**AngularJS**

https://www.codementor.io/angularjs/tutorial/angularjs-interview-questions-sample-answers

http://career.guru99.com/top-25-angular-js-interview-questions/

https://www.toptal.com/angular-js/interview-questions

<http://stackoverflow.com/questions/tagged/angularjs>

<http://java.dzone.com/articles/angularjs-interview-questions>

**Performance** ---

http://www.binpress.com/tutorial/speeding-up-angular-js-with-simple-optimizations/135

**Angular life cycle ---**

<http://www.angular-buzz.com/angular-js-architecture-life-cycle/>

Good Angular Google Maps tutorial

<https://www.youtube.com/watch?v=6RdrGWWc_4s>

Youtube: webtunnings objected oriented javascript (youtube)

Different ways of structuring angularJs application?

https://medium.com/opinionated-angularjs/scalable-code-organization-in-angularjs-9f01b594bf06

Ans) 1) File Pattern 2) Folder Pattern 3) Module Pattern

2) What is $resource (angular-resource)?

Ans) A factory which creates a resource object that lets you interact with RESTful server-side data sources.

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

app.factory("Post", function($resource) {

return $resource("/api/posts/:id");

});

We can use this way:

app.controller("PostIndexCtrl", function($scope, Post) {

Post.query(function(data) {

$scope.posts = data;

});

});

3) What is the difference between factory and service?

4) How singleton can be acheived in Javascript(Using {} it can be acheived) ?

5) What is MVC? What is the difference between server side and client side mvc?

6) What are interceptors (http interceptors) in Angular?

**Ans)** <https://djds4rce.wordpress.com/2013/08/13/understanding-angular-http-interceptors/>

Angular JS  built in service  $http  is used to make http server requests.  More often than not you would find yourself in a situation where you would want to run hooks for the http calls, i.e execute some logic before or after the http call. For example appending the auth token  to every api request or generic http response error handling. For this $http interceptors become quite handy. One more very important use of interceptors is to log http requests made to external API’s which can be used for analytics.

**What are Pros and Cons of Angular?**

<http://www.tekritisoftware.com/angular-js>

AngularJS is a client side MVC framework for developing dynamic web apps and perfectly suits single page applications. It lets you use HTML as your template language and lets you extend HTML's syntax to express your application's components clearly and any DOM manipulations to be done should go to Directives.

PROS:

1) HTML Directives,

2) Builtin Dependency Injection,

3) 2-way Data Binding,

4) HTML Partials,

5) Built in REST communication using $http and $resource services

6) Lots of third party libraries to add additional features

CONS:

1) AngularJS requires everything to be loaded at the application start up!!!

There is no way of optimizing the script loading, and doing an on demand or lazy loading for parts of your application

2) Everything you do, have to be done the AngularJS way. AngularJS is very opinionated about everything (Any one who works with AngularJS,

may learn some good design patterns, but will not really improve his JavaScript skills.)

3) Two ways binding - digest cycles

  a) Memory – each object has two different copies in memory in order to perform a dirty check (the current one and the “dirty” one)

  b) Performance – on each digest cycle, a deep object comparison is made meaning equality checks over all of the data that the view depends on.

What is the latest  version of angular and what are the new things introduced in angular (1.2+)?

Diff Between Watch & watchCollection

Ans - <http://jsfiddle.net/luisperezphd/2zj9k872/>

Directive execution Order or Directive execution

<https://www.undefinednull.com/2014/07/07/practical-guide-to-prelink-postlink-and-controller-methods-of-angular-directives/>

**Angular 2**

1. **Project setup**
2. **Diff component, directives**
3. **OOPS**
4. **Routing**

**Innominds---**

Security Considerations while developing application?

Performance things that you need to keep in mind while developing web applications?

What are http interceptors?

What are polyfills in HTML5?

What is require.js and what is its purpose ?

RequireJS is a JavaScript file and module loader.

Using a modular script loader like RequireJS will improve the speed and quality of your code.

It is optimized for in-browser use, but it can be used in other JavaScript environments, like Rhino and Node.

http://localhost:9001/ITE/index.html#/dashboard

Access-Control-Allow-Origin

org.codehaus.mojo

**General Interview Preparation:**

11:10: {"result":[{"record":{"conversationid":"test\_conversationid","api":"test\_api","version":"test\_version","difference":"test\_difference"}}]}

1) How the application is structured in Angular?

https://medium.com/opinionated-angularjs/scalable-code-organization-in-angularjs-9f01b594bf06

Ans) 1) File Pattern 2) Folder Pattern 3) Module Pattern

We are using Yeoman to structure our angular js application (Sample Steps as below)

Install generator-angular:

npm install -g generator-angular

Make a new directory, and cd into it:

mkdir my-new-project && cd $\_

Run yo angular, optionally passing an app name:

yo angular [app-name]

Run grunt for building and grunt serve for preview

2) How did you distribute components in Angular?

For distribution, register the new project with Bower (a web library package manager), executing the following:

$ bower register [component-name] [component-github]

Now the component is available to the world via the bower package manager, and can be installed:

$ bower install [component-name]

**3) What is the difference between compile and link?**

Compiler is an angular service which traverses the DOM looking for attributes. The compilation process happens into two phases.

1. **Compile:** traverse the DOM and collect all of the directives. The result is a linking function.
2. **Link:** combine the directives with a scope and produce a live view. Any changes in the scope model are reflected in the view, and any user interactions with the view are reflected in the scope model. Making the scope model a single source of truth.

4) What are the different strategies you can communicate between controllers in Angular?

Ans) Usage of events on $rootScope and using Services/Factories by means of dependency injection

5) How would you structure your application using models?

6) What are the benefits in using controller Functions besides using scope?

Use can reduce digest() cycle

7) Can you explain the digest cycle and why do you need it?

$scope.$watch('aModel', function(newValue, oldValue) {

//update the DOM with newValue

});

The $digest cycle starts as a result of a call to $scope.$digest().

It’s the $digest cycle where the watchers are fired. When a watcher is fired, AngularJS evaluates the scope model,

and if it has changed then the corresponding listener function is called. So, our next question is when and how this $digest cycle starts.

$scope.$apply(), which in turn calls $rootScope.$digest(). As a result of this, a digest cycle starts at the $rootScope,

and subsequently visits all the child scopes calling the watchers along the way.

http://jsfiddle.net/luisperezphd/2zj9k872/

8) Have you used any function programming concepts? Example?

9) What is the use of function programming concepts vs OOP?

10) $broadCast vs $emit which is more performant and why ?

Refer: http://stackoverflow.com/questions/11252780/whats-the-correct-way-to-communicate-between-controllers-in-angularjs

Ans) I highly advise not to use $rootScope.$broadcast + $scope.$on but rather $rootScope.$emit+ $rootScope.$on.

The former can cause serious performance problems as raised by @numan. That is because the event will bubble down through all scopes.

However, the latter (using $rootScope.$emit + $rootScope.$on) does not suffer from this and can therefore be used as a fast

communication channel!

From the angular documentation of $emit:

Dispatches an event name upwards through the scope hierarchy notifying the registered

Since there is no scope above $rootScope, there is no bubbling happening. It is totally safe to use $rootScope.$emit()/ $rootScope.$on()

as an EventBus.

However, there is one gotcha when using it from within Controllers. If you directly bind to $rootScope.$on()

from within a controller, you'll have to clean up the binding yourself when your local $scope gets destroyed. This is because controllers (in contrast to services) can get instantiated multiple times over the lifetime of an application which would result into bindings summing up eventually creating memory leaks all over the place :)

General UI Developer Question refer to link below:

<http://rileyh.com/ui-developer-interview-questions-answers/>

what is the use of promise ?

Promises in AngularJS are provided view the built-in $q service. They provide a way to execute asynchronous functions in series by registering them with a promise object.

Promise.then().Success or failure

Diff between Synchronous & asynchronous

Difference between value and constant in AngularJS ?

How to preload images into the application?

Good example of using transclude and transclude function to modify the transcluded value?

<http://jsfiddle.net/rp1nt21m/1/>

<http://stackoverflow.com/questions/11703086/how-can-i-transclude-into-an-attribute>

**AngularJS Questions Reference and Links**

**Speeding up AngularJS apps with simple optimizations**

<http://www.binpress.com/tutorial/speeding-up-angular-js-with-simple-optimizations/135>

**What is the difference between the $parse, $interpolate and $compile services?**

<http://stackoverflow.com/questions/17900588/what-is-the-difference-between-the-parse-interpolate-and-compile-services>

* **$compile** - it can take the whole markup and turn it into a linking function that, when executed against a certain scope will turn a piece of HTML text into dynamic, live DOM with all the directives (here: ng-src) reacting to model changes. One would invoke it as follows: $compile(imgHtml)($scope) and would get a DOM element with all the DOM event bounds as a result. $compile is making use of $interpolate (among other things) to do its job.
* **$interpolate** knows how to process a string with embedded interpolation expressions, ex.: /path/{{name}}.{{extension}}. In other words it can take a string with interpolation expressions, a scope and turn it into the resulting text. One can think of the $interpolationservice as a very simple, string-based template language. Given the above example one would use this service like: $interpolate("/path/{{name}}.{{extension}}")($scope) to get the path/image.jpg string as a result.
* **$parse** is used by $interpolate to evaluate individual expressions (name, extension) against a scope. It can be used to both *read* and *set* values for a given expression. For example, to evaluate the name expression one would do: $parse('name')($scope) to get the "image" value. To set the value one would do: $parse('name').assign($scope, 'image2')

**onLoad vs onDOMContentLoaded**

<http://javascript.info/tutorial/onload-ondomcontentloaded>

What is angular-translate module in Angular ?

## Multi-language support

http://www.ng-newsletter.com/posts/angular-translate.html

$translateProvider is used to teach your app different languages and e.g. tell your app which language to use etc.

Teaching your app a language with $translateProvider is pretty easy. Just inject it in your module configuration and make use of the translations() method.

angular.module('myApp', ['pascalprecht.translate'], ['$translateProvider', function ($translateProvider) {

// register translation table

$translateProvider.translations({

'HEADLINE\_TEXT':'Hey Guys, this is a headline!',

'SOME\_TEXT': 'A text anywhere in the app.'

});

}]);

You can register a translation table just by passing an object hash where a key represents a translation id and a value the concrete translation. In the example above there are two registered translations, HEADLINE\_TEXT and SOME\_TEXT.

angular.module('myApp').controller('Ctrl', ['$translate', '$scope', function ($translate, $scope) {

$scope.translatedText = $translate('HEADLINE\_TEXT');

}]);

How to create an IE Only stylesheet?

<https://css-tricks.com/how-to-create-an-ie-only-stylesheet/>

List of IE bugs?

<http://www.positioniseverything.net/explorer.html>

<https://css-tricks.com/ie-css-bugs-thatll-get-you-every-time/>

**Pega Systems**

Gajanan :

In which language cordova developed?

What is AngularJS & Ionic Framework What are difference in this 2 framework.

What is difference between Web Developer & Web Application

Who Developed Ionic Framework

1. What is $scope in angular.js?
2. How to create custom filter in Angular.js?
3. How to create Custom Directive in Angular.js?
4. What is scope isolation in Custom Directive of Angular.js?
5. What is difference between $emit and $broadcast in Angular.js?
6. How to pass event between controller to controller in angular.js?
7. What is $link and $compile in Angular.js?
8. What is routing in Angular.js?
9. What is difference between factory,service and provider in angular.js?
10. What is the differ in Angular.js , How Achive this?
11. What is $ngResource in Angular.js
12. What does the ng-transclude directive in case of custom directive in Angular.js

# [**$evalAsync vs $timeout**](http://stackoverflow.com/questions/17301572/angularjs-evalasync-vs-timeout)

https://docs.angularjs.org/api/ng/type/$rootScope.Scope

|  |  |
| --- | --- |
|  | * if code is queued using **$evalAsync from a directive**, it should run after the DOM has been manipulated by Angular, but before the browser renders * if code is queued using **$evalAsync from a controller**, it should run before the DOM has been manipulated by Angular (and before the browser renders) -- rarely do you want this * if code is queued using **$timeout**, it should run after the DOM has been manipulated by Angular, and after the browser renders (which may cause flicker in some cases) |

* **$timeout(callback)** will wait for the current digest cycle to be done (i.e. angular update all model and the DOM), then it will execute its callback - potentially affecting angular model - then launch a full $apply on the root $scope, and redigest everything.
* **$evalAsync(callback)**, on the other hand, will add the callback to the current, or next, digest cycle. Which means if you are within a digest cycle (for instance in a function called from some ng-click directive), this will not wait for anything, the code will be executed right away. If you are within an asynchronous call, for instance a setTimeout, a new digest cycle ($apply) will be triggered.