Reuse

General part

Explain generally about:

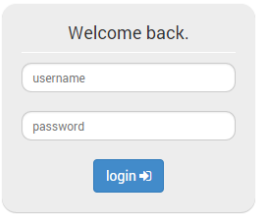
- Reuse in Object-Oriented Programming.
- Mechanisms for reuse in Java.
- Mechanisms for reuse in JavaScript
- Mechanisms for reuse in general
- Mechanisms for reusing ideas

Practical part

In angular we have some very useful possibilities for creating reusable source code. Here you must create examples on reusable code in angular using filters, directives, and services/factories.

1. Create a filter that will produce the name of a person in the form `"lastname, firstname"`.
   Given a person object on the form `{ firstName: 'Peter', lastName: 'Smith' }`
   it should output `"Smith, Peter"` when used like this `{{ person | name }}`
   Create an angular application that shows the filter in use.

2. Create a directive named login-form that encapsulate the following:

   You do not need to implement the same style as in the figure, just the functionality. It must be possible to use the directive with the following html snippet:

   ```
   <login-form></login-form>
   ```

   You should provide an angular application that shows the directive in use. If you have time provide a header attribute so the header can be defined in html:

   ```
   <login-form header="Welcome back."></login-form>
   ```

3. Create a service or factory which provide three functions
   - titleCase("my example service") should return: "My Example Service",
   - camelCase("my example service") should return: "MyExampleService",
   - dashCase("my example service") should return: "my-example-service".

   Create a controller that uses the service and an angular application that shows all functions in use.

4. Write the following unit tests, using Jasmine and Karma:
   - Write a unit test that tests the filter implemented in 1)
   - Write a unit test that tests the service implemented in 3)
   - Write a unit test that test the controller implemented in 3)

Setting up the project for testing:

Add a folder called test, and create an empty JavaScript file called karma.conf.js in the folder.

Copy this content into the file:

```
module.exports = function(config) {
  config.set({

   basePath : '../',
    frameworks: ['jasmine'],

    files: [
      'bower_components/angular/angular.js',
      'app.js',
      'test/*.js'
    ],
    exclude: [],

    preprocessors: {},

    plugins : [
      'karma-chrome-launcher',
      'karma-jasmine'
    ],

    colors: true,
    logLevel: config.LOG_INFO,

    autoWatch: true,

    browsers: ['Chrome']

  });
};
```

Make sure that you include all the files you want to test in the **files** section

Do the following:

```
npm install jasmine-core
npm install karma
npm install karma-chrome-launcher
npm install karma-jasmine
```

Add your test file(s) to test folder and include tests as sketched below:

```
describe("Tests", function () {

    beforeEach(module("MyApp"));

    describe("Test", function () {

        it('should return true', function(){
            expect("Hello World").toBe("Hello World");

        })
    });
})
```