**Angular js**

Angular js is a java script framework to build an application that run on web browser using html

Two basic requirement to add angular to web page is

1 -Script tag pointing to angular.js, angular has no dependency on other libraries.

2-add ng –app some where in your application , it’s a special directive that angular looks for in html

And if it is find angular will bootstrap itself , so it initializes it self and jumps into action for managing the page .

There should be only one ng-app directive per page

Examples are in Angular application , script.js file

**Benefits of Angular js**

1-Dependency Injection

2-TwoWay Data Binding

3-Testing

4-Model View controller ()

5- Directives (Controlling the behavior of DOM element ), filters

**JavaScript patterns**

1-function as abstraction

Call function from another function basically,

Basically some routine add logic or some routine to add value before executing the main function.

2- function as build object

Some times we need object that as data and message attach to it . in Java script we define a model by creating a java script function. it may has some private implementation details from which other are not aware of implementation they can access by creating the object.

3- function to avoid a global variable,

Global variable is evil in JavaScript. java script is a dynamic type language , global variable can be override by some one else as they are accessible to any where in files

**Module**

A module is a container for different part your application i.e controller,services ,directive filters etc

You can think of module as a main() method in other type of application.

To create module we use angular object angula.module(“”,[])

[] array in module specify the dependency , empty array means it has node dependency .

Angular supports modula approach .

**Controller**

Controller is a central peace of angular framework , controller is an encharge of many area of application, controller is responsible for building a model for the view to display , a model contains data that we need to work with controller to what ever needed to grap that data ,

For basics facts of controller

The job of controller is to control the information that want to put on page or in editing scenario to save information that user is typing on page

1-controller directive in html

Ng-controller is an attribute that we place on page , ng- controller is directive same as ng-app

When we use ng-controller in particular div it’s a incharge of that div or tag ,

2-bqscically controllers are function that angular invoke

3-controller takes $scope parameter.

$ sing prefix is a sign to indicate that component you r working with is component provide by angular

Scope is bascically going to be a model

4- Attach model to scope .

**Controller capabilities**

1-multiple controller

2-complex object

3-Nest Controllers

**Two Way Binding**

Keeps the model and view in sync at all time, that is change in model updates the view and change in view updates the model.

Binding expression updates the view when model changes

Ng-model directive updates the model when view changes

**Filters**

Filter can do three different thing

Format , Sort and filters data

Filters can be used with regular expression or a directive

To apply filters use(|) characters

{{expression | filtersname:parameter}}

Bulid in filters

Lower, upper,

number(format the number as text , include comma as thousand separator and numbe of decimal place can be specified.),

currency(format the number as currency ,$ is default)

date (formate date to a stringbased on request format)

limitTofilters :can be used o limit the number of rows or character in string.

{{expression|limitTO : limit : begin }}

**Sorting Data In Angular**

To sort data in angular we use

orderby filter

{{ordeby\_expression |orderby :expression:reverse}}

Example:ng-repeat”employee in employees | order:salary: false”

**Custom Filters**

Is a function that returns a function

Use to filter function to create a custom filter.

**Ng-init**

The ng-init directive allows you to evaluate an expression in the current scope.

In the real world application you should use a controller instead of ng-init to initialize value on a scope.

Ng- init should only be used for aliasing special properties of ng-repeat directive.

**Services**

Basically we need to fetch data from web server and were server communicate with sql database or any data source To send data we need

So above can be achieve by $http service , basically it’s a object , object with method that we can use make http

1-encapsulate HTTP communication methods like get, post put, delete

2-can ask for $http inside a controller

Just add as another parameter to controller function

$scope.user =$http.get(“/user/1234”)

Will nt assign a data to scope immediately bcz communication is synchronous what it returns object immediately is promise . promise promise you to return some result in future may be even error messag

So we use promise.then , then will call function when data is ready

$http.get(“https/github.com/users/robconery”).then(onusercomplete)

The method in .then method only onvoked when .get is successful response.

If the server is down or error we will call another call back function to show error .

$http.get(“https/github.com/users/robconery”).then(onusercomplete,onerror)

Built in service bacically

$http,$log

$http service is used to make http request to emote server .basically for ajax call

$http service is a function that has a single input parameter i.e configuration object

$http({

Method:’Get’

[Url:”service](Url:) the need to be called that basccally in json format data or we convert that in json”

})

For $log service : $log.info(response ) that will show information in console of web browser.

What is service in angular?/

Service in angular simply an object that provide some sort of service or function that can be reused in angular application. Angular service object has property and method like other java script object

Why we need service ???

Basically to maintain single responsibility principle for controller ,service encapsulate reuable logic that does not belong to any where else (ie Directive,Filters,View , Model and controller )

Benefits of Using services ????

1 reusability (contain logic that you want to use in entire application )

If you want ajax call you can use http service I, simply by injecting in obect that need it

2 dependency injection -Another benefit of service is that they can simply injected into controllers or other services thet need them

3 Testability

GitHubApi

Available from javascrtpt in a browser.

Return Json(easy to convert into objects)

No authentication and key required.

**Custom service**

We use a factory method to create a custom service and register that with angular

App.factory(“Name of service ”,anonymous function ) --(usually service in angular is state less )

And anonymous function returns java script object , which contain one function as use practice in example . when we use return it return object .

Now inject the service you created in any controller that you need

**AnchorScroll Service**

$anchorscroll service is used to jump to a specified element on a page .

We need to give the id where we need to jump.

We use location service . $location.hash(“we pass is where we need to scroll”) and location service has hash method which append hat ever you provide to that to URl.

Then we call anchorscroll() method below $location.hash() anchor scroll read the hash fragment in the url and automatically scroll to that element on page

$anchorscroll service has property yOffset which will leave that much space between the element and browse .

**Routing**

To use routing feature we need to use routing module that is present in separate java script file.

**LAyoutTemple**

<a href="#/home">Home</a>

# Is to tel the browser that we don’t want to navigate from index.html the startup page instead we want partial template to inject in location in index.cshtml

In the div having <ng-view > directive

**RouteConfiguration**

All routing feature present in a diffent module and that module is ngRoute so we pass that module in an array as dependency in angular.module

.cofig(function is use to specify the configurartion for application) in this we inject $routeProvider

$routeProvider has .when(/home) function which inject tempelate in ng-view DERICTIVE div element or any dom element where ng- view will present

We can have N number of .when function depending upon the route

You can remove the # from URl by Using $routeProvider.html5Mode(true).

**PageReload Issue**

When we reload the page basically we lost the stylesheet of that web page. Getting exception unexpected token . to fix this issue what we need to do is move <base href=”/”/> at top in head section.

**Controller as Syntax**

Till we are using $scope object as mechanism to make member of controller available in the view there is another way to achieve this is controller as syntax we need to inject $scope in controller function. At place of $scope.message we use this.message

In view we use controller name as ng-controller =“maincontroller as mainctrl” mainctrl is reference pointing to instance of maincontroller. In expression {{ mainctrl .message}}

We can not use .this with .then of http.get () bcz .this pont to window object not to the instance of controller to handle this we use “var vm=this” and that vm is attach with property of controller

Controller as syntax made our code more readable as compared to scope. Particularly when we deal with nested controller.

Controller as syntax is new technique and released in 1.2.0 $scope is old technique and is available since initial version

It still using $scope behind the seen , alias name of controller is reference of controller pointing to instance. angular taking that reference and attaching that reference to $scope

**casInsensitiveMatch**

route are by default case censitive to make it insensitive we use casInsensitiveMatch=true; inside the .when when using .config.

for whole route to make case insencitive we use $routrProvider. casInsensitiveMatch=true;

for inline tempelate we write tempelate : <h1>Inline tempelate </h1>property instead of tempelateUrl:

**Route Reload Method**

Angular route service reload() method is useful when you want to reload the current route instead of entire application. We need to inject $route service within a controller where we want to use reload method.

Var reloaddata= function()

{route.reload(); }

**$Scope vs $rootScope**

$scope is available to particular controller in which it is injected but $rootScope is available to each controller independent of controller in which it is injected .

**Cancel Route Change**

Inject $scope in controller and call .$on() method $scope.on(); when route occurs in angular an event occur that even is $routeChangeStart we handle that event using $on($routeChangeStart, function(event,next, current )) here event is event object that contain event info,next is route we want to move , current is current route

To cancel the route change we use event object and we call event.preventDefault() to cancel the route change

**Learning while coding**

**Angular factory method is used to creates a service**

**employeeForm=employee-form(snake case)**

**CSS Class for decorating forms**

**Container 🡺adds a margin and centers the page within the browser. You can directly place in body element**

**Form-group🡺 used within a form to group together a label and the input fields**

**Form –controls 🡺placed on input fields**

**Form-horizontal🡺place the label left of controls , this is responsible to make everything appear on rows**

**Control-label🡺group a label and input fields**

**Col-sm-3, col-sm-9🡺use 12 column grid**

**<fieldset> tag🡺** Group related elements in a form

**<legend>** The HTML <legend> tag s used to define a caption for <fieldset> tag.

**Data Binding**

**$scope🡺data is stored in the $scope variable in controller , what ever attach to the $scope will have access to that data in markup and the way to access that data in a form control is with ng-model attribute**

**ng-model 🡺 placed on a form control for two way binding**

**ng-bind or {{}} are for one way binding**

**ng-submit 🡺 placed on the form tag itself, and it specifies the callback function usually on scope object that is called when button is clicked**

**ng-click 🡺 placed on submit button**

**using both on same page will submit data twice**

**$scope.fn 🡺 attach cal back function directly on your scope object**

**Routing**

**$routeprovider is used for routing**

**$location.path()**

Whatever the changes are made in the URL in the address bar are reflected into $ location service and Changes made by $location are reflected into the browser address bar.

// get the current path

$location.path();

// change the path

$location.path('/newValue')

**Angular.copy()🡺 creates a deep copy of angular object that helps in when you are working with editable changes**

**After submitting of we want to navigate back to previous page we use window service**

**Syntax🡺$window.history.back();**

**$routeParams🡺 hold the ID from the query string when particular record requested**

**$modal🡺 it is used to show template in a popup (modal form ‘ui.bootstrap’ is used as a dependencies in a root modal)**

**$modal.open() open method returns modal instance ie result**

**To show template with rich UI we will use class modal-header, modal-body, modal-footer in a template which we will show in popup**

**When tempelate is shown in popup we can not use $window service to go back , we use $modalInstance.dismiss() method to go back or $modalInstance.close()**

**Difference between $modalInstance.dismiss() and $modalInstance.close()**

**Dismiss is similar as cancel, close is similar to , okay lets do our processing , we don’t have any**

**Some important angularjs-Ui bootstrap directive**

**1-Datepicker <**datepicker>**</**datepicker>

**2- timepicker**

**3-rating directive**

**4-collapse and accordion directive (**collapse="!editableEmployee.isUnderNonCompete")

**5-popover and tooltip**

**Validation at Angular**

**While validating form fields , first form should have a name so that it can be referenced some where.**

**novalidate attribute is used at form level to insure that browser itself not validate the form**

**Require is html5 attribute and ng-require is angular attribute, require attribute does not take any value**

**Angular JS CSS class**

**Ng-pristine 🡺 will be placed on a fields that has been touched yet  
ex- in css class use property like input.ng-pristine**

**Ng-dirty🡺 as soon as user enter value or modify a fields the field become dirty and ng-dirty is assigned to the fields**

**Ng-valid and ng-invalid placed on a control based on a validation setting**

**Class=”help-block” is a bootstrap class that help you to style the text that associated with an input controls**

**Custom Directive**

**When we create custom directive there is attribute require in which we pass** require: '^form', this is a form controller

FormController keeps track of all its controls and nested forms as well as the state of them, such as being valid/invalid or dirty/pristine.

**link: function (scope, el, attrs, formCtrl) {**

**link function is called when directive is used**