Aprameyah
Technologies Pvt. Ltd.

Angular JS

Aprameyah
Technologies Pvt. Ltd.

About Me

- Maruthi R Janardhan
  - Been doing java since jdk 1.2
  - Been with IBM, ANZ, HCL-HP, my own startup Leviossa..
  - Total 16 years programming C, C++, Java, Javascript, Ruby, Perl, Python, PHP, etc

Aprameyah
Technologies Pvt. Ltd.

What Is It

- Its an open-source javascript web application framework maintained by Google and by a community of individual developers and corporations to address many of the challenges encountered in developing single-page applications.

- Wikipedia

Aprameyah
Technologies Pvt. Ltd.

MVC in Angular

- Models - Javascript Object
- Controller - Javascript Functions
- View - html + angular's template language
    - Views are declarative and not imperative unlike in many javascript frameworks

Aprameyah
Technologies Pvt. Ltd.

Setup Angular

```
<!DOCTYPE html>
<html lang="en">
<head>
    <script src="scripts/angular.js"></script>
</head>
<body ng-app="myapp">
<div ng-controller="HelloController" >
    <h2>{{helloTo.title}} !</h2>
</div>
<script>
angular.module("myapp", [])
    .controller("HelloController", function($scope) {
        $scope.helloTo = {};
        $scope.helloTo.title = "Hello World, AngularJS";
    } );
</script>
</body>
</html>
```

View

Model

Controller Function

Execution Flow

- HTML loads into browser and so does angular.js
- Angular scans the document views and apps to hook them up
- Controller functions are executed and view rendered with expression evaluation
- Event handlers are registered as part of parsing of some of the directives

Aprameyah
Technologies Pvt. Ltd.

Angular Vs Jquery

- DOM Manipulation is decoupled from application logic

- Angular is more heavily targeted at SPAs unlike complex DOM Manipulation thats needed to achieve the same in JQuery

- UI design is declaratively connected to the behaviour code

Aprameyah
Technologies Pvt. Ltd.

Expressions

- {{4+10}}, {{'Hello'+'World'}}, {{user.name}}, {{items.index}}

- Angular expressions do not have access to global variables like window, document or location

  - Instead use services like $window and $location in functions called from expressions. Such services provide mockable access to globals.

Aprameyah
Technologies Pvt. Ltd.

Directives

- A Directive is a marker on a HTML tag that tells Angular to run some Javascript code when the element is loaded

- Example ng-repeat iterates over a list or keys of an object and repeats the content of the element

  - this line is repeated

Aprameyah
Technologies Pvt. Ltd.

ng-show/ng-hide

- ng-show and ng-hide directives enable and disable visibility of fields. Expression is a boolean value

```
<div ng-show=“loading">I am visible when $scope.loading=true</
img></div>
```

Aprameyah
Technologies Pvt. Ltd.

Use ng-repeat (1)

- Create a UserController thats going to create a hardcoded list of user objects (with name and age fields) in $scope

```
$scope.users=[];
$scope.users[0] = {name:"Krishna",age:8};
$scope.users[1] = {name:"Hari",age:28};
```

- Use ng-repeat to render this list

Aprameyah
Technologies Pvt. Ltd.

Solution Html

```
<div ng-controller="UserController">
        <table style="width: 400px">
            <tr ng-repeat="user in users">
                <td>{{user.name}}</td>
                <td>{{user.age}}</td>
            </tr>
        </table>
        Showing {{users.length}} users
</div>
```

- Download myotherdb.zip from github and unzip

- Change hibernate.cfg.xml to refer to this new directory instead of mydb and restart tomcat

- Test the url to get some data: http://localhost:8080/AngularDynWeb/rest/user

Aprameyah
Technologies Pvt. Ltd.

Angular Services

- Angular has many built in services to perform basic tasks
    - $http - facilitates communication with the remote HTTP servers via the browser's XMLHttpRequest object or via JSONP.

```
var promise = $http.get('/someUrl');
promise.success(function(data, status, headers, config) {
  // this callback will be called asynchronously
  // when the response is available
}).
error(function(data, status, headers, config) {
  // called asynchronously if an error occurs
  // or server returns response with an error status.
});
```

    - $log - Simple service for logging. provides different level of logging methods $log.warn(), $log.debug(), $log.error()

Aprameyah
Technologies Pvt. Ltd.

Dependency Injection

- Services needed by the controller are dependency injected

```
app.controller('UserController',function($http, $log, $scope){})
```

Aprameyah
Technologies Pvt. Ltd.

use ng-show and $http(1)

- Use $http service to load the list of users from rest/user url.

- Use ng-show to render the images/loading.gif till the data is loaded. Once data is loaded hide it. For this setup a boolean field in scope in the controller

Aprameyah Technologies Pvt. Ltd.

Solution

```
<div ng-controller="UserController">
        <div ng-show="loading"><img src="images/loading.gif"></img></div>
        <table style="width: 400px">
            <tr ng-repeat="user in users">
                <td>{{user.name}}</td>
                <td>{{user.age}}</td>
            </tr>
        </table>
        {{error}}
        Showing {{users.length}} users
</div>

var app = angular.module('Airlines',[]);
app.controller('UserController',function($http, $log, $scope){
    $scope.users=[];
    $scope.loading = true;
    $log.debug("Getting users...");
    $http.get('rest/user').
      success(function(data, status, headers, config) {
              $scope.users = data;
              $scope.loading = false;
      }).
      error(function(data, status, headers, config) {
          $scope.loading = false;
          $scope.error = status;
      });
});
```

Scope Hierarchy

- There is a root scope, and the root scope has one or more child scopes

- Each view has its own $scope (which is a child of the root scope), so whatever variables one view controller sets on its $scope variable, those variables are invisible to other controllers.

```
<div ng-controller="myController1">
   {{data.theVar}}
</div>
<div ng-controller="myController2">
    {{data.theVar}}
</div>
```

Aprameyah
Technologies Pvt. Ltd.

Event Listeners

- DOM events are attached using these event directives
- ng-click
- ng-dbl-click
- ng-mousedown
- ng-mouseup
- ng-mouseenter
- ng-mouseleave
- ng-mousemove
- ng-mouseover
- ng-keydown
- ng-keyup
- ng-keypress
- ng-change

Aprameyah
Technologies Pvt. Ltd.

Using Event Directives

- we can evaluate expressions or call scope defined functions in response to DOM events

- ng-click="someboolean=true"
- ng-click="somefunc()"
  - in controller: $scope.somefunc = function(){}

Aprameyah
Technologies Pvt. Ltd.

Forms

- Form fields can be bound to model fields using ng-model.

```
<form name="userForm">
    <input type="text" ng-model="user.name"></input>
    <input type="number" ng-model="user.age"></input>
</form>
```

- Two way data binding happens between form fields and scope object. The model is updated as we type in the form fields

Aprameyah
Technologies Pvt. Ltd.

use ng-click and forms (2)

- Create a button “Add User” and in ng-click set a boolean value: showForm=true
- Create a form for adding users in the same html below the table and put this in a div. Use ng-show to only show this form if the showForm field is true
- Create a button in form. use ng-click to call a function addUser(user)
- Define the addUser function to add the user using $http.post to url “rest/user”

```
$http.post("rest/user",user)
```

    - After successfully adding the user, add that user to the list used by userlist above the form and also hide the form used for adding
- Provide a cancel button to hide the form without adding

```
app.controller('UserController',function($http,
$log, $scope){
       var controller = this;
       $scope.users=[];
       $scope.loading = true;
       $log.debug("Getting users...");
       $http.get('rest/user')....


    $scope.addUser = function(user){
        $http.post("rest/user",user)
        .success(function(data){
            $log.debug(data);
            $scope.users.push(user);
        });
    }
});
```

```
<div ng-controller="UserController">
        <!-- User listing part goes here -->
<input type="button" value="Add User" ng-
click="showAddForm=true"></input>
<div ng-show="showAddForm">
<form novalidate>
<input type="text" ng-model="user.name"></
input>
<input type="number" ng-model="user.age"></
input>
<input type="button" value="Cancel" ng-
click="showAddForm=false"></input>
<input type="button" value="Save" ng-
click="addUser(user)"></input>
</form>
</div>
</div>
```

Aprameyah
Technologies Pvt. Ltd.

Issues With Post

- $http.post() is sending json data to the server while the server is expecting form url encoded data.

- For now, lets replace $http.post with jquery post method $.post() which sends form url encoded data by default

```
$.post("rest/user",user,function(data){
    $log.debug(data);
    $scope.users.push(user);
});
```

Aprameyah
Technologies Pvt. Ltd.

Data Binding on Changes

- When external callbacks modify angular $scope data, binding does not update.
- Data binding in angular works based on three fundamental functions:
    - $watch()
        - The $scope.watch() function creates a watch of some variable.
    - $digest()
        - iterates through all the watches in the $scope and calls the value function for each watch. If the value has changed the listener function for that watch is called.
    - $apply()
        - The $scope.$apply() function takes a function as parameter which is executed, and after that $scope.$digest() is called internally.

edit user(3)

- Add a link to each row of display and set an ng-click event handler function that will accept the user of that row, set it in $scope.user and set the form to be visible
    - Note: Form hidden fields are not needed since it refers to loaded user model
    - Notice how the data in the list is getting updated as we type
- Provide a new form button “update” that will call a different update routine that does $http.put with the data

Aprameyah
Technologies Pvt. Ltd.

Filters

- Angular supports a new syntax of transforming data via filters.
  - {{ 12345.67 | currency }}
  - {{ "12345.67" | currency:"USD" }}
  - {{ 1288323623006 | date }}
- Standard filters: https://docs.angularjs.org/api/ng/filter

Aprameyah
Technologies Pvt. Ltd.

Custom Filters

- We can write our own filters too. (Ideally this should have been called a map function though)

- Filters can be written and registered on the app. Later referenced in expressions

```
app.filter('checkmark', function() {
        return function(input) {
          return input ? '\u2713' : '\u2718';
        };
   });

{{data | checkmark}}
```

Aprameyah
Technologies Pvt. Ltd.

use filters (4)

- Use date filters on another column to print user.joinDate formatted as dd/mm/yyyy

- Add date field to user form. Enter date yyyy-MM-dd format

- Write a filter that returns a color based on user age and use it on the row that displays the user

```
<tr ng-repeat="user in users" style="background-color: {{user |
colorCodeAge}}">
```

```
var app = angular.module('Airlines',[]);
    app.filter('colorCodeAge', function() {
        return function(user) {
            if(user!==undefined && user.age!==undefined)
                if(user.age<18){
                    return '#E6E6E6';
                }else{
                    return '#CCD6F5';
                }
            return '#CCD6F5';
        };
    });
```

Aprameyah
Technologies Pvt. Ltd.

Implement delete(5)

- We need a new link for delete
- A new controller function that calls $http.delete
- On successful deletion the object should be removed from the collection in $scope (Hint: use splice() function to remove item from array)

List Filters

- Angular provides built in list filters that match the search string to all properties of the object in list: ng-repeat="friend in friends | filter:searchText"

- Custom filters that act as list filters can be implemented with angular: ng-repeat="friend in friends | nameFilter:searchName

```
app.filter('nameFilter', function($log) {
   return function(friends, searchName) {
      //return a list thats filtered based on params passed
   };
});
```

Aprameyah
Technologies Pvt. Ltd.

Implement Custom Filter (6)

- Implement a text box on top of the users list that is bound to a model field
- Filter the user list based on that model field using a custom implemented filter
- Change it to use built-in filter

- ng-repeat="u in users | userNameFilter:searchName

```
app.filter('userNameFilter', function($log) {
    return function(users, searchName) {
        $log.debug(users);
        $log.debug("Filtering for :"+searchName);
        if(users!==undefined && searchName!==undefined && searchName!==""){
            var tempUsers = [];
            for(index in users){
                if(users[index]!==undefined && users[index].name.indexOf(searchName)!=-1){
                  tempUsers.push(users[index]);
                }
            }
            return tempUsers;
        }
        return users;
    };
});
```

Aprameyah
Technologies Pvt. Ltd.

Includes

- ng-include directive fetches, compiles and includes an external HTML fragment.

-

- If the expression is a string literal, then use single quotes around it

Aprameyah Technologies Pvt. Ltd.

Directives

- We can create our own directives
  - Element directives

```
myapp.directive('message', function() {
        var directive = {};
        directive.restrict = 'E';
        directive.template = “<div>{{textToInsert}}</div>";
        return directive;
});

<message></message>
```

  - Attribute directives

```
myapp.directive('message', function() {
        var directive = {};
        directive.restrict = 'A';
        directive.template = “<div>{{textToInsert}}</div>";
        return directive;
});

<div message></div>
```

Aprameyah
Technologies Pvt. Ltd.

Controllers For Directives

- Directives can have their own controllers

```
app.directive('directiveName', function(){
      return {
         //Configuration object for directive definition
         restrict: 'E',
         templateUrl: 'directive-template.html',
         controller: function(){

         },
         controllerAs: ctrl
      }
});
```

Isolating $scope for directive

The directive.scope.user property is set to "=user". That means, that the directive.scope.user property is bound to the property in the scope property (not in the isolate scope) with the name passed to the user attribute of the <userinfo> element. Used like this: <userinfo user="user1"></userinfo>

```
myapp.directive('userinfo', function() {
    var directive = {};
    directive.restrict = 'E';
    directive.template = "User : {{dispUser.firstName}}
{{dispUser.lastName}}";
    directive.scope = {
        dispUser : "=user"
    }
    return directive;
})
```

Aprameyah
Technologies Pvt. Ltd.

Implement Product Listing(7)

- Use product service at rest/products url to implement a products view in this layout (There are images in the images/products/ folder based on ids):

```
<div style="float: left;"><img src="image" width="100px"></img></div>
<div style="height: 100px">
<h2>Samsung Galaxy</h2>
$300.0
Stock: 3
</div>
```

- Move this into an include first

- Then move this into a directive so that it can render any product like this: <product product="prod"></product>

```
<div ng-controller="ProductController as productController">
        <div ng-show="loadingProducts"><img src="images/loading.gif"></img></div>

        <shopping-cart>
            <div ng-repeat="prod in products">
                <product product="prod"></product>
            </div>
        </shopping-cart>
    </div>

app.directive('shoppingCart', function() {
        var directive = {};
        directive.restrict = 'E';
        directive.templateUrl = "templates/cart.html";
        directive.transclude = true;
        directive.scope = {
                prods : "=prods"
        }
        return directive;
    });
```