



Angular 5 CT Specialization

Angular Material Lab 8

January 2019

CitiusTech has prepared the content contained in this document based on information and knowledge that it reasonably believes to be reliable. Any recipient may rely on the contents of this document at its own risk and CitiusTech shall not be

responsible for any error and/or omission in the preparation of this document. The use of any third party reference should not be regarded as an indication of an endorsement, an affiliation or the existence of any other kind of relationship between CitiusTech and such third party

Document Control

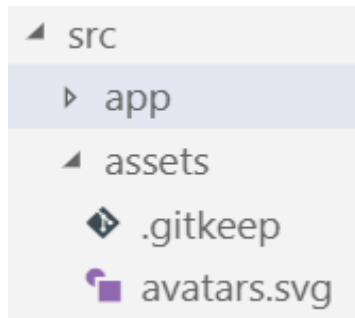
Version	1.0
Created by	Karthikeyan J
Updated by	Karthikeyan J
Reviewed by	
Date	January, 2019
Earlier versions	

Table of Contents

Copying an <i>icon set</i> into the project.....	3
Injecting <i>MatIconRegistry</i> & <i>DomSanitizer</i> services.....	3
Registering the icon set file with <i>MatIconRegistry</i> & generating a safe URL using <i>DomSanitizer</i>	4
Navigating to a specific contact/user using <i>routerLink</i>	5
Adding a method in the <i>data service</i> to get a user/contact by <i>id</i>	5
Reading route parameters using <i>ActivatedRoute</i> service & using Angular material's <i><mat-spinner></i> component to display a progress indicator	6

Copying an *icon set* into the project

1. Copy the desired icon set file into the **src\assets** folder of your project as shown below:



Injecting *MatIconRegistry* & *DomSanitizer* services

2. Open **contact-manager.component.ts** file & add the following import statements into it:

```
import {MatIconRegistry} from '@angular/material';  
import {DomSanitizer} from '@angular/platform-browser';
```

3. Add the following parameters to the constructor of the component class:

```
private maticonregistry_svc:MatIconRegistry, private  
      domsanitizer_svc:DomSanitizer
```

Registering the icon set file with *MatIconRegistry* & generating a safe URL using *DomSanitizer*

4. Add the following code inside the constructor of the **ContactManagerAppComponent** class:

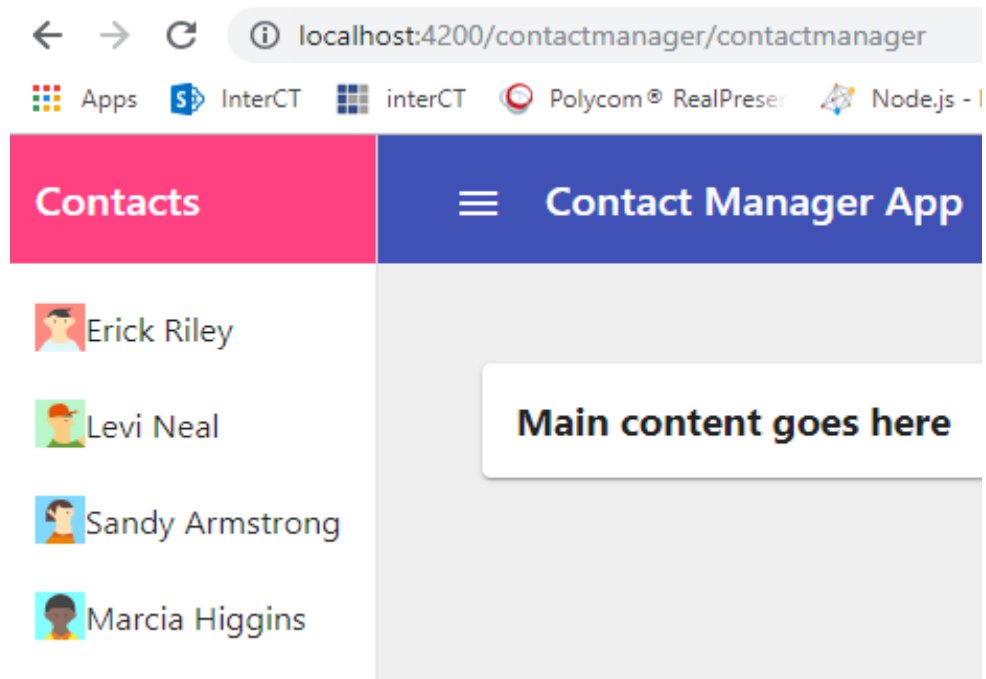
```
this.maticonregistry_svc.  
  addSvgIconSet(this.domsanitizer_svc.bypassSecurityTrustResourceUrl('  
    ../assets/avatars.svg'));
```

5. Open the root module & add the following import statement into it:
`import {HttpClientModule} from '@angular/common/http';`
6. Also add the **HttpClientModule** into the **imports** array of the root module.

7. Open **toolbar.component.html** file & add the **<mat-icon>** element inside the **<mat-list-item>** element as follows:

```
<mat-icon svgIcon="{{user.avatar}}"></mat-icon>
```

8. Serve the application & navigate to **http://localhost:4200/contactmanager**. Observe that the SVG icons are now displayed against each contact name as shown below:



Navigating to a specific contact/user using *routerLink*

9. Open **contact-manager-routing.module.ts** file & add the following before the last path inside the **children** array:

```
{path: ':id', component: MainContentComponent},
```
10. Open **toolbar.component.html** file & add the following directive to the **<a>** element:

```
routerLink="{{user.id}}"
```

Adding a method in the *data service* to get a user/contact by *id*

11. Open **app/services/user.service.ts** file & add the following method into it:

```
public GetUserId(id:number):Observable<User>
{
    return this.http.get<User>('https://angular-material-
    api.azurewebsites.net/users/' +id);
}
```

Reading route parameters using *ActivatedRoute* service & using Angular material's *<mat-spinner>* component to display a progress indicator

1. Open **main-content.component.ts** file & add the following import statements:

```
import {ActivatedRoute, Params} from '@angular/router';  
import {UserService} from '../services/user.service';  
import {Observable} from 'rxjs';  
import {User} from '../models/user';
```

2. Add the following parameters to the constructor of the **MainContentComponent** class:

```
private activatedroute:ActivatedRoute,private svc:UserService
```

3. Add the following inside the **MainContentComponent** class:

```
public userinfo:Observable<User>;  
public fetching:boolean;  
public user:User;
```

4. Add the following inside the **ngOnInit()** method of the **MainContentComponent** class:

```
this.activatedroute.paramMap.subscribe(
  (paramlist) => {
    let id = -1;
    this.fetching = true;

    if(paramlist.get('id') != undefined)
    {
      id = parseInt(paramlist.get('id'));
    }

    if(id != -1)
    {
      setTimeout(() => {
        this.fetching = false;
        this.userinfo = this.svc.GetUserById(id);
        this.userinfo.subscribe(
          (user) => this.user = user,
          (error) => console.log("Error in retrieving user info: ",
                                error)
        );
      },500);
    }
  }
);
```

THIS CODE CONTAINS LOGIC TO FETCH THE ROUTE PARAMETER USING THE *ActivatedRoute* SERVICE & SIMULATES A SMALL DELAY OF CALLING THE *GetUserById()* METHOD OF THE DATA SERVICE. THE BOOLEAN VARIABLE *fetching* IN THIS CASE IS USED TO HIDE OR DISPLAY THE *<mat-spinner>* COMPONENT IN THE TEMPLATE.

5. Open **main-content.component.html** file. Delete the existing markup & add the following markup instead:

```
<div *ngIf="fetching == true ; else contents">
  <mat-spinner></mat-spinner>
</div>

<ng-template #contents>
  <mat-card>
    <h3>Main content goes here</h3>
  </mat-card>
</ng-template>
```


6. Replace the markup inside **<mat-card>** element with the following:

```
<mat-card-header>
  <mat-icon mat-card-avatar
    svgIcon="{{user.avatar}}"></mat-icon>
  <mat-card-title>
    {{user.name}}
  </mat-card-title>
  <mat-card-subtitle>
    Birthday: {{user.birthDate | date:'longDate'}}
  </mat-card-subtitle>
</mat-card-header>

<mat-card-content>
  {{user.bio}} <br/> <br/>
  <mat-divider [inset]="true"></mat-divider>
  <div *ngFor="let n of user.notes">
    <span>{{n.id}}, {{n.title}}</span> <br/>
    <span>{{n.date | date}}</span>
    <br/> <br/>
  </div>
</mat-card-content>
```

7. Serve the application & navigate to **<http://localhost:4200/contactmanager>**. Notice that the progress is displayed by the **<mat-spinner>** component since as per the logic written above. Click on a contact name & observe the URL on the browser. Also notice that when a contact is clicked, the **<mat-spinner>** is displayed to simulate a delay before the details of the selected contact is displayed in the main content area.