**1.2 The Beginning of All Routes 230 pts**

Let's get started creating routes for this super-duper NoteWrangler app! The night is late and the coffee is gettin' cold. Let's set up our app so we can start wranglin' some notes!

1. Inside the app.js file, declare an Angular module for our NoteWrangler app. For now it will have no dependencies.

2. What is the name of the module you need to include to use routes? Update the app.js module declaration to include it.

3. Define the module’s config() function and inject $routeProvider.

Define config() function on our applications module and pass in $routeProvider.

angular.module('NoteWrangler', ['ngRoute'])

.config(['$routeProvider', function($routeProvider) {}]);

4. Create a simple route for when /notes is in the URL, link to the index.html template in the templates/pages/notes/ directory.

Use $routeProvider to watch the notes route and show the notes index page.

angular.module('NoteWrangler', ['ngRoute'])

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

$routeProvider.when('/notes', {

templateUrl: 'templates/pages/notes/index.html'

});

}]);

5. Now, add a new div tag to our index.html with an attribute directive that will include the rendered template for the current route.

app.js:

angular.module('NoteWrangler',['ngRoute'])

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

$routeProvider.when('/notes',{

templateUrl: '/templates/pages/notes/index.html'

});

}]);

index.html:

<!DOCTYPE html>

<html lang="en" ng-app="NoteWrangler">

<head>

<meta charset="utf-8">

<title>Note Wrangler</title>

<meta name="viewport" content="width=device-width, initial-scale=1">

<link rel="stylesheet" href="css/application.css" />

</head>

<body>

<div class="nav-wrapper has-dropdown">

<div class="nav-content">

<div class="wrapper">

<div class="nav-content-layout">

<div class="nav-list">

<a href="#/"

class="list-item"

ng-click="selected = 'notes'"

ng-class="{'active': selected == 'notes'}"> Notes </a>

<a href="#/users"

class="list-item"

ng-click="selected = 'users'"

ng-class="{'active': selected == 'users'}"> Users </a>

</div>

</div>

</div>

</div>

</div>

<div class="hero-wrapper">

<div class="hero-content">

<div class="hero"></div>

</div>

</div>

<div class="main-wrapper">

<div ng-view></div>

</div>

<!-- Load Js libs -->

<script src="./js/vendor/jquery.js"></script>

<script src="./js/vendor/angular.js"></script>

<script src="./js/vendor/angular-route.js"></script>

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

<!-- Controllers -->

<script src="./js/controllers/notes-create-controller.js"></script>

<script src="./js/controllers/notes-edit-controller.js"></script>

<script src="./js/controllers/notes-index-controller.js"></script>

<script src="./js/controllers/notes-show-controller.js"></script>

<script src="./js/controllers/users-index-controller.js"></script>

<script src="./js/controllers/users-show-controller.js"></script>

</body>

</html>

# 1.3 404, Route Not Found 250 pts

The [**Correct**](http://campus.codeschool.com/courses/staying-sharp-with-angular-js/) method is used to set a route when no other route definition is matched.

Sol: otherswise

# 1.4 A New Route and a Root Route 240 pts

Practice using when() by creating a route for new notes and a route for the root of the application.

1. Create a new route which maps /notes/new to the notes edit.html template.

2. Create a route for / that **redirects** to the /users index page.

Add another call to when() on the $routeProvider for the / route and provide an options object with the redirectTo value set to '/users'.

.when('/', {

redirectTo: '/users'

})

3. Create a default route that will redirect the user to /notes when an undefined route is used.

App.js:

angular.module('NoteWrangler', ['ngRoute'])

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

$routeProvider

.when('/notes', {

templateUrl: 'templates/pages/notes/index.html'

})

.when('/users', {

templateUrl: 'templates/pages/users/index.html'

})

.when('/notes/new', {

templateUrl: 'templates/pages/notes/edit.html'

})

.when('/', {

redirectTo: '/users'

})

.otherwise({redirectTo: '/notes'})

;

}]);

# 1.6 $http Methods I 250 pts

What $http method is used for updating existing data?

Sol: $http.put

# 1.7 $http Methods II 250 pts

What $http method is used for retrieving data?

Sol: $http.get

# 1.8 Llamas in Space 250 pts

A brand new llama recruit has signed up to go into space and he's ready to be added to the list on the server. Make a $http() function call to make the proper request to the server at the /llamas.json URL.

Sol:

var newLlamaRecruit = {

llamaOne: 'Larry Llama'

};

$http({method:'post',url:'/llamas.json',data:newLlamaRecruit});

# 1.9 Inside Route Controller 240 pts

We want to display notes from the server within our template. Let's get started by creating a controller in our routes file.

1. Create a controller function inside this /notes route.

2. Give the newly created controller an alias of notesCtrl so that we can access it from within the notes index template.

3. We'll need to make a call to the server to retrieve the list of notes data, so we'll need to use the $http service in the controller we just created. Inject the $http service into the controller.

4. Use the $http() function to make an HTTP request to the /notes URL. Upon success, save the resulting data to a notes variable **on** the controller.

You want to use the $http service to make a GET request to the /notes route and save the success data to controller.notes.

.when('/notes', {

templateUrl: 'templates/pages/notes/index.html',

controller: function($http) {

var controller = this;

$http({method: 'GET', url: '/notes'}).success(function(data) {

controller.notes = data;

});

},

controllerAs: 'notesCtrl'

})

App.js:

angular.module('NoteWrangler', ['ngRoute'])

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

$routeProvider

.when('/', {

redirectTo: '/users'

})

.when('/notes', {

templateUrl: 'templates/pages/notes/index.html',

controller: function($http){

var controller=this;

$http({method: 'get', url: '/notes'}).success(function(data){

controller.notes = data;});},

controllerAs: 'notesCtrl',

})

.when('/users', {

templateUrl: 'templates/pages/users/index.html'

})

.when('/notes/new', {

templateUrl: 'templates/pages/notes/edit.html'

})

.otherwise({

redirectTo: '/notes'

});

}]);

# 1.10 Outside Controller 250 pts

In order for a new note to be created by a user on the new notes page, we need to add a controller and use the $http service to create a new note. This time around, we'll use an external controller for the new notes page.

1. Attach the NotesCreateController to the new notes route.

2. Give the controller an alias we can reference in the template.

3. Use the ngModel directive to save the data from each form input and textarea in a model called note. Remember the data key on the model matches the form input name. Look at the description textarea in the notes/new.html template for an example.

4. Use the ngClick directive to call the saveNote() function of the controller when the save button is clicked. Remember to pass the note model to the function.

App.js:

angular.module('NoteWrangler', ['ngRoute'])

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

$routeProvider

.when('/', {

redirectTo: '/users'

})

.when('/notes', {

templateUrl: 'templates/pages/notes/index.html',

controller: function($http){

var controller = this;

$http({method: 'GET', url: '/notes'}).success(function(data){

controller.notes = data;

});

},

controllerAs: 'notesCtrl'

})

.when('/users', {

templateUrl: 'templates/pages/users/index.html'

})

.when('/notes/new', {

templateUrl: 'templates/pages/notes/edit.html',

controller: 'NotesCreateController',

controllerAs: 'ncCtrl'

})

.otherwise({

redirectTo: '/notes'

});

}]);

New.html:

<div class="new-note">

<div class="new-note-container">

<form class="form">

<fieldset>

<label for="title">Title</label>

<input name="title" ng-model="note.title" />

</fieldset>

<fieldset>

<label for="description">Description</label>

<textarea name="description" ng-model="note.description"></textarea>

</fieldset>

<fieldset>

<label for="content">Content</label>

<textarea name="content" ng-model="note.content"></textarea>

</fieldset>

<button class="btn" ng-click="ncCtrl.saveNote(note)">Save</button>

</form>

</div>

</div>

# 1.11 Getting a Note's id From the URL 250 pts

We can gain access to the note's id in the following URL: http://example.com/#/notes/42 by injecting the [**Correct**](http://campus.codeschool.com/courses/staying-sharp-with-angular-js/) service into the NotesShowController.

Sol: $routeParams

# 1.12 Displaying a Single Note 240 pts

Use the routeParams service to fetch a single note based on the id given in the URL from the server URL /notes/:id.

An example of a possible URL for this might be: http://example.com/#/notes/42.

1. Inject the routeParams service into the NotesShowController so that we get access to the id in the URL.

angular.module('NoteWrangler').controller('NotesShowController', ['$routeParams', function($routeParams) {

}]);

2. Inject the $http service into the NotesShowController so we can use it to fetch a note.

3. Use the $http() function to fetch a note from the /notes/:id URL. Upon success, save the resulting data to a note variable **on** the controller.

#### notes-show-controller.js:

angular.module('NoteWrangler')

.controller('NotesShowController', ['$routeParams','$http', function($routeParams, $http) {

var controller = this;

$http({method: 'get', url: '/notes/'+$routeParams.id}).success(function(data){

controller.note=data;});

}]);