

# Report lab 10

## Ex 23

```
<html ng-app="countryApp">
  <head>
    <meta charset="utf-8">
    <title>Angular.js Example</title>
    <script
src="//cdnjs.cloudflare.com/ajax/libs/angular.js/1.2.1/angular.min.js"></script>
    <script>
      var countryApp = angular.module('countryApp', []);
      countryApp.controller('CountryCtrl', ['$scope', '$http', function (scope,
http){
        http.get('countries.json').success(function(data) {
          scope.countries = data;
        });
      }]);
    </script>
  </head>
  <body ng-controller="CountryCtrl">
    Search:<input ng-model="query" type="text"/>
    <table>
      <tr>
        <th>Country</th>
        <th>Population</th>
      </tr>
      <tr ng-repeat="country in countries | filter:query | orderBy:'population'">
        <td>{{country.name}}</td>
        <td>{{country.population}}</td>
      </tr>
    </table>
  </body>
</html>
```

The ng-app directive defines the name of the AngularJS application, in this case, it's "countryApp".

The ng-controller directive defines the controller that will handle the logic for the HTML section below it.

The ng-model directive creates a two-way data binding between the input field and the query variable in the controller.

The ng-repeat directive loops through each country object in the countries array, and creates a table row for each one.

The filter and orderBy filters are used to filter and sort the data based on the query variable and the population property of each country object.

The {{ }} expression is used to display the values of the name and population properties of each country object.

The JavaScript code defines the countryApp module and the CountryCtrl controller. The http.get() method is used to retrieve the data from a countries.json file, and the retrieved data is stored in the \$scope.countries variable, which is then used by the HTML section.

## Ex 36

```
<html ng-app="countryApp">
  <head>
    <meta charset="utf-8">
    <title>Angular.js Example</title>
    <script
src="//cdnjs.cloudflare.com/ajax/libs/angular.js/1.2.10/angular.min.js"></script>
    <script src="//cdnjs.cloudflare.com/ajax/libs/angular.js/1.2.10/angular-
route.min.js"></script>
    <script>
      var countryApp = angular.module('countryApp', ['ngRoute']);

      countryApp.config(function($routeProvider) {
        $routeProvider.
          when('/', {
            templateUrl: 'country-list.html',
            controller: 'CountryListCtrl'
          }).
          when('/:countryName', {
            templateUrl: 'country-detail.html',
            controller: 'CountryDetailCtrl'
          }).
          otherwise({
            redirectTo: '/'
          });
      });

      countryApp.controller('CountryListCtrl', function ($scope, $http){
```

```

    $http.get('countries.json').success(function(data) {
        $scope.countries = data;
    });
});

countryApp.controller('CountryDetailCtrl', function ($scope, $routeParams){
    $scope.name = $routeParams.countryName;
});
</script>
</head>
<body>
    <div ng-view></div>
</body>
</html>

```

The HTML file contains a reference to the AngularJS library and the ngRoute module. The ng-app directive is used to define the root element of the application.

The AngularJS module "countryApp" is defined, and the ngRoute module is injected as a dependency.

The \$routeProvider service is used to configure the routing of the application. There are two routes defined:

The first route ('/') maps to the 'CountryListCtrl' controller and loads the 'country-list.html' template.

The second route('/:countryName') maps to the 'CountryDetailCtrl' controller and loads the 'country-detail.html' template. The ':countryName' part of the URL is a route parameter that is used to display information about a specific country.

The 'CountryListCtrl' controller is defined and uses the \$http service to load data from a 'countries.json' file. The retrieved data is then bound to the \$scope object using the 'ng-repeat' directive in the 'country-list.html' template, which displays a list of countries.

The 'CountryDetailCtrl' controller is defined and uses the \$routeParams service to retrieve the 'countryName' parameter from the URL. This parameter is then bound to the \$scope object, and displayed in the 'country-detail.html' template.

Finally, the 'ng-view' directive is used to display the appropriate template based on the current route.

## Ex 40

```
<html ng-app="countryApp">
  <head>
    <meta charset="utf-8">
    <title>Angular.js Example</title>
    <script
src="//cdnjs.cloudflare.com/ajax/libs/angular.js/1.2.10/angular.min.js"></script>
    <script src="//cdnjs.cloudflare.com/ajax/libs/angular.js/1.2.10/angular-
route.min.js"></script>
    <script>
      var countryApp = angular.module('countryApp', ['ngRoute']);

      countryApp.config(function($routeProvider) {
        $routeProvider.
          when('/', {
            templateUrl: 'country-list.html',
            controller: 'CountryListCtrl'
          }).
          when('/:countryName', {
            templateUrl: 'country-detail.html',
            controller: 'CountryDetailCtrl'
          }).
          otherwise({
            redirectTo: '/'
          });
      });

      countryApp.factory('countries', function($http){
        return {
          list: function(callback){
            $http.get('countries.json').success(callback);
          }
        };
      });

      countryApp.controller('CountryListCtrl', function ($scope, countries){
        countries.list(function(countries) {
          $scope.countries = countries;
        });
      });

      countryApp.controller('CountryDetailCtrl', function ($scope, $routeParams,
$http){
        $http.get('countries.json').success(function(data) {
```

```

        $scope.country = data.filter(function(entry){
            return entry.name === $routeParams.countryName
        })[0];
    });
});
</script>
</head>
<body>
    <div ng-view></div>
</body>
</html>

```

This is an updated version of the previous AngularJS application, with some changes in the way data is loaded and managed. The application still uses routing to display different views based on the URL path, but now uses a factory to load and manage the country data.

The 'countries' factory is defined, which uses the \$http service to load data from a 'countries.json' file. It returns an object with a single method, 'list', which takes a callback function as an argument. When the data is successfully loaded, the callback function is invoked with the data as an argument.

The 'CountryListCtrl' controller is defined and uses the 'countries' factory to load data. When the data is loaded, it is bound to the \$scope object using the 'ng-repeat' directive in the 'country-list.html' template, which displays a list of countries.

The 'CountryDetailCtrl' controller is defined and uses the \$routeParams service to retrieve the 'countryName' parameter from the URL. It then uses the 'countries' factory to load the data, and filters the data to find the country with the matching name. The matching country is then bound to the \$scope object and displayed in the 'country-detail.html' template.

Finally, the 'ng-view' directive is used to display the appropriate template based on the current route.

