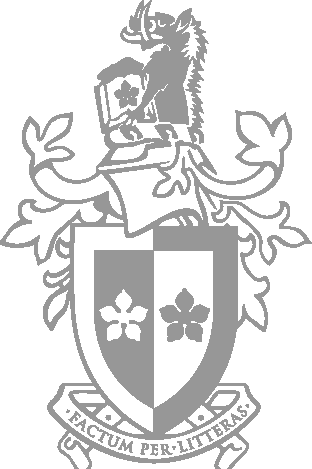
Faculty of Science, Engineering and Technology





**Interface Design and Development**

Distinction Task 2: Creating Config

**Overview**

Module enables you to create a config for your web application. In this task you will create a config that will display unit information using a view template in your web application.

**Purpose:** Learn how to use module, and create config for routing.

**Task:** Create a web app that allows users to click of a unit to show more information. When the user clicks on a specific unit, the web app will pass the unit code to the routed controller to retrieve the unit’s information and populate the view template for display.

**Time:** This is a Distinction Task and can be completed at any stage. Do ensure that you should continue on to Topic 6 (Pass & Credit Task) and return to this task at a later date if it is not completed on time.

**Resources:**

* + - Lecture notes #3, #4 and #5

***Submission Details***

You must submit the following files to Blackboard:

* Unit information main source code (units2.html).
* Unit information view template source code (units2template.html).
* Unit module and config (which include the route and necessary controller) source code (appunits2.js).
* Screenshot of the web app.

Make sure that your task has the following in your submission:

* The routing web application is HTML5 compliant.
* Demonstrates understanding in using the AngularJS framework.
* Demonstrates use of AngularJS config and route.



**Instructions**

Implement the unit information web app that display detailed unit information. In this web application, you will need to implement a config, route and controller to show the detailed information of a selected unit.

1. Start by creating a new HTML file in Brackets.
2. Implement the basic outline of an AngularJS web app with the appropriate scripts.

**Note:** It is a good practice to write the controller in a separate file.

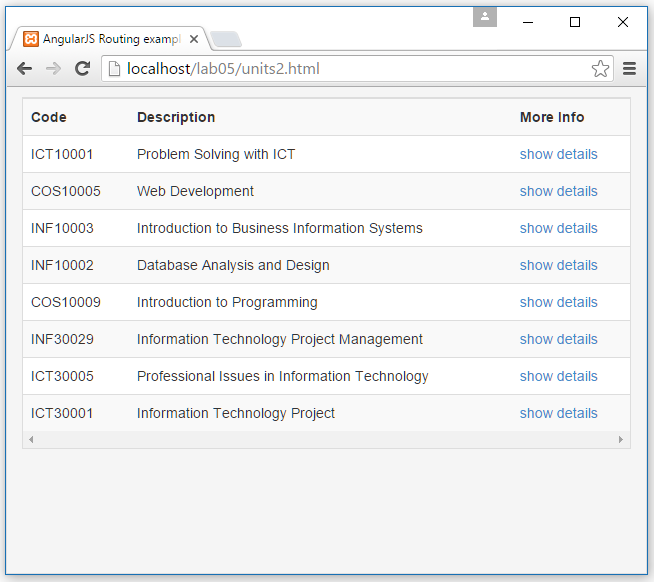
<script src="js/ appunits2.js"></script>

1. Create the web application HTML file with the following login.
   * Initialise the array of objects with the following unit information

|  |  |  |  |
| --- | --- | --- | --- |
| Unit codes | Units | Credit points | Type |
| ICT10001 | Problem Solving with ICT | 12.5 | Core |
| COS10005 | Web Development | 12.5 | Core |
| INF10003 | Introduction to Business Information Systems | 12.5 | Core |
| INF10002 | Database Analysis and Design | 12.5 | Core |
| COS10009 | Introduction to Programming | 12.5 | Core |
| INF30029 | Information Technology Project Management | 12.5 | Core |
| ICT30005 | Professional Issues in Information Technology | 12.5 | Core |
| ICT30001 | Information Technology Project | 12.5 | Core |

Table 1: Unit Information

* + Display the table that list the unit information from lab tasks 3.2. Search and filter is optional for this task.



* + A ngView that will contain the respective view component.

**Note:** Mark up the code using Bootstrap is optional for this task. Bootstrap components will discuss in lecture 6.

1. Create the unit detail view templates that will list the unit information. We will limit the information to unit code, description, credit point and type.

**Note:** The unit code is used to retrieve detail information from the database. However for this task, we will retrieve the unit information from the model via ngRepeat with the unit code as filter.

1. Create the controller file appcalendar.js with the module template.

Web App: **appunits2.js**

————————————————

var app = angular.module('myApp', [**'ngRoute'**]);

app.config(['**$routeProvider**', function(**$routeProvider**) {

$routeProvider.

when('<menu option/parameter>', {

templateUrl: '<path / filename>',

controller: '<name of controller>'

});

}]);

app.controller('<name of controller>',

function(**$scope**, **$routeParams**) {

<your code which may include creating new model

in scope from the parameters passed via route>

});

1. Your web app should now be complete. Please add comments in your code to explain what it does. Lastly, make sure you test it on the browser to make sure that it works as you expect.

**Note:** Testing your work as a file i.e. file://c:/documents/lab05/units2.html will only work in Firefox. You need to test web apps **with config/route on a web server when using Chrome**, Internet Explorer and Edge. Brackets uses a web server for its live view in Chrome.

