Reusable "components" with AngularJS

General part

Explain generally about:

- Mechanisms for reuse in Java.
- Mechanisms for reuse with Angular
- Mechanisms for reuse in general
- Mechanisms for reusing ideas

Practical part

Start Code for this exercise: *You should use the project found in reusableControlsAngular.zip as the starting point for this exercise. It is a basic Maven Web Project, but this is a Client-side only exercise, so you won't need the src-folder.*

In angular we have some very useful possibilities for creating reusable source code. Here you must create examples of reusable code 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.
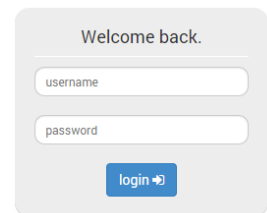
2. Create a directive named login-form that encapsulate the following:

   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".

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