



Outline Overview

1. Configure angular develop environment install system require.
2. Backend environment create configure.
 - 2.1. XAMPP / WAMP.
 - 2.2. Docker compose.
3. PostgreSQL database Configure.
 - 3.1. Create our tutorial database.
4. Create Angular fore frontend project with angular cli.
5. Create CakePHP3.5.x fore backend project with composer.
 - 5.1. Configure CakePHP3.5.x fore RESTFUL API.
 - 5.2. Users RESTFUL API.
6. Angular Component.
 - 6.1. Project structure.
 - 6.2. File structure.
 - 6.3. Data binding component.
 - 6.4. Making template.
 - 6.5. User component.
 - 6.6. Install paginate, toaster notification, date picker plugin.
 - 6.7. User register / add user component.
 - 6.8. User update / user edit component.
 - 6.9. User delete component.
 - 6.10. User filter / user search component.
 - 6.11. Authorize / Authentication Component.
 - 6.12. Guard / routing guard.
7. HW
8. Q & A

System Requirement

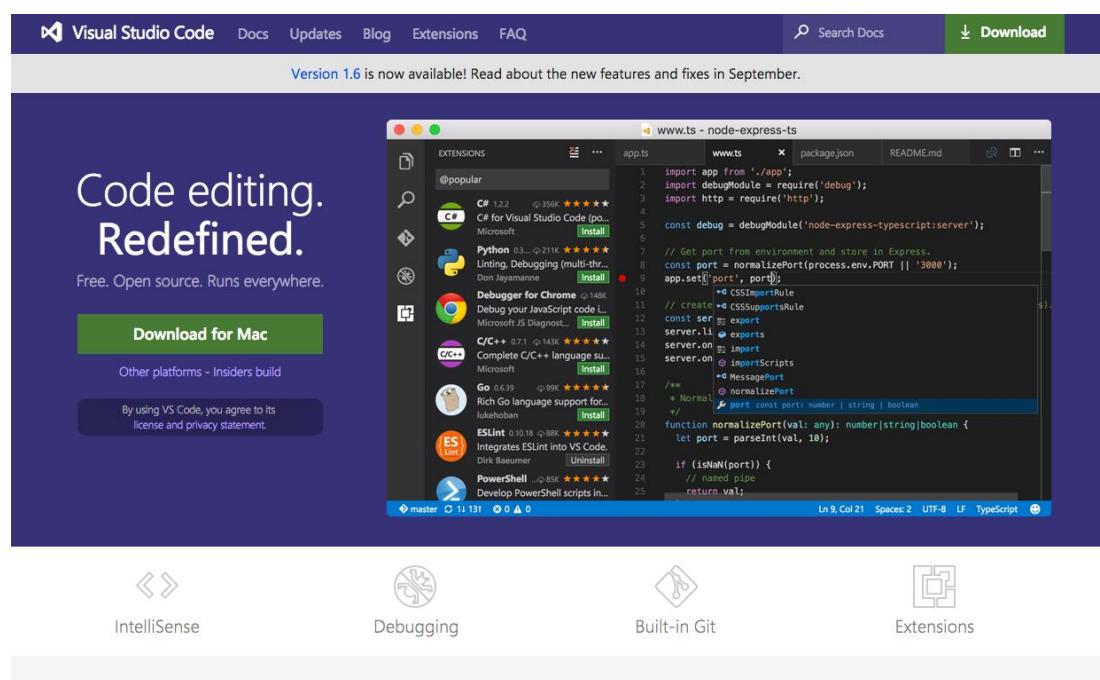
NODE JS V 6.9.x or greater

NPM 5.x or greater

Download: <https://nodejs.org>

Visual Studio Code

Download: <https://code.visualstudio.com/>



Check Environment

```
$ node --v
```

```
$ npm --v
```



A screenshot of a terminal window titled 'MINGW64/d/ANGULAR/PAKGON_TRAINING'. It shows two command-line entries: 'node --v' which outputs 'v8.9.1', and 'npm --v' which outputs '6.0.0'. The terminal has standard window controls (minimize, maximize, close) at the top right.

```
OK_Antikorn@DESKTOP-3DAZQ21 MINGW64 /d/ANGULAR/PAKGON_TRAINING
$ node --v
v8.9.1
OK_Antikorn@DESKTOP-3DAZQ21 MINGW64 /d/ANGULAR/PAKGON_TRAINING
$ npm --v
6.0.0
OK_Antikorn@DESKTOP-3DAZQ21 MINGW64 /d/ANGULAR/PAKGON_TRAINING
$
```

If your NPM version less than 5.x

Run the following command for updating npm

```
$ npm install -g npm@latest
```

Install Typescript on your system

```
$ npm install -g typescript
```

Install Angular-CLI in global environment

```
$ npm install -g @angular/cli@6.2.3
```

Checking for angular-cli to be work

```
$ ng -v
```



```
MINGW64/d/ANGULAR/PAKGON_TRAINING
$ ng -v
Angular CLI: 6.2.3
Node: 8.9.1
OS: win32 x64
Angular:
...
Package          Version
@angular-devkit/architect    0.8.3
@angular-devkit/core         0.8.3
@angular-devkit/schematics   0.8.3
@schematics/angular          0.8.3
@schematics/update           0.8.3
rxjs                6.2.2
typescript          2.9.2
```

How to use the Angular-CLI

```
$ ng new your-project-name (create new angular project)
```

```
$ cd your-project-name
```

```
$ ng serve (run for angular server)
```

You can configure the default HTTP port and the one used by the LiveReload server with two command line options.

```
$ ng serve [ --host 192.168.0.1 ] [ --port 4201 ] [ --live-reload-port 5000 ]
```

New angular project with angular-CLI

```
$ mkdir DEMO_CAKEPHP3_ANGULAR6
```

```
$ cd DEMO_CAKEPHP3_ANGULAR6
```

```
$ ng new ANGULAR6-FRONTEND --routing
```

```
MINGW64/d/ANGULAR/PAKGON_TRAINING/DEMO_CAKEPHP3_ANGULAR6
$ ng new ANGULAR6-FRONTEND --routing
  ANGULAR6-FRONTEND angular.json (3647 bytes)
CREATE ANGULAR6-FRONTEND/package.json (1323 bytes)
CREATE ANGULAR6-FRONTEND/README.md (103 bytes)
CREATE ANGULAR6-FRONTEND/tsconfig.json (408 bytes)
CREATE ANGULAR6-FRONTEND/tsconfig.lib.json (102 bytes)
CREATE ANGULAR6-FRONTEND/tsconfig.polymorphic.json (245 bytes)
CREATE ANGULAR6-FRONTEND/.gitignore (303 bytes)
CREATE ANGULAR6-FRONTEND/src/favicon.ico (5430 bytes)
CREATE ANGULAR6-FRONTEND/src/index.html (303 bytes)
CREATE ANGULAR6-FRONTEND/src/main.ts (373 bytes)
CREATE ANGULAR6-FRONTEND/src/polyfills.ts (3194 bytes)
CREATE ANGULAR6-FRONTEND/src/test.ts (642 bytes)
CREATE ANGULAR6-FRONTEND/src/styles.css (80 bytes)
CREATE ANGULAR6-FRONTEND/src/assets/icon.png (300 bytes)
CREATE ANGULAR6-FRONTEND/src/karma.conf.js (964 bytes)
CREATE ANGULAR6-FRONTEND/src/tsconfig.app.json (166 bytes)
CREATE ANGULAR6-FRONTEND/src/tsconfig.spec.json (256 bytes)
CREATE ANGULAR6-FRONTEND/src/tslint.json (314 bytes)
CREATE ANGULAR6-FRONTEND/src/assets/.gitkeep (0 bytes)
CREATE ANGULAR6-FRONTEND/src/environments/environment.prod.ts (51 bytes)
CREATE ANGULAR6-FRONTEND/src/environments/environment.ts (662 bytes)
CREATE ANGULAR6-FRONTEND/src/app/app-routing.module.ts (101 bytes)
CREATE ANGULAR6-FRONTEND/src/app/app.module.ts (301 bytes)
CREATE ANGULAR6-FRONTEND/src/app/app.component.html (1173 bytes)
CREATE ANGULAR6-FRONTEND/src/app/app.component.spec.ts (1145 bytes)
CREATE ANGULAR6-FRONTEND/src/app/app.component.ts (221 bytes)
CREATE ANGULAR6-FRONTEND/src/app/app.component.css (0 bytes)
CREATE ANGULAR6-FRONTEND/e2e/protractor.conf.js (752 bytes)
CREATE ANGULAR6-FRONTEND/e2e/tsconfig.e2e.json (213 bytes)
CREATE ANGULAR6-FRONTEND/e2e/src/app/e2e-spec.ts (313 bytes)
CREATE ANGULAR6-FRONTEND/e2e/src/app.po.ts (208 bytes)
```

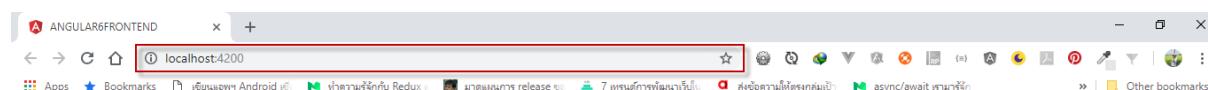
How to run your angular project

```
$ ng serve --open
```

```
MINGW64/d/ANGULAR/PAKGON_TRAINING/DEMO_CAKEPHP3_ANGULAR6/ANGULAR6-FRONTEND
$ cd ANGULAR6-FRONTEND/
$ ng serve --open
  Angular Live Development Server is listening on localhost:4200, open your browser on http://localhost:4200/ **

Date: 2018-10-25T11:26:29.278Z
Hash: 3e9861ba081263791c36
Time: 8656ms
chunk {main} main.js, main.js.map (main) 12.9 kB [initial] [rendered]
chunk {polyfills} polyfills.js, polyfills.js.map (polyfills) 227 kB [initial] [rendered]
chunk {runtime} runtime.js, runtime.js.map (runtime) 6.22 kB [entry] [rendered]
chunk {styles} styles.js, styles.js.map (styles) 15.6 kB [initial] [rendered]
chunk {vendor} vendor.js, vendor.js.map (vendor) 3.55 MB [initial] [rendered]
[...]: Compiled successfully.
```

Open up your browser and type:



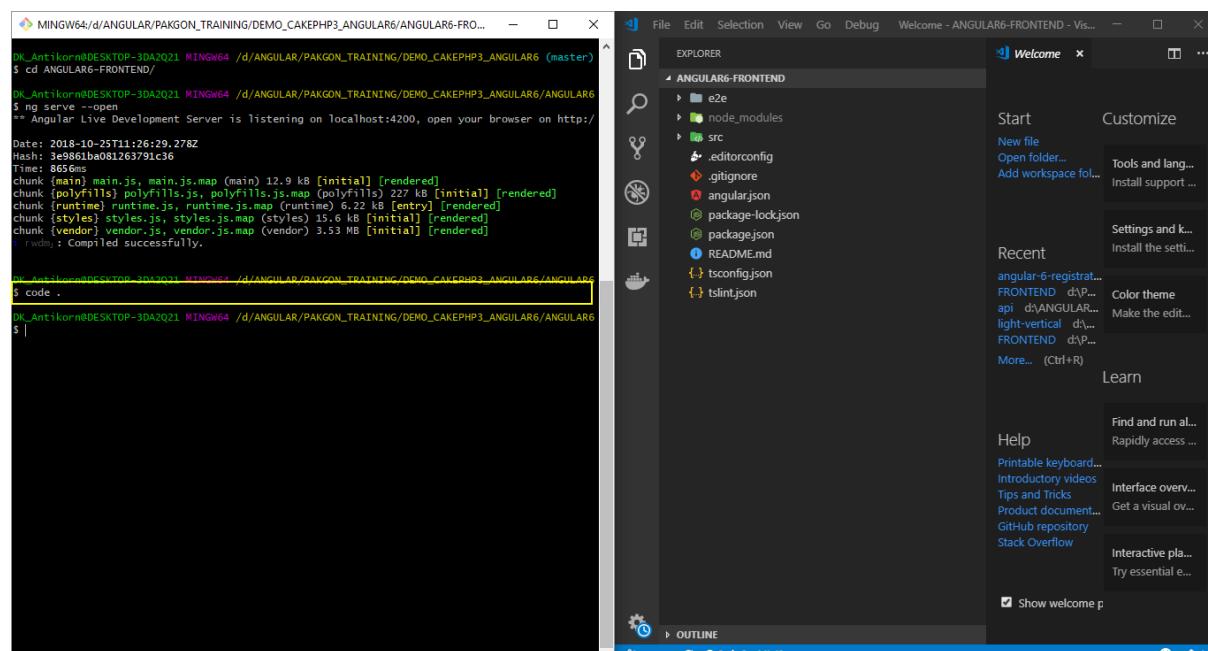
Welcome to ANGULAR6-FRONTEND!



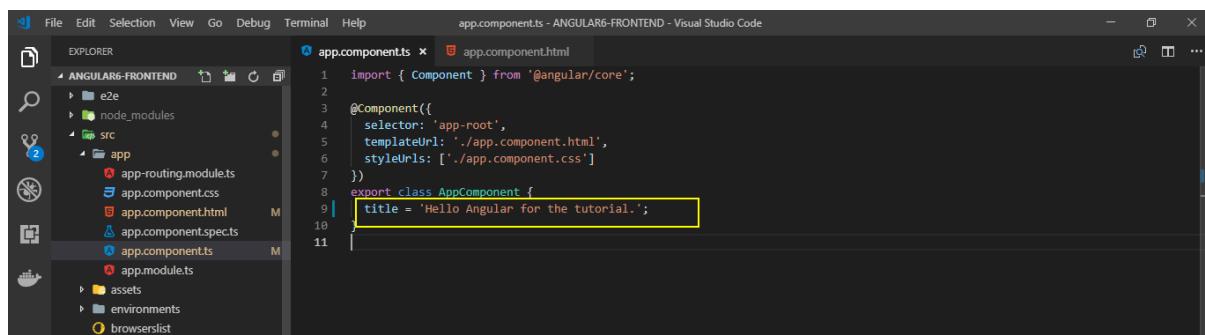
Here are some links to help you start:

- [Tour of Heroes](#)
- [CLI Documentation](#)
- [Angular blog](#)

Open up your angular project with Visual studio code (VS Code)



Open src/app/app.component.ts and modified as in figure.



```
import { Component } from '@angular/core';
@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {
  title = 'Hello Angular for the tutorial.';
}
```

Back to the Browser



Generate with ng command you can generate class, component, service,

Bulma CSS Framework

Download:

<http://bulma.io/>

<https://github.com/jgthms/bulma>

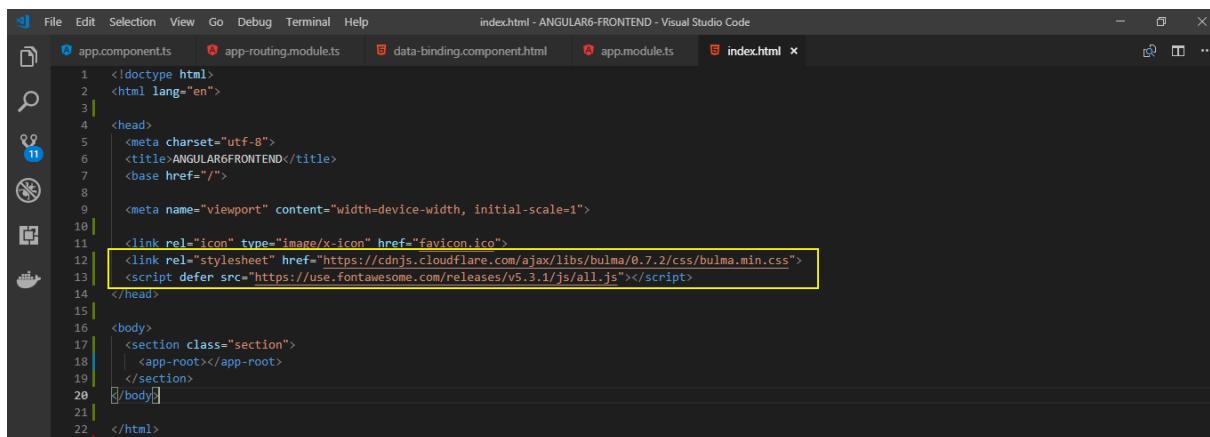
<https://bulma.io/documentation/overview/start/>

Font Awesome: Web Icon, Application Icon

Download:

<http://fontawesome.io/>

Adding for each line to src/index.html



```

1  <!doctype html>
2  <html lang="en">
3  |
4  <head>
5  <meta charset="utf-8">
6  <title>ANGULAR6FRONTEND</title>
7  <base href="/">
8  |
9  <meta name="viewport" content="width=device-width, initial-scale=1">
10 |
11 <link rel="icon" type="image/x-icon" href="favicon.ico">
12 <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/bulma/0.7.2/css/bulma.min.css">
13 <script defer src="https://use.fontawesome.com/releases/v5.3.1/js/all.js"></script>
14 </head>
15 |
16 <body>
17 | <section class="section">
18 | | <app-root></app-root>
19 | </section>
20 | </body>
21 |
22 </html>

```

Generate for component

```
$ ng generate component component-name [--spec=false] [--is] [--it]
```

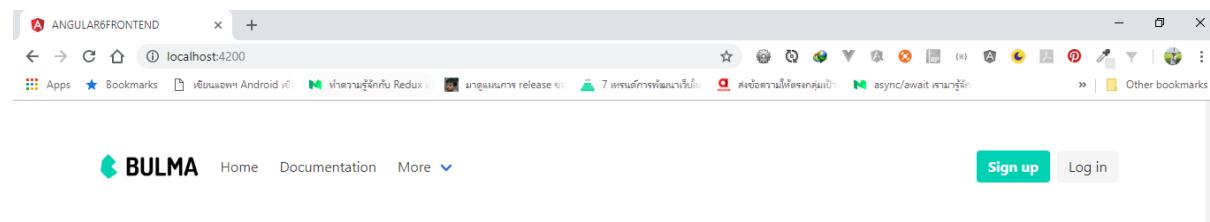
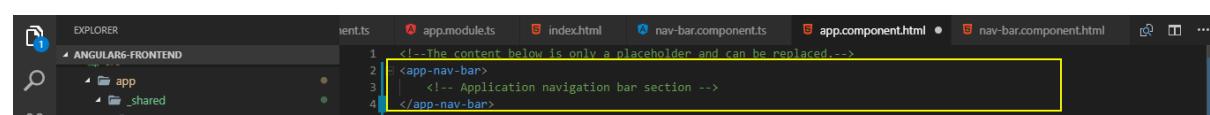
```
$ ng g c component component-name [--spec=false] [--is] [--it]
```

Generate Navigation Bar Component

```
$ ng g c _shared/nav-bar --spec false
```

```
MINGW64/d/ANGULAR/PAKGON_TRAINING/DEMO_CAKEPHP3_ANGULAR6/ANGULAR6-FRONTEND
$ ng g c _shared/nav-bar --spec false
CREATE src/app/_shared/nav-bar/nav-bar.component.html (26 bytes)
CREATE src/app/_shared/nav-bar/nav-bar.component.ts (272 bytes)
CREATE src/app/_shared/nav-bar/nav-bar.component.css (0 bytes)
UPDATE src/app/app.module.ts (710 bytes)
```

Adding html code to app/_shared/nav-bar/nav-bar.component.html

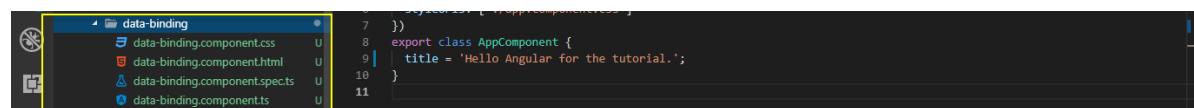


Generate data binding component

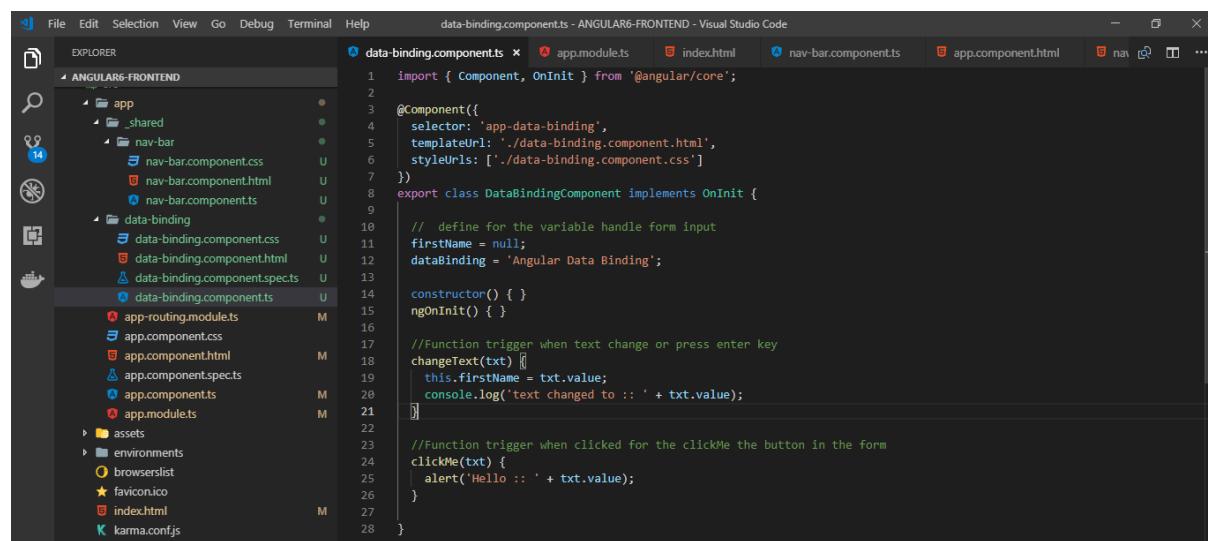
\$ ng generate component data-binding

```
AK_Antikorn@DESKTOP-3DAZQ21 MINGW64 /d/ANGULAR/PAKGON_TRAINING/DEMO_CAKEPHP3_ANGULAR6/FRONTEND (master)
$ ng g c data-binding
CREATE src/app/data-binding/data-binding.component.html (31 bytes)
CREATE src/app/data-binding/data-binding.component.spec.ts (664 bytes)
CREATE src/app/data-binding/data-binding.component.ts (292 bytes)
DELETE src/app/data-binding/data-binding.component.css (0 bytes)
UPDATE src/app/app.module.ts (497 bytes)
OK_Antikorn@DESKTOP-3DAZQ21 MINGW64 /d/ANGULAR/PAKGON_TRAINING/DEMO_CAKEPHP3_ANGULAR6/FRONTEND (master)
$
```

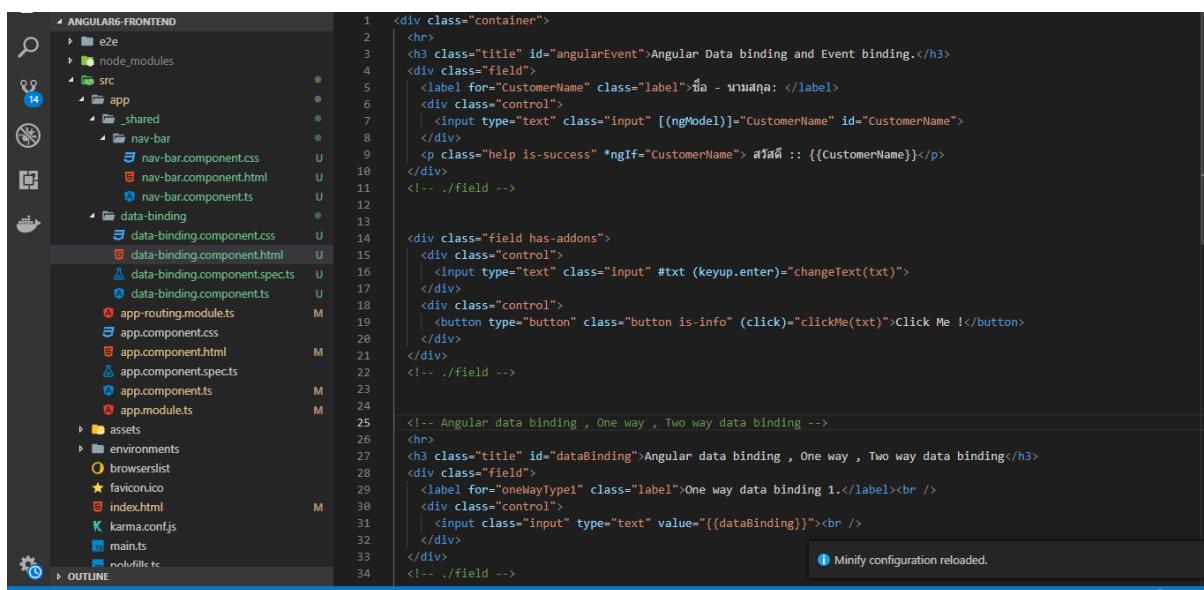
Component file structure.



Open app/data-binding/data-binding.component.ts and then adding code.



Open app/data-binding/data-binding.component.html and then adding code.



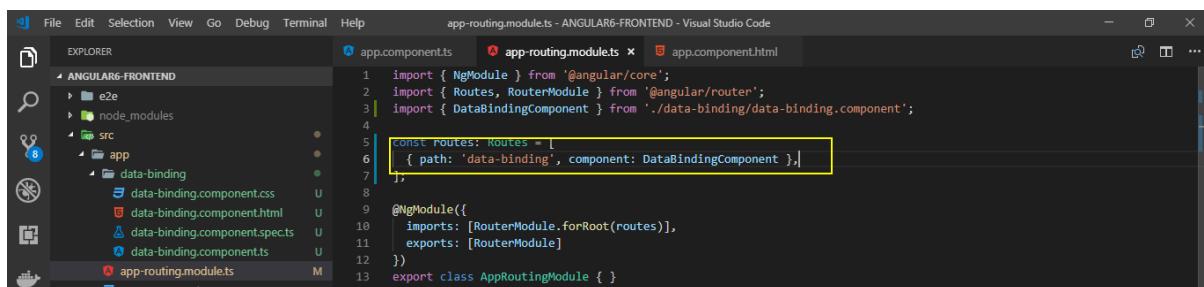
```

1 <div class="container">
2   <hr>
3   <h3 class="title" id="angularEvent">Angular Data binding and Event binding.</h3>
4   <div class="field">
5     <label for="CustomerName" class="label">ໜີ້ - ນາມສະກຳ: </label>
6     <div class="control">
7       <input type="text" class="input" [(ngModel)]="CustomerName" id="CustomerName">
8     </div>
9     <p class="help is-success" *ngIf="CustomerName" style:: {{CustomerName}}</p>
10    </div>
11    <!-- ./field -->
12
13
14 <div class="field has-addons">
15   <div class="control">
16     <input type="text" class="input" #txt (keyup.enter)="changeText(txt)">
17   </div>
18   <div class="control">
19     <button type="button" class="button is-info" (click)="clickMe(txt)">Click Me !</button>
20   </div>
21 </div>
22 <!-- ./field -->
23
24
25 <!-- Angular data binding , One way , Two way data binding -->
26 <hr>
27 <h3 class="title" id="dataBinding">Angular data binding , One way , Two way data binding</h3>
28 <div class="field">
29   <label for="oneWayType1" class="label">One way data binding 1.</label><br />
30   <div class="control">
31     <input class="input" type="text" value="{{dataBinding}}"><br />
32   </div>
33 </div>
34 <!-- ./field -->

```

Minify configuration reloaded.

Add routing path in app/app-routing.module.ts

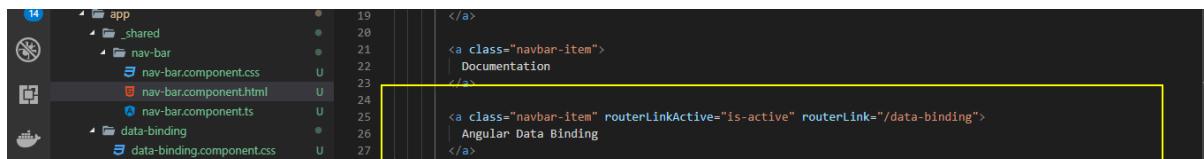


```

1 import { NgModule } from '@angular/core';
2 import { Routes, RouterModule } from '@angular/router';
3 import { DataBindingComponent } from './data-binding/data-binding.component';
4
5 const routes: Routes = [
6   { path: 'data-binding', component: DataBindingComponent },
7 ];
8
9 @NgModule({
10   imports: [RouterModule.forRoot(routes)],
11   exports: [RouterModule]
12 })
13 export class AppRoutingModule { }

```

And add code router link navigate to app/nav-bar/nav-bar.component.html

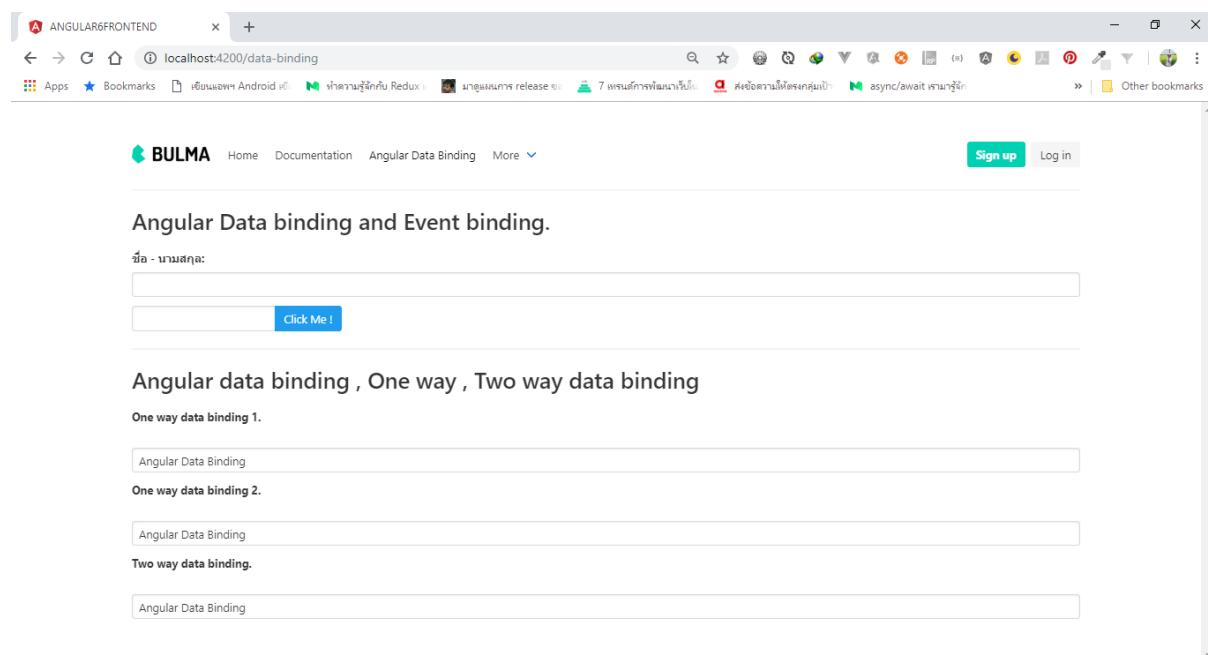


```

19   </a>
20
21   <a class="navbar-item">
22     Documentation
23   </a>
24
25   <a class="navbar-item" routerLinkActive="is-active" routerLink="/data-binding">
26     Angular Data Binding
27   </a>

```

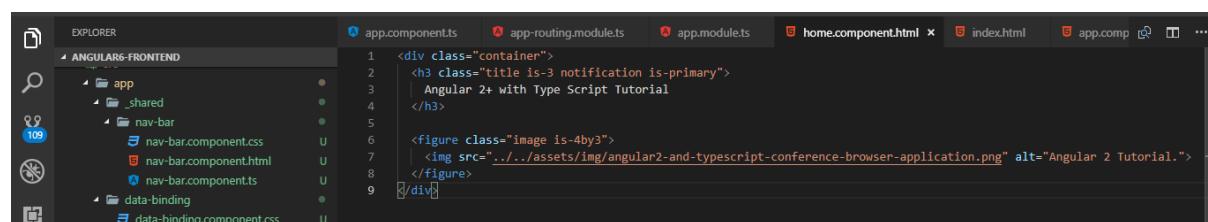
Open your browser and type URL: <http://localhost:4200/data-binding>



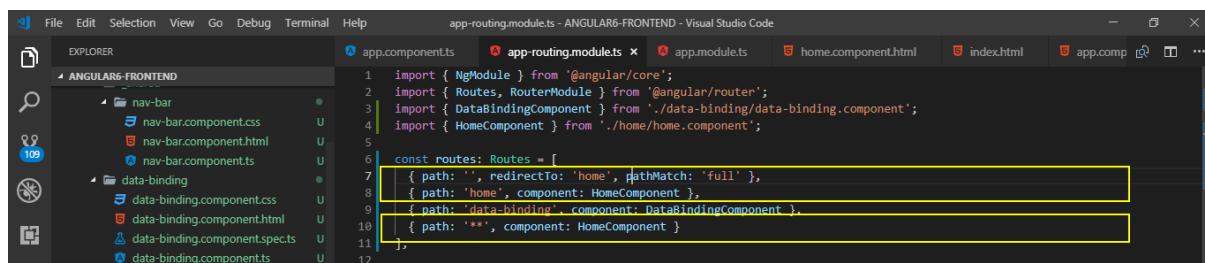
Generate home component.

```
OK_Artikorn@DESKTOP-30A2021 MINGW64 /d/ANGULAR/PAKGON_TRAINING/DEMO_CAKEPHP3_ANGULAR6/ANGULAR6-FRONTEND (master)
$ ng g c home --spec=false
CREATE src/app/home/home.component.html (32 bytes)
CREATE src/app/home/home.component.ts (261 bytes)
CREATE src/app/home/home.component.css (0 bytes)
UPDATE src/app/app.module.ts (771 bytes)
```

Add code to app/home/home.component.html



And add router link path in app/app-routing.module.ts

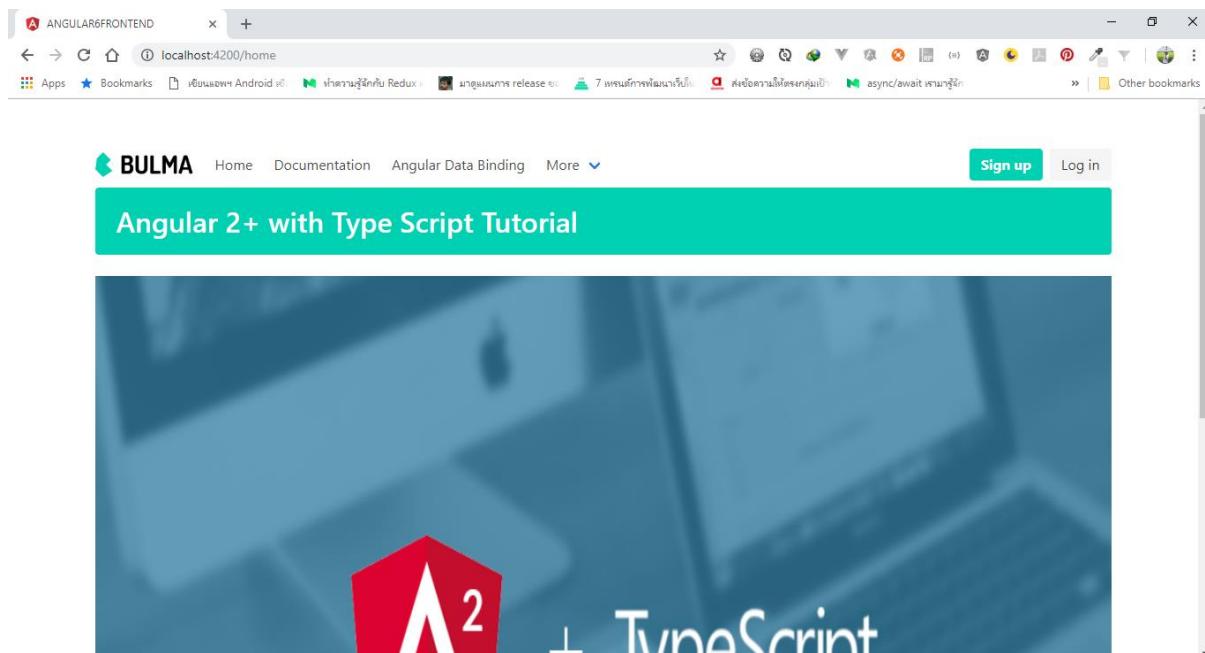


```

File Edit Selection View Go Debug Terminal Help
app-routing.module.ts - ANGULAR6-FRONTEND - Visual Studio Code
EXPLORER
ANGULAR6-FRONTEND
nav-bar
  nav-bar.component.css
  nav-bar.component.html
  nav-bar.component.ts
data-binding
  data-binding.component.css
  data-binding.component.html
  data-binding.component.spec.ts
  data-binding.component.ts
app.component.ts
app.module.ts
home.component.html
index.html
app.comp ...
...
1 import { NgModule } from '@angular/core';
2 import { Routes, RouterModule } from '@angular/router';
3 import { DataBindingComponent } from './data-binding/data-binding.component';
4 import { HomeComponent } from './home/home.component';
5
6 const routes: Routes = [
7   { path: '', redirectTo: 'home', pathMatch: 'full' },
8   { path: 'home', component: HomeComponent },
9   { path: 'data-binding', component: DataBindingComponent },
10  { path: '**', component: HomeComponent }
11 ],
12

```

Open your browser:



BACK-END CAKE PHP version 3.5.x

```
$ cd d:
```

```
$ cd xampp/htdocs/
```

```
$ composer create-project --prefer-dist cakephp/app:3.5.* DEMO_CAKEPHP3_BACKEND/www
```

```
DESKTOP-MTNGW64:~/ANGULAR/PAKGON_TRAINING/DEMO_CAKEPHP3_ANGULAR6$ composer self-update && composer create-project --prefer-dist cakephp/app:3.5.* DEMO_CAKEPHP3_BACKEND/www
Installing cakephp/app (3.5.1)
- Installing cakephp/app (3.5.1): Downloading (100%)
Created project in DEMO_CAKEPHP3_BACKEND/www
Loading composer repositories with package information
Updating dependencies (including require-dev)
Package operations: 18 installs, 0 updates, 0 removals
- Installing cakephp/plugin-installer (1.1.0): Loading from cache
- Installing aura/intl (3.0.0): Loading from cache
- Installing psr/http-message (1.0.1): Loading from cache
- Installing zendframework/zend-diactoros (1.8.6): Loading from cache
- Installing psr/log (1.0.2): Loading from cache
- Installing cakephp/cakephp (3.5.17): Loading from cache
- Installing symfony/polyfill-mbstring (v1.14.0): Loading from cache
- Installing symfony/polyfill-iconv (v1.14.0): Loading from cache
- Installing symfony/yaml (v3.4.17): Loading from cache
- Installing symfony/debug (v4.1.6): Loading from cache
- Installing symfony/console (v3.4.17): Loading from cache
Install successful in 00:00:00, 100% (0 bytes)

```

Vhost Configuration: D:\xampp\apache\conf\extra\httpd-vhosts.conf

```
<VirtualHost *:80>
```

```
DocumentRoot "D:/ANGULAR/PAKGON_TRAINING/DEMO_CAKEPHP3_ANGULAR6/CAKEPHP3-BACKEND/www"
```

```
ServerName demo.cakephp-angular.local
```

```
ErrorLog "logs/demo.cakephp-angular.local-error.log"
```

```
CustomLog "logs/demo.cakephp-angular.local-access.log" common
```

```
<Directory "D:/ANGULAR/PAKGON_TRAINING/DEMO_CAKEPHP3_ANGULAR6/CAKEPHP3-BACKEND/www">
```

```
    AllowOverride All
```

```
    Order Allow,Deny
```

```
    Allow from all
```

```
    Require all granted
```

```
</Directory>
```

```
</VirtualHost>
```

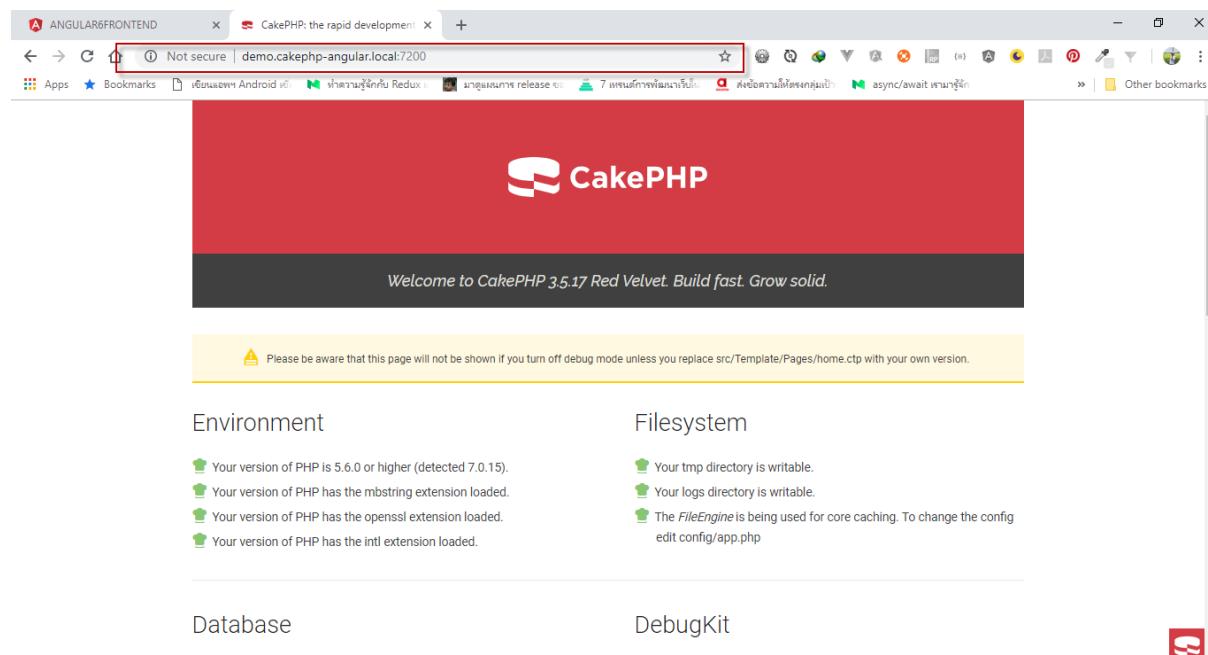
Host File: C:\Windows\System32\drivers\etc\hosts

127.0.0.1 demo.cakephp-angular.local

::1 demo.cakephp-angular.local

Restart your apache in XAMPP

Open up your browser: <http://demo.cakephp-angular.local:7200>



Docker version

D:\ANGULAR\PAKGON_TRAINING\DEMO_CAKEPHP3_ANGULAR6\docker-compose.yml

```

version: '3.2'
services:
  cakephp-backend:
    image: sarawut/php7-apache
    volumes:
      - D:/ANGULAR/PAKGON_TRAINING/DEMO_CAKEPHP3_ANGULAR6/CAKEPHP3-BACKEND/www:/var/www/html
    ports:
      - "7200:80"
      - "7243:443"
    depends_on:
      - db
  db:
    image: postgres:9.4-alpine
    ports:
      - "5435:5432"
    environment:
      POSTGRES_PASSWORD: password

```

Host File: C:\Windows\System32\drivers\etc\hosts

192.168.99.100 demo.cakephp-angular.local

Run docker compose

\$ docker-compose up -d (create and configure environment and run each container)

\$ docker-compose start (start container where configure in docker-compose.yml file)

```

MINGW64/d/ANGULAR/PAKGON_TRAINING/DEMO_CAKEPHP3_ANGULAR6
$ docker-compose up -d
Starting db ... done
Starting cakephp-backend ... done

```

CakePHP RESTFUL API

Adding line to config/routes.php



```

41     * 'action' markers.
42     *
43     */
44 Router::defaultRouteClass(DashedRoute::class);
45
46 Router::scope('/', function (RouteBuilder $routes) {
47
48     // ...
49     $routes->setExtensions(['json', 'xml', 'ajax']);
50
51     /**
52      * Here, we are connecting '/' (base path) to a controller called 'Pages',
53      * its action called 'display', and we pass a param to select the view file

```

Adding header line to src/AppController.php

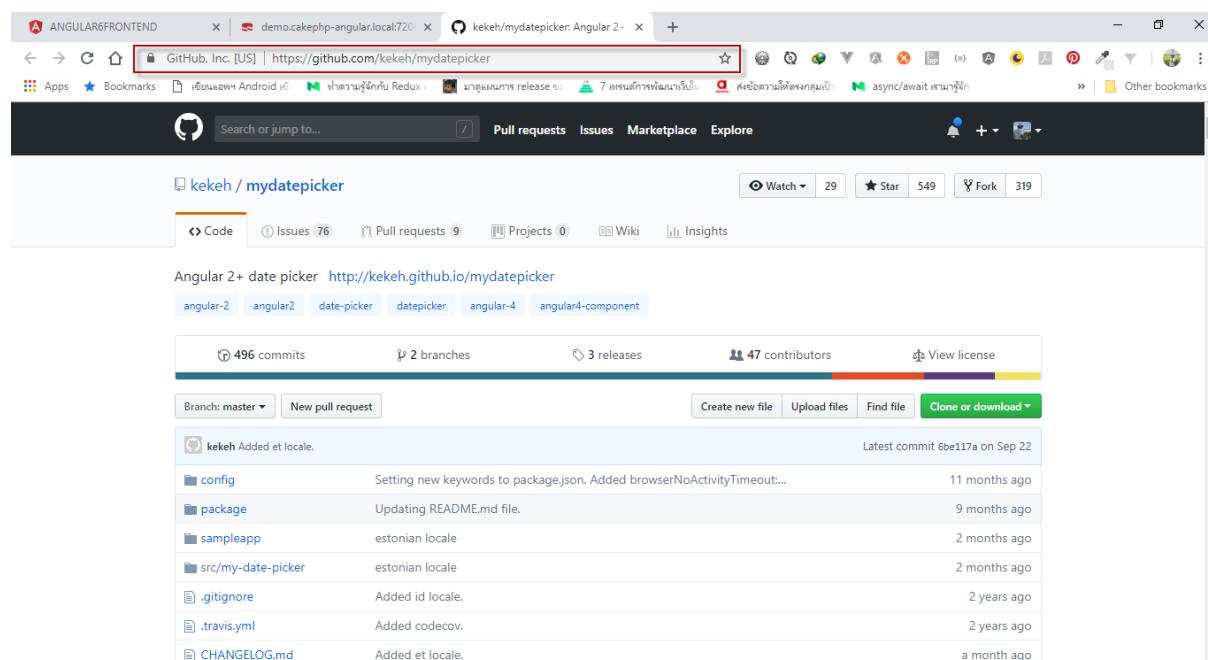


```

13 |     * @license https://opensource.org/licenses/mit-license.php MIT License
14 |
15 |     namespace App\Controller;
16 |
17 |     use Cake\Controller\Controller;
18 |     use Cake\Event\Event;
19 |
20 |     header('Access-Control-Allow-Origin: http://localhost:4200');
21 |     header('Access-Control-Allow-Headers: Origin, Content-Type, Authorization, X-Auth-Token, X-Requested-With');
22 |     header('Access-Control-Allow-Methods: GET, POST, PUT, PATCH, DELETE, HEAD, OPTIONS');
23 |
24 |     /**

```

Add Date picker plugin to angular project: <https://github.com/kekeh/mydatepicker>



kekeh / mydatepicker

Angular 2+ date picker <http://kekeh.github.io/mydatepicker>

496 commits 2 branches 3 releases 47 contributors View license

Branch: master New pull request Create new file Upload files Find file Clone or download

kekeh Added et locale. Latest commit 6be117a on Sep 22

File	Description	Time Ago
config	Setting new keywords to package.json. Added browserNoActivityTimeout...	11 months ago
package	Updating README.md file.	9 months ago
sampleapp	estonian locale	2 months ago
src/my-date-picker	estonian locale	2 months ago
.gitignore	Added id locale.	2 years ago
.travis.yml	Added codecov.	2 years ago
CHANGELOG.md	Added et locale.	a month ago

Angular Service

Generate users service

```
$ ng g service _services/users --spec false
```

```
File: /Users/.../ANGULAR6-FRONTEND/_services/users.service.ts - ANGULAR6-FRONTEND - Visual Studio Code
$ ng g service _services/users --spec false
src/app/_services/users.service.ts (134 bytes)
```

Adding each line code to app/_services/users.service.ts.

```
File Edit Selection View Go Debug Terminal Help users.service.ts - ANGULAR6-FRONTEND - Visual Studio Code
EXPLORER
ANGULAR6-FRONTEND
  e2e
  node_modules
  src
    app
      _services
        users.service.ts M
        _shared
        data-binding
        home
        register
          register.component.css
          register.component.html M
          register.component.specs M
          register.component.ts
          app-routing.modules.ts
          app.component.css
          app.component.html
          app.component.ts
          app.components.ts
          app.module.ts
        assets
        environments
          environment.prod.ts
          environment.ts
        favicon.ico
        index.html
        karma.conf
OUTLINE
Ln 26, Col 29 Spaces: 2 UTF-8 LF TypeScript 3.1.1 2
```

```
import { Injectable } from '@angular/core';
import { HttpClient, HttpRequest, HttpResponse, HttpHeaders, HttpParams, HttpHeaders } from '@angular/common/http';
import { Http, Response, Headers, RequestOptions } from '@angular/http';
import { Observable } from 'rxjs';
import { environment } from './../../../environments/environment';

@Injectable({
  providedIn: 'root'
})
export class UserService {
  constructor(private http: HttpClient) {}

  //Function load all user are in the api application
  loadAllUsers() {...}

  //Function load a user infomation whare match parameter id
  findUserById(id) {...}

  /**
   * Function add / register for user profile
   * @author sarawut.b
   * @param user as a object of user information
   */
  addUserProfile(user: any) {
    return this.http.post(`${environment.apiUrl}/users/addUserProfile.json`, user);
  }

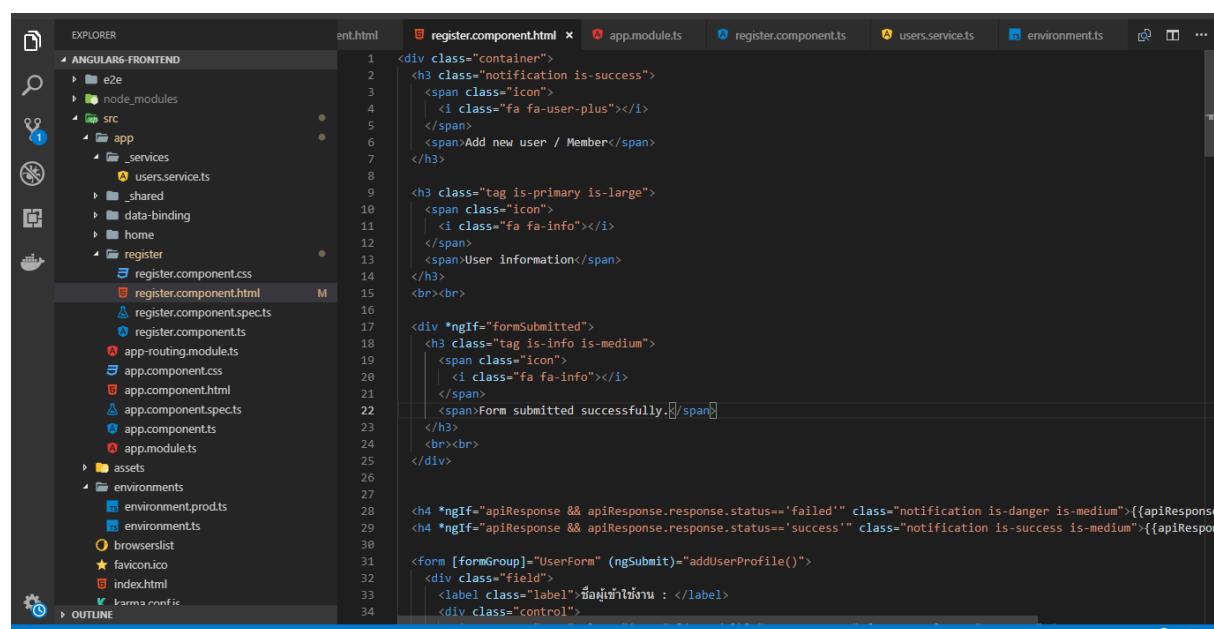
  /**
   * Function update for user profile
   */
}
```

Generate register component

\$ ng g c register --spec false

```
MINGW64:/d/ANGULAR/PAKGON_TRAINING/DEMO_CAKEPHP3_ANGULAR6/ANGULAR6-FRONTEND
$ ng g c register --spec false
CREATE src/app/register/register.component.html (27 bytes)
CREATE src/app/register/register.component.spec.ts (642 bytes)
ERASE src/app/register/register.component.ts (27 bytes)
CREATE src/app/register/register.component.css (0 bytes)
UPDATE src/app/app.module.ts (861 bytes)
OK_Antikorn@DESKTOP-3DAZQ21 MINGW64 /d/ANGULAR/PAKGON_TRAINING/DEMO_CAKEPHP3_ANGULAR6/ANGULAR6-FRONTEND (master)
$
```

Adding each line code to app/register/register.component.html



```

<div class="container">
  <h3 class="notification is-success">
    <span class="icon">
      | <i class="fa fa-user-plus"></i>
    </span>
    <span>Add new user / Member</span>
  </h3>

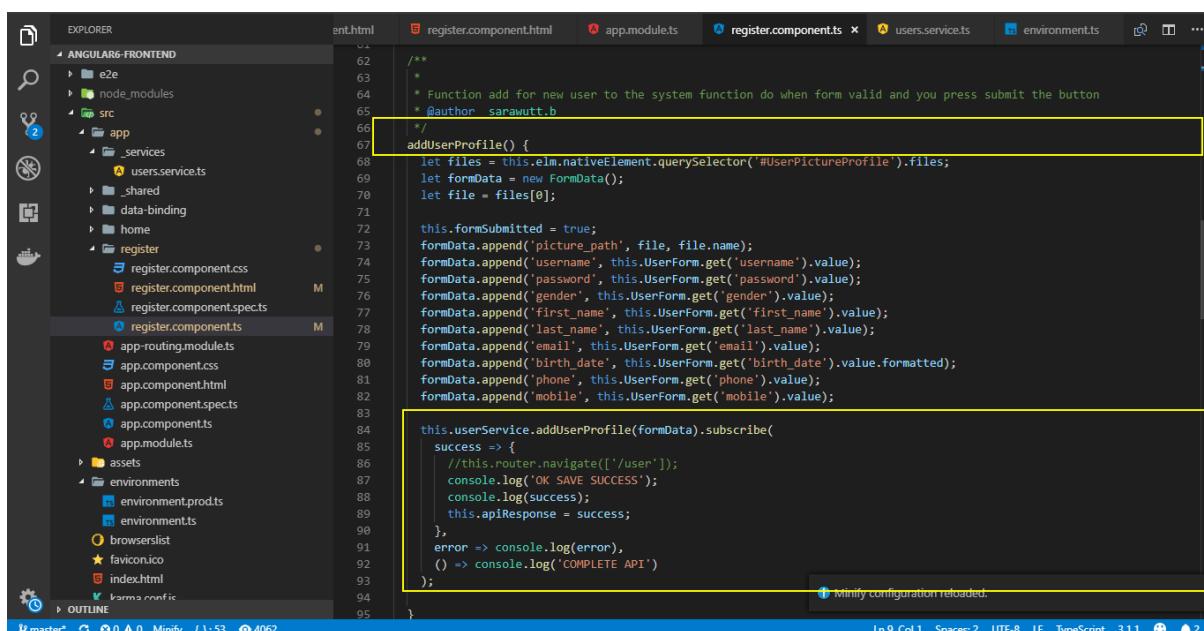
  <h3 class="tag is-primary is-large">
    <span class="icon">
      | <i class="fa fa-info"></i>
    </span>
    <span>User Information</span>
  </h3>
  <br><br>

  <div *ngIf="formSubmitted">
    <h3 class="tag is-info is-medium">
      <span class="icon">
        | <i class="fa fa-info"></i>
      </span>
      <span>Form submitted successfully.</span>
    </h3>
    <br><br>
  </div>

  <h4 *ngIf="apiResponse && apiResponse.response.status=='failed'" class="notification is-danger is-medium">{{apiResponse.response.message}}
  <h4 *ngIf="apiResponse && apiResponse.response.status=='success'" class="notification is-success is-medium">{{apiResponse.response.message}}
</h4>

<form [formGroup]="UserForm" (ngSubmit)="addUserProfile()">
  <div class="field">
    <label class="label">ชื่อผู้ใช้:</label>
    <div class="control">
```

Adding each line code to app/register/register.component.ts



```

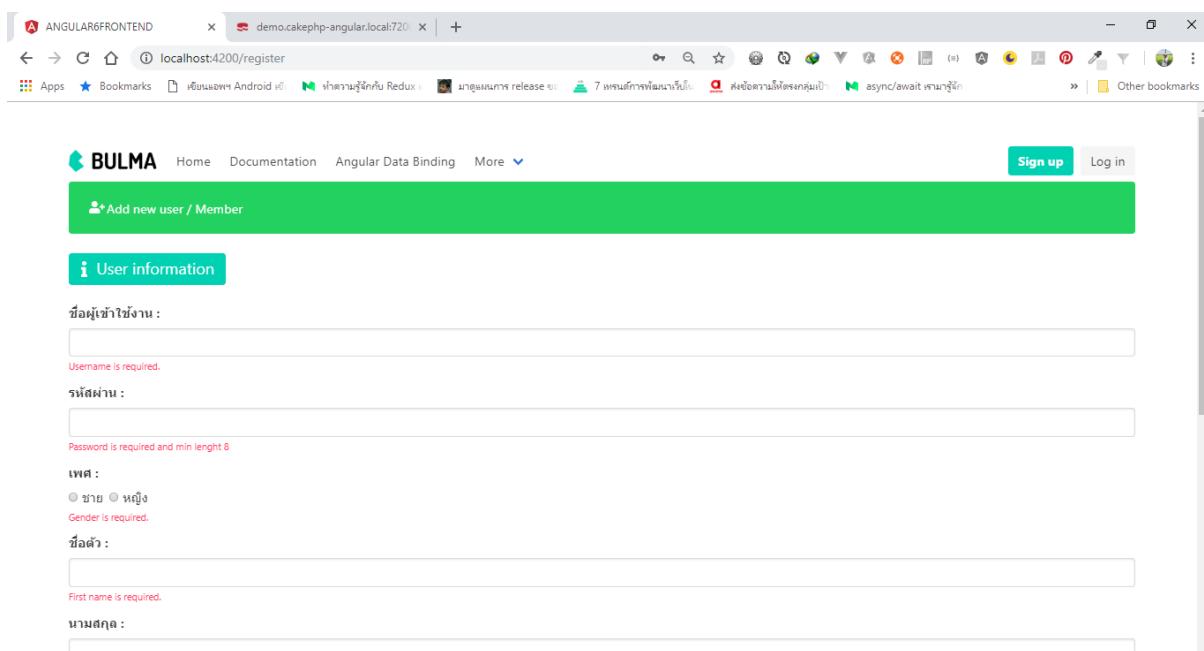
    /**
 * Function add for new user to the system function do when form valid and you press submit the button
 * @author saramut.b
 */
addUserProfile() {
  let files = this.elm.nativeElement.querySelector('#UserPictureProfile').files;
  let formData = new FormData();
  let file = files[0];

  this.formSubmitted = true;
  formData.append('picture_path', file, file.name);
  formData.append('username', this.UserForm.get('username').value);
  formData.append('password', this.UserForm.get('password').value);
  formData.append('gender', this.UserForm.get('gender').value);
  formData.append('first_name', this.UserForm.get('first_name').value);
  formData.append('last_name', this.UserForm.get('last_name').value);
  formData.append('email', this.UserForm.get('email').value);
  formData.append('birth_date', this.UserForm.get('birth_date').value.formatted);
  formData.append('phone', this.UserForm.get('phone').value);
  formData.append('mobile', this.UserForm.get('mobile').value);

  this.userService.addUserProfile(formData).subscribe(
    success => {
      //this.router.navigate(['/user']);
      console.log('OK SAVE SUCCESS');
      console.log(success);
      this.apiResponse = success;
    },
    error => console.log(error),
    () => console.log('COMPLETE API')
  );
}

```

Open your browser:



Add function addUserProfile() on to UsersController.php code each bellow.

```

109 * @author sarawutt.b
110 */
111 public function addUserProfile() {
112     $response = ['message' => __('Initial for add a user profile'), 'status' => 'failed'];
113     if ($this->request->is('post')) {
114         $params = $this->request->getData();
115         if (!($this->Users->exists(['username' => $params['username']]))) {
116             $user = $this->Users->newEntity();
117
118             $resultOK = []; //Keep file handler
119             $file_path = "";
120             $file_path = 'uploadfile/img';
121             if (!empty($params['picture_path']['name'])) {
122                 $resultOK = $this->uploadFiles(null, $file_path, $params['picture_path']);
123                 if (isset($resultOK['errors'])) {
124                     $response = ['message' => __("({$resultOK['errors'][0]}). Please, try again."), 'status' => 'failed'];
125                 }
126             }
127
128             $params['picture_path'] = $resultOK['uploadPaths'][0];
129             $params['create_uid'] = 99;
130             $user = $this->Users->patchEntity($user, $params);
131             if ($this->Users->save($user)) {
132                 $response = ['message' => __('The user has been saved.'), 'status' => 'success'];
133             } else {
134                 $response = ['message' => __('The username %s is already exist. Please try again.', $params['username']), 'status' => 'failed'];
135             }
136         }
137     }
138     $this->set(compact('response'));
139     $this->set('_serialize', ['response']);
140 }

```

HW1:

1. Adding for the link for signup button navigate to register page.

Add Toaster plugin to angular project:

<https://www.npmjs.com/package/ng6-toastr-notifications>

\$ npm install ng6-toastr-notifications --save

```

OK_Antikorn@DESKTOP-IDAJD21_MTNOW64 /d/ANGULAR/PAKGON_TRAINING/DEMO_CAKEPHP3_ANGULAR6/ANGULAR6-FRONTEND (master)
$ npm install ng6-toastr-notifications --save
[!] WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.4 (node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.4: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"})
+ ng6-toastr-notifications@1.0.4
added 1 package in 12.39s
[+] no known vulnerabilities found [39016 packages audited]

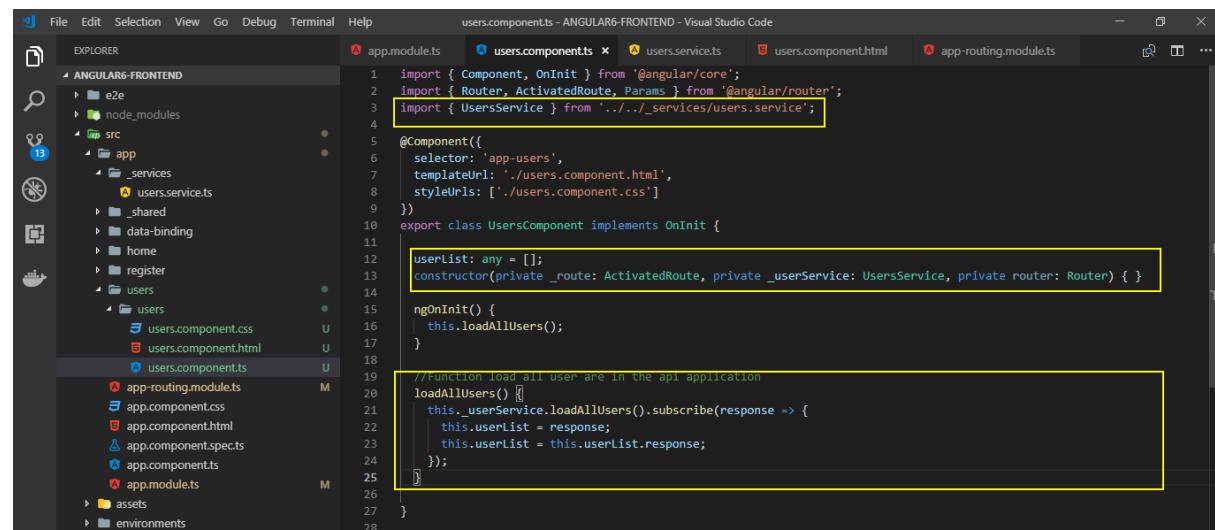
```

Generate for users component

```
$ ng g c users/users --spec false
```

```
$ ng g c users/users --spec false
src/app/users/users.component.ts (265 bytes)
CREATE src/app/users/users.component.ts (265 bytes)
CREATE src/app/users/users.component.css (0 bytes)
UPDATE src/app/app.module.ts (1427 bytes)
```

Add component code load user information on to app/users/users.component.ts



The screenshot shows the Visual Studio Code interface with the following details:

- Project Structure:** The left sidebar shows the project structure under "ANGULAR6-FRONTEND". It includes folders like e2e, node_modules, src, and app. The app folder contains sub-folders _services, _shared, home, register, and users. The users folder contains files users.component.css, users.component.html, and users.component.ts.
- Code Editor:** The main editor area displays the content of users.component.ts. The code imports necessary modules and defines a component "app-users" that loads all users from a service. A function "loadAllUsers()" uses RxJS to subscribe to the service's response and update the component's state.

```

1 import { Component, OnInit } from '@angular/core';
2 import { Router, ActivatedRoute, Params } from '@angular/router';
3 import { UserService } from '../../../../../_services/users.service';
4
5 @Component({
6   selector: 'app-users',
7   templateUrl: './users.component.html',
8   styleUrls: ['./users.component.css']
9 })
10 export class UsersComponent implements OnInit {
11
12   userList: any = [];
13   constructor(private _route: ActivatedRoute, private _userService: UserService, private router: Router) { }
14
15   ngOnInit() {
16     this.loadAllUsers();
17   }
18
19   //Function load all user are in the api application
20   loadAllUsers() {
21     this._userService.loadAllUsers().subscribe(response => {
22       this.userList = response;
23       this.userList = this.userList.response;
24     });
25   }
26
27 }
28

```

Add user component template on file app/users/users.component.html

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows the project structure under "ANGULAR6-FRONTEND". Key files include app.module.ts, users.service.ts, users.component.ts, users.component.html, users.component.css, users.component.tsx, app-routing.module.ts, app.component.ts, app.component.html, app.component.spec.ts, and app.component.tsx.
- Code Editor:** The active file is "users.component.html". The code is a template for a user list page, featuring a header with a warning notification, a search bar, and a table with columns for user ID, first name, last name, email, and actions (Edit, Delete). It includes Angular directives like *ngFor, *ngIf, and routerLink.
- Bottom Status Bar:** Displays file statistics: Line 11, Col 73, Spaces: 2, UTF-8, LF, HTML, and a file icon.

HW2:

1. Add users service (type script) for load all user information.

Add CakePHP loadAllUsers() function UsersController.php for service users data.

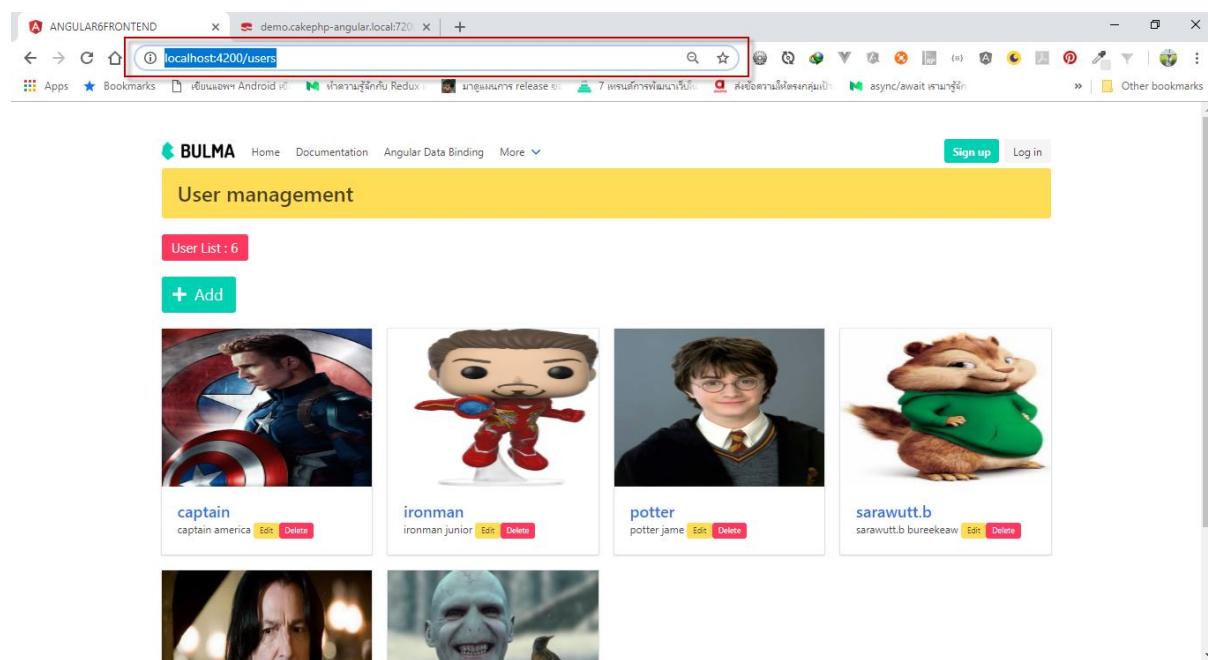
The screenshot shows a code editor with the following details:

- Code Editor:** The file is "UserController.php". It contains a PHP class "UserController" with a public method "loadAllUsers()". The method uses the CakePHP \$this->Users->find() method to query the database and return all users, ordered by username, first name, and last name.
- Code Content:**

```

84 /**
85  * Function list all user on the system
86  * @author sarawutt.b
87  */
88
89 public function loadAllUsers() {
90     $response = $this->Users->find()->order(['username'=>'asc','first_name'=>'asc','last_name'=>'asc'])->all();
91     $this->set(['response' => $response, '_serialize' => ['response']]);
92 }
93 
```

And then open your browser:



User Filter with pipe

\$ ng g pipe users/pipe/user-filter --spec=false

```
OK Artilorin@DESKTOP-1D42021 MINGW64 /d/ANGULAR/PAKGON TRAINING/DEMO CAKEPHP3 ANGULAR6/ANGULAR6-FRONTEND (master)
$ ng g pipe users/pipe/user-filter --spec=false
CREATE src/app/users/pipe/user-filter.pipe.spec.ts (204 bytes)
CREATE src/app/users/pipe/user-filter.pipe.ts (209 bytes)
UPDATE src/app/app.module.ts (1409 bytes)
```

Open app/users/pipe/user-filter.pipe.ts and then adding for each code.

```
import { Pipe, PipeTransform } from '@angular/core';

@Pipe({
  name: 'userFilter'
})
export class UserFilterPipe implements PipeTransform {

  /**
   * Function user filter by user property
   * @author sarawutt.b
   * @param value as array of user Object
   * @param filterBy as a string of search key
   */
  transform(value: any, filterBy: string): any {
    filterBy = filterBy ? filterBy.toLocaleLowerCase() : null;
    return filterBy ? value.filter(data => {
      return data.username.toString().toLocaleLowerCase().includes(filterBy) ||
        data.first_name.toLocaleLowerCase().includes(filterBy) ||
        data.last_name.toLocaleLowerCase().includes(filterBy)
    }) : value;
  }
}
```

Open app/users/users.component.html

```

<div class="container">
  <div class="notification is-warning title is-3">
    User management
  </div>
  <div class="tag is-danger is-large sub-title is-5">User List : {{userList.length}}</div>
  <br><br>

  <div class="columns">
    <div class="column">
      <form #userForm="ngForm" (ngSubmit)="searchUser()">
        <div class="field has-addons">
          <div class="control is-expanded">
            <div class="control">
              <input type="text" class="input" name="searchTxt" [(ngModel)]="searchTxt" #name="ngModel">
            </div>
          </div>
          <div class="control">
            <button type="submit" class="button is-info [disabled]={{userForm.form.invalid}}">
              <span class="icon">
                <i class="fa fa-search">/i
              </span>
              <span>Search</span>
            </button>
          </div>
        </div>
      </form>
    </div>
  </div>

  <div class="columns">...
  </div>

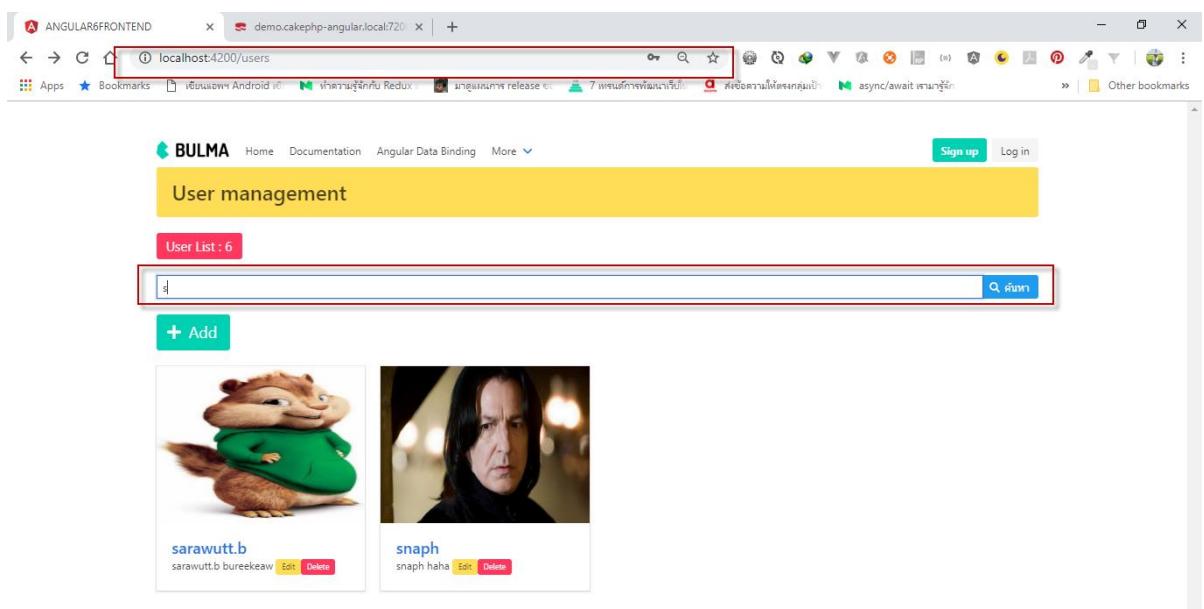
  <div class="columns is-multiline">
    <div class="column is-one-quarter" *ngFor="let user of userList | userFilter: searchTxt;">
      <div class="card">
        <div class="card-image">
          <figure class="image is-4by3">
            
          </figure>
        </div>
      </div>
    </div>
  </div>

```

Open app/users/users.component.ts and then adding line to the component.

```
searchTxt: string;
```

Open your browser:

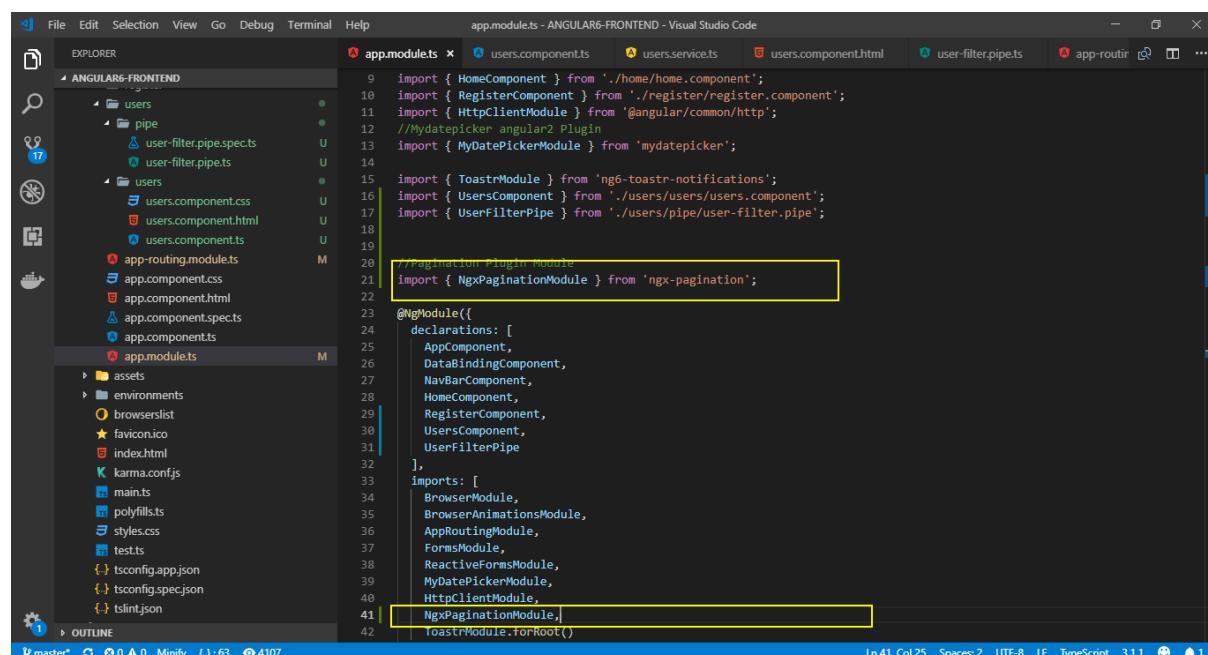


How to Pagination and Paging

\$ npm install ngx-pagination --save

```
OK Antikorn@DESKTOP-3D42031 MINGW64 /d/ANGULAR/PAKGON_TRAINING/DEMO_CAKEPHP3_ANGULAR6/ANGULAR6-FRONTEND (master)
$ npm install ngx-pagination --save
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.4 (node_modules\fs-events):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.4: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"})
+ ngx-pagination@3.2.1
added 1 package from 1 contributor in 10.65s
[+] no known vulnerabilities found [39015 packages audited]
```

And then open app/app.module.ts and adding code import class.



```
File Edit Selection View Go Debug Terminal Help app.module.ts - ANGULAR6-FRONTEND - Visual Studio Code
EXPLORER
ANGULAR6-FRONTEND
  users
    pipe
      user-filter.pipe.specs
      user-filter.pipe.ts
    users
      users.component.css
      users.component.html
      users.component.ts
    app-routing.module.ts
    app.component.css
    app.component.html
    app.component.specs
    app.component.ts
  app.module.ts
  assets
  environments
  favicon.ico
  index.html
  karma.conf.js
  main.ts
  polyfills.ts
  styles.css
  test.ts
  tsconfig.app.json
  tsconfig.spec.json
  tslint.json
  OUTLINE
app.module.ts
9   import { HomeComponent } from './home/home.component';
10  import { RegisterComponent } from './register/register.component';
11  import { HttpClientModule } from '@angular/common/http';
12 //Mydatepicker angular2 Plugin
13  import { MyDatePickerModule } from 'mydatepicker';
14
15  import { ToastrModule } from 'ng6-toastr-notifications';
16  import { UsersComponent } from './users/users/users.component';
17  import { UserFilterPipe } from './users/pipe/user-filter.pipe';
18
19 //Pagination Plugin module
20  import { NgxPaginationModule } from 'ngx-pagination';
21
22 @NgModule({
23   declarations: [
24     AppComponent,
25     DataBindingComponent,
26     NavBarComponent,
27     HomeComponent,
28     RegisterComponent,
29     UsersComponent,
30     UserFilterPipe
31   ],
32   imports: [
33     BrowserModule,
34     BrowserAnimationsModule,
35     AppRoutingModule,
36     FormsModule,
37     ReactiveFormsModule,
38     MyDatePickerModule,
39     HttpClientModule,
40     NgxPaginationModule,
41   ],
42   providers: [
43     ToastrModule.forRoot()
44   ]
45 })
```

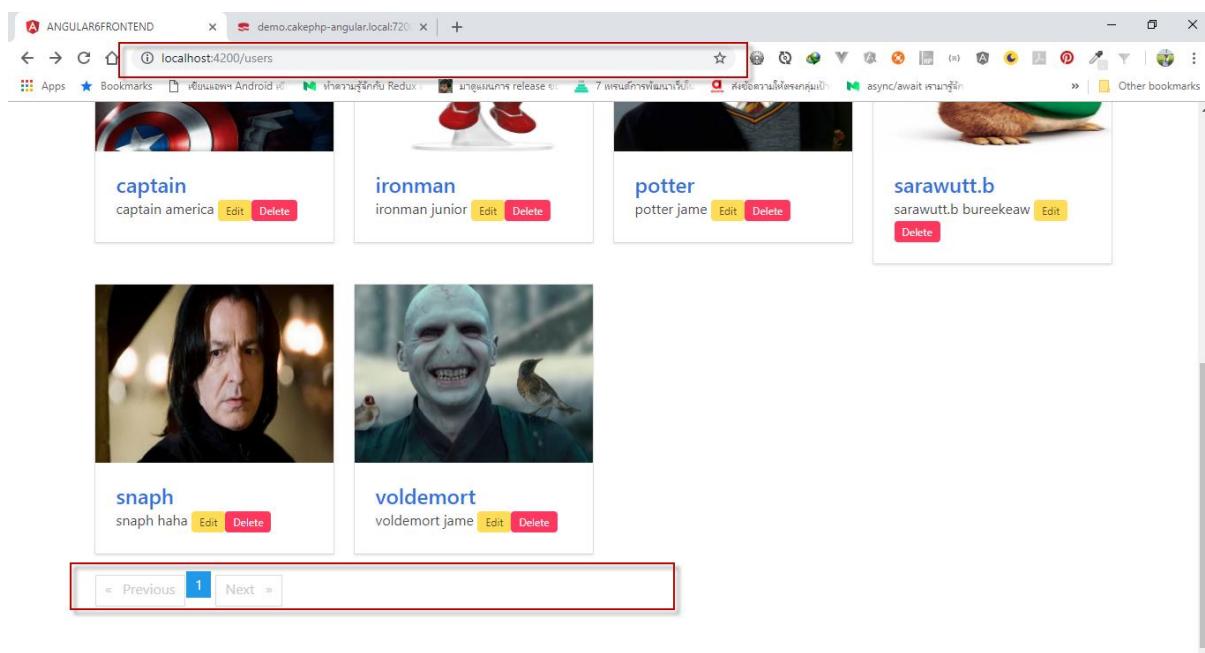
Open app/users/users/users.component.html and then adding each below line.

```
47
48 <div class="columns is-multiline">
49   <div class="column is-one-quarter" *ngFor="let user of userList | userFilter: searchText | paginate: {itemsPerPage: 10, currentPage: p}">
```

And then add pagination in bottom in the same file.

```
100
101 <div class="columns">
102   <div class="column">
103     <!-- Pagination Page -->
104     <pagination-controls class="pagination is-right" role="navigation" aria-label="pagination" (pageChange)="p = $event"></pagination-controls>
105   </div>
106 </div>
107
108 </div>
```

Open your browser:

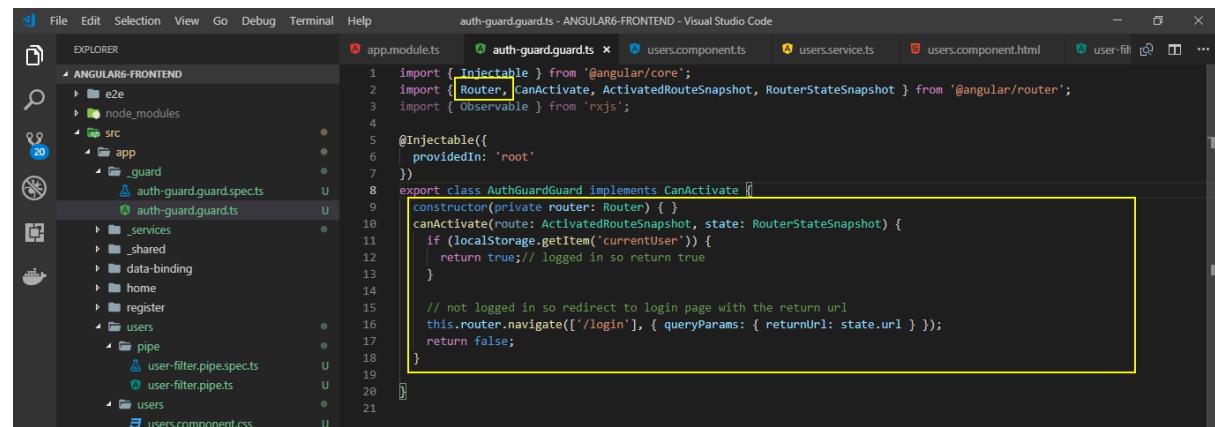


Authentication and Authorization

```
$ ng g guard _guard/auth-guard --spac=false
```

```
  ng g guard _guard/auth-guard --spec=false  
  ng g guard _guard/auth-guard.spec.ts  
CREATE src/app/_guard/auth-guard.guard.ts (419 bytes)
```

Open app/_guard/auth.guard.ts and then adding each below source code.



Generate service for Authentication service

```
$ ng g service _services/authentication --spec=false
```

Open app/_services/authentication.service.ts and then adding each line.

```

import { Injectable } from '@angular/core';
import { Http, Headers, Response } from '@angular/http';
import { HttpClient, HttpRequest, HttpHeaders, HttpResponse, HttpParams, HttpHeaders } from '@angular/common/http';
import { map, filter, catchError, mergeMap } from 'rxjs/operators';
import { environment } from '../../../../../environments/environment';

@Injectable({
  providedIn: 'root'
})
export class AuthenticationService {
  constructor(private http: HttpClient) {}

  /**
   * login(username: string, password: string) {
   *   return this.http.post('/${environment.apiUrl}/users/login.json', { username: username, password: password })
   *     .pipe(map(user => {
   *       console.log(user.response.token);
   *       // login successful if there's a jwt token in the response
   *       if (user && user.response.token) {
   *         // store user details and jwt token in local storage to keep user logged in between page refreshes
   *         localStorage.setItem('currentUser', JSON.stringify(user.response.data));
   *       }
   *       return user;
   *     }));
   * }
   */
  logout() {
    // remove user from local storage to log user out
    localStorage.removeItem('currentUser');
  }
}

```

Generate for Login Component

```
$ ng g c login --spec false
```

Open app/login/login.component.html and then typing code on each below line.

```
File Edit Selection View Go Debug Terminal Help login.component.ts - ANGULAR6-FRONTEND - Visual Studio Code

authentication.service.ts users.service.ts users.component.html login.component.html login.component.ts

25 // reset login status
26 this.authenticationService.logout();
27 // get return url from route parameters or default to '/'
28 // this.toastr.successToastr('ລົງທະບຽນສໍາເລັດ', 'Success!');
29 this.returnUrl = this.route.snapshot.queryParams['returnUrl'] || '/';
30 }

31 /**
32 * ...
33 */
34 login() {
35   this.loading = true;
36   this.authenticationService.login(this.model.username, this.model.password).subscribe(
37     user => {
38       console.log(user);
39       let status = user.response.status || 'failed';
40       if (status == 'failed') { (parameter) user: any
41         this.toastr.errorToastr(user.response.message, 'Oops!');
42       } else {
43         this.toastr.successToastr(user.response.message, 'Success!');
44       }
45
46       this.router.navigate([this.returnUrl]);
47     },
48     error => {
49       this.toastr.errorToastr('Bad Request Please try again!', 'Oops!');
50       this.loading = false;
51     });
52   }
53 }

54 }

55 }

56 }

57 }

58 }
```

Open app/app-routing.module.ts and then adding for each line.

```

import { NgModule } from '@angular/core';
import { Routes, RouterModule } from '@angular/router';
import { HomeComponent } from './home/home.component';
import { DataBindingComponent } from './data-binding/data-binding.component';
import { RegisterComponent } from './register/register.component';
import { UsersComponent } from './users/users.component';
import { LoginComponent } from './login/login.component';

const routes: Routes = [
  { path: '', redirectTo: 'home', pathMatch: 'full' },
  { path: 'home', component: HomeComponent, canActivate: [AuthGuardGuard] },
  { path: 'data-binding', component: DataBindingComponent },
  { path: 'register', component: RegisterComponent },
  { path: 'users', component: UsersComponent, canActivate: [AuthGuardGuard] },
  { path: 'login', component: LoginComponent },
  { path: '**', component: HomeComponent }
];

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

```

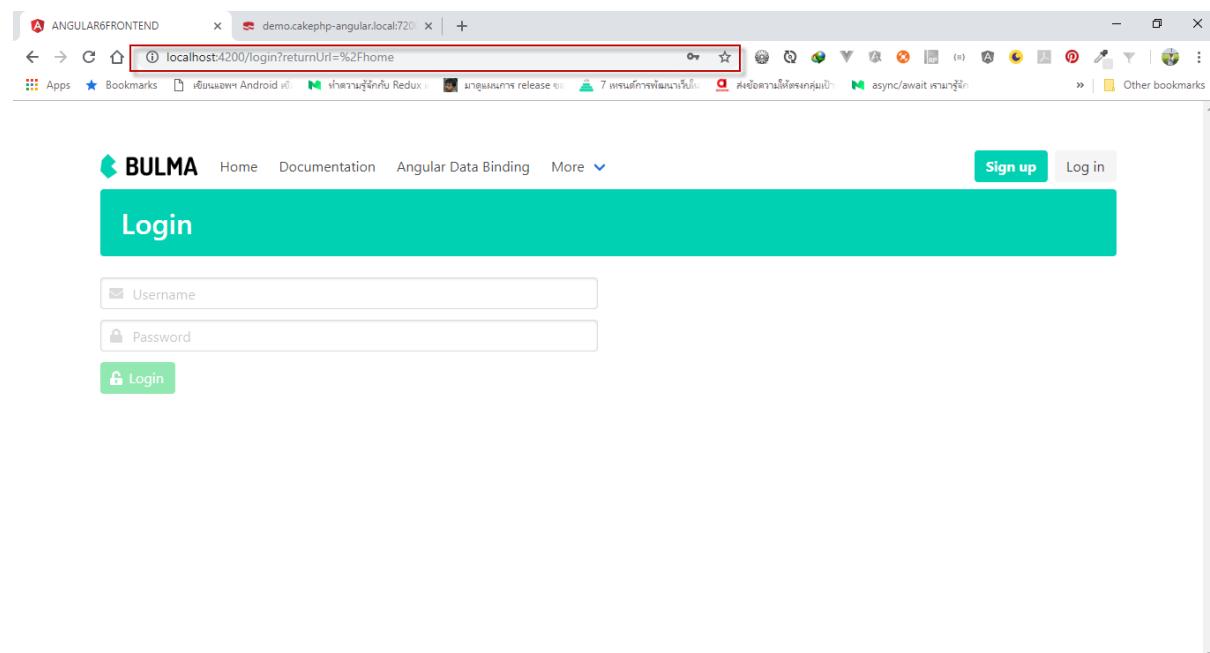
Adding login() function to UsersController.php and then adding each code.

```

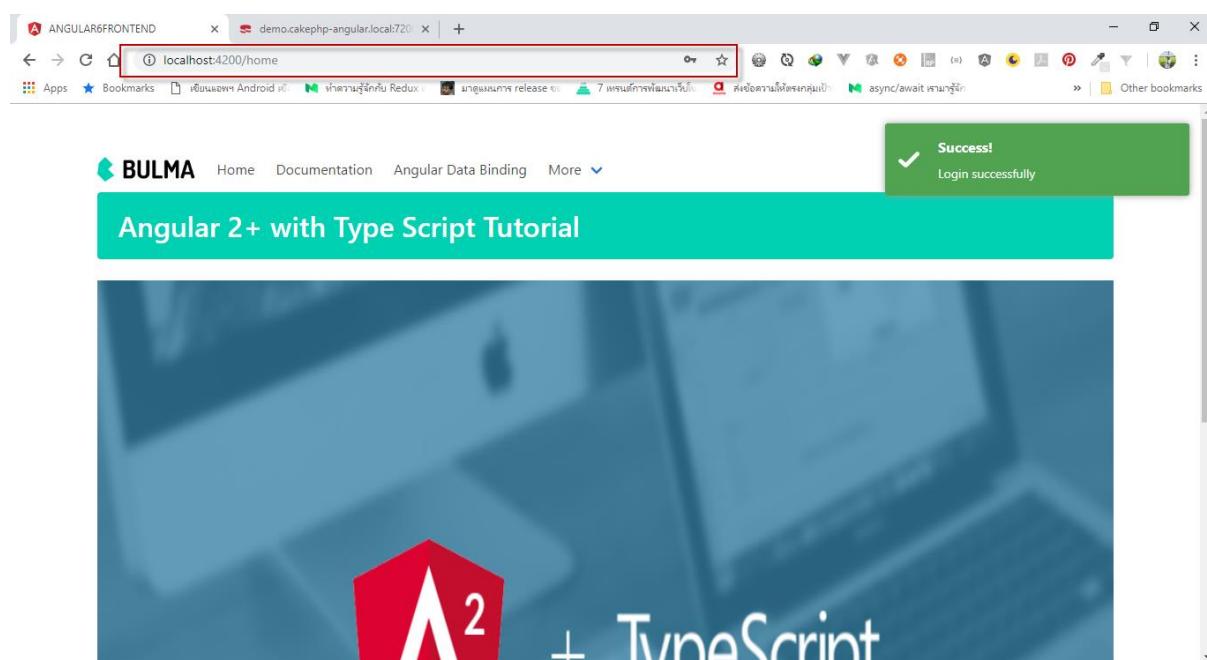
30 |
31 |     /**
32 |     * Login function is Authorized valid login
33 |     * @author Sarawut.b
34 |     * @return void
35 |    */
36 |    public function login() {
37 |        $response = [];
38 |        if ($this->request->is('post')) {
39 |            $user = $this->Auth->identify();
40 |            if ($user) {
41 |                $this->Auth->setUser($user);
42 |                $response = ['status' => 'success', 'message' => __('Login successfully'), 'data' => $this->Auth->user(), 'token' => 'jwt-token'];
43 |            } else {
44 |                $response = ['status' => 'failed', 'message' => __('Invalid username or password Please try again!')];
45 |            }
46 |        }
47 |
48 |        $this->set(compact('response'));
49 |        $this->set('_serialize', ['response']);
50 |

```

Open your browser:



Login page:



Home Page:

HW3

1. Add router link to login button when click navigate to login page.
2. Change Login button and navigate to Logout when already login.

HW4

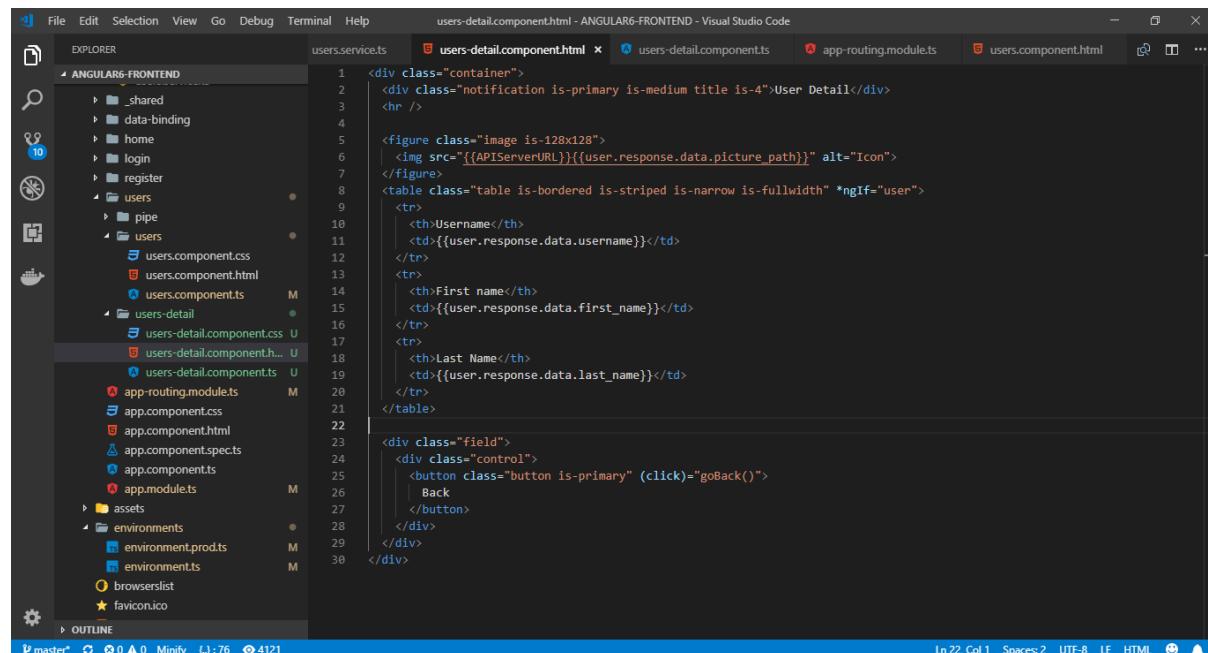
1. User Detail Component.
2. User Edit Component.
3. User Delete.

Generate User Detail Component

\$ ng g c users/user-detail --spec=false

```
OK Antonn@DESKTOP-3D42021 MINGW64 /d/ANGULAR/PAKGON-TRAINING/DEMO_CAKEPHP3_ANGULAR6/ANGULAR6-FRONTEND (master)
$ ng g c users/users-detail --spec=false
  File created: src/app/users/users-detail/user-detail.component.html (31 bytes)
  File created: src/app/users/users-detail/user-detail.component.ts (292 bytes)
  File created: src/app/users/users-detail/user-detail.component.css (0 bytes)
  File updated: src/app/app.module.ts (1705 bytes)
```

Open app/users/user-detail/user-detail.component.html and adding each code.



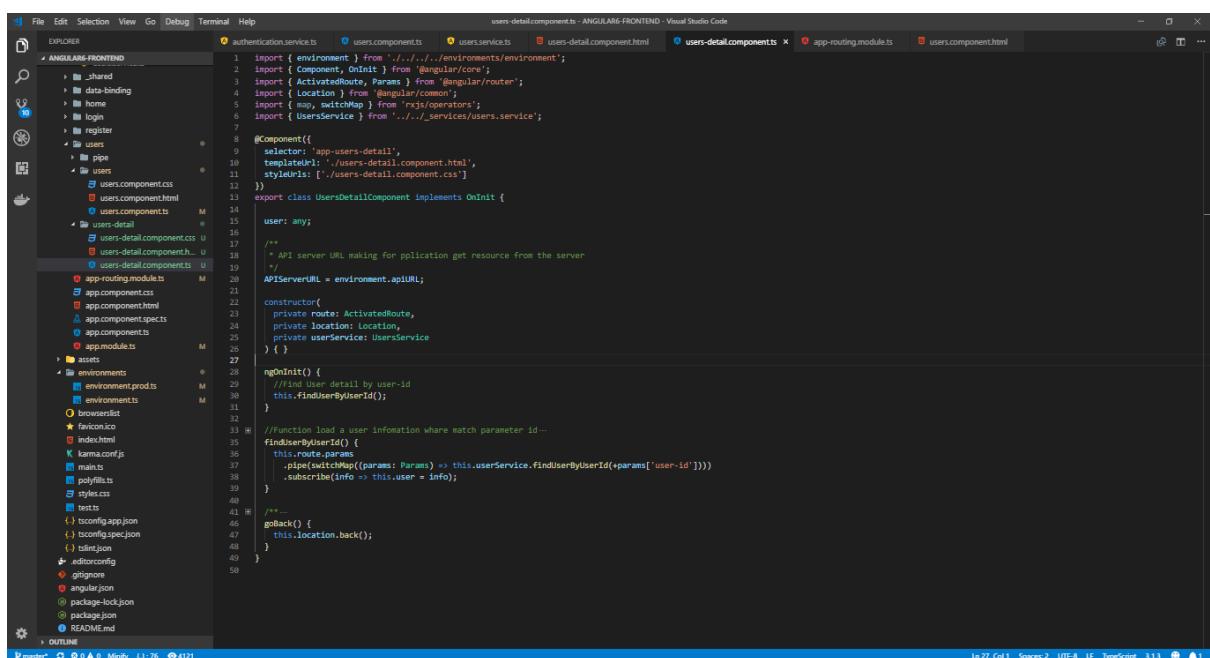
The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer (Left):** Shows the project structure under "ANGULAR6-FRONTEND". The "users" folder contains "users", "pipe", and "users-detail" subfolders. "users-detail" contains "users-detail.component.css", "users-detail.component.html", and "users-detail.component.ts". Other files like "users.service.ts", "app-routing.module.ts", etc., are also listed.
- Code Editor (Center):** Displays the content of "users-detail.component.html". The code includes HTML for a container, notification, image, table, and a back button.
- Status Bar (Bottom):** Shows the current branch ("master"), file count (0), line count (76), character count (4121), and encoding ("UTF-8").

```
<div class="container">
  <div class="notification is-primary is-medium title is-4">User Detail</div>
  <hr />
  <figure class="image is-128x128">
    
  </figure>
  <table class="table is-bordered is-striped is-narrow is-fullwidth" *ngIf="user">
    <tr>
      <th>Username</th>
      <td>{{user.response.data.username}}</td>
    </tr>
    <tr>
      <th>First name</th>
      <td>{{user.response.data.first_name}}</td>
    </tr>
    <tr>
      <th>Last Name</th>
      <td>{{user.response.data.last_name}}</td>
    </tr>
  </table>
  <div class="field">
    <div class="control">
      <button class="button is-primary" (click)="goBack()">
        Back
      </button>
    </div>
  </div>
</div>
```

WORKSHOP ANGULAR 6+ FRONT WITH CAKEPHP3+ BACKEND VERSION 1.0

Open app/users/user-detail/user-detail.component.ts and then adding code block.

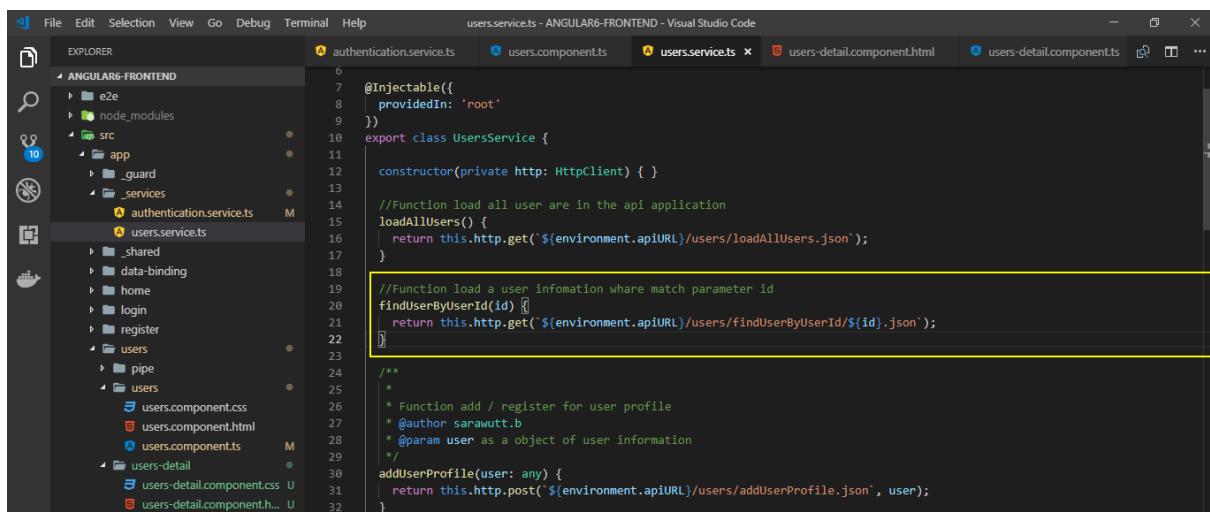


```

1 import { environment } from '../../../../../environments/environment';
2 import { ActivatedRoute, Params } from '@angular/router';
3 import { Location } from '@angular/common';
4 import { map, switchMap } from 'rxjs/operators';
5 import { UserService } from '../../../../../services/users.service';
6
7 @Component({
8   selector: 'app-users-detail',
9   templateUrl: './users-detail.component.html',
10  styleUrls: ['./users-detail.component.css']
11})
12 export class UsersDetailComponent implements OnInit {
13
14   user: any;
15
16   /**
17    * API server URL making for application get resource from the server
18   */
19   APIServerURL = environment.apiURL;
20
21   constructor(
22     private route: ActivatedRoute,
23     private location: Location,
24     private userService: UserService
25   ) {}
26
27   ngOnInit() {
28     //Find user detail by user-id
29     this.findUserByUserId();
30   }
31
32   //Function load a user infomation whare match parameter id
33   findUserById(id) {
34     this.userService
35       .findUserById(params: Params)
36       .pipe(switchMap((params: Params) => this.userService.findUserByUserId(+params['user-id'])))
37       .subscribe(info => this.user = info);
38   }
39
40   /**
41    * goBack()
42    */
43   goBack() {
44     this.location.back();
45   }
46
47 }
48
49 }
50

```

Open app/_services/users.service.ts and adding code block.

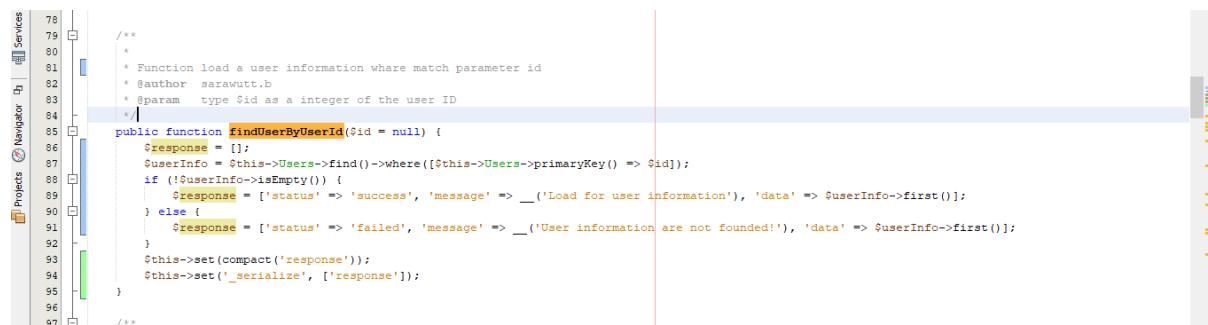


```

1 import { Injectable } from '@angular/core';
2 import { HttpClient } from '@angular/common/http';
3 import { Observable } from 'rxjs';
4
5 @Injectable({
6   providedIn: 'root'
7 })
8 export class UserService {
9
10   constructor(private http: HttpClient) { }
11
12   //Function load all user are in the api application
13   loadAllUsers(): Observable<any> {
14     return this.http.get(`${environment.apiURL}/users/loadAllUsers.json`);
15   }
16
17   //Function load a user infomation whare match parameter id
18   findUserById(id): Observable<any> {
19     return this.http.get(`${environment.apiURL}/users/findUserByUserId${id}.json`);
20   }
21
22   /**
23    *
24    * Function add / register for user profile
25    * @author sarawut.b
26    * @param user as a object of user information
27    */
28   addUserProfile(user: any): Observable<any> {
29     return this.http.post(`${environment.apiURL}/users/addUserProfile.json`, user);
30   }
31
32 }

```

CakePHP backend add findUserById() function into UsersController.php

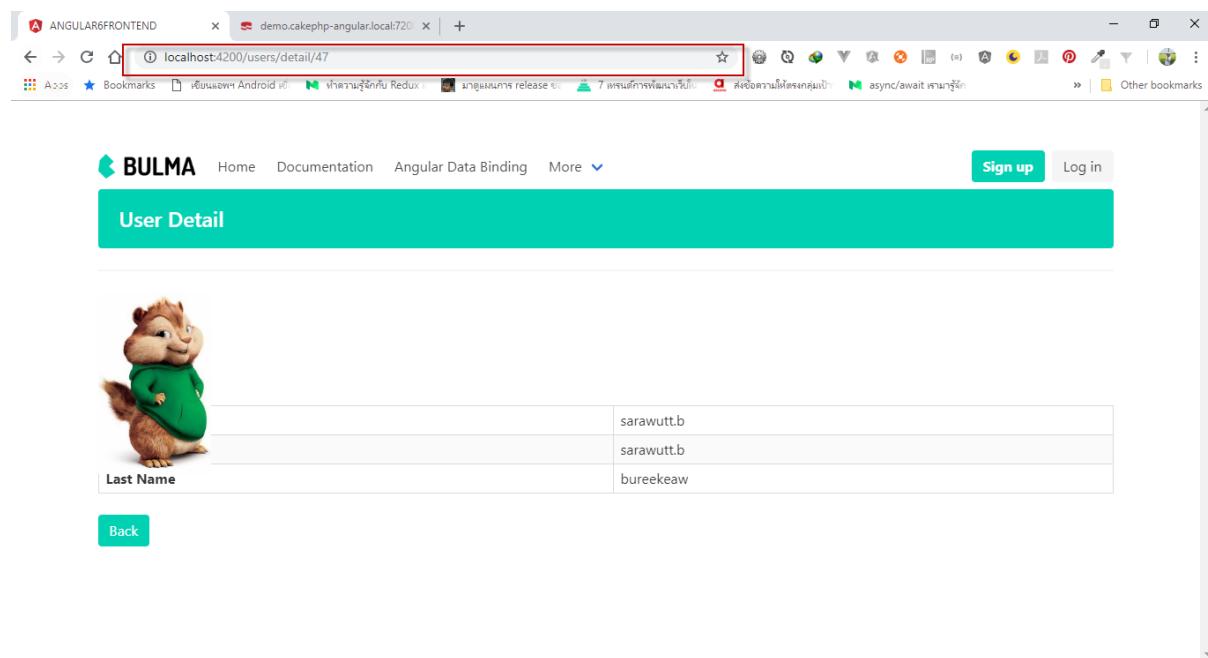


```

78 /**
79 *
80 * Function load a user information where match parameter id
81 * @author sarawutt.b
82 * @param type $id as a integer of the user ID
83 */
84
85 public function findUserById($id = null) {
86     $response = [];
87     $userInfo = $this->Users->find()->where([$this->Users->primaryKey() => $id]);
88
89     if (!$userInfo->isEmpty()) {
90         $response = ['status' => 'success', 'message' => __('Load for user information'), 'data' => $userInfo->first()];
91     } else {
92         $response = ['status' => 'failed', 'message' => __('User information are not founded!'), 'data' => $userInfo->first()];
93     }
94     $this->set(compact('response'));
95     $this->set('_serialize', ['response']);
96 }
97 /**

```

Open your browser:



The screenshot shows a web browser window with the URL localhost:4200/users/detail/4. The page title is "User Detail". The user information displayed is:

First Name	sarawutt.b
Last Name	bureekaw

Generate Users Edit Component

\$ ng g c users/users-edit --spec false

```

D:\K\ht\korn\Desktop\3DA2022\WING64\4\ANGULAR\PAKGON_TRAINING\DEMO_CAKEPHP3_ANGULAR6\ANGULAR6-FRONTEND (master)
$ ng g c users/users-edit --spec false
CREATE src/app/users/users-edit/users-edit.component.html (29 bytes)
CREATE src/app/users/users-edit/users-edit.component.ts (284 bytes)
CREATE src/app/users/users-edit/users-edit.component.css (0 bytes)
UPDATE src/app/app.module.ts (1807 bytes)

```

WORKSHOP ANGULAR 6+ FRONT WITH CAKEPHP3+ BACKEND VERSION 1.0

Open app/users/users-edit/users-edit.component.html and adding on each file.

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows the project structure under "ANGULAR6-FRONTEND". The "users-edit" folder is selected.
- Editor:** The file "users-edit.component.html" is open, displaying its content. The code includes HTML for notifications, form submission handling, and a form group for user information.
- Bottom Status Bar:** Shows "Ln 11, Col 36" and other standard status bar information.

```

<div class="container">
  <h3 class="notification is-primary">Update User profile / Member profile</h3>
  <h3 class="tag is-danger is-large">User information</h3>
  <br><br>

  <div *ngIf="formSubmitted">
    <h3 class="tag is-info is-medium">
      <span class="icon">
        | <i class="fa fa-info"></i>
      </span>
    <span>Form submitted successfully.</span>
  </div>

  <h4 *ngIf="apiResponse && apiResponse.response.status=='failed'" class="notification is-danger is-medium">{{apiResponse}}
  <h4 *ngIf="apiResponse && apiResponse.response.status=='success'" class="notification is-success is-medium">{{apiResponse}}
  <br><br>
  <form [formGroup]="UserForm" (ngSubmit)="onUpdateFormSubmit()">

    <input type="hidden" [(ngModel)]="user.id" formControlName="id" />
    <div class="field">
      <label class="label">ชื่อผู้ใช้งาน : </label>
      <div class="control">
        | | <input type="text" class="input" [(ngModel)]="user.username" formControlName="username" />
      </div>
      <p *ngIf="UserForm.get('username').invalid" [ngClass]="'help is-danger'">Username is required.</p>
    </div>

    <div class="field">
      <label class="label">นามสกุล : </label>
      <div class="control">
        | | <input type="text" class="input" [(ngModel)]="user.first_name" formControlName="first_name" />
      </div>
    </div>
  </form>

```

Open app/user/user-edit/user-edit.component.ts and then adding code.

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows the project structure under "ANGULAR6-FRONTEND". The "users-edit" folder is selected.
- Editor:** The file "user-edit.component.ts" is open, displaying its content. It defines a component for user editing, including imports for Angular modules and services, and a constructor for setting up the route and location.
- Bottom Status Bar:** Shows "Ln 12, Col 3" and other standard status bar information.

```

import { Component, OnInit } from '@angular/core';
import { FormGroup, FormControl, Validators } from '@angular/forms';
import { Router, ActivatedRoute } from '@angular/router';
import { Location } from '@angular/common';
import { UserService } from '../../_services/users.service';
import { ToastrManager } from 'ng6-toastr-notifications';

@Component({
  selector: 'app-user-edit',
  templateUrl: './user-edit.component.html',
  styleUrls: ['./user-edit.component.css']
})
export class UserEditComponent implements OnInit {

  formSubmitted = false;
  apiResponse: any;

  updateStatus: Boolean = false;
  user: any = { id: null, username: '', first_name: '', last_name: '', email: '', phone: '', mobile: '' };

  UserForm = new FormGroup({
    id: new FormControl(null, Validators.required),
    username: new FormControl(null, [Validators.required, Validators.maxLength(10)]),
    first_name: new FormControl(null, Validators.required),
    last_name: new FormControl(null, Validators.required),
    email: new FormControl(null, [Validators.required, Validators.email]),
    phone: new FormControl(),
    mobile: new FormControl(),
  });

  constructor(
    private route: ActivatedRoute,
    private router: Router,
    private location: Location,
  ) {
  }
}

```

Open app/_services/users.service.ts and adding below code.

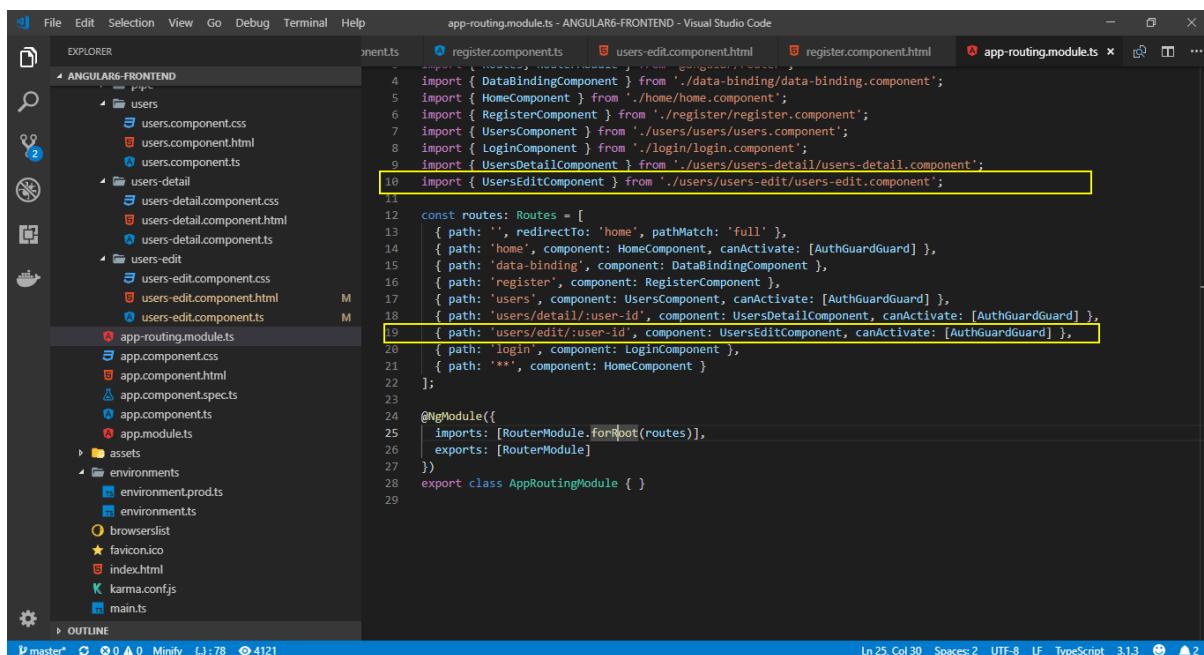


```

register.component.css      34  /**
35   *
36   * Function update for user profile
37   * @author sarawuttb.b
38   * @param user as a object of user information
39   */
40 updateUserProfile(user: any) {
41   return this.http.put(`${environment.apiUrl}/users/updateUserProfile/${user.id}.json`, user);
42 }
43

```

Open app/app-routing.ts and adding code for route to user edit component.



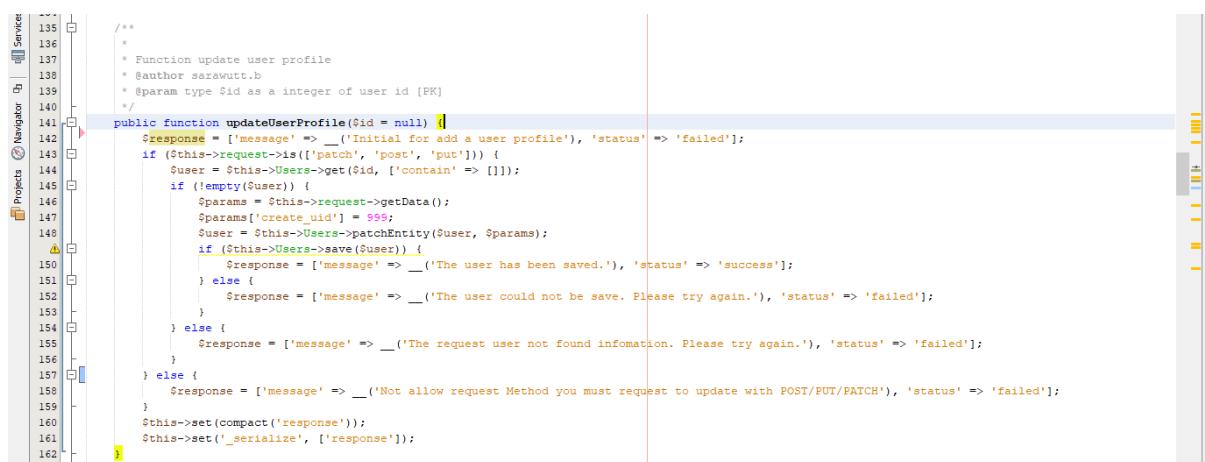
```

File Edit Selection View Go Debug Terminal Help
app-routing.module.ts - ANGULAR6-FRONTEND - Visual Studio Code
EXPLORER register.component.ts users-edit.component.html register.component.html app-routing.module.ts ...
1 import { NgModule } from '@angular/core';
2 import { RouterModule, Routes } from '@angular/router';
3 import { HomeComponent } from './home/home.component';
4 import { DataBindingComponent } from './data-binding/data-binding.component';
5 import { LoginComponent } from './login/login.component';
6 import { RegisterComponent } from './register/register.component';
7 import { UsersComponent } from './users/users/users.component';
8 import { UsersDetailComponent } from './users/users-detail/users-detail.component';
9 import { UsersEditComponent } from './users/users-edit/users-edit.component';
10
11 const routes: Routes = [
12   { path: '', redirectTo: 'home', pathMatch: 'full' },
13   { path: 'home', component: HomeComponent, canActivate: [AuthGuardGuard] },
14   { path: 'data-binding', component: DataBindingComponent },
15   { path: 'register', component: RegisterComponent },
16   { path: 'users', component: UsersComponent, canActivate: [AuthGuardGuard] },
17   { path: 'users/detail/:user-id', component: UsersDetailComponent, canActivate: [AuthGuardGuard] },
18   { path: 'users/edit/:user-id', component: UsersEditComponent, canActivate: [AuthGuardGuard] },
19   { path: 'login', component: LoginComponent },
20   { path: '**', component: HomeComponent }
21 ];
22
23 @NgModule({
24   imports: [RouterModule.forRoot(routes)],
25   exports: [RouterModule]
26 })
27 export class AppRoutingModule { }
28

```

Ln 25, Col 30 Spaces: 2 UTF-8 LF TypeScript 3.1.3

CakePHP backend adding updateUserProfile() function into UsersController.php

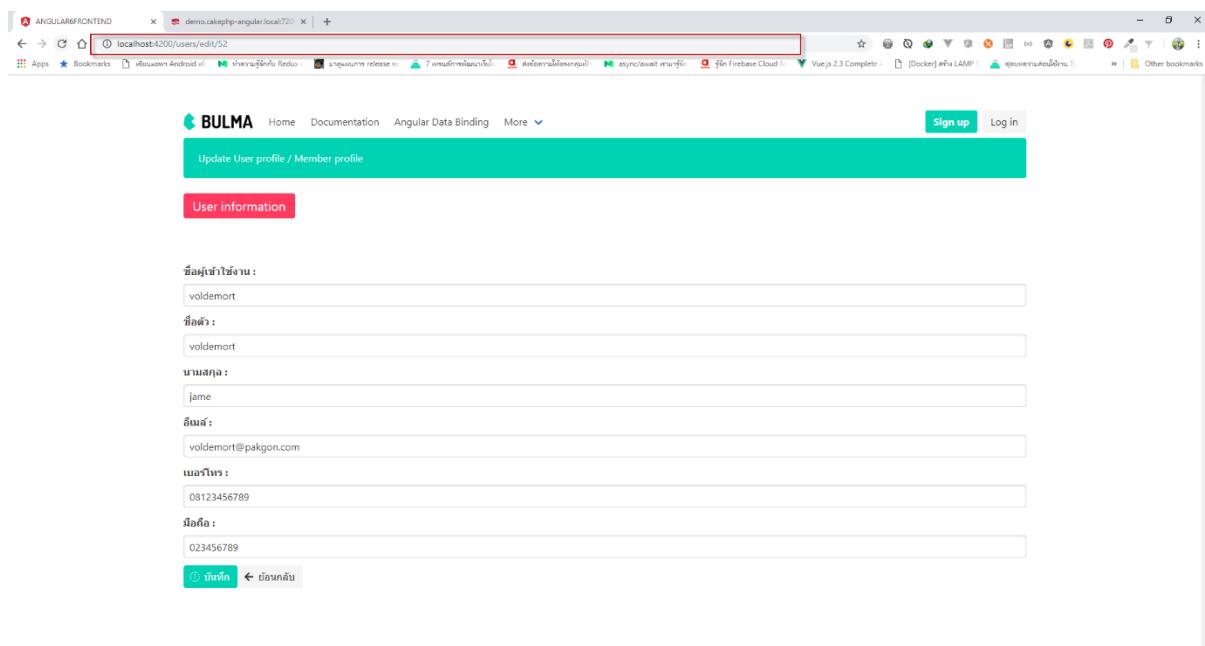


```

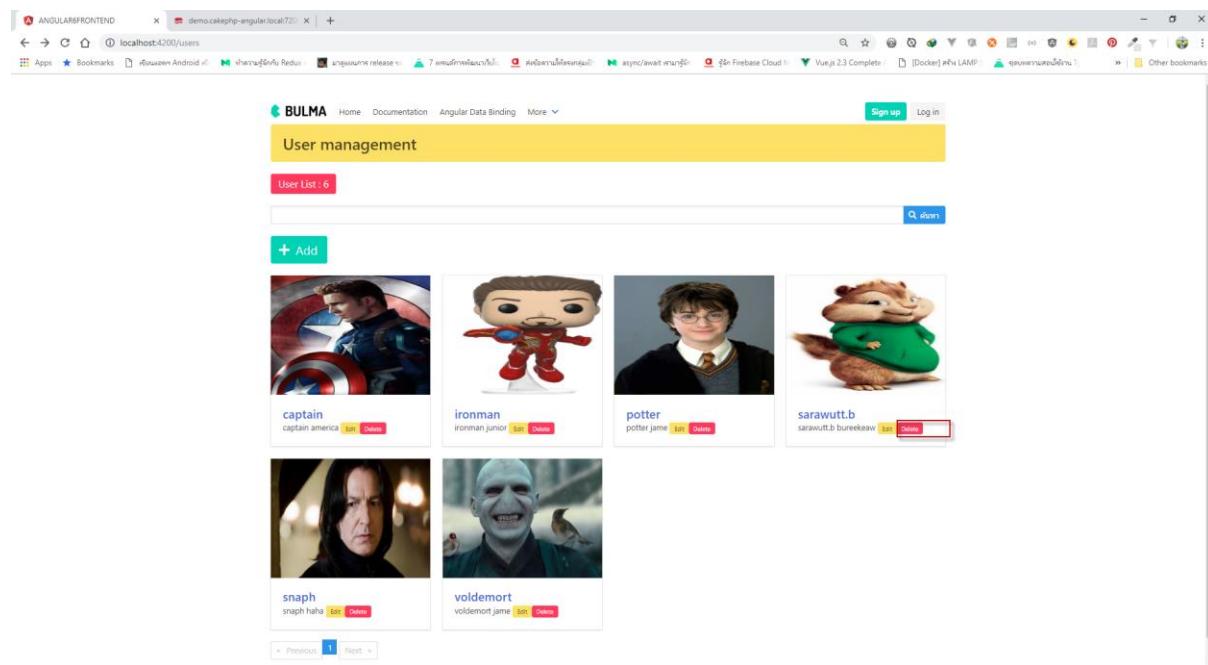
135 /**
136 * Function update user profile
137 * @author sarawutt.b
138 * @param type $id as an integer of user id [PK]
139 */
140
141 public function updateUserProfile($id = null) {
142     $response = ['message' => __('Initial for add a user profile'), 'status' => 'failed'];
143     if ($this->request->is(['patch', 'post', 'put'])) {
144         $user = $this->Users->get($id, ['contain' => []]);
145         if (!empty($user)) {
146             $params = $this->request->getData();
147             $params['create_uid'] = 999;
148             $user = $this->Users->patchEntity($user, $params);
149             if ($this->Users->save($user)) {
150                 $response = ['message' => __('The user has been saved.'), 'status' => 'success'];
151             } else {
152                 $response = ['message' => __('The user could not be save. Please try again.'), 'status' => 'failed'];
153             }
154         } else {
155             $response = ['message' => __('The request user not found infomation. Please try again.'), 'status' => 'failed'];
156         }
157     } else {
158         $response = ['message' => __('Not allow request Method you must request to update with POST/PUT/PATCH'), 'status' => 'failed'];
159     }
160     $this->set(compact('response'));
161     $this->set('_serialize', ['response']);
162 }

```

Open your browser:



Delete User



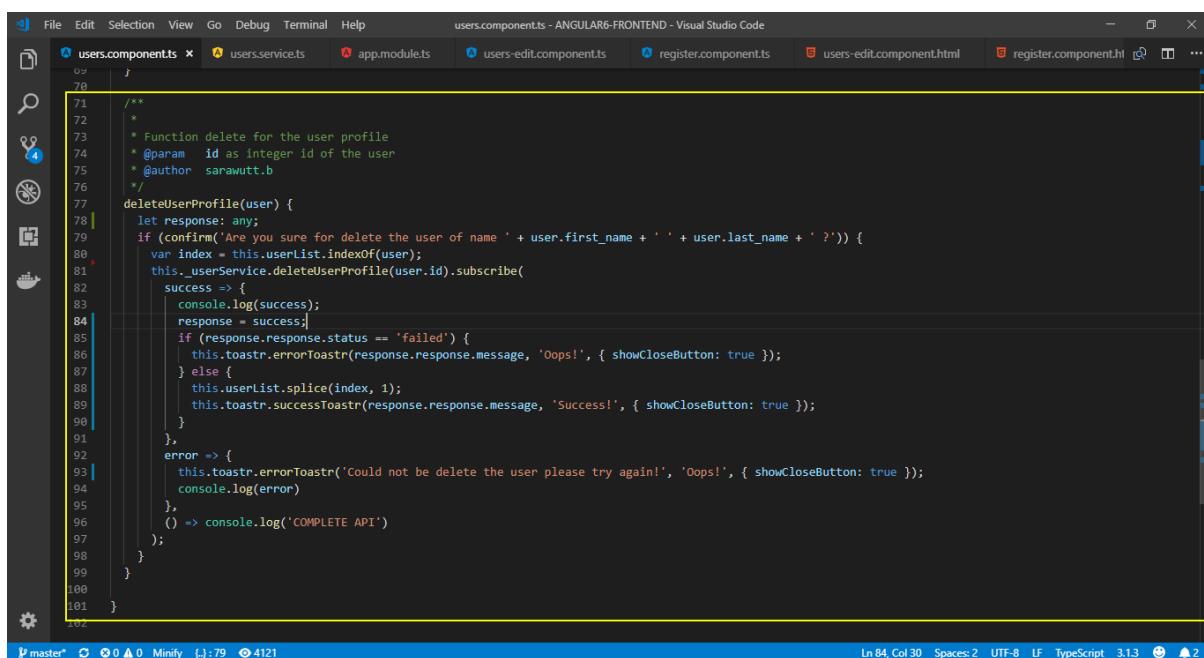
Open app/users/users/user.component.html and adding below source code.

```

</div>
<div class="columns is-multiline">
  <div class="column is-one-quarter" *ngFor="let user of userList | userFilter: searchTxt | paginate: {itemsPerPage: 6}">
    <div class="card">
      <div class="card-image">
        <figure class="image is-4by3">
          
        </figure>
      </div>
      <div class="card-content">
        <div class="media">
          <div class="media-content">
            <p class="title is-4">
              <a [routerLink]="'[detail',user.id]" class="is-success">{{user.username}}</a>
            </p>
            <p class="subtitle is-6">{{user.first_name}} {{user.last_name}}</p>
            <a [routerLink]="'[edit',user.id]" class="tag is-warning">Edit</a>
            <a (click)="deleteUserProfile(user)" class="tag is-danger">Delete</a>
          </div>
        </div>
      </div>
    </div>
  </div>
<div class="columns">
  <div class="column">
    <!-- Pagination Page -->
    <pagination-controls class="pagination is-right" role="navigation" aria-label="pagination" (pageChange)="p = $event"></pagination-controls>
  </div>
</div>

```

Open app/users/users/user.component.ts



```

 71 /**
 72 *
 73 * Function delete for the user profile
 74 * @param id as integer id of the user
 75 * @author sarawutt.b
 76 */
 77 deleteUserProfile(user) {
 78   let response: any;
 79   if (confirm('Are you sure for delete the user of name ' + user.first_name + ' ' + user.last_name + '?')) {
 80     var index = this.userList.indexOf(user);
 81     this.userService.deleteUserProfile(user.id).subscribe(
 82       success => {
 83         console.log(success);
 84         response = success;
 85         if (response.response.status == 'failed') {
 86           this.toastr.errorToastr(response.response.message, 'Oops!', { showCloseButton: true });
 87         } else {
 88           this.userList.splice(index, 1);
 89           this.toastr.successToastr(response.response.message, 'Success!', { showCloseButton: true });
 90         }
 91       },
 92       error => {
 93         this.toastr.errorToastr('Could not be delete the user please try again!', 'Oops!', { showCloseButton: true });
 94         console.log(error)
 95       },
 96       () => console.log('COMPLETE API')
 97     );
 98   }
 99 }
100 }
101

```

Open app/_services/users.service.ts

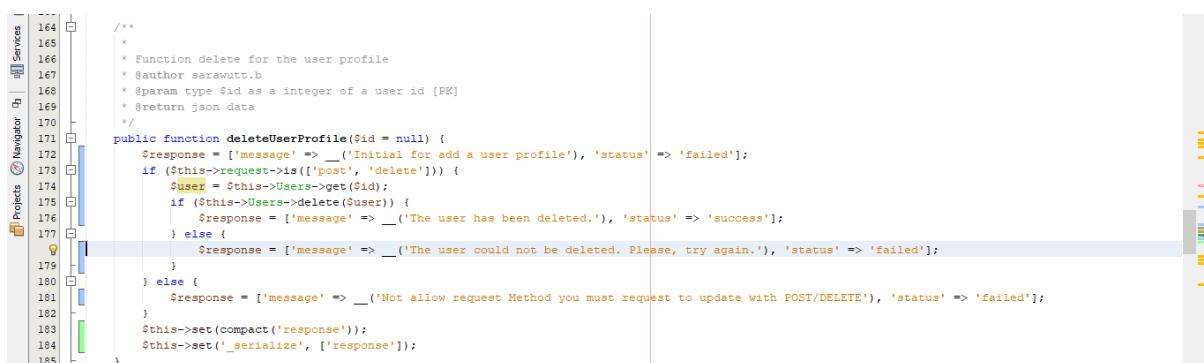


```

43 /**
44 *
45 * Function delete for the user profile
46 * @param id as integer id of the user
47 * @author sarawutt.b
48 */
49 deleteUserProfile(id: any) {
50   return this.http.post(`${environment.apiUrl}/users/deleteUserProfile/${id}.json`, JSON.stringify({ id: id }));
51 }
52

```

CakePHP backend adding deleteUserProfile() function into UsersController.php

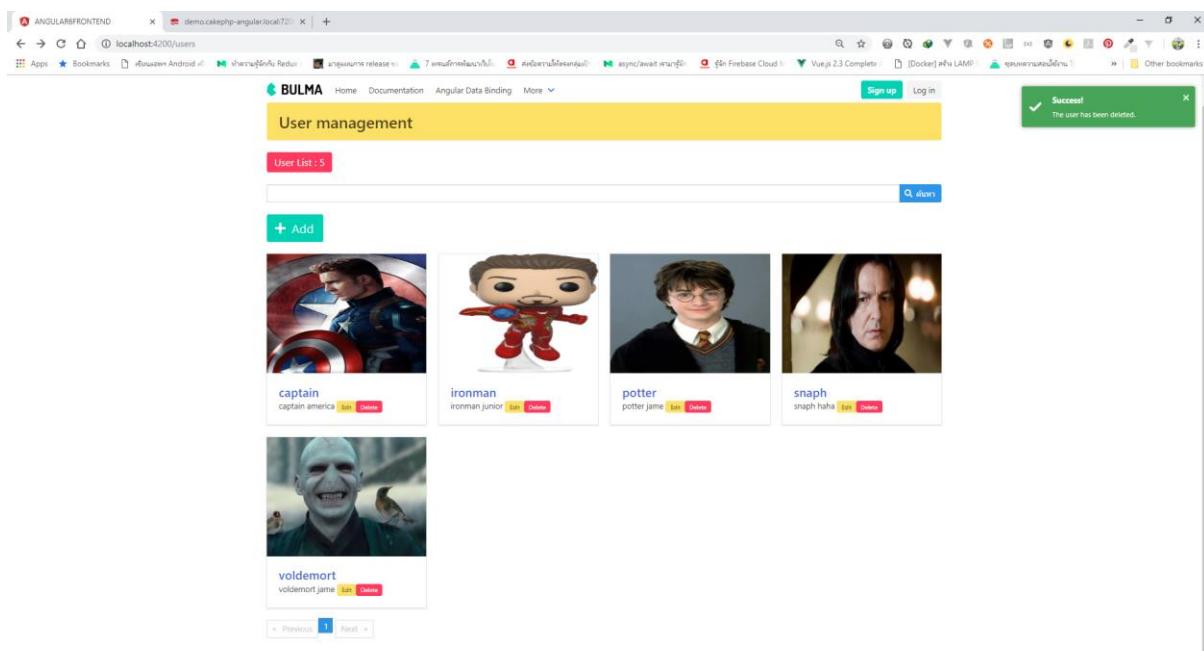


```

164 /**
165  * Function delete for the user profile
166  * @author sarawutt.b
167  * @param type $id as a integer of a user id [PK]
168  * @return json data
169 */
170
171 public function deleteUserProfile($id = null) {
172     $response = ['message' => __('Initial for add a user profile'), 'status' => 'failed'];
173     if ($this->request->is(['post', 'delete'])) {
174         $user = $this->Users->get($id);
175         if ($this->Users->delete($user)) {
176             $response = ['message' => __('The user has been deleted.'), 'status' => 'success'];
177         } else {
178             $response = ['message' => __('The user could not be deleted. Please, try again.'), 'status' => 'failed'];
179         }
180     } else {
181         $response = ['message' => __('Not allow request Method you must request to update with POST/DELETE'), 'status' => 'failed'];
182     }
183     $this->set(compact('response'));
184     $this->set('_serialize', ['response']);
185 }

```

Open your Browser:



HW5:

1. Modifier for navigation bar display for dynamic menu.



TIP & Trick**Angular In-Memory Web API (Mock Backend API)**

Angular provides in-memory web API to process HTTP request in test environment. In case we don't have actual server URL, we can use angular in-memory web API for testing our angular Http methods. It provides a dummy URL that can be replaced by actual REST web service URL later. It returns an Observable of HTTP Response object in the manner of a RESTy web api. In our example we are using in-memory web API to get and post data. To use it in our angular application we need to follow below steps.

```
$ npm i angular2-in-memory-web-api --save-dev
```

Minimum requirement must to implement createDb ()

Create /src/app/book-data.ts to mock REST back-end data API.

```

1 import { InMemoryDbService } from 'angular2-in-memory-web-api';
2
3 export class BookData implements InMemoryDbService {
4   createDb() {
5     let books = [
6       { id: '1', name: 'Angular 2 with Angular-CLI', category: 'Angular', writer: 'sarawutt.b' },
7       { id: '2', name: 'AngularJS simple Application on CRUD', category: 'Angular', writer: 'sarawutt.b' },
8       { id: '3', name: 'Angular 2 with Typescript', category: 'Angular', writer: 'sarawutt.b' },
9       { id: '4', name: 'Core Java OOP', category: 'Java', writer: 'sarawutt.b' },
10      { id: '5', name: 'JSP & Servlet', category: 'Java', writer: 'sarawutt.b' },
11      { id: '6', name: 'JPA', category: 'Java', writer: 'sarawutt.b' },
12      { id: '7', name: 'Hibernate', category: 'Hibernate', writer: 'sarawutt.b' }
13    ];
14    return {books};
15  }
16 }

```

Open /src/app.module.ts and then adding two this line for loading mock API.

```

21 //Memory DB mock for fake back-end demo or test
22 import { InMemoryWebApiModule } from 'angular2-in-memory-web-api';
23 import { BookData } from './book-data';
24 InMemoryWebApiModule.forRoot(BookData)

```

In import section of /src/app.module.ts

```

151
152 InMemoryWebApiModule.forRoot(BookData)

```

Date Picker Plugin

<https://github.com/kekeh/mydatepicker>

```
MINGW64:/d/Angular II/angular4-app
$ npm install mydatepicker --save
+ mydatepicker@2.0.27
added 1 package in 24.33s

dk_antikorn@DKANTIKORN MINGW64 /d/Angular II/angular4-app (master)
$ npm install ngx-pagination --save
+ ngx-pagination@3.0.1
added 1 package in 23.343s

dk_antikorn@DKANTIKORN MINGW64 /d/Angular II/angular4-app (master)
$
```

Page Pagination

<https://github.com/michaelbromley/ngx-pagination>

```
MINGW64:/d/Angular II/angular4-app
$ npm install mydatepicker --save
+ mydatepicker@2.0.27
added 1 package in 24.33s

dk_antikorn@DKANTIKORN MINGW64 /d/Angular II/angular4-app (master)
$ npm install ngx-pagination --save
+ ngx-pagination@3.0.1
added 1 package in 23.343s

dk_antikorn@DKANTIKORN MINGW64 /d/Angular II/angular4-app (master)
$
```

Angular 6 Toaster Notifications Module

<https://www.npmjs.com/package/ng6-toastr-notifications>

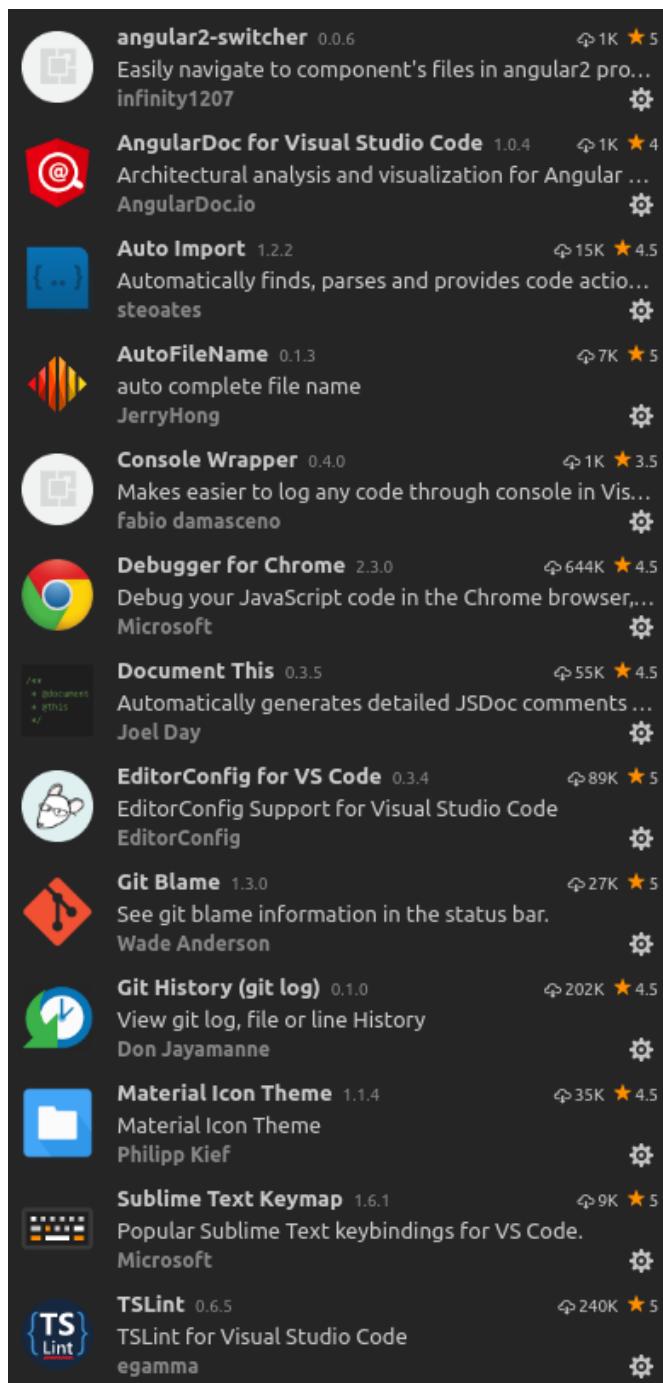
\$ npm install ng6-toastr-notifications --save

Angular-CLI Command Reference

Command Type	Command
Component	ng g component my-new-component
Directive	ng g directive my-new-directive
Pipe	ng g pipe my-new-pipe
Service	ng g service my-new-service
Class	ng g class my-new-class
Interface	ng g interface my-new-interface
Enum	ng g enum my-new-enum

VS Code Extension suggest to install

1. Angular v4 TypeScript Snippets By JohnPapa.
2. AutoFileName By Jerry Hong.
3. Autolimport By steotes.
4. TypeScript Toolbox.
5. vscode-icons.



NetBean Plugin.

Type Script

<https://github.com/Everlaw/nbts/releases>

The screenshot shows a GitHub release page for the repository 'nbts'. The release is titled 'v3.0.1: Fix getEmitOutput not honoring compile settings' and was released by 'Jeffrey-easyesi' on July 25. It includes three assets: 'netbeanstypescript-3.0.1.nbm' (107 KB), 'Source code (zip)', and 'Source code (tar.gz)'. A note states: 'getCompilationSettings needs to return a flat object. getEmitOutput uses the spread operator, so it won't pick up settings from a prototype.' A download link is also present.

Angular CLI

<https://github.com/Chris2011/NBAngularCLI>

The screenshot shows a GitHub repository page for 'NBAngularCLI' created by 'Chris2011'. The repository has 32 commits, 2 branches, 2 releases, and 1 contributor. The latest commit was made on Sep 24. A table lists the files modified in the latest commit:

File	Description	Time Ago
nbproject	Removed some imports.	a month ago
screenshots	Moved files one folder up.	7 months ago
src/net/chrizly/netbeans/plugins/nbangularcli	Removed some imports.	a month ago
.gitignore	Moved files one folder up.	7 months ago
.travis.yml	Moved files one folder up.	7 months ago
LICENSE	Moved files one folder up.	7 months ago
README.md	Updated Readme.	a month ago
build.xml	Removed some imports.	a month ago

Angular Code snipped Netbean Plugin

<https://github.com/Chris2011/netbeans-angular2-code-templates>

Code templates for Angular 2 in Netbeans. Templates exists for TypeScript and HTML.

46 commits	2 branches	4 releases	1 contributor	MIT	
Branch: master	New pull request	Create new file	Upload files	Find file	Clone or download
Chris2011 Signed module. Latest commit b6572ab on Oct 27, 2017					
images	Added screencapture for html snippets.	3 years ago			
nbproject	Signed module.	a year ago			
src/org/chrisle/netbeans/plugins/angular2...	Fixed TypeScript templates.	a year ago			
.gitignore	Updated gitignore file.	2 years ago			
LICENSE	Initial commit	3 years ago			
README.md	Added (old screenshot) string.	a year ago			
build.xml	Moved files.	2 years ago			
manifest.mf	Changed version back to 1.1 because I didn't have one.	a year ago			

Automate Compilation with Gulp Task Manage:

\$ npm install -g gulp

\$ npm install gulp gulp-typescript --save-dev

Create java script filename **gulpfile.js** at the root of your project with the following source code:

```

1 var gulp = require('gulp');
2 var ts = require('gulp-typescript');
3 var tsProject = ts.createProject('tsconfig.json');
4
5 gulp.task('build', function() {
6   var tsResult = tsProject.src().pipe(ts(tsProject));
7   return tsResult.js.pipe(gulp.dest('./built'));
8 });
9

```

Run task on your terminal

\$ gulp build

Watch and automate

```

1 var gulp = require('gulp');
2 var ts = require('gulp-typescript');
3 var tsProject = ts.createProject('tsconfig.json');
4
5 gulp.task('build', function() {
6   var tsResult = tsProject.src().pipe(ts(tsProject));
7   return tsResult.js.pipe(gulp.dest('./built'));
8 });
9
10 gulp.task('watch', ['build'], function() {
11   gulp.watch('./**/*.ts', ['build']);
12 });

```

Run the task on your terminal for build and then what file changed and then rebuild automated

\$ gulp watch

Chrome Extension

Augury

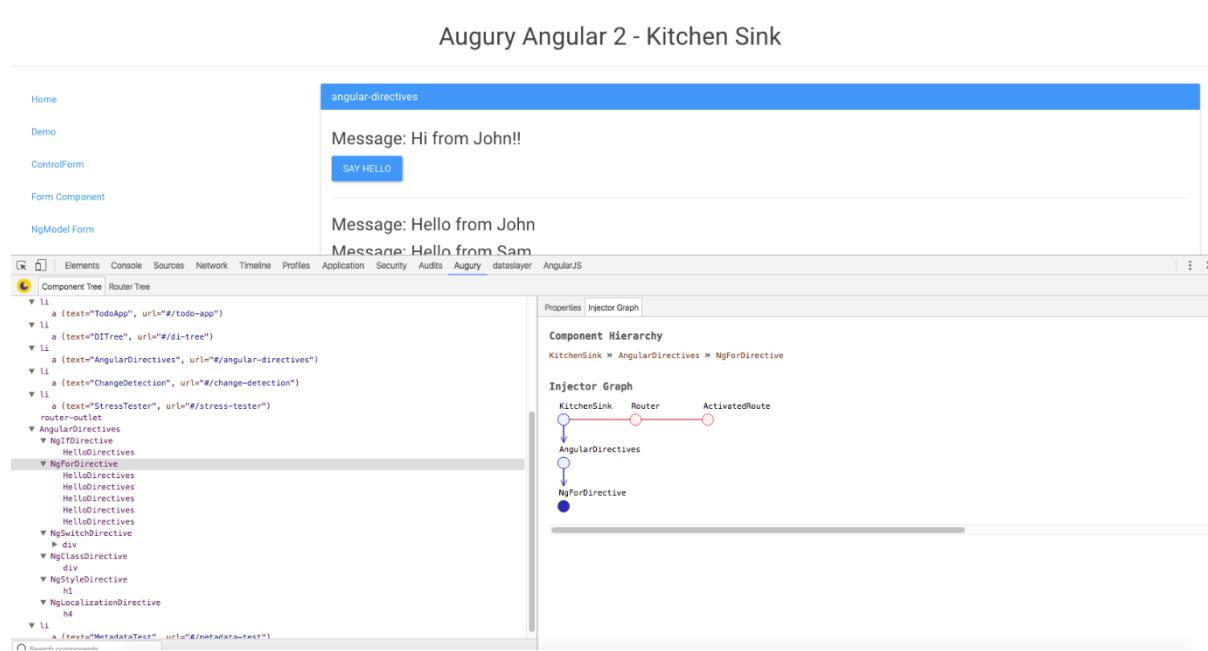
Augury is a Google Chrome developer extension for debugging and profiling Angular 2 applications. It is an open source project, developed by google and rangle.io. Using Augury, we will get details about component trees, router trees and component class properties.

Augury Angular 2 - Kitchen Sink

Features of Augury

Component Tree

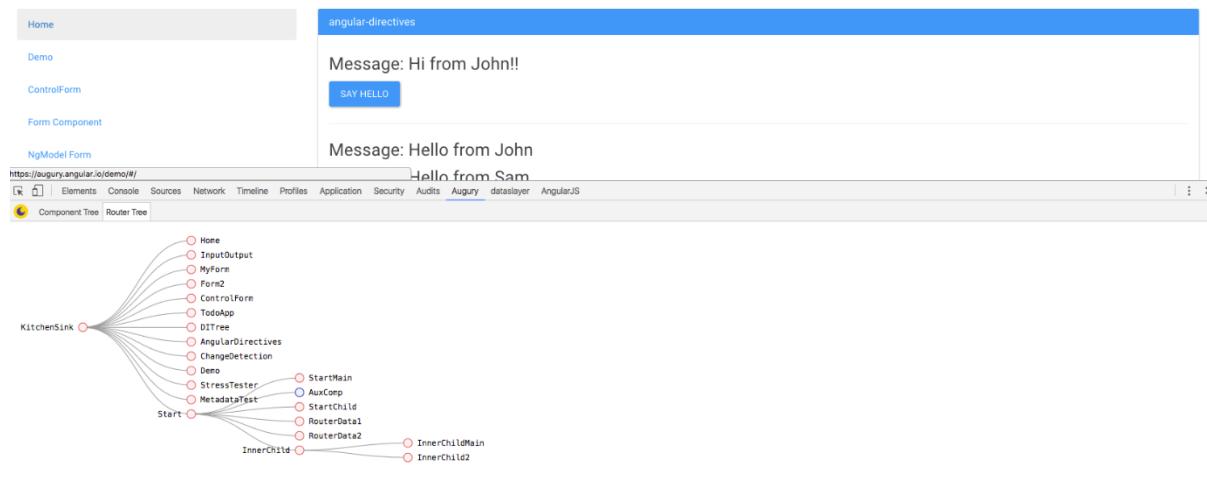
As we know that Angular 2 applications are built on components. So there should be nesting of components to meet that requirements. While nesting the components, it will create a tree structure. So if we need to analyze or understand properties of some of the components, it will be difficult to get the information from code. Here we can use Augury to get the information, **by simply selecting the component from Augury console.**



Router Tree

Router tree will help us to get information about the routes in our Angular 2 application. Augury will visualize the application route in a tree structure. So it will be easy to understand. By selecting each route in the tree, you will get more information about the same.

Augury Angular 2 - Kitchen Sink



Update npm and nodeJS

```
$ npm install -g npm stable
```

```
$ npm install -g node or npm install -g n
```

Update Angular-CLI

```
$ npm upgrade @angular/cli@latest
```

OR

```
$ npm uninstall -g @angular/cli
```

```
$ npm cache verify
```

```
$ npm install -g @angular/cli@latest
```

Appendix Angular CLI

Create Angular Project

1. `ng new [project name]`
 - standard for angular project
2. `ng new [project name] --minimal`
 - no testing file included
 - Inline template code for components
3. `ng new [project name] --inline-template`
4. `ng new [project name] --inline-style`
5. `ng new [project name] --style=scss` Possible values for `--style` attribute:
 - a. `css` (default)
 - b. `scss`
 - c. `less`
 - d. `sess`
 - e. `styl`
6. `ng new [project name] --routing`
7. `ng new [project name] --prefix=myapp`
8. `ng new [project name] --skip-git`

Build Project

1. `ng build`
2. `ng build --env=prod` (build with env of the production with develop mode not to minify)
3. `ng build --prod --env=prod` (build with production env and minify all package)

jQuery with Angular

STEP1

Install jQuery package::

```
$ npm install --save jquery
```

```
$ npm install --save --D @types/jquery
```

STEP2

Import in needed component

```
import * as $ from jquery;
```

OR

```
import $ = require('jquery');
```

OR adding import from angular-cli.json

```
"scripts": [  
    "./node_modules/jquery/dist/jquery.min.js",  
]
```

In each component

```
declare var $:any;
```

```
declare var jQuery:any;
```

STEP3

USE as same in ES5

Generate Component

All you can generate Component / Service / Model / Pipe / Class / Guard / Directive / Module

ng generate component products / (ng g c products)

ng generate component shop/products / (ng g c shop/products)

--module = app / --m = app

--spec = false / --spec false

--style=false / --style false

--inline style = --is

--inline HTML = --it

ต้องการสร้าง	คำสั่งที่ใช้ในการสร้าง	คำสั่งแบบสั้น
Component	ng generate component my-new-component	ng g c my-new-component
Directive	ng generate directive my-new-directive	ng g d my-new-directive
Pipe	ng generate pipe my-new-pipe	ng g p pipe my-new-pipe
Service	ng generate service my-new-service	ng g s my-new-service
Class	ng generate class my-new-class	ng g cl my-new-class
Interface	ng generate interface my-new-interface	ng g i my-new-interface
Enum	ng generate enum my-new-enum	ng g e enum my-new-enum

Generate with file type

\$ ng g class models/users --type=model (Create output file src/app/models/users.model.ts)

Reference

(Angular CLI)	https://github.com/angular/angular-cli
(Angular CLI Wiki)	https://github.com/angular/angular-cli/wiki
(Angular Redux)	https://github.com/ivanderbu2/angular-redux
(Angular Material)	https://material.angular.io/
	https://github.com/angular/material2
(Type Script)	https://tutorialzine.com/2016/07/learn-typescript-in-30-minutes
(JWT and Authenticate)	https://github.com/auth0/angular2-jwt
(RxJS and Reactive Programming)	https://www.babelcoder.com/blog/posts/rxjs-observables
(Mock server API)	https://github.com/typicode/json-server
(NGX-PIPES (template Helper))	https://github.com/danrevah/ngx-pipes#object
(My Datepicker (Date picker Plugin))	https://github.com/kekeh/mydatepicker
(Bulma CSS Framework)	http://bulma.io/documentation/overview/start/
(Font Awesome ICON)	http://fontawesome.io/
(Angular PWAs)	https://www.smashingmagazine.com/2018/09/pwa-angular-6/
(Netbean Plugin)	https://github.com/Everlaw/nbts/releases

Our tutorial repository

https://github.com/dkantikorn/DEMO_CAKEPHP3_ANGULAR6