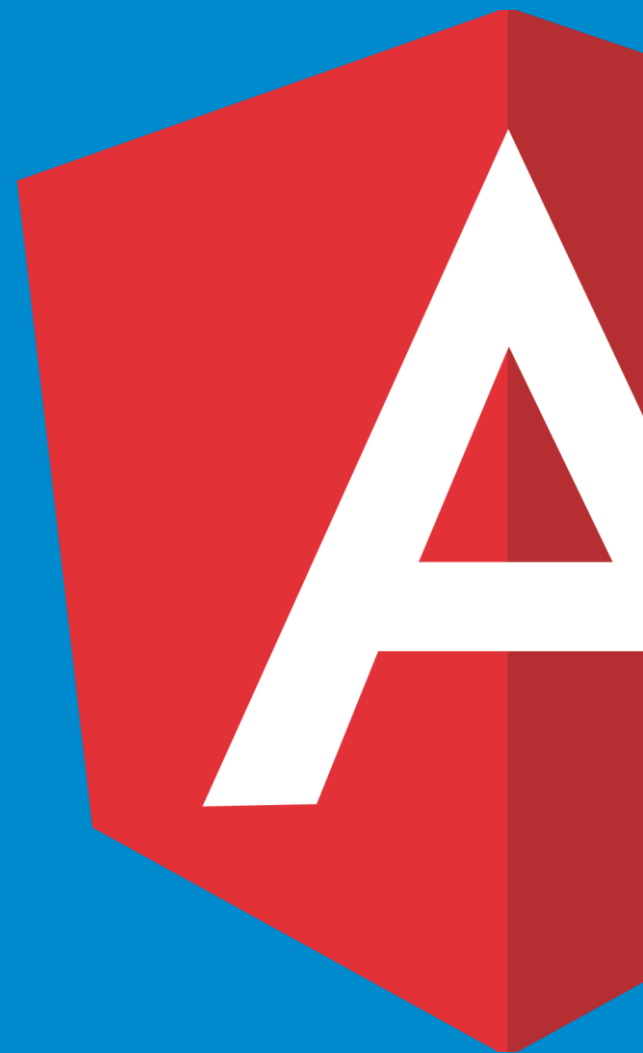


Web-school

Home task 5

# Angular



## Task 5. Modules & Services



Do not use http module for the task.

1. Create service for courses. Implement next methods:
  2. Get list
  3. Create course
  4. Get item by id
  5. Update item
  6. Remove item
1. Implement static collection of courses in the service.
2. Remove static array in your course component. Use service instead. Call service for items in appropriate lifecycle hook.
3. Implement delete action.
4. Add confirmation modal on delete item. ([Do you really want to delete this course? Yes/No](#)). You can use simple prompt window, third party package, your own solution to render modal.



5. Create authentication service. Implement next methods:

1. Login (stores fake user info and token to local storage)
2. Logout (wipes fake user info and token from local storage)
3. IsAuthenticated (boolean)
4. GetUserInfo (returns user login)

Use `authService` in the header component. Implement action on logout (wipe user info, `console.log` with action (Logout) name)

6. Implement Login page:

VIDEO COURSES 1

**Login**

E-mail: \*required  
Enter e-mail

Password: \*required  
Enter password

[Forgot password?](#) Login

Copyright © Videocourses. All rights reserved

## Details

1. **Reuse header/footer components.**

Implement logic to hide breadcrumbs and user info block. Call “IsAuthenticated” and act corresponding to the result. (Use `*ngIf`.)

2. **Create markup, module, and component for login page**

Do not implement any validation here. W/o `NgForm` for this step. Just bind inputs to class properties.

On “Login” button click - use login method of your authService. Then just console log ‘logged in successfully’. Fake for now.



You do not have to use routing, still there are no restrictions. W/o routing you can use construction below:

Place `<login-page></login-page>` into App component.

## Evaluation criteria

- 2 Added some simple markup + css for login page. Added services but not working.
- 3 Partially done with critical issues
- 4 Added implementations for services, login page.
- 5 Did all above without major issues

### Extra task



Proceed with unit testing. Now you have your first services. It's time to cover them with unit tests. Related materials provided in the micro lecture after general.