Ads – SPA with AngularJS Lab

The **goal of this lab** is to continue the development of the Online Ads AngularJS application: add **login and logout functionality**.

Problem 5. Login and Logout

Step 1. Implement the **authService**. It is responsible for performing login / logout / register and keeping the information about the currently logged in user (anonymous site visitor / normal user / administrator):

authService.js

```
'use strict';

app.factory('authService',
  function ($http, baseServiceUrl) {
      return {
          login: function(userData, success, error) {
              var request = {
                  method: 'POST',
                  url: baseServiceUrl + '/api/user/login',
                  data: userData
              };
              $http(request).success(function(data) {
                  sessionStorage['currentUser'] = JSON.stringify(data);
                  success(data);
              }).error(error);
          },

          // TODO: implement "register" function (just like the login)

          logout: function() {
              delete sessionStorage['currentUser'];
          },

          getCurrentUser : function() {
              var userObject = sessionStorage['currentUser'];
              if (userObject) {
                  return JSON.parse(sessionStorage['currentUser']);
              }
          },

          isAnonymous : function() {
              return sessionStorage['currentUser'] == undefined;
          },

          isLoggedIn : function() {
              // TODO: implement this (similar to isAnonymous())
          },

          isNormalUser : function() {
              var currentUser = this.getCurrentUser();
              return (currentUser != undefined) && (!currentUser.isAdmin);
          },
```

```
        isAdmin : function() {
            // TODO: implement this (similar to isNormalUser())
        },

        getAuthHeaders : function() {
            var headers = {};
            var currentUser = this.getCurrentUser();
            if (currentUser) {
                headers['Authorization'] = 'Bearer ' + currentUser.access_token;
            }
            return headers;
        }
    }
  }
);
```

10 score

Step 2. Create the **login form** (the HTML template for the login):

login.html

```
<div class="box">
  <h2>Login</h2>
  <p>
    <label for="username">Username:</label>
    <input type="text" id="username" ng-model="userData.username" required />
  </p>
  <!-- TODO: similar code for the password field -->
  <p>
    <a ng-click="login(userData)" class="button">Login</a>
    <a href="#/register" class="link">Register</a>
  <p>
</div>
```

5 score

Step 3. Implement the **LoginController** to process the login form and login a user by calling the Login REST service:

LoginController.js

```
'use strict';

app.controller('LoginController',
    function ($scope, $rootScope, $location, authService, notifyService) {
        $scope.login = function(userData) {
            authService.login(userData,
                function success() {
                    notifyService.showInfo("Login successful");
                    $location.path("/");
                },
                function error(err) {
                    // TODO: display the error message returned by the server
                }
            );
        };
```

```
    }
);
```

5 score

Step 4. Modify the HTML template (view) for the **left sidebar** to display different content for anonymous site visitors, for logged-in users and for administrators:

left-sidebar.html

```
<!-- Navigation box for anonymous site visitors -->
<div ng-if="authService.isAnonymous()" class="box">
  <h2>Navigation</h2>
  <ul class="sidebar-menu">
      <li><a href="#/">Home</a></li>
      <li><a href="#/login">Login</a></li>
      <!-- TODO: add "Register" link -->
  </ul>
</div>

<!-- Login / register box for anonymous site visitors -->
<div class="box" ng-if="authService.isAnonymous()">
  <p>To publish a new ad, please login.</p>
  <!-- TODO: add "Login" button -->
  <a href="#/register" class="button">Register</a>
</div>

<!-- Navigation box for logged in normal users (not admins) -->
<div ng-if="authService.isNormalUser()" class="box">
  <h2>Navigation</h2>
  <ul class="sidebar-menu">
      <li><a href="#/">Home</a></li>
      <li><a href="#/user/ads">My Ads</a></li>
      <!-- TODO: add "Publish New Ad" link -->
      <!-- TODO: add "Edit Profile" link -->
      <li><a ng-click="logout()">Logout</a></li>
  </ul>
</div>

<!-- Navigation box for logged in admins -->
<div ng-if="authService.isAdmin()" class="box">
  <!-- TODO: implement the "Admin" navigation sidebar -->
</div>
```

10 score

Step 5. Implement the "logout" logic in the **AppController**. It will be used by "logout" buttons at the site header and at the left sidebar:

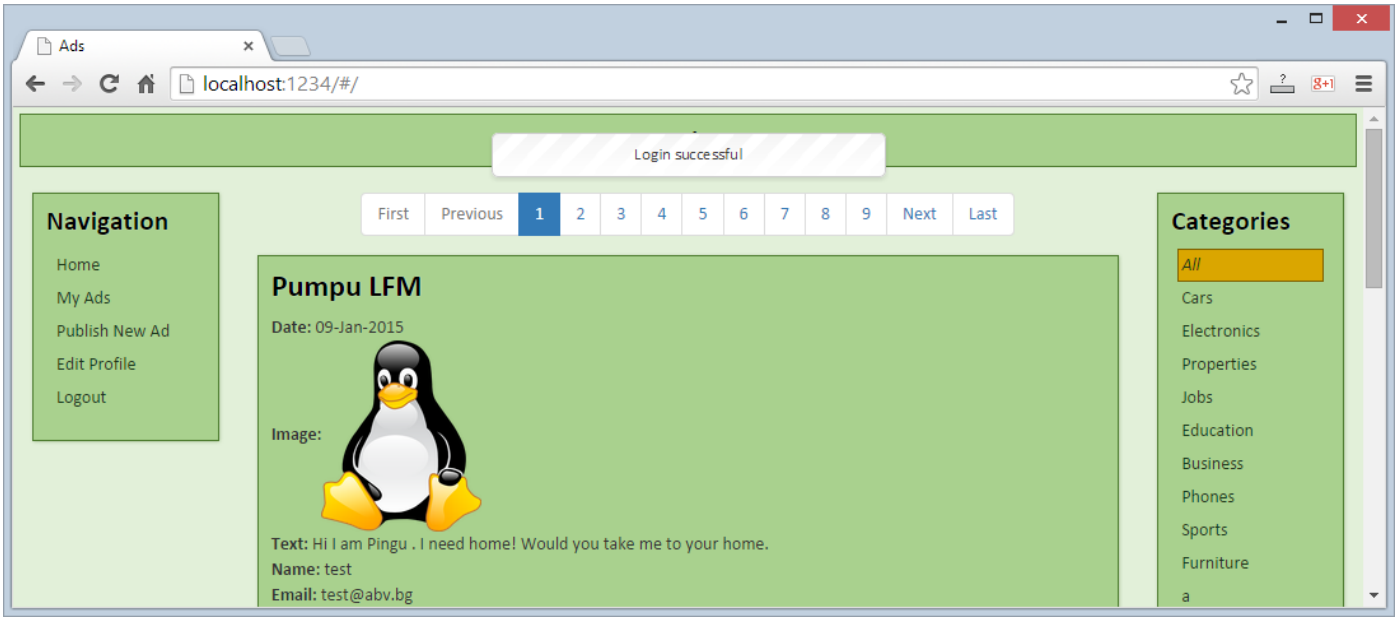
AppController.js

```
'use strict';

// The AppController holds the presentation logic for the entire app (common all
screens)
app.controller('AppController',
```

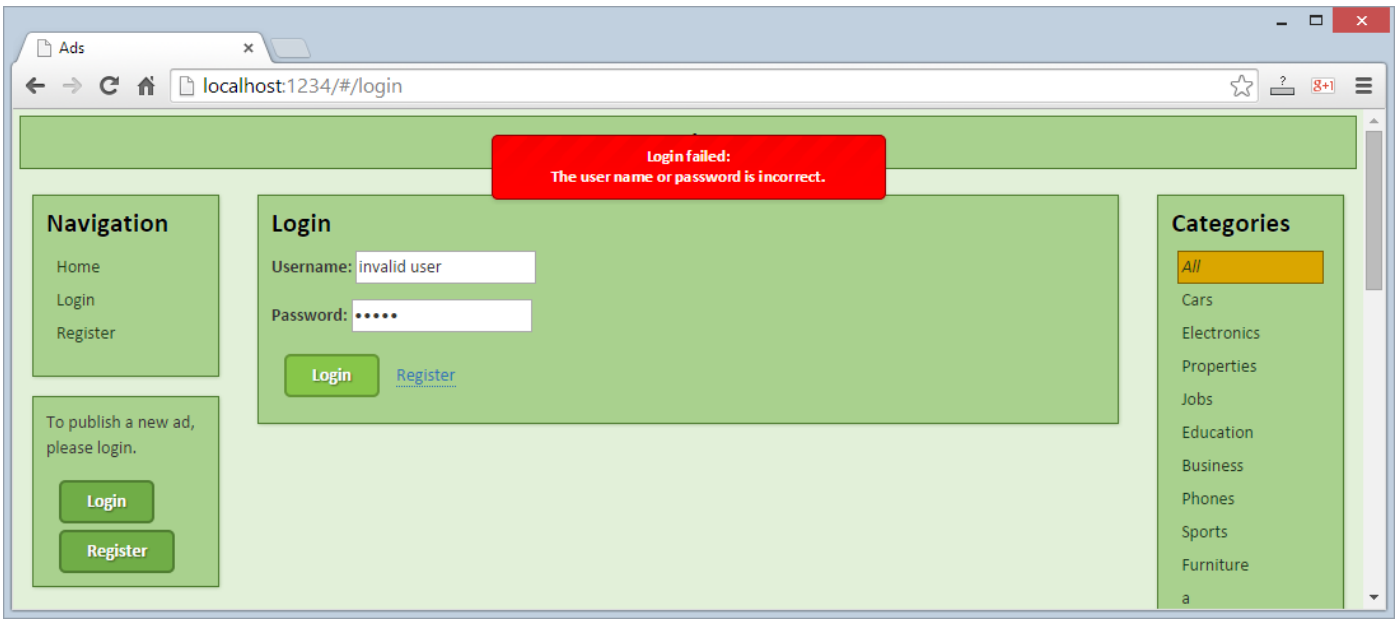
```
    function ($scope, authService) {
        // Put the authService in the $scope to make it accessible from all screens
        $scope.authService = authService;

        // Implement the "logout" button click event handler
        $scope.logout = function() {
            authService.logout();
            // TODO: display "Logout successful" notification
            // TODO: redirect to the application home page
        };
    }
);
```

10 score

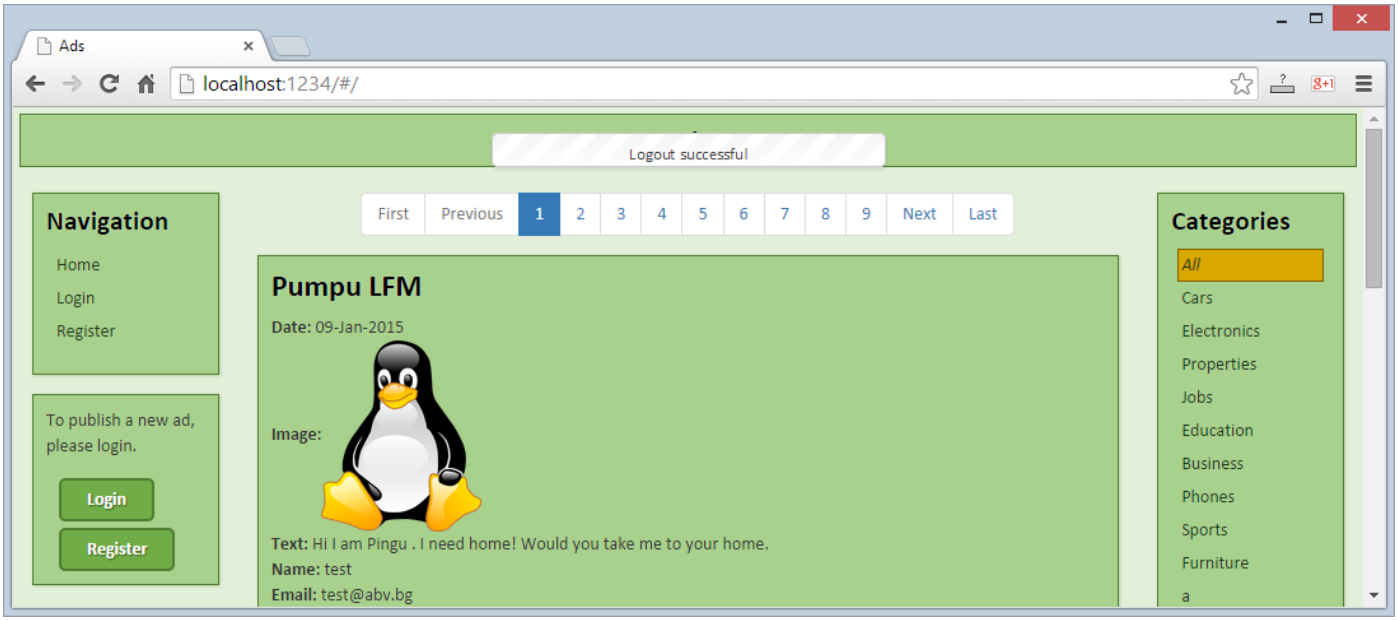
Step 6. Test your application to ensure **login** and **logout work correctly**:



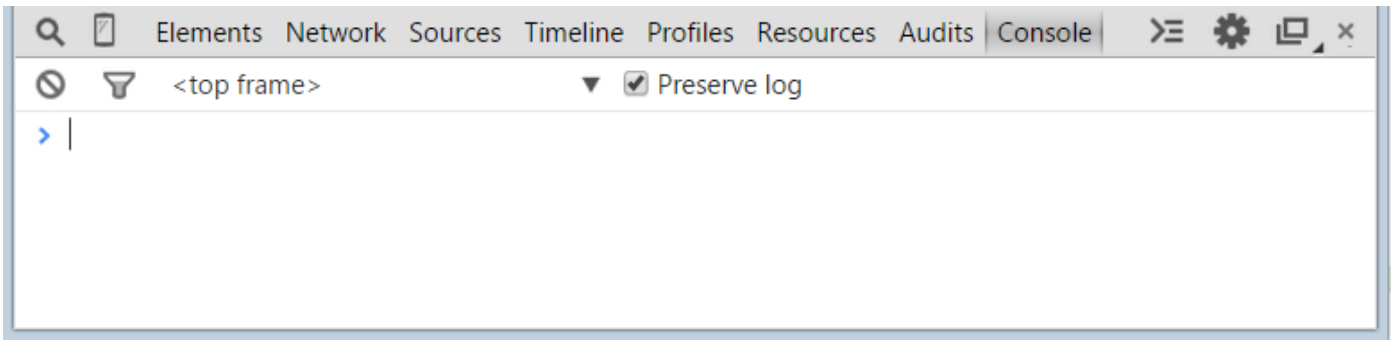
- Check also the **incorrect login** functionality:



- Check also the **logout** functionality:



- Check for errors at the Web browser's development console:



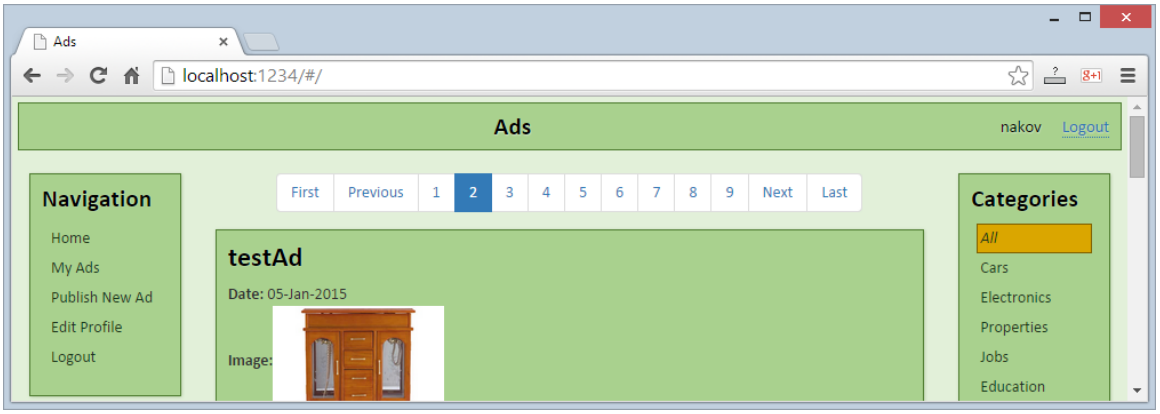
10 score

Step 7. Implement the **site header**. It should display different content for anonymous and logged in users:

header.html

```
Ads<span ng-if="pageTitle"> &ndash; {{pageTitle}}</span>

<div ng-if="authService.isLoggedIn()" class="user-area">
  <span>{{authService.getCurrentUser().username}}</span>
  <a ng-click="logout()" class="link">Logout</a>
</div>
```

Test the site header:



10 score