} <!-- This shows data in a Broser-->
</div>
```

Git:
```
09a8b01  Create and display identity sample data
4 files changed, 26 insertions(+), 1 deletion(-)
```

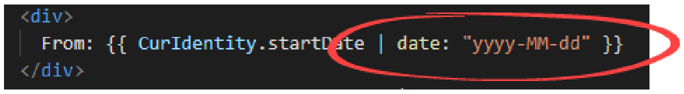
Format the list using Bootstrap

Note: In an **Identity Manager – Standard web portal** all layout is done using **"Elemental UI"** (by One Identity) based on a layout framework **"Angular material"** (by Google). We will show this detailed in later parts of our Angular Training.
For this, first baby steps we use another framework called **"Bootstrap"** which is another, less complex framework, every web-developer should have recognized.

- Integrate Bootstrap as a CSS layout:
  https://getbootstrap.com/
- From Home page integrate link and script into the globale ***index.html***. This makes Bootstrap available for all Angular apps in this Dev-space.
- Format the sample data using
  - Card: https://getbootstrap.com/docs/5.2/components/card/#about
  - Grid: https://getbootstrap.com/docs/5.2/layout/grid/#example
- **Opportunities to display date values**

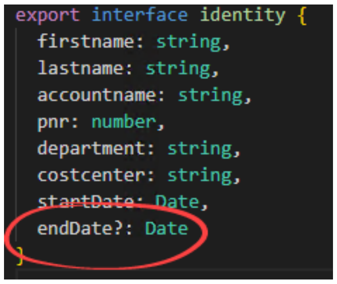
Note: Trying to display the date values, it is an opportunity to show especially date values not set. This is a standard business behavior for unlimited employee contracts. To handle this problem (you will not see all entries, because displaying is broken for undefined values), some data changes are needed.

- Format the date using a pipe format https://angular.io/api/common/DatePipe



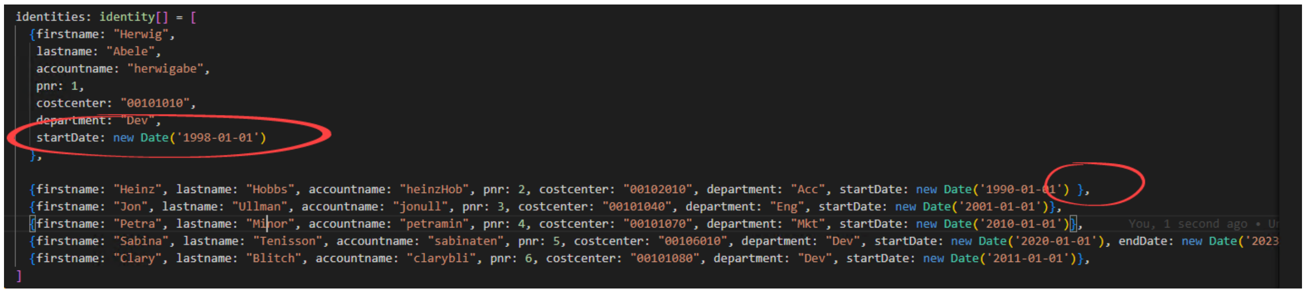
```
<div>
  From: {{ CurIdentity.startDate | date: "yyyy-MM-dd" }}
</div>
```

- Change interface in ***identities.ts*** making the ***endDate*** property optional. In an interface the **"?"** can be used. This effects an interface as well (interfaces does have all properties mandatory).
  Optional with "?" →



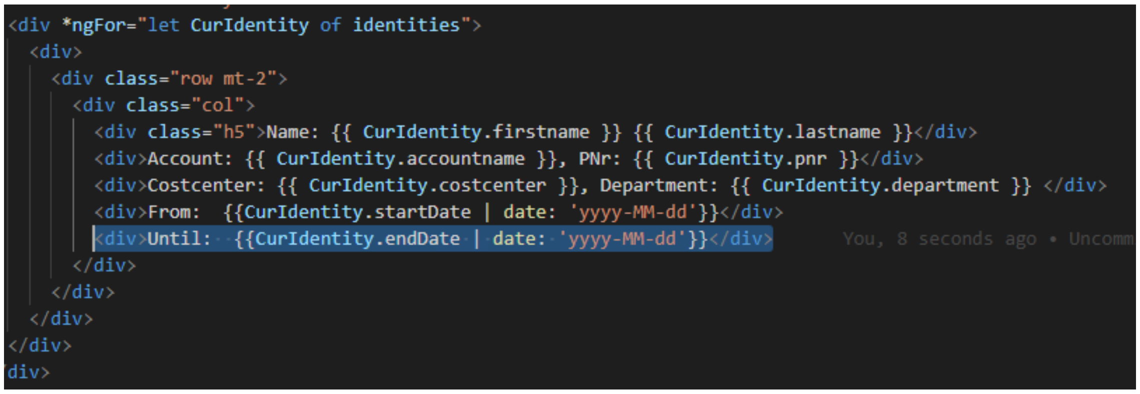
```
export interface identity {
  firstname: string,
  lastname: string,
  accountname: string,
  pnr: number,
  department: string,
  costcenter: string,
  startDate: Date,
  endDate?: Date
}
```

- Removing all endDate='' values in the Data source (***app.component.ts***).

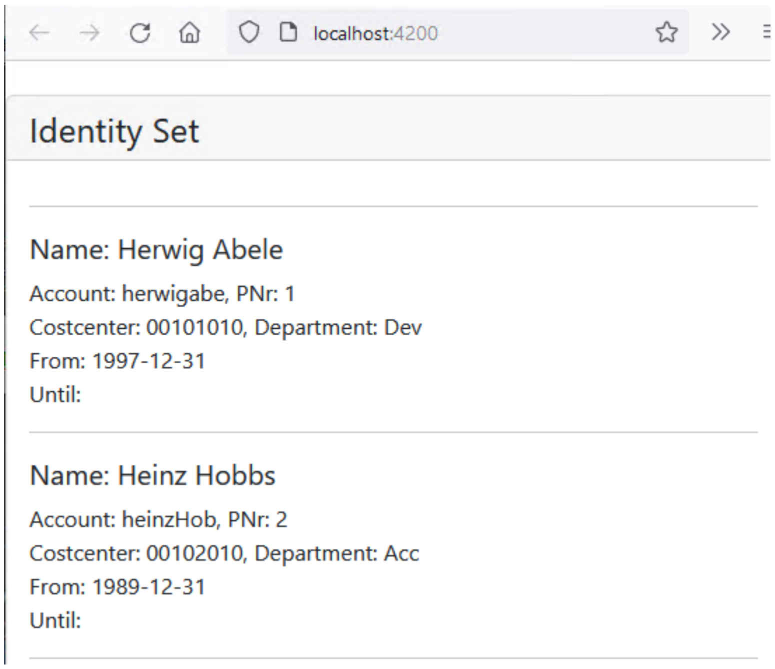


```
identities: identity[] = [
  {firstname: "Herwig",
   lastname: "Abele",
   accountname: "herwigabe",
   pnr: 1,
   costcenter: "00101010",
   department: "Dev",
   startDate: new Date('1998-01-01')
  },

  {firstname: "Heinz", lastname: "Hobbs", accountname: "heinzHob", pnr: 2, costcenter: "00102010", department: "Acc", startDate: new Date('1990-01-01') },
  {firstname: "Jon", lastname: "Ullman", accountname: "jonull", pnr: 3, costcenter: "00101040", department: "Eng", startDate: new Date('2001-01-01')},
  {firstname: "Petra", lastname: "Mihor", accountname: "petramin", pnr: 4, costcenter: "00101070", department: "Mkt", startDate: new Date('2010-01-01')},        You, 1 second ago • Ur
  {firstname: "Sabina", lastname: "Tenisson", accountname: "sabinaten", pnr: 5, costcenter: "00106010", department: "Dev", startDate: new Date('2020-01-01'), endDate: new Date('2023
  {firstname: "Clary", lastname: "Blitch", accountname: "clarybli", pnr: 6, costcenter: "00101080", department: "Dev", startDate: new Date('2011-01-01')},
]
```

- Formatting date values in ***app.component.html*** using the pipe formatter



```
<div *ngFor="let CurIdentity of identities">
  <div>
    <div class="row mt-2">
      <div class="col">
        <div class="h5">Name: {{ CurIdentity.firstname }} {{ CurIdentity.lastname }}</div>
        <div>Account: {{ CurIdentity.accountname }}, PNr: {{ CurIdentity.pnr }}</div>
        <div>Costcenter: {{ CurIdentity.costcenter }}, Department: {{ CurIdentity.department }} </div>
        <div>From:  {{CurIdentity.startDate | date: 'yyyy-MM-dd'}}</div>
        <div>Until:  {{CurIdentity.endDate | date: 'yyyy-MM-dd'}}</div>        You, 8 seconds ago • Uncomm
      </div>
    </div>
  </div>
</div>
div>
```


localhost:4200

Identity Set

Name: Herwig Abele
Account: herwigabe, PNr: 1
Costcenter: 00101010, Department: Dev
From: 1997-12-31
Until:

Name: Heinz Hobbs
Account: heinzHob, PNr: 2
Costcenter: 00102010, Department: Acc
From: 1989-12-31
Until:

Something similar to this...

Git:
```
3b6e1f5  Integrating Bootstrap and formatting Date values
 5 files changed, 28 insertions(+), 14 deletions(-)
```

Implement a button to delete list entries

- Define a button with a click – event in ***app.component.html***

```
<div class="col">
  <button class = "btn btn-danger btn-sn" (click)="deleteIdentity(CurIdentity)">x Entfernen</button>
</div>
```

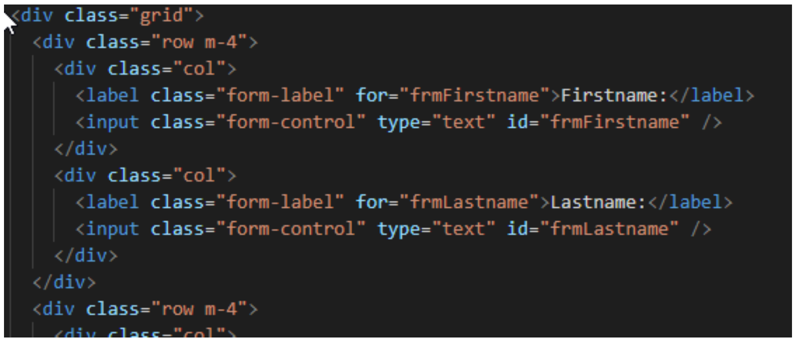
- In ***app.component.ts*** define a function to filter the current entry out of the collection

```
deleteIdentity(myIdentity: identity): void {
  this.identities = this.identities.filter(i => i != myIdentity)
}
```

Git:
```
96474b5 Implement a button to delete single list entries of the sample data set
 3 files changed, 14 insertions(+)
```

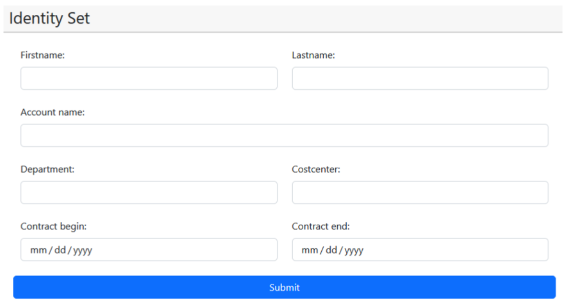
Implement input fields to create a new list entry

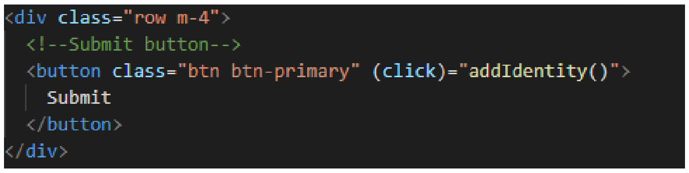
- Add FormModule (to work with forms) to ***app.module.ts***
- Create a form in ***app.component.html***
  https://getbootstrap.com/docs/5.2/forms/overview/#overview



```html
<div class="grid">
  <div class="row m-4">
    <div class="col">
      <label class="form-label" for="frmFirstname">Firstname:</label>
      <input class="form-control" type="text" id="frmFirstname" />
    </div>
    <div class="col">
      <label class="form-label" for="frmLastname">Lastname:</label>
      <input class="form-control" type="text" id="frmLastname" />
    </div>
  </div>
  <div class="row m-4">
    <div class="col">
```

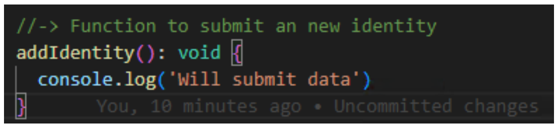
- Create a button to submit changes in ***app.component.html***





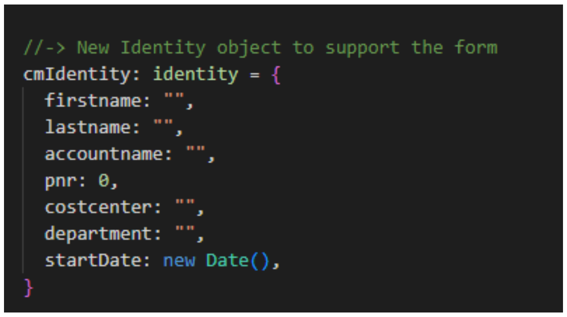
```html
<div class="row m-4">
  <!--Submit button-->
  <button class="btn btn-primary" (click)="addIdentity()">
    Submit
  </button>
</div>
```

- Create a new function in ***app.component.ts*** to store data objects to be implemented below



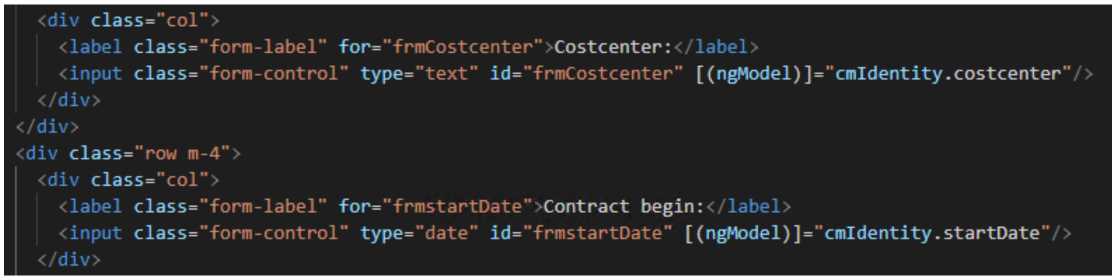
```typescript
//-> Function to submit an new identity
addIdentity(): void {
  console.log('Will submit data')
}       You, 10 minutes ago • Uncommitted changes
```

- Create a new data object in ***app.components.ts***



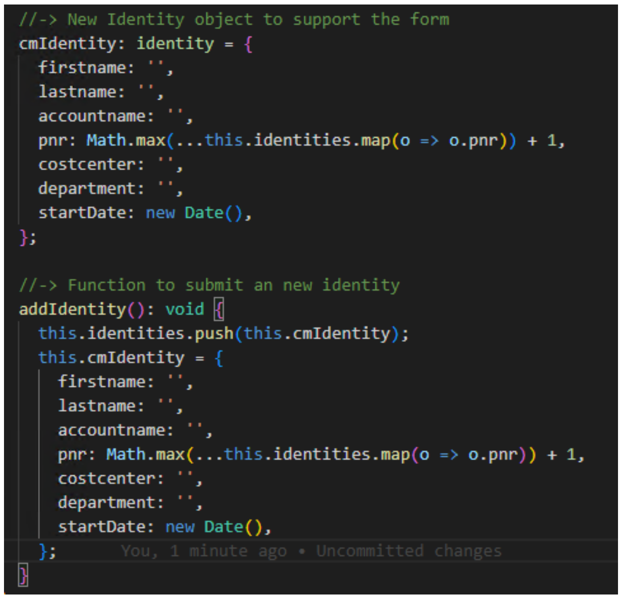
```typescript
//-> New Identity object to support the form
cmIdentity: identity = {
  firstname: "",
  lastname: "",
  accountname: "",
  pnr: 0,
  costcenter: "",
  department: "",
  startDate: new Date(),
}
```

- Bind the data object to form in ***app.components.html***



```html
  <div class="col">
    <label class="form-label" for="frmCostcenter">Costcenter:</label>
    <input class="form-control" type="text" id="frmCostcenter" [(ngModel)]="cmIdentity.costcenter"/>
  </div>
</div>
<div class="row m-4">
  <div class="col">
    <label class="form-label" for="frmstartDate">Contract begin:</label>
    <input class="form-control" type="date" id="frmstartDate" [(ngModel)]="cmIdentity.startDate"/>
  </div>
```

- In ***app.component.ts*** implement function to store data into data set
- Update ***cmidentity*** and ***addIdentity()*** in ***app.component.ts*** to create a new pnr as max of all pnr
  https://stackoverflow.com/questions/4020796/finding-the-max-value-of-an-attribute-in-an-array-of-objects



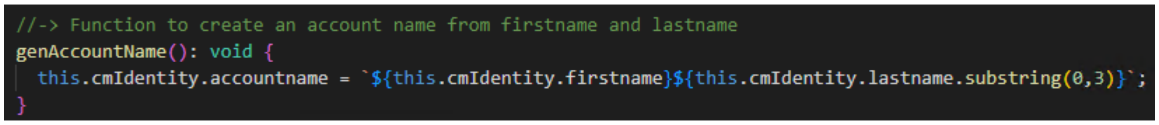
```
//-> New Identity object to support the form
cmIdentity: identity = {
  firstname: '',
  lastname: '',
  accountname: '',
  pnr: Math.max(...this.identities.map(o => o.pnr)) + 1,
  costcenter: '',
  department: '',
  startDate: new Date(),
};

//-> Function to submit an new identity
addIdentity(): void {
  this.identities.push(this.cmIdentity);
  this.cmIdentity = {
    firstname: '',
    lastname: '',
    accountname: '',
    pnr: Math.max(...this.identities.map(o => o.pnr)) + 1,
    costcenter: '',
    department: '',
    startDate: new Date(),
  };       You, 1 minute ago • Uncommitted changes
}
```

Git:
```
9b97a0 Form to insert new identity records implemented
 4 files changed, 176 insertions(+), 34 deletions(-)
```

Create an account name out of Firstname and Lastname

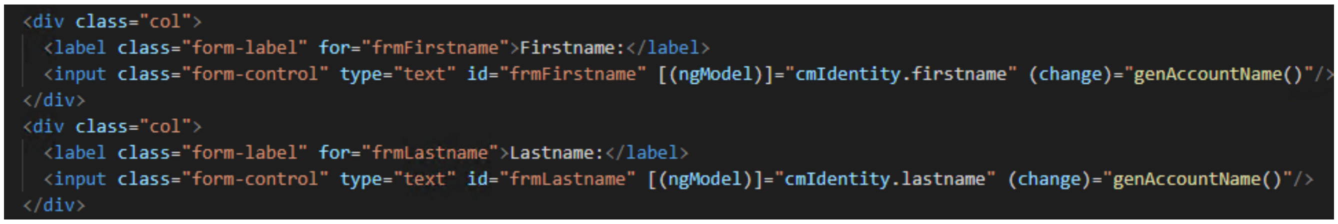
Note: This is an exercise similar to an Identity Manager data template. The functions needs to build the account name from firstname + first 3 char of a lastname . This is a change happens to the ***cmIdentity*** object account name.

- Create a new function in ***app.component.ts***



```
//-> Function to create an account name from firstname and lastname
genAccountName(): void {
  this.cmIdentity.accountname = `${this.cmIdentity.firstname}${this.cmIdentity.lastname.substring(0,3)}`;
}
```

- Call the function by changes in firstname, lastname in ***app.component.html***



```
<div class="col">
  <label class="form-label" for="frmFirstname">Firstname:</label>
  <input class="form-control" type="text" id="frmFirstname" [(ngModel)]="cmIdentity.firstname" (change)="genAccountName()"/
</div>
<div class="col">
  <label class="form-label" for="frmLastname">Lastname:</label>
  <input class="form-control" type="text" id="frmLastname" [(ngModel)]="cmIdentity.lastname" (change)="genAccountName()"/>
</div>
```

Git:
```
a200ff4 Template to account name creation (from first and lastname) added
3 files changed, 9 insertions(+), 3 deletions(-)
```

***** End of videos 2.1-x *****

Add a REST based Web Service

-

Note: This is the end of all Angular code development we show in the official “Identity Manager - Angular Web Development” video series (on YouTube):

The purpose of this lectures was to show how Angular as a development Framework works by sample. This is of course not sufficient enough to learn, how to work with Angular. Therefore at least an Angular course and some experience is needed before a new developer could/should try to understand and to modify the heavily more complex ***One Identity – Identity Manager – Standard Web Application, Angular based***.

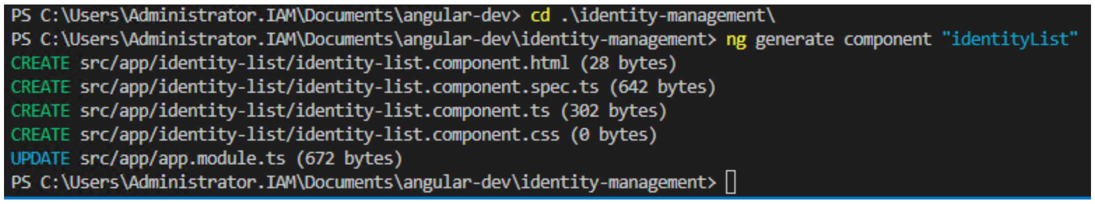
Nevertheless, in the following steps we will add more functionality to this small exercise which lives only in this zipped git project. This means, students can try to follow the steps using the One Identity - Technical Enablement Git repository from Github which will be continuously extend after the video course from time to time. This will show some more native TypeScript and Angular techniques.

Further TypeScript / Angular code development

Separate listing and creation of identities using components

Note: To handle all functionality in just one application/module is not common. It was helpful to lower the complexity during the first steps but now we like to:
- Separate listing and creation of identities using components.
- Insert routing and a menu.
- Allow changing of available identity records.

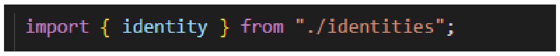
- Create a new component identityList



```
PS C:\Users\Administrator.IAM\Documents\angular-dev> cd .\identity-management\
PS C:\Users\Administrator.IAM\Documents\angular-dev\identity-management> ng generate component "identityList"
CREATE src/app/identity-list/identity-list.component.html (28 bytes)
CREATE src/app/identity-list/identity-list.component.spec.ts (642 bytes)
CREATE src/app/identity-list/identity-list.component.ts (302 bytes)
CREATE src/app/identity-list/identity-list.component.css (0 bytes)
UPDATE src/app/app.module.ts (672 bytes)
PS C:\Users\Administrator.IAM\Documents\angular-dev\identity-management> 
```

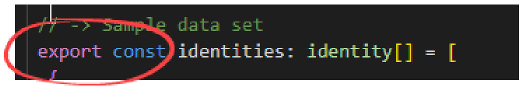
- Move display of records from ***app.component.html*** to ***identity-list.component.html***

- Create a ***new mock-identities.ts*** document in folder app.
  - Import the identity object into ***mock-identities.ts***



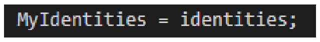
```
import { identity } from "./identities";
```

  - Move the Identity array from ***app.components.ts*** to ***mock-identities.ts***
  - Extend the array by exporting it as a data object (const)



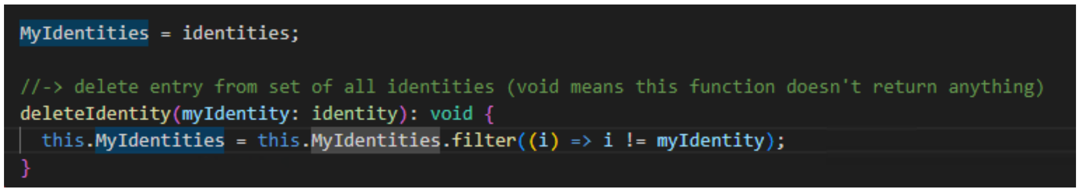
```
// -> Sample data set
export const identities: identity[] = [
  {
```

- Import identities in ***identity-list.component.ts***
- create an object “MyIdentities” and assign the imported identities.



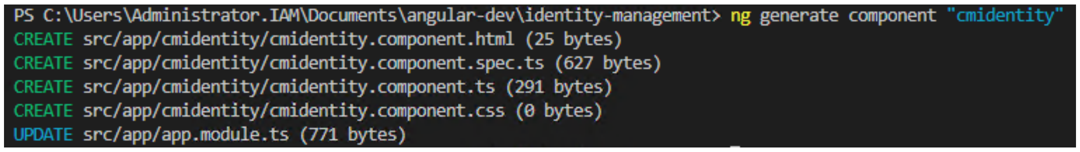
```
MyIdentities = identities;
```

- Move function used to delete records to ***identity-list.component.ts***. You might need to change “identities” in the function references to “MyIdentities”. The “MyI” is helpful to separate the local variable from the imported data object.



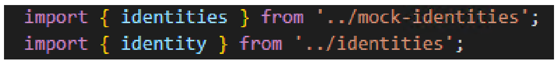
```
MyIdentities = identities;

//-> delete entry from set of all identities (void means this function doesn't return anything)
deleteIdentity(myIdentity: identity): void {
  this.MyIdentities = this.MyIdentities.filter((i) => i != myIdentity);
}
```

- Create a new component cmIdentity. This is for creating and modifying of an identity object.



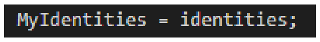
```
PS C:\Users\Administrator.IAM\Documents\angular-dev\identity-management> ng generate component "cmidentity"
CREATE src/app/cmidentity/cmidentity.component.html (25 bytes)
CREATE src/app/cmidentity/cmidentity.component.spec.ts (627 bytes)
CREATE src/app/cmidentity/cmidentity.component.ts (291 bytes)
CREATE src/app/cmidentity/cmidentity.component.css (0 bytes)
UPDATE src/app/app.module.ts (771 bytes)
```

- Into the new ***cmidentity.component.ts***
  - Import identity from identities
  - Import identities from mock-identities



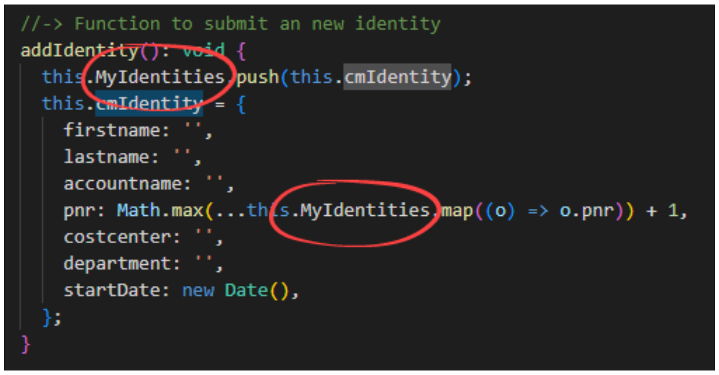
```
import { identities } from '../mock-identities';
import { identity } from '../identities';
```

  - Create a "MyIdentities" variable and assign identities.



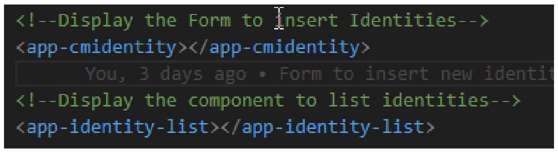
```
MyIdentities = identities;
```

  - Copy all content from ***app.component.ts*** which is related to adding an identity record (cmidentity..., GetIdentity()..., GenAccountName()…)
  - Replace references to "identities" by "MyIdentities"



```
//-> Function to submit an new identity
addIdentity(): void {
  this.MyIdentities.push(this.cmIdentity);
  this.cmIdentity = {
    firstname: '',
    lastname: '',
    accountname: '',
    pnr: Math.max(...this.MyIdentities.map((o) => o.pnr)) + 1,
    costcenter: '',
    department: '',
    startDate: new Date(),
  };
}
```

- From ***app.component.html*** copy the complete form section into the ***cmidentity.component.html***
- Reference the new selector "app-cmidentity" as a new a root selector in app.component.html



```
<!--Display the Form to Insert Identities-->
<app-cmidentity></app-cmidentity>
      You, 3 days ago • Form to insert new identit
<!--Display the component to list identities-->
<app-identity-list></app-identity-list>
```

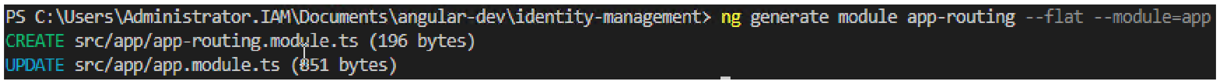
- After solving all errors you should see exactly the picture you started with!

Note: Of course this was a lot of work taking no visible effect. In this part of the lecture we separated the single application in different components and we moved the mock-data to a separate file we are able to include in these components. This was a big step into the direction of big applications.

Git:
```
eb365f7 Separated the application from listing and creating objects. Separation of moc-data.
 13 files changed, 287 insertions(+), 187 deletions(-)
```

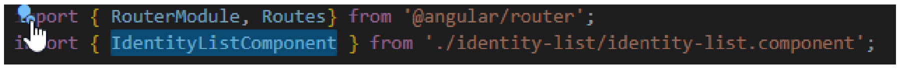
Insert routing and a menu and more Layout

- Add a flat routing module to the application. **--Flat** adds a ".ts" file directly in the root of the application. The flag **--module=app** inserts the new module directly into the app.module.ts



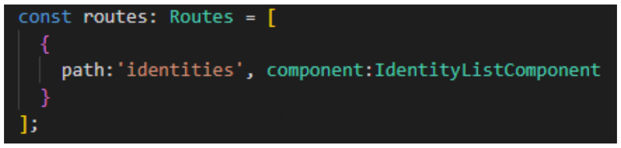
```
PS C:\Users\Administrator.IAM\Documents\angular-dev\identity-management> ng generate module app-routing --flat --module=app
CREATE src/app/app-routing.module.ts (196 bytes)
UPDATE src/app/app.module.ts (851 bytes)
```

- In ***app-routing.module.ts***
  - delete the CommonModule reference – not needed.
  - Import and initiate RouterModule
  - Import Routs
  - Import IdentityListComponent



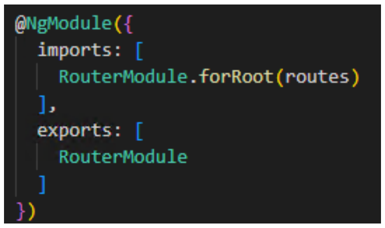
```
import { RouterModule, Routes} from '@angular/router';
import { IdentityListComponent } from './identity-list/identity-list.component';
```

- In ***app-routing.module.ts*** insert a new Routes array object
  - Add one route with path and related component



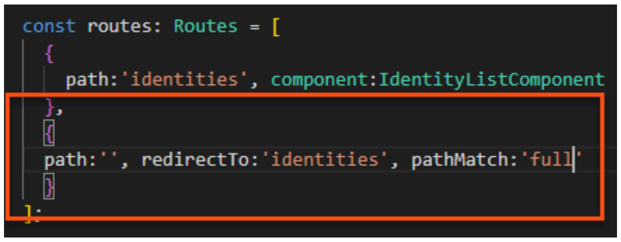
```
const routes: Routes = [
  {
    path:'identities', component:IdentityListComponent
  }
];
```

  - Initiate the RouterModule (imports) – (use forRoot for a menu in the app root)
  - Make the RouterModule available (export)
  - Delete the declarations (not needed)



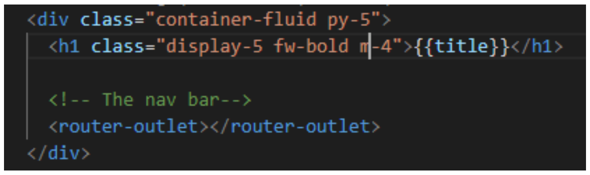
```
@NgModule({
  imports: [
    RouterModule.forRoot(routes)
  ],
  exports: [
    RouterModule
  ]
})
```

- In ***app.component.html*** replace the app-cmidentity and pp-identity-list component tags by on single router-outlet tag.
- To see a preview you need to use URL: http://localhost:4200/identities
- To allow http://localhost:4200 to redirect to URL: http://localhost:4200/identities (default start page) add a redirector in ***app-routing.module.ts*** (another extra routes entry.



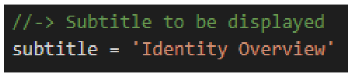
```
const routes: Routes = [
  {
    path:'identities', component:IdentityListComponent
  },
  {
  path:'', redirectTo:'identities', pathMatch:'full'
  }
];
```

- In ***app.component.html*** insert from bootstrap, parts of the "Jumbotron" template https://getbootstrap.com/docs/5.2/examples/jumbotron/
  - Ensure the app header is displayed
  - Ensure the router is included into the outer div.
  - Add m-4 to the title



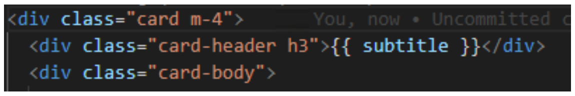
```
<div class="container-fluid py-5">
  <h1 class="display-5 fw-bold m-4">{{title}}</h1>

  <!-- The nav bar-->
  <router-outlet></router-outlet>
</div>
```

- In ***identity-list.component.ts*** enter a variable for a subtitle



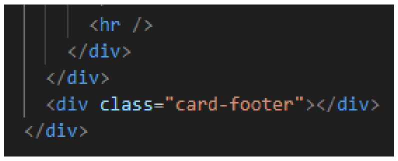
```
//-> Subtitle to be displayed
subtitle = 'Identity Overview'
```

- In ***identity-list.component.htm***l
  - Change the card class mt-4 to m-4 (this displays a border around the whole module)
  - Change the header using the new defined subtitle variable
  - Move the horizontal ruler downward to ensure it gets displayed after each entry
  - Add a card footer



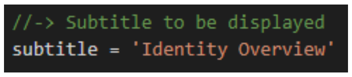
```
<div class="card m-4">      You, now • Uncommitted c
  <div class="card-header h3">{{ subtitle }}</div>
  <div class="card-body">
```

…



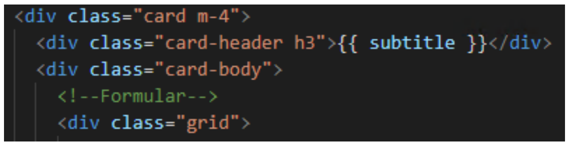
```
        <hr />
      </div>
    </div>
    <div class="card-footer"></div>
  </div>
```

- In ***cmidentity.component.ts*** enter a variable for a subtitle



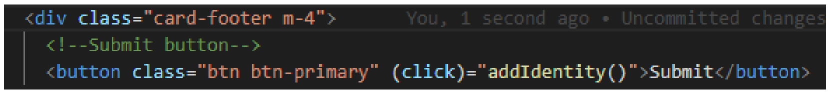
```
//-> Subtitle to be displayed
subtitle = 'Identity Overview'
```

- In ***identity-list.component.html***
  - Enter div with class card and a header similar to ***list-identity.component.ts***
  - Change the header using the new defined subtitle variable
  - Change the div containing the submit button to a footer

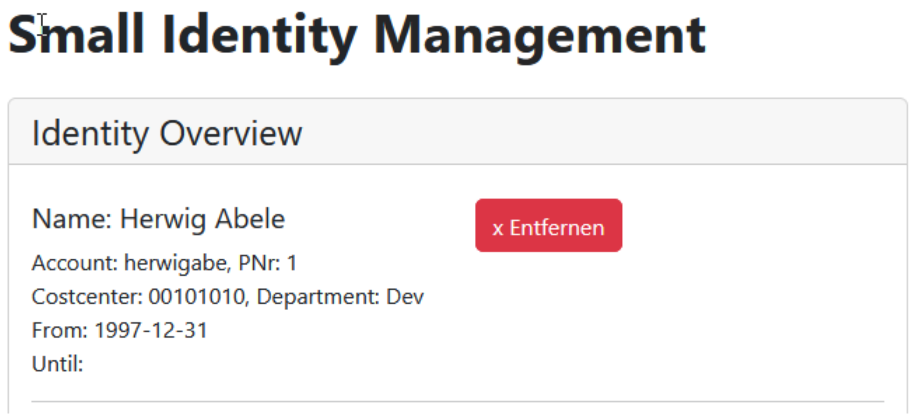


```
<div class="card m-4">
  <div class="card-header h3">{{ subtitle }}</div>
  <div class="card-body">
    <!--Formular-->
    <div class="grid">
```

…



```
<div class="card-footer m-4">      You, 1 second ago • Uncommitted changes
  <!--Submit button-->
  <button class="btn btn-primary" (click)="addIdentity()">Submit</button>
```



Small Identity Management

Identity Overview

Name: Herwig Abele

Account: herwigabe, PNr: 1

Costcenter: 00101010, Department: Dev

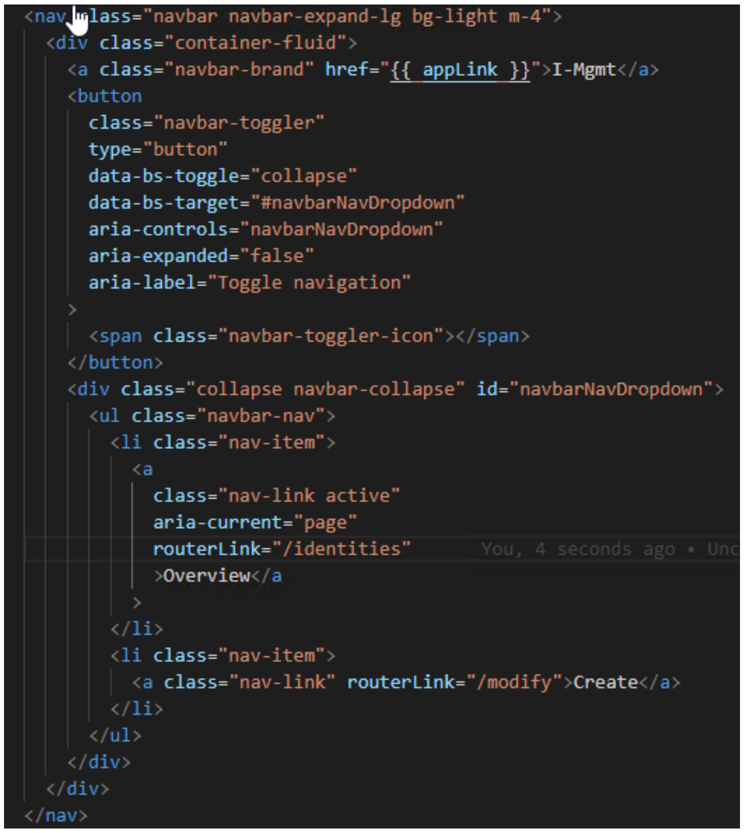
From: 1997-12-31

Until:

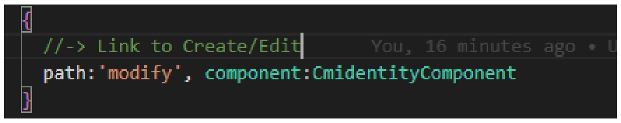
Git:
```
7041d60 Insert routing and a menu and more Layout
 8 files changed, 133 insertions(+), 58 deletions(-)
```

Add navbar to visualize the menu

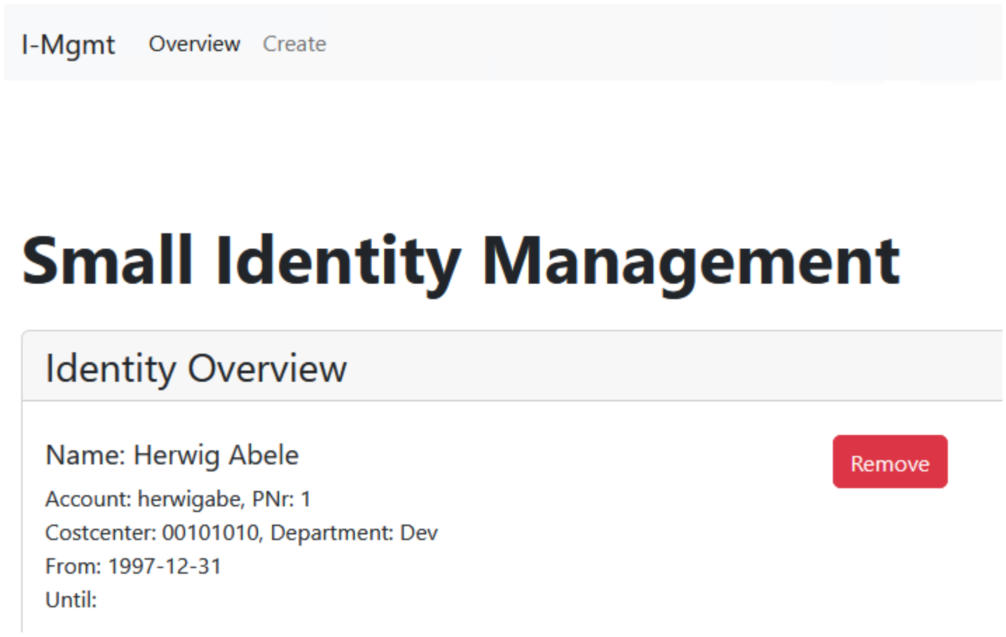
- Minor correction: In ***identity-list.component.html*** change button name to "Remove" (old bug to be solved).
- Open the Bootstrap documentation and search for navbar https://getbootstrap.com/docs/5.2/components/navbar/#how-it-works
- Copy code for a navbar you like into ***app.component.html*** (above all other code)
- In ***app.component.html***
  - Delete components you don't need (all beside two standard menu entries)
  - Add "m-4" to the top class
  - Rename "navbar" to something like "I-Mgmt"
  - Configure navbar href to {{appLink}}
  - Rename first navigation entry to "Overview" and the second navmenu to "Modify"
  - Replace Overview href with routerLink="/identities"
  - Replace Modify href with routerLink="/modify"


```
<nav class="navbar navbar-expand-lg bg-light m-4">
  <div class="container-fluid">
    <a class="navbar-brand" href="{{ appLink }}">I-Mgmt</a>
    <button
      class="navbar-toggler"
      type="button"
      data-bs-toggle="collapse"
      data-bs-target="#navbarNavDropdown"
      aria-controls="navbarNavDropdown"
      aria-expanded="false"
      aria-label="Toggle navigation"
    >
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarNavDropdown">
      <ul class="navbar-nav">
        <li class="nav-item">
          <a
            class="nav-link active"
            aria-current="page"
            routerLink="/identities"        You, 4 seconds ago • Unc
            >Overview</a
          >
        </li>
        <li class="nav-item">
          <a class="nav-link" routerLink="/modify">Create</a>
        </li>
      </ul>
    </div>
  </div>
</nav>
```

- In app.component.ts add a variable appLink="http://localhost:4200"
- In app-routing.module.ts add an entry for modify page



```
{
  //-> Link to Create/Edit        You, 16 minutes ago • U
  path:'modify', component:CmidentityComponent
}
```



Small Identity Management

Identity Overview

Name: Herwig Abele

Account: herwigabe, PNr: 1

Costcenter: 00101010, Department: Dev

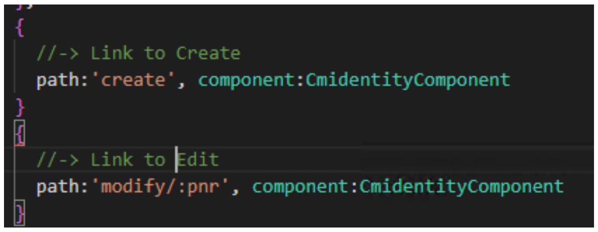
From: 1997-12-31

Until:

Git:
```
2fd4c25 Add navbar to visualize the menu
 5 files changed, 47 insertions(+), 10 deletions(-)
```

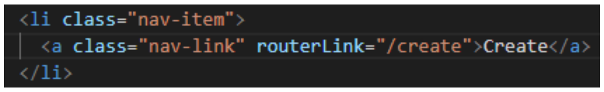
Allow modifying of available identity records

- In ***app-routing.modules.ts***
- Rename the route path "modify" to "create"
- Add a new route "modify" allowing a rout parameter "/:pnr" (to edit a record we need to identify the record by a unique identifier which is our personnel number. As component we use the same component as for creating a record (form))



```
{
  //-> Link to Create
  path:'create', component:CmidentityComponent
}
{
  //-> Link to Edit
  path:'modify/:pnr', component:CmidentityComponent
}
```

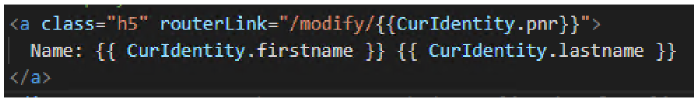
- In ***app.component.html*** replace the "modify" path in the "create" menu with path "create"



```
<li class="nav-item">
  <a class="nav-link" routerLink="/create">Create</a>
</li>
```

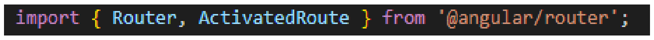
- In ***identity-list.component.html***
  - Find the div tag wraps firstname and lastname and switches "div" to "a". The a tag makes it clickable.

- Add a routerLink for modify and parameter pnr and use an interpolation binding to assign the current selected personnel number {{CurIdentity.pnr}}



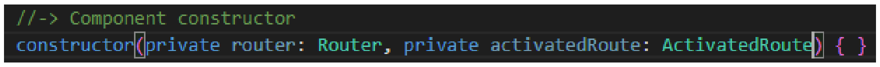
```
<a class="h5" routerLink="/modify/{{CurIdentity.pnr}}">
  Name: {{ CurIdentity.firstname }} {{ CurIdentity.lastname }}
</a>
```

- To load the right record into the form add to ***cmidentity.component.ts***
  - Import for Router and ActivatedRoute



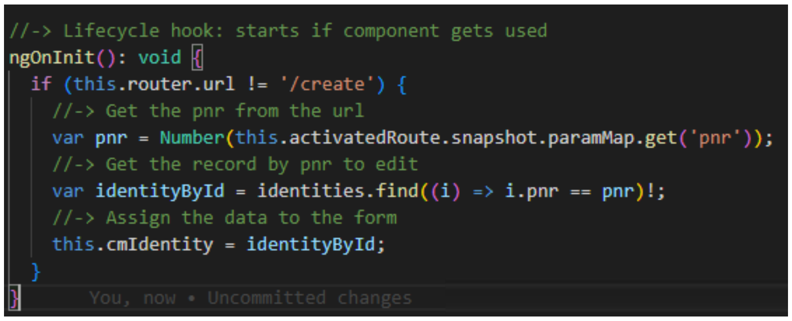
```
import { Router, ActivatedRoute } from '@angular/router';
```

  - Initiate these in the constructor. This time we will use “private” because its only used in this module.



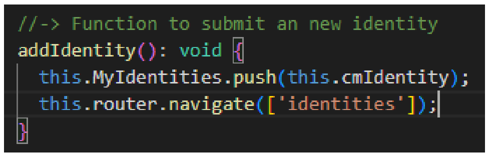
```
//-> Component constructor
constructor(private router: Router, private activatedRoute: ActivatedRoute) { }
```

  - Add to ngOnInit() a variable “pnr” getting parameter pnr from the activated route. Ensure the returned value is a number.
  - Add another variable get the record with this pnr. Add a “Non-null assertion operator “ => ! to ensure the value can’t be null.
  - Assign this record to cmIdentity
  - Ensure that all of this only happens for edited records and not for created



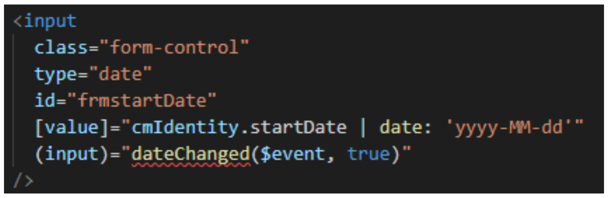
```
//-> Lifecycle hook: starts if component gets used
ngOnInit(): void {
  if (this.router.url != '/create') {
    //-> Get the pnr from the url
    var pnr = Number(this.activatedRoute.snapshot.paramMap.get('pnr'));
    //-> Get the record by pnr to edit
    var identityById = identities.find((i) => i.pnr == pnr)!;
    //-> Assign the data to the form
    this.cmIdentity = identityById;
  }
}       You, now • Uncommitted changes
```

- Ensure the Form jumps back to the overview list after the created / changed object is submitted



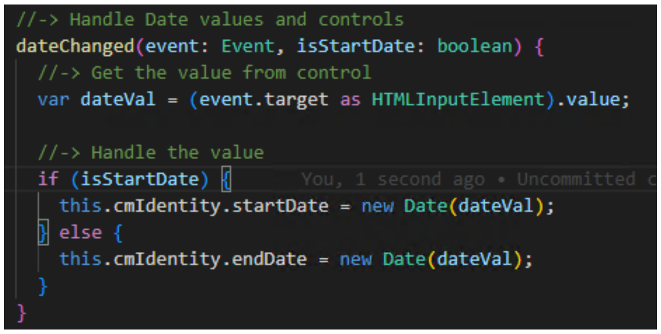
```
//-> Function to submit an new identity
addIdentity(): void {
  this.MyIdentities.push(this.cmIdentity);
  this.router.navigate(['identities']);
}
```

- To solve the problem with the non displayed date values (for both date values), establish a one way data binding modify file ***cmidentity.component.html***:
  - Replace the two way data binding by a [value] handler
  - Add a pipe format for an iso date to the date value “|date:’yyyy-MM-dd’”
  - Add an event (input) to each date input and name the function “dateChanged($event)
  - $event offers the event called, but you need as well a parameter which identifies the date field (startDate // endDate). Therefor, we use a boolean flag which is true for the startDate.



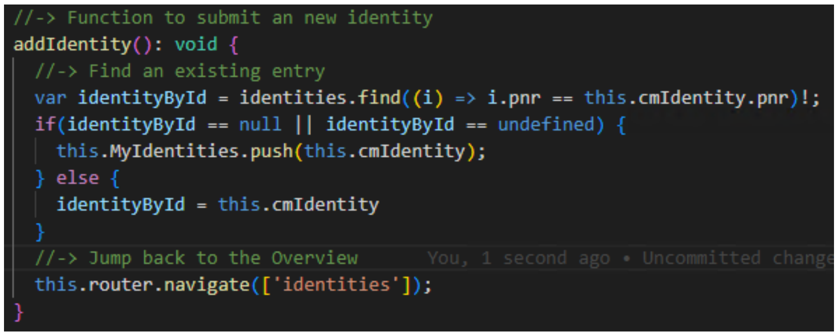
```
<input
  class="form-control"
  type="date"
  id="frmstartDate"
  [value]="cmIdentity.startDate | date: 'yyyy-MM-dd'"
  (input)="dateChanged($event, true)"
/>
```

- Define the new function in ***cmidentity.component.ts***
  - We need to get the value out of the touched HTML control
  - We need to store the right date value depending on the flag we inserted



```
//-> Handle Date values and controls
dateChanged(event: Event, isStartDate: boolean) {
  //-> Get the value from control
  var dateVal = (event.target as HTMLInputElement).value;

  //-> Handle the value
  if (isStartDate) {        You, 1 second ago • Uncommitted c
    this.cmIdentity.startDate = new Date(dateVal);
  } else {
    this.cmIdentity.endDate = new Date(dateVal);
  }
}
```

- To ensure we don't create everytime a new object, if we modify an object wenn need to separate a create from an edit submit. In ***cmIdentity.component.ts***:
  - Find function addIdentity()
  - From ngOnInit copy the line where the variable identityById gets created
  - Modified the copied line by assigning "this.cmIdentity.pnr" as the current object value
  - Remove the "!", we need null values if they happen
  - For null or undefined lets push a new element and for all others just store the available one.



```
//-> Function to submit an new identity
addIdentity(): void {
  //-> Find an existing entry
  var identityById = identities.find((i) => i.pnr == this.cmIdentity.pnr)!;
  if(identityById == null || identityById == undefined) {
    this.MyIdentities.push(this.cmIdentity);
  } else {
    identityById = this.cmIdentity
  }
  //-> Jump back to the Overview      You, 1 second ago • Uncommitted change
  this.router.navigate(['identities']);
}
```

Git:
```
8dc6fc7 Allow modifying of available identity records
 7 files changed, 51 insertions(+), 28 deletions(-)
```

Addendum

A1: Git log – Phase 2.1

```
Git:
a200ff4 Template to account name creation (from first and lastname) added
 3 files changed, 9 insertions(+), 3 deletions(-)

29b97a0 Form to insert new identity records implemented
 4 files changed, 176 insertions(+), 34 deletions(-)

96474b5 Implement a button to delete single list entries of the sample data set
 3 files changed, 14 insertions(+)

3b6e1f5 Integrating Bootstrap and formating Date values
 5 files changed, 28 insertions(+), 14 deletions(-)

09a8b01 Create and display identity sample data
 4 files changed, 26 insertions(+), 1 deletion(-)

9fde8ec Create an interface for an identity - data structure we need
 3 files changed, 20 insertions(+), 483 deletions(-)

30b5e4a Empty Angular project IDENTITY-MANAGEMENT
 28 files changed, 21959 insertions(+)

1122248 Initial empty folder for Angular projects -> all protected by one Git repository
 1 file changed, 1 insertion(+)
```

A2: Code Listing Development Phase 2.1

A2.1: Index.html

```html
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>IdentityManagement</title>
  <base href="/">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="icon" type="image/x-icon" href="favicon.ico">
  <!-- Bootstrap Integration Part 1 -->
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0-beta1/dist/css/
bootstrap.min.css" rel="stylesheet"
integrity="sha384-0evHe/X+R7YkIZDRvuzKMRqM+OrBnVFBL6DOitfPri4tjfHxaWutUpFmBp4vmVor"
crossorigin="anonymous">

</head>
<body>
  <!-- Bootstrap Integration Part 2 -->
  <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0-beta1/dist/js/bootstrap.bundle.min.js"
integrity="sha384-pprn3073KE6tl6bjs2QrFaJGz5/SUsLqktiwsUTF55Jfv3qYSDhgCecCxMW52nD2"
crossorigin="anonymous"></script>
  <app-root></app-root>
</body>
</html>
```

A2.2: app/app-component.html

```
<div class="card mt-4">
  <div class="card-header h3">Insert Identity </div>
</div>
<div>
  <!--Formular-->
  <div class="grid">
    <div class="row m-4">
      <div class="col">
        <label class="form-label" for="frmFirstname">Firstname:</label>
        <input class="form-control" type="text" id="frmFirstname"
[(ngModel)]="cmIdentity.firstname" (change)="genAccountName()"/>
      </div>
      <div class="col">
        <label class="form-label" for="frmLastname">Lastname:</label>
        <input class="form-control" type="text" id="frmLastname"
[(ngModel)]="cmIdentity.lastname" (change)="genAccountName()"/>
      </div>
    </div>
    <div class="row m-4">
      <div class="col">
        <label class="form-label" for="frmpnr">Personnel number:</label>
        <input class="form-control" type="number" id="frmpnr"
[(ngModel)]="cmIdentity.pnr"/>
      </div>
      <div class="col">
        <label class="form-label" for="frmAccountname">Account name:</label>
        <input class="form-control" type="text" id="frmAccountname"
[(ngModel)]="cmIdentity.accountname"/>
      </div>
    </div>
    <div class="row m-4">
      <div class="col">
        <label class="form-label" for="frmDepartment">Department:</label>
        <input class="form-control" type="text" id="frmDepartment"
[(ngModel)]="cmIdentity.department"/>
      </div>
      <div class="col">
        <label class="form-label" for="frmCostcenter">Costcenter:</label>
        <input class="form-control" type="text" id="frmCostcenter"
[(ngModel)]="cmIdentity.costcenter"/>
      </div>
    </div>
    <div class="row m-4">
      <div class="col">
        <label class="form-label" for="frmstartDate">Contract begin:</label>
        <input class="form-control" type="date" id="frmstartDate"
[(ngModel)]="cmIdentity.startDate"/>
      </div>
      <div class="col">
        <label class="form-label" for="frmendDate">Contract end:</label>
        <input class="form-control" type="date" id="frmendDate"
[(ngModel)]="cmIdentity.endDate"/>
      </div>
    </div>
    <div class="row m-4">
      <!--Submit button-->
      <button class="btn btn-primary" (click)="addIdentity()">
        Submit
      </button>
    </div>

    <div class="card mt-4">
      <div class="card-header h3">Identity Set</div>
      <div class="card-body">
        <!--Loop to step through all entries of the data array-->
        <div *ngFor="let CurIdentity of identities">
          <hr />
          <div>
            <div class="row mt-2">
              <div class="col">
                <!--Display values-->
                <div class="h5">
                  Name: {{ CurIdentity.firstname }} {{ CurIdentity.lastname }}
                </div>
                <div>
                  Account: {{ CurIdentity.accountname }}, PNr:
                  {{ CurIdentity.pnr }}
                </div>
                <div>
                  Costcenter: {{ CurIdentity.costcenter }}, Department:
                  {{ CurIdentity.department }}
                </div>
                <div>
                  From: {{ CurIdentity.startDate | date: "yyyy-MM-dd" }}
                </div>
                <div>Until: {{ CurIdentity.endDate | date: "yyyy-MM-dd" }}</div>
              </div>
              <div class="col">
                <!--Button to delete list entries-->
                <button
                  class="btn btn-danger btn-sn"
                  (click)="deleteIdentity(CurIdentity)"
                >
                  x Entfernen
                </button>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>
```

A.2.3: app/app.module.ts

```
//-> needed to use the 2 way data binding
import { NgModule } from '@angular/core';
```

```
//-> oob to render the content
import { BrowserModule } from '@angular/platform-browser';
import { AppComponent } from './app.component';
//-> Needed to work with forms
import { FormsModule } from '@angular/forms';

@NgModule({
  declarations: [
    AppComponent
  ],
  imports: [
    //-> Some imports needs to be preloaded to become usable
    BrowserModule,
    FormsModule,
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

A.2.4: app/identities.ts

```
import { DecimalPipe } from "@angular/common";

// -> Data structure for a single identity
// -> created as an interface to get values required

export interface identity {
  firstname: string,
  lastname: string,
  accountname: string,
  pnr: number,
  department: string,
  costcenter: string,
  startDate: Date,
  endDate?: Date
}
```

A.2.5: app/app.component.ts

```
import { Component } from '@angular/core';
import { identity } from './identities';

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css'],
})

// -> AppCompontent feature set
export class AppComponent {
  title = 'Small Identity Management';

  // -> Sample data set
```

```
identities: identity[] = [
  {
    firstname: 'Herwig',
    lastname: 'Abele',
    accountname: 'herwigabe',
    pnr: 1,
    costcenter: '00101010',
    department: 'Dev',
    startDate: new Date('1998-01-01'),
  },
  {
    firstname: 'Heinz',
    lastname: 'Hobbs',
    accountname: 'heinzHob',
    pnr: 2,
    costcenter: '00102010',
    department: 'Acc',
    startDate: new Date('1990-01-01'),
  },
  {
    firstname: 'Jon',
    lastname: 'Ullman',
    accountname: 'jonull',
    pnr: 3,
    costcenter: '00101040',
    department: 'Eng',
    startDate: new Date('2001-01-01'),
  },
  {
    firstname: 'Petra',
    lastname: 'Minor',
    accountname: 'petramin',
    pnr: 4,
    costcenter: '00101070',
    department: 'Mkt',
    startDate: new Date('2010-01-01'),
  },
  {
    firstname: 'Sabina',
    lastname: 'Tenisson',
    accountname: 'sabinaten',
    pnr: 5,
    costcenter: '00106010',
    department: 'Dev',
    startDate: new Date('2020-01-01'),
    endDate: new Date('2023-05-01'),
  },
  {
    firstname: 'Clary',
    lastname: 'Blitch',
    accountname: 'clarybli',
    pnr: 6,
    costcenter: '00101080',
    department: 'Dev',
    startDate: new Date('2011-01-01'),
    },
  ];

  //-> delete entry from set of all identities (void means this function doesn't return
anything)
  deleteIdentity(myIdentity: identity): void {
    this.identities = this.identities.filter((i) => i != myIdentity);
  }

  //-> New Identity object to support the form
  cmIdentity: identity = {
    firstname: '',
    lastname: '',
    accountname: '',
    pnr: Math.max(...this.identities.map(o => o.pnr)) + 1,
    costcenter: '',
    department: '',
    startDate: new Date(),
  };

  //-> Function to submit an new identity
  addIdentity(): void {
    this.identities.push(this.cmIdentity);
    this.cmIdentity = {
      firstname: '',
      lastname: '',
      accountname: '',
      pnr: Math.max(...this.identities.map(o => o.pnr)) + 1,
      costcenter: '',
      department: '',
      startDate: new Date(),
    };
  }

  //-> Function to create an account name from firstname and lastname
  genAccountName(): void {
    this.cmIdentity.accountname = `${this.cmIdentity.firstname}$
{this.cmIdentity.lastname.substring(0,3)}`;
  }
}
```

- In NX select the Workspace
- Integrate Bootstrap
  https://getbootstrap.com/docs/5.1/getting-started/introduction/

git log --oneline --shortstat