Week 4

Quiz 32

1. True or False. In Component-based Architecture, a component has well-defined public API. 1 / 1 point

- (●) True
- ( ) False

Correct

2. The default label (property name) that refers to the component's controller in the template is: 1 / 1 point

- ( ) component
- ( ) controller
- (●) $ctrl
- ( ) this

Correct

3. True or false? You must always provide a controller for every component you create. 1 / 1 point

- ( ) True
- (●) False

Correct
True. If you don't provide one, Angular will create an empty one for you automatically.

4. What new method was introduced to the component lifecycle methods in Angular 1.5.8 that allows you to hook into each turn of the digest cycle? 1 / 1 point

- ( ) $onDigest
- ( ) $onChanges
- ( ) doCheck
- (●) $doCheck

Correct
Correct. (All component lifecycle methods start with the '$')

Quiz 33

1. In implementing the Publish-Subscribe design pattern, what does Angular use for the "channel" portion of the pattern?

1 / 1 point

- ○ Global object (window)
- ○ Shared Services
- ◉ $scope
- ○ $timeout

Correct

2. If your component is set up to listen for some event using the $rootScope, what must you ensure to do in such a setup? For example, what must be done (if ANYTHING!) when you see the following in your component:

1 / 1 point

```
1  $ctrl.$onInit = function () {
2    $rootScope.$on('alert:turnOn', handler);
3
4    function handler(event, data) {
5      //...
6    }
7  }
8
```

- ○ Aside from implementing the handler function, not much left to do. Angular takes care of the rest.
- ○ The handler function must call the $emit method with the event name of 'alert:turnOff'.
- ○ Weeeeeee! Ok, clearly not the right answer, but so much fun! 😀
- ◉ We must deregister the handler when this view is destroyed. We can do this by declaring a local to controller variable called 'cancelHandler' and changing this line 2 to this:

```
1  cancelHandler = $rootScope.$on('alert:turnOn', handler);
```

Then, add another lifecycle method to the component like this:

```
1  $ctrl.$onDestroy = function () {
2    cancelHandler();
3  };
```

Correct

3. Is it ever OK to call $rootScope.$emit(...)?

1 / 1 point

- ○ No, never.
- ◉ Yes, calling $emit on the $rootScope would make ONLY the $rootScope in the application have a chance at catching that event.

Correct
While this is true, if you're finding yourself doing something like that, it probably means your architecture is messed up... USUALLY, you shouldn't be forced into such communications.

Quiz 34

1. Given the following snippet of code, which controllers would be declared in the 'myApp' module?

1 / 1 point

```
angular.module('myApp', [])
.controller("MyController1", MyController1);

angular.module('myApp')
.controller("MyController2", MyController2);

angular.module('myApp', [])
.controller("MyController3", MyController3);
```

- ○ MyController1
- ○ MyController2
- ◉ MyController3
- ○ MyController1 and MyController3
- ○ MyController2 and MyController3
- ○ None. This code will cause an error.

2. When importing several *artifacts* of a module using the <script> tags in the HTML, dependencies of some service must be declared FIRST and only then the service itself.

1 / 1 point

- ○ True
- ◉ False

Correct
Correct. Individual artifacts of a module can be imported in any order. Angular automatically figures out which ones need to be instantiated first.

3. When importing your Angular code using the <script> tags in HTML, the order in which you list them does not matter as long as you specify the angular libraries code first (like angular.min.js, etc.)

1 / 1 point

- ○ True
- ◉ False

Correct
Correct. While individual module artifacts can be listed in any order, the module declarations MUST be listed prior to any code that *uses* that module in order to attach artifacts to it like controller, components, etc.

Quiz 35

1. In Single Page Application (SPA) model, the responsibility of routing falls onto

1 / 1 point

- ○ the server
- ◉ the browser
- ○ Cisco

Correct
It's the browser, though Javascript, that's responsible for routing.

2. Given the following HTML code, referred to as snippet #1:

1 / 1 point

```
1  <section>
2    <ui-view>
3
4    </ui-view>
5  </section>
```

and Javascript code, referred to as snippet #2

```
1  angular.module('App')
2  .config(RoutesConfig);
3
4  RoutesConfig.$inject = ['$stateProvider', '$urlRouterProvider'];
5  function RoutesConfig($stateProvider, $urlRouterProvider) {
6    $stateProvider
7      .state('viewA', {
8        url: '/viewA',
9        templateUrl: 'viewA.html'
10     })
11
12     .state('viewB', {
13       url: '/viewB',
14       templateUrl: 'viewB.html'
15     });
16
17   $urlRouterProvider
18     .otherwise('/viewB');
19 }
```

What would appear on on line 3 of snippet #1 if the user were to attempt to go to a URL in the form of http://someHost/#/Weeeeeeeeee?

- ○ viewA.html
- ○ viewB.html
- ○ Contents of viewA.html
- ◉ Contents of viewB.html
- ○ Browser would display a 404 error because there is no such URL configured.
- ○ Seriously, this is a professional course, can you stop with the whole 'Weeee' thing? (Yaakov's response: "I refuse!")

Quiz 36

1. Given the following snippet of Javascript code:

1/1 point

```
...
.state('home', {
  url: '/',
  templateUrl: 'home.html',
  controller: 'HomeCtrl as home'
});

```

and the following as part of the home.html file:

```
<div>
  {{ HomeCtrl.getExcitement(); }}
</div>
```

What would the user see? Assume there is method called getExcitement() defined on the implementation of the controller declared on the state. The method returns... you guessed it: "Weeeee".

- ○ Weeeee
- ○ Nothing. It will display an error on the console.
- ◉ Nothing. Absolutely nothing anywhere: not on the web page, not on the console.

Quiz 37

1. Given the following snippet of Javascript code:

1/1 point

```
...
.state('home', {
  url: '/',
  templateUrl: 'home.html',
  controller: 'HomeCtrl as home',
  resolve: {
    data1: 'Weeeee',
    data2: ['Service', function (Service) {
      return Service.methodReturnsPromise();
    }]
  }
});

```

and the following implementation of HomeCtrl:

```
HomeCtrl.$inject = ['data2'];
function HomeCtrl(data2) {
  ...
}
```

Which of the following statements are true?

- ☐ An error will occur because HomeCtrl forgot to inject 'data1'.
- ☑ 'home' view state will not be transitioned to until the promise returned by 'methodReturnsPromise()' is resolved.

  Correct
  Correct!

- ☑ 'home' view state will not EVER show if the promise returned by 'methodReturnsPromise()' is rejected or an error occurs during 'data2' resolution.

  Correct
  Correct!

Quiz 38

1. Given the following snippet of Javascript code:

1 / 1 point

```
1   ...
2   .state('home', {
3     url: '/{param1}',
4     templateUrl: 'home.html',
5     controller: 'HomeCtrl as home',
6     resolve: {
7       data1: 'Weeeee',
8       data2: ['Service', function (Service) {
9         return Service.methodReturnsPromise(???);
10      }]
11    }
12  });
13
```

We need to pass the value of param1 into the 'methodReturnsPromise'. What do we need to do to get that done?

- ○ Replace line 9 with:

```
1   return Service.methodReturnsPromise(param1);
2
```

- ○ Replace line 9 with:

```
1   return Service.methodReturnsPromise('param1');
2
```

- ◉ Replace line 8 and 9 with:

```
1   data2: ['$stateParams', 'Service', function ($stateParams, Service) {
2         return Service.methodReturnsPromise($stateParams.param1);
3
```

- ○ Replace line 9 with:

```
1   return Service.methodReturnsPromise({'param1'});
2
```

Quiz 39

1. Given the following snippet of Javascript code:

1 / 1 point

```
...
.state('items.detail', {
  url: '/detail/{param1}',
  templateUrl: 'detail.html',
  ...
  }
});
```

Assuming the 'items' view state has a templateUrl with the value of 'items.html', what tag should be somewhere inside of the 'items.html' for this child state setup to make sense.

- ○ `<ui-router child='true'></ui-router>`
- ◉ `<ui-view></ui-view>`
- ○ `<ui-child></ui-child>`

Correct
Correct!

2. 'resolve' properties declared on the parent view state can be *injected* into a controller declared for the child state *without* having to declare 'resolve' property on the child state definition.

1 / 1 point

- ◉ True
- ○ False

Correct
Correct! Those properties are automatically available to the child states.

Quiz 40

1. True or False? All ui-router events are fired on the $scope

1 / 1 point

- ○ True
- ◉ False

Correct
Correct! They are always fired on the $rootScope.

2. In order to respond to errors that occur during the transition from one view state to another (including errors from resolves), you have to..

1 / 1 point

- ○ Wrap transitions in an try/catch block.
- ○ Nothing to do, ui-router automatically handles errors.
- ◉ Set up a listener for $stateChangeError event and handle the error inside of that listener.

Correct
Correct! In fact, that's pretty much the only straightforward way of getting that done.