

Tipo : Guía de Laboratorio
Capítulo : Angular JS
Duración : 120 minutos

I. OBJETIVO

Crea el método de autenticación para AngularJs, haciendo uso de la WebApi creada previamente.

II. REQUISITOS

Los siguientes elementos de software son necesarios para la realización del laboratorio:

- Windows 10 (como mínimo Windows 8.1)
- Extension Postman de Google Chrome.
- Microsoft Sql Server 2014 Management Studio (como mínimo la versión 2008)
- Visual Studio 2017

III. EJECUCIÓN DEL LABORATORIO

Crear una cuenta de Microsoft Azure

Actividades

1. En el proyecto **Cibertec.Angular** procede a crear la carpeta: **Code** y en ella agrega la siguiente clase: **ConfigurationFile.cs** con el siguiente código:

```
namespace Cibertec.Angular.Code
{
    public class ConfigurationFile
    {
        public string WebApiUrl { get; set; }
    }
}
```

2. En el fichero appsettings.json agrega el siguiente objeto json:

```
{
  "ConfigurationFile": {
    "WebApiUrl": "http://localhost/WebApi"
  },
  "Logging": {
    "IncludeScopes": false,
    "LogLevel": {
      "Default": "Warning"
    }
  }
}
```

Recuerde que esta información debe de concordar con la url de la WebApi que publicaste.

3. En el fichero **Startup.cs** en la configuración de la inyección de dependencias agrega la siguiente línea:

```
services.Configure<ConfigurationFile>(Configuration.GetSection("Configura  
tionFile"));
```

Quedando el método de la siguiente manera:

```
public void ConfigureServices(IServiceCollection services)  
{  
    // Add framework services.
```

```
services.Configure<ConfigurationFile>(Configuration.GetSection("Configura  
tionFile"));
```

```
services.AddMvc();  
}
```

Posteriormente usaremos esta configuración para obtener la url de la WebApi desde el archivo de configuración de configuración de nuestra aplicación web.

4. Procedemos a crear nuestros archivos javascript para administrar el acceso al login, en el folder **"app"** creamos la carpeta **"shared"** y agregamos el fichero **"data.service.js"** con el siguiente código:

```
(function () {  
    angular  
        .module('app')  
        .factory('dataService', dataService);  
  
    dataService.$inject = ['$http'];  
  
    function dataService($http) {  
        var service = {};  
        service.getData = getData;  
        service.postData = postData;  
        service.putData = putData;  
        service.deleteData = deleteData;  
  
        return service;  
  
        function getData(url) {  
            return $http.get(url);  
        }  
        function postData(url, data) {  
            return $http.post(url, data);  
        }  
        function putData(url, data) {  
            return $http.put(url, data);  
        }  
        function deleteData(url) {  
            return $http.delete(url);  
        }  
    }  
})();
```

5. En la misma carpeta agregamos el fichero **settings.service.js** con el siguiente código:

```
(function () {  
    'use strict';  
  
    angular  
        .module('app')  
        .factory('configService', configService);  
  
    function configService() {  
        var service = {};
```

```

    var apiUrl = undefined;
    var isLoggedIn = false;
    service.setLogin = setLogin;
    service.getLogin = getLogin;
    service.setApiUrl = setApiUrl;
    service.getApiUrl = getApiUrl;

    return service;

    function setLogin(state) {
        isLoggedIn = state;
    }

    function getLogin() {
        return isLoggedIn;
    }

    function getApiUrl() {
        return apiUrl;
    }

    function setApiUrl(url) {
        apiUrl = url;
    }
}
})();

```

6. Agregamos en la misma carpeta el archivo **"authentication.service.js"** con el siguiente código:

```

(function () {
    angular
        .module('app')
        .factory('authenticationService', authenticationService);

    authenticationService.$inject = ['$http', '$state',
    'localStorageService', 'configService'];

    function authenticationService($http, $state, localStorageService,
    configService) {
        var service = {};
        service.login = login;
        service.logout = logout;
        return service;

        function login(user) {
            var url = configService.getApiUrl() + '/Token';
            var data = "username=" + user.userName + "&password=" +
            user.password;
            $http.post(url,
                data,
                {
                    headers: {
                        'Content-Type': 'application/x-www-form-
            urlencoded'
                    }
                })
            .then(function (result) {
                $http.defaults.headers.common.Authorization = 'Bearer '
            + result.data.access_token;
                localStorageService.set('userToken',
                {

```

```

        token: result.data.access_token,
        userName: user.userName
    });
    configService.setLogin(true);
    $state.go("home");
    },
    function error(response) {
        $state.go("login");
    }
    );
}

function logout() {
    $http.defaults.headers.common.Authorization = '';
    localStorageService.remove('userToken');
    configService.setLogin(false);
}

}
})();

```

7. En la raíz de la carpeta “**app**” agregamos el fichero “**app.config.js**” con el siguiente código:

```

(function () {
    'use strict';
    angular.module('app').run(run);

    run.$inject = ['$http', '$state', 'localStorageService',
    'configService'];
    function run($http, $state, localStorageService, configService) {
        var user = localStorageService.get('userToken');
        if (user && user.token) {
            $http.defaults.headers.common.Authorization = 'Bearer ' +
            localStorageService.get('userToken').token;
            configService.setLogin(true);
        }
        else $state.go('login');
    }
})();

```

8. Seguido crea en la misma raíz de la carpeta “**app**” el fichero “**app.controller.js**” con el siguiente código:

```

(function () {
    'use strict';
    angular.module('app').controller('applicationController',
    applicationController);

    applicationController.$inject = ['$scope', 'configService',
    'authenticationService', 'localStorageService'];

    function applicationController($scope, configService,
    authenticationService, localStorageService) {
        var vm = this;
        vm.validate = validate;
        vm.logout = logout;

        $scope.init = function(url) {
            configService.setApiUrl(url);
        }

        function validate() {
            return configService.getLogin();
        }
    }
}

```

```

    }

    function logout() {
        authenticationService.logout();
    }

}

})();

```

9. Creamos la carpeta **public** y dentro de ella la carpeta **login**.

10. Dentro de la carpeta **login** crea el fichero: index.html con el siguiente código:

```

<div ng-controller="loginController as vm">
    <div ng-if="vm.showError" class="alert alert-danger">
        <strong>Error!</strong> El usuario o contraseña son
incorrectos.
    </div>
    <form id="loginModel" ng-submit="vm.login()" class="form-
horizontal">
        <div class="form-group">
            <label for="userName" class="control-label col-sm-
2">Usuario:</label>
            <div class="col-sm-4">
                <input id="userName" type="text" class="form-control"
ng-model="vm.user.userName" /><br />
            </div>
        </div>
        <div class="form-group">
            <label for="password" class="control-label col-sm-
2">Contraseña:</label>
            <div class="col-sm-4">
                <input id="password" type="password" class="form-
control" ng-model="vm.user.password" /><br />
            </div>
            <div class="col-sm-offset-2 col-sm-10">
                <button type="submit" class="btn btn-
primary">Submit</button>
            </div>
        </form>
    </div>

```

11. En la misma carpeta crear el fichero **"login.controller.js"** con el siguiente código:

```

(function () {
    'use strict';
    angular.module('app')
        .controller('loginController', loginController);

    loginController.$inject = ['$http', 'authenticationService',
'configService', '$state'];

    function loginController($http, authenticationService,
configService, $state) {
        var vm = this;
        vm.user = {};
        vm.title = 'Login';
        vm.login = login;

        init();

        function init() {

```

```

        if (configService.setLogin()) $state.go("home");
        authenticationService.logout();
    }

    function login() {
        authenticationService.login(vm.user);
    }
}
})();

```

12. No olvidarse de eliminar el fichero login.html de la raíz de la carpeta “app”.
13. En la carpeta “**wwwroot/libraries/angular**” insertar la librería **angular-local-storage** que se encuentra en la carpeta compartida.

14. En el HomeController agregamos las siguiente líneas de código:

```

using Microsoft.AspNetCore.Mvc;
using Cibertec.Angular.Code;
using Microsoft.Extensions.Options;

namespace Cibertec.Angular.Controllers
{
    public class HomeController : Controller
    {
        private readonly ConfigurationFile _config;
        public HomeController(IOptions<ConfigurationFile> config)
        {
            _config = config.Value;
        }
        public IActionResult Index()
        {
            ViewBag.WebApiUrl = _config.WebApiUrl;
            return View();
        }

        public IActionResult Error()
        {
            return View();
        }
    }
}

```

15. En el layout procedemos a modificar el código de la siguiente manera:

- En la etiqueta de inicio **body**:


```

<body ng-controller="applicationController as vm"
      ng-init="init('@ViewBag.WebApiUrl')">

```
- Agregamos las librerías creadas como parte de la carga inicial de la pagina:


```

<script src="~/libraries/angular/angular.js"></script>
<script src="~/libraries/angular/angular-ui-router.js"></script>
<script src="~/libraries/angular/angular/angular-local-storage.js"></script>

<script src="~/libraries/jquery/jquery.js"></script>
<script src="~/libraries/bootstrap/js/bootstrap.min.js"></script>

<script src="~/app/app.js"></script>
<script src="~/app/app.routes.js"></script>
<script src="~/app/app.config.js"></script>
<script src="~/app/app.controller.js"></script>

<script src="~/app/shared/data.service.js"></script>
<script src="~/app/shared/authentication.service.js"></script>

```

```
<script src=~/.app/shared/settings.service.js"></script>

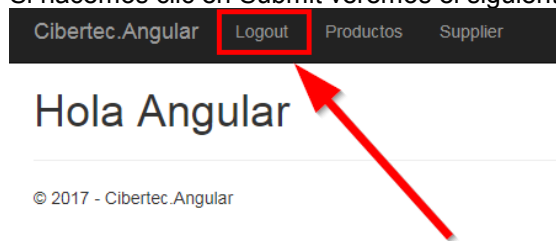
<script src=~/.app/public/login/login.controller.js"></script>
```

16. Como parte de la navegación dejamos las siguientes líneas:

```
<li><a ng-if=!vm.validate()" ui-sref="login" ui-sref-
active="active">Login</a></li>
<li><a ng-if=vm.validate()" ng-click="vm.logout()"
href="#" ui-sref-active="active">Logout</a></li>
```

17. Al ejecutar la aplicación y haciendo clic en la opción Login, veremos la siguiente pantalla:

18. Si hacemos clic en Submit veremos el siguiente resultado:



El botón de **Login** desaparecerá y será reemplazado por un **Logout**.