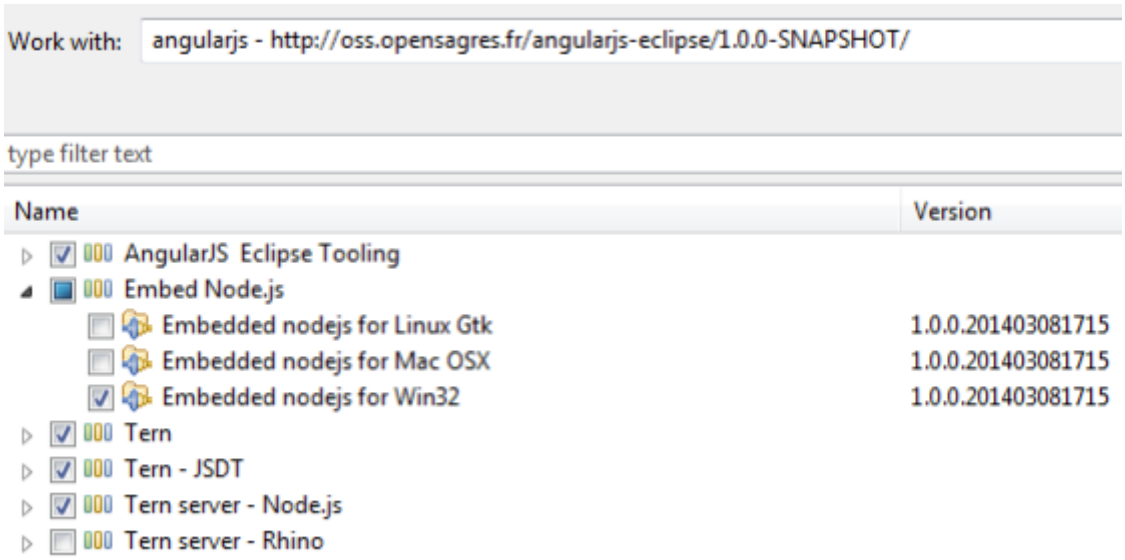
Install and Configure AngularJS + Contact List

Part 1: Install AngularJS plugin and set up a new project – 5 Marks

I am using Eclipse as my IDE, if you wish to use another, feel free.

- Install the AngularJS plugin - Go to help – install new software – and install the snapshot provided at http://oss.opensagres.fr/angularjs-eclipse/1.0.0-SNAPSHOT/
- Install the following:



- Choose File -> New -> Static Web Project
  - If asked to use Web Perspective, select Yes.
- Right-click on your project and select Configure -> Convert to AngularJS Project
- Create a new folder called "Lab 6", and within it add the following (for now, empty) files:
  - Index.html
  - Todo.css
  - Todo.js
- Copy in the code from the "Add Some Control" tutorial on https://angularjs.org/ to your 3 files and run the html file in your browser
  - What does this code do? – 1 Mark
  - What does 'archive' do? – 1 Mark
  - How is the strike-through added to the ticked items? – 1 Mark
  - Add another default toDo item – 1 Mark
  - Add a new paragraph containing the value of the input text that appears in real-time. – 1 Mark

Part 2: Contact List using controllers and Partials – 5 Marks

Set up a new AngularJS project in Eclipse with the following files – 1 Mark

- index.html
- app.js
- controllers.js
- partials/add.html
- partials/index.html
- partials/info.html
- partials/remove.html

Download this week's code starter and copy the code to the appropriate locations. This is the starter code for a simple Contact list app using HTML5, Bootstrap and AngularJS. In this app, we will list all contacts in a list, and give the user an option to delete any contact in the list or add a new contact. Each contact has a name and phone property.

Complete the app.js and controllers.js files – 2 + 2 Marks

App.js

This is our incomplete route provider – complete the code to add the add and remove controller routes. Also, add in the default route to partials/index.html.

Complete the code for the routeProvider

Controllers.js

Look at the top level controller for our app. The data is stored as a simple JavaScript object. AngularJS doesn't require you to extend any special classes, or stick to one particular inheritance pattern, so you're free to use whatever patterns you like. This keeps the business logic clean, and makes code far easier to test.

Complete the controllers for adding and removing contacts.