Source Code Of Authentication

Configuration Class

1. @Configuration
2. @EnableWebSecurity
3. **public** **class** **BasicAuthConfiguration** {
4. @Bean
5. **public** InMemoryUserDetailsManager **userDetailsService**() { **UserDetails** user = User.withUsername("user") .password("{noop}password")
6. .roles("USER")
7. .build(); **return** **new**
8. **InMemoryUserDetailsManager**(user);
9. }
10. @Bean
11. **public** SecurityFilterChain **filterChain**(HttpSecurity http) **throws** Exception {
12. http.csrf()
13. .disable()
14. .authorizeRequests()
15. .antMatchers(HttpMethod.OPTIONS, "/\*\*")
16. .permitAll()
17. .antMatchers("/login")
18. .permitAll()
19. .anyRequest()
20. .authenticated()
21. .and()
22. .httpBasic();
23. **return** http.build();
24. }

2.Controller Class

@RestController

@CrossOrigin

public class UserController {

@RequestMapping("/login")

public boolean login(@RequestBody User user) {

return

user.getUserName().equals("user") && user.getPassword().equals("password");

}

@RequestMapping("/user")

public Principal user(HttpServletRequest request) {

String authToken = request.getHeader("Authorization")

.substring("Basic".length()).trim();

return () -> new String(Base64.getDecoder()

.decode(authToken)).split(":")[0];

}

}

3.Login Page Using Angular

<html ng-app="app">

<body>

<div ng-view></div>

<script src="//code.jquery.com/jquery-3.1.1.min.js"></script>

<script src="//code.angularjs.org/1.6.0/angular.min.js"></script>

<script src="//code.angularjs.org/1.6.0/angular-route.min.js"></script>

<script src="app.js"></script>

<script src="home/home.controller.js"></script>

<script src="login/login.controller.js"></script>

</body>

</html>

4.View Page

(function () {

'use strict';

angular

.module('app', ['ngRoute'])

.config(config)

.run(run);

config.$inject = ['$routeProvider', '$locationProvider'];

function config($routeProvider, $locationProvider) {

$routeProvider.when('/', {

controller: 'HomeController',

templateUrl: 'home/home.view.html',

controllerAs: 'vm'

}).when('/login', {

controller: 'LoginController',

templateUrl: 'login/login.view.html',

controllerAs: 'vm'

}).otherwise({ redirectTo: '/login' });

}

run.$inject = ['$rootScope', '$location', '$http', '$window'];

function run($rootScope, $location, $http, $window) {

var userData = $window.sessionStorage.getItem('userData');

if (userData) {

$http.defaults.headers.common['Authorization']

= 'Basic ' + JSON.parse(userData).authData;

}

$rootScope

.$on('$locationChangeStart', function (event, next, current) {

var restrictedPage

= $.inArray($location.path(), ['/login']) === -1;

var loggedIn

= $window.sessionStorage.getItem('userData');

if (restrictedPage && !loggedIn) {

$location.path('/login');

}

});

}

})();

5.LoginPage.Html

<h2>Login</h2>

<form name="form" ng-submit="vm.login()" role="form">

<div>

<label for="username">Username</label>

<input type="text" name="username"

id="username" ng-model="vm.username" required />

<span ng-show="form.username.$dirty

&& form.username.$error.required">Username is required</span>

</div>

<div>

<label for="password">Password</label>

<input type="password"

name="password" id="password" ng-model="vm.password" required />

<span ng-show="form.password.$dirty

&& form.password.$error.required">Password is required</span>

</div>

<div class="form-actions">

<button type="submit"

ng-disabled="form.$invalid || vm.dataLoading">Login</button>

</div>

</form>

6.Using Angular

(function () {

'use strict';

angular

.module('app')

.controller('LoginController', LoginController);

LoginController.$inject = ['$location', '$window', '$http'];

function LoginController($location, $window, $http) {

var vm = this;

vm.login = login;

(function initController() {

$window.localStorage.setItem('token', '');

})();

function login() {

$http({

url: 'http://localhost:8082/login',

method: "POST",

data: {

'userName': vm.username,

'password': vm.password

}

}).then(function (response) {

if (response.data) {

var token

= $window.btoa(vm.username + ':' + vm.password);

var userData = {

userName: vm.username,

authData: token

}

$window.sessionStorage.setItem(

'userData', JSON.stringify(userData)

);

$http.defaults.headers.common['Authorization']

= 'Basic ' + token;

$location.path('/');

} else {

alert("Authentication failed.")

}

});

};

}

})();

7.Home.API Pages

(function () {

'use strict';

angular

.module('app')

.controller('HomeController', HomeController);

HomeController.$inject = ['$window', '$http', '$scope'];

function HomeController($window, $http, $scope) {

var vm = this;

vm.user = null;

initController();

function initController() {

$http({

url: 'http://localhost:8082/user',

method: "GET"

}).then(function (response) {

vm.user = response.data.name;

}, function (error) {

console.log(error);

});

};

$scope.logout = function () {

$window.sessionStorage.setItem('userData', '');

$http.defaults.headers.common['Authorization'] = 'Basic';

}

}

})();

8.Index.Html Pages

<!DOCTYPE html>

<html>

<head>

<base href="/" />

<script src="node\_modules/core-js/client/shim.min.js"></script>

<script src="node\_modules/zone.js/dist/zone.js"></script>

<script src="node\_modules/systemjs/dist/system.src.js"></script>

<script src="systemjs.config.js"></script>

<script>

System.import('app').catch(function (err) { console.error(err); });

</script>

</head>

<body>

<app>Loading...</app>

</body>

</html>

9.Login.Component

@Component({

selector: 'login',

templateUrl: './app/login/login.component.html'

})

export class LoginComponent implements OnInit {

model: any = {};

constructor(

private route: ActivatedRoute,

private router: Router,

private http: Http

) { }

ngOnInit() {

sessionStorage.setItem('token', '');

}

login() {

let url = 'http://localhost:8082/login';

let result = this.http.post(url, {

userName: this.model.username,

password: this.model.password

}).map(res => res.json()).subscribe(isValid => {

if (isValid) {

sessionStorage.setItem(

'token',

btoa(this.model.username + ':' + this.model.password)

);

this.router.navigate(['']);

} else {

alert("Authentication failed.");

}

});

}

}