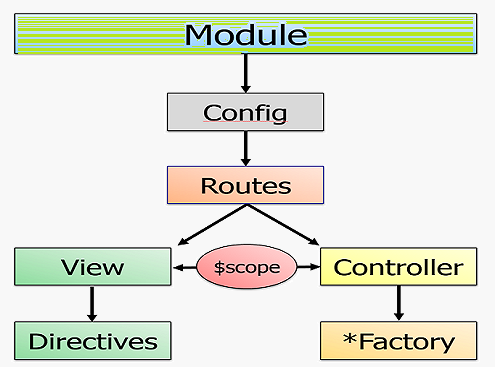
**<https://markwillcollins.silvrback.com/7-things-i-wish-i-knew-about-angularjs?utm_content=buffer9f368&utm_medium=social&utm_source=linkedin.com&utm_campaign=buffer>**

**Angularjs Interview Questions and Answers**

What is the [angular js](http://www.code-sample.com/2014/05/angularjs-interview-questions-and.html)?

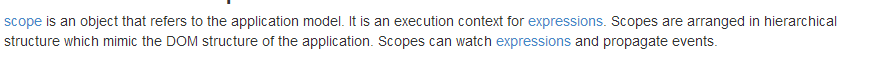
Angular js is a web framework and maintain by Google. This is usefully for both client side and server side web application and the main features are given below.  
[](http://3.bp.blogspot.com/-VRhv1lVxCfs/U4pHf3JLirI/AAAAAAAAAcI/9Rq9EvpeM_s/s1600/angular+js.png)

 1. Its support two way data-binding.  
     2. Its support MVC pattern.  
     3. Its support static template and angular template.  
     4. You can add custom directives.  
     5. Its support REST full services.  
     6. Its support form validations.  
     7. Its support both client and server communication.  
     8. Its support dependency injection.  
     9. Applying Animations.  
     10. Event Handlers  
  
What are the main features in angular js?

Ans : The main key features are

1. Scope,
2. Controller,
3. Model,
4. View,
5. Services,
6. Data Binding,
7. Directives,
8. Filters,
9. Validation and
10. Testable.

What about a scope in angular js ?

[](http://2.bp.blogspot.com/-jQIjSyaiafM/U4pEf9rlP9I/AAAAAAAAAb8/3TiJmA2UZJU/s1600/What+are+Scopes.png)

How to initialize a select box with options on page load in angular js ?

Ans : initialize work with the help of "ng-init"  
<div ng-controller="[myApps/myDashboard/myAccount](http://www.code-sample.com/2014/05/angularjs-interview-questions-and.html" \t "_blank)" ng-switch on="!!myAccounts" ng-       init="loadMyDatayData()">

How to enable and disable buttons with using condition?

[](http://3.bp.blogspot.com/-sK4Lm7ACRUU/U4oxbkIQzaI/AAAAAAAAAaw/ubs1H8jnuiI/s1600/Enable+Disable+Buttons+Conditionally.png)

Ans . By using the "ng-show" and "ng-disabled" directive, Please see the below example.

<div ng-app="myapp">

    <div ng-controller="myScope">

       First Name :  <input type="text" ng-model="fName" /><br>

       Last Name :  <input type="text" ng-model="lName" /><br>

        <button ng-disabled="!fName || !lName">Submit</button>

    </div>

</div>

<script>

var $scope;

var app = angular.module('myapp', []);

function myScope($scope) {

}

</script>

How to using looping through a collection for each item in angular js?

[](http://4.bp.blogspot.com/-6HMMWRQchLI/U4oxb2mTSAI/AAAAAAAAAbA/qjS0PxER_5w/s1600/loop+through+a+collection.png)

Ans. By using the "ng-repeat" directive, Please see the below example.

<script>

    function Main($scope) {

$scope.getWelcomeEntities = function(){return [{Name:'for a loop in Angular JS.'}];};

    };

</script>

<body ng-app>

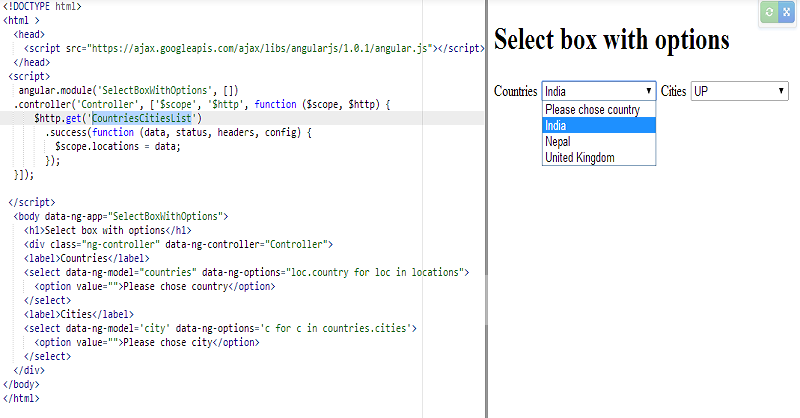
  <div ng-repeat="item in getWelcomeEntities()">

    code sample {{item.Name}}

  </div>

</body>

How to bind selection box with options in the angular js?

[](http://4.bp.blogspot.com/-8EKuGWzoQRA/U4oxcSW3eyI/AAAAAAAAAbM/rVdn6z0hGr8/s1600/option+binding+1.png)

[](http://4.bp.blogspot.com/-vp8LUq6_IaQ/U4oxci7LPCI/AAAAAAAAAbI/1HcfWQU4-rE/s1600/option+binding+2.png)

Ans:

<!DOCTYPE html>

<html>

  <head>

    <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.0.1/angular.js"></script>

  </head>

 <script>

   angular.module('SelectBoxWithOptions', [])

.controller('Controller', ['$scope', '$http', function ($scope, $http) {

      $http.get('**CountriesCitiesList.json'**)

.success(function (data, status, headers, config) {

$scope.locations = data;

});

}]);

 </script>

  <body data-ng-app="SelectBoxWithOptions">

    <h1>Select box with options</h1>

    <div class="ng-controller" data-ng-controller="Controller">

    <label>Countries</label>

    <select data-ng-model="countries" data-ng-options="loc.country for loc in locations">

      <option value="">Please chose country</option>

    </select>

    <label>Cities</label>

    <select data-ng-model='city' data-ng-options='c for c in countries.cities'>

      <option value="">Please chose city</option>

    </select>

</div>

</body>

</html>

**CountriesCitiesList.json** is the next file used in $http.get('CountriesCitiesList.json') for display list.

[

  {

    "code": 1,

    "country": "Germany",

    "cities": [

      "Berlin",

      "Munich"

    ]

  },

  {

    "code": 2,

    "country": "Italy",

    "cities": [

      "Rome",

      "Venice"

    ]

  },

  {

    "code": 3,

    "country": "United Kingdom",

    "cities": [

      "London",

      "Manchester"

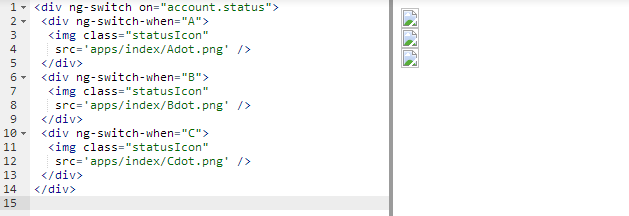
    ]

  }

]

How to display images based on the status(A, B, and C) in angular js?

Ans :

[](http://3.bp.blogspot.com/-LwtoXYHjbUw/U4o2zpOvCOI/AAAAAAAAAbY/A7S4DZ6PRgA/s1600/images+based+on+the+status.png)

<div ng-switch on="account.status">  
 <div ng-switch-when="A">  
  <img class="statusIcon"   src='apps/index/Adot.png' />  
 </div>  
 <div ng-switch-when="B">  
  <img class="statusIcon"  src='apps/index/Bdot.png' />  
 </div>  
 <div ng-switch-when="C">  
  <img class="statusIcon"  src='apps/index/Cdot.png' />  
 </div>  
</div>

## HTML5 mode

In HTML5 mode, the $location service getters and setters interact with the browser URL address through the HTML5 history API, which allows for use of regular URL path and search segments, instead of their hashbang equivalents. If the HTML5 History API is not supported by a browser, the $location service will fall back to using the hashbang URLs automatically. This frees you from having to worry about whether the browser displaying your app supports the history API or not; the $location service transparently uses the best available option.

* Opening a regular URL in a legacy browser -> redirects to a hashbang URL
* Opening hashbang URL in a modern browser -> rewrites to a regular URL

When you have html5Mode enabled, the # character will no longer be used in your urls. The #symbol is useful because it requires no server side configuration. Without #, the url looks much nicer, but it also requires server side rewrites.

When you use HTML5 history API mode, you will need different links in different browsers, but all you have to do is specify regular URL links, such as: <a href="/some?foo=bar">link</a>

When a user clicks on this link,

* In a legacy browser, the URL changes to /index.html#!/some?foo=bar
* In a modern browser, the URL changes to /some?foo=bar

In cases like the following, links are not rewritten; instead, the browser will perform a full page reload to the original link.

* Links that contain target element  
  Example: <a href="/ext/link?a=b" target="\_self">link</a>
* Absolute links that go to a different domain  
  Example: <a href="http://angularjs.org/">link</a>
* Links starting with '/' that lead to a different base path when base is defined  
  Example: <a href="/not-my-base/link">link</a>

## Using $location outside of the scope life-cycle

$location knows about Angular's [scope](https://docs.angularjs.org/api/ng/type/$rootScope.Scope) life-cycle. When a URL changes in the browser it updates the $location and calls $apply so that all [$watchers](https://docs.angularjs.org/api/ng/type/$rootScope.Scope#$watch) / [$observers](https://docs.angularjs.org/api/ng/type/$compile.directive.Attributes#$observe) are notified. When you change the $location inside the $digest phase everything is ok; $location will propagate this change into browser and will notify all the [$watchers](https://docs.angularjs.org/api/ng/type/$rootScope.Scope#$watch) / [$observers](https://docs.angularjs.org/api/ng/type/$compile.directive.Attributes#$observe). When you want to change the $location from outside Angular (for example, through a DOM Event or during testing) - you must call $apply to propagate the changes.

## $location.path() and ! or / prefixes

A path should always begin with forward slash (/); the $location.path() setter will add the forward slash if it is missing.

Note that the ! prefix in the hashbang mode is not part of $location.path(); it is actually hashPrefix.

## Issue: JavaScript errors don't throw within resolve functions

If you are having issues where a trivial error wasn't being caught because it was happening within the resolve function of a state, this is actually the intended behavior of promises per the [spec](http://wiki.commonjs.org/wiki/Promises/A).

If the callback throws an error, the returned promise will be moved to failed state.

The promise is rejected with the error. Additionally the [$stateChangeError](https://github.com/angular-ui/ui-router/wiki#state-change-events) event is triggered. It's very important to always set up your $stateChangeError handler early in the project to avoid headaches.

Diff b/w ng-view and ui-view

Ng-view => it is used to load template. Nested view not possible with ng-view

<nav>

<a href="#/tab1">Tab 1</a>

<a href="#/tab2">Tab 2</a>

<a href="#/tab3">Tab 3</a>

</nav>

<div class="tab">

<h3>Tab 1</h3>

The CSS transform property lets you modify the coordinate space of the CSS visual formatting model. Using it, elements can be translated, rotated, scaled, and skewed according to the values set.

</div>

angular

.module('MyApp', [

'ngRoute',

'ngAnimate',

'MyApp.controllers'

])

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

$routeProvider

.when("/tab1",

{

templateUrl: "tab1.html"

})

.when("/tab2",

{

templateUrl: "tab2.html"

})

.when("/tab3",

{

templateUrl: "tab3.html"

})

$routeProvider.otherwise({

redirectTo: '/tab1'

});

}]);

angular.module('MyApp.controllers', [])

.controller('MainController', function($rootScope, $window, $scope){

$scope.slide = 'slide-left';

$rootScope.back = function(){

$scope.slide = 'slide-right';

$window.history.back();

}

$rootScope.forward = function(){

$scope.slide = 'slide-left';

$window.history.forward();

}

})

Ui-view => it is used to load nested view in page.

<nav>

<a ui-sref="tab1">Tab 1</a>

<a ui-sref="tab2">Tab 2</a>

<a ui-sref="tab3">Tab 3</a>

</nav>

<div class="tab">

<h3>Tab 1</h3>

The CSS transform property lets you modify the coordinate space of the CSS visual formatting model. Using it, elements can be translated, rotated, scaled, and skewed according to the values set.

</div>

angular

.module('MyApp', [

'ui.router',

'ngAnimate',

'MyApp.controllers'

])

.config(function($urlRouterProvider, $stateProvider){

$stateProvider

.state('tab1',

{

url : "/tab1",

templateUrl: "tab1.html"

})

.state('tab2',

{

url : "/tab2",

templateUrl: "tab2.html"

})

.state('tab3',

{

url : "/tab3",

templateUrl: "tab3.html"

});

$urlRouterProvider.otherwise('/tab1');

});

angular.module('MyApp.controllers', [])

.controller('MainController', function($rootScope, $window, $scope){

$scope.name = "Don Coyote"

$scope.slide = '';

$rootScope.$on('$stateChangeStart', function(){

$scope.slide = $scope.slide || 'slide-left'

});

$rootScope.back = function(){

$scope.slide = 'slide-right';

$window.history.back();

}

$rootScope.forward = function(){

$scope.slide = 'slide-left';

$window.history.forward();

}

})