1DA1611:A
Advanced Internet Programming

Project nr. 10 Report: 2023-04-27

Illia Priadko (309062)

Plan:

For each example given:

- Describe the basic components of AngularJS application
- Provide an explanation of how this code is applied in practice

Given task:

Tasks - Angular JS Examples

Please run and describe the functionalities of the following AngularJS examples

https://curran.github.io/screencasts/introToAngular/exampleViewer/#/

Examples 22, 33, 35 – Priadko Illia

1. Introduction

The examples from the webpage are given as code snippets and an <iframe> element that shows it in practice, sufficient to provide good insight into how the source code is both written and how it works.

I have chosen to append the comments and descriptions to the code snippets themselves, for easier reference and readability, via HTML and JS provided comment syntax respectively.

2. Example analysis:

Example 22 -- Adding search using Angular filters: Example Viewer (curran.github.io)

Focus of this example is the **filter**.
Documentation: AngularJS: API: filter

This element is used to select a subset of items from a given array (in our case – countries given in countries.json) and return a new array with only the selected elements included.

Example of usage with explanation:

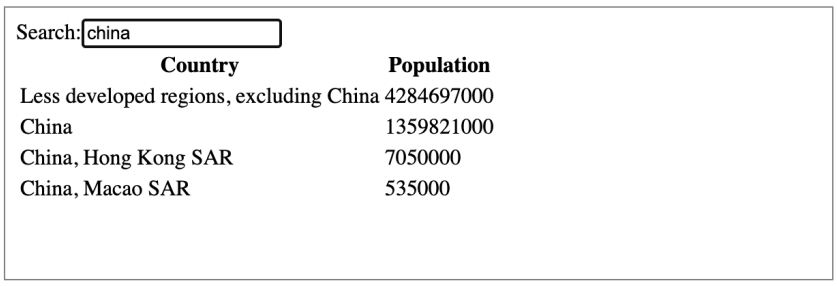
```
<html ng-app="countryApp">
<!-- The ng-app directive is used to define the main AngularJS module for the application, which
is named countryApp. -->

<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>
        // the AngularJS module is defined using the angular.module function. The [] parameter is
used to specify that there are no dependencies for this module. A controller named CountryCtrl is
also defined using the countryApp.controller method.
        var countryApp = angular.module('countryApp', []);
        // The CountryCtrl controller has two dependencies, $scope and $http. The $http
dependency is used to make a GET request to a file named countries.json. When the request is
successful, the response data is assigned to the $scope.countries property, making it available
to the view.
        countryApp.controller('CountryCtrl', ['$scope', '$http', function (scope, http) {
            http.get('countries.json').success(function (data) {
                scope.countries = data;
            });
        }]);
    </script>
</head>

<!-- In the <body> section of the HTML file, the ng-controller directive is used to associate the
CountryCtrl controller with the <body> element. -->
<body ng-controller="CountryCtrl">
    <!-- A search box is defined using an <input> element with the ng-model directive set to
query. This creates a two-way binding between the value of the search box and the $scope.query
property. -->
    Search:<input ng-model="query" type="text" />
    <!-- A table is defined using the <table> element. The table header is defined using the <th>
element and the table rows are generated using the ng-repeat directive. -->
    <table>
        <tr>
            <th>Country</th>
            <th>Population</th>
        </tr>
        <!-- The ng-repeat directive iterates over the countries array, which was defined in the
CountryCtrl controller. The | filter:query filter is applied to the ng-repeat directive, which
filters the list of countries based on the value of $scope.query. Each row of the table displays
the name and population of a country using the {{}} syntax to display the properties of the
country object. -->
        <tr ng-repeat="country in countries | filter:query">
            <td>{{country.name}}</td>
            <td>{{country.population}}</td>
        </tr>
    </table>
</body>

</html>
```

Please note, the the formatting of the .json file's elements is as follows:

```
  {
    "name": "countryname",
    "population": Population_as_integer
  },
```

Preview:



Example 33 -- Preparing for routing - making a simple country listing: [Example Viewer (curran.github.io)]

Please note that most elements that were described previously, so the comments in the following snippet will mostly associate with the new elements.

As this snippet does the same thing fetching a .json file via a GET request, the substantial difference is lack of filtering. As described in the name, it is no more than a simple country listing, using the HTML elements <ul> and <li>.

```
<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>

        // angular.module() function is used to create a new module called "countryApp" and
assign it to the variable countryApp. The empty array as the second argument indicates that this
module has no dependencies.
        var countryApp = angular.module('countryApp', []);
        // controller has two parameters: $scope and $http
        countryApp.controller('CountryListCtrl', function ($scope, $http) {
            $http.get('countries.json').success(function (data) {
                $scope.countries = data;
            });
        });
    </script>
</head>

<!-- associate the controller "CountryListCtrl" with the current HTML element. -->

<body ng-controller="CountryListCtrl">
    <!-- ng-repeat directive to loop through each country in the countries array and display its
name in an <li> tag with a {{}} expression -->
    <ul>
        <li ng-repeat="country in countries">{{country.name}}</li>
    </ul>
</body>

</html>
```

JSON formatting is as follows:

```
  {
    "name": "China",
    "population": 1359821000,
    "flagURL": "//upload.wikimedia.org/wikipedia/commons/f/fa/Flag_of_the_People%27s_Republic_of_China.svg",
    "capital": "Beijing",
    "gdp": 12261
  },
```

Preview:

- China
- India
- United States of America

Example 35 -- Moving templates for routes into separate files: [Example Viewer (curran.github.io)]

This example introduces routing: a functionality to navigate between different views/templates without reloading the entire page from the start. ngRoute is a separately distributed module.

```
<html ng-app="countryApp">

<head>
    <meta charset="utf-8">
    <title>Angular.js Example</title>
    <!-- The script section now also includes the routing module of AngularJS-->
    <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']);
        // $routeProvider sets up routes that the application will use to navigate between
different views. In this case, it defines two routes: one for the country list and another for
the country details, stored as separate .html files in the same directory.
        countryApp.config(function ($routeProvider) {
            $routeProvider.
                when('/', {
                    // The templateUrl property specifies the HTML template to use
                    templateUrl: 'country-list.html',
                    // controller property specifies the AngularJS controller that will manage
the data and behavior of each view
                    controller: 'CountryListCtrl'
                }).
                when('/:countryName', {
                    templateUrl: 'country-detail.html',
                    controller: 'CountryDetailCtrl'
                }).
                otherwise({
                    redirectTo: '/'
                });
        });
        // CountryListCtrl controller retrieves the list of countries from a JSON file using the
$http service and sets the $scope.countries variable to the retrieved data
        countryApp.controller('CountryListCtrl', function ($scope, $http) {
            $http.get('countries.json').success(function (data) {
                $scope.countries = data;
            });
        });
        // CountryDetailCtrl controller logs the parameters passed to it using the defined
$routeParams service
        countryApp.controller('CountryDetailCtrl', function ($scope, $routeParams) {
            console.log($routeParams);
        });
    </script>
</head>

<body>
    <!-- ng-view directive will be replaced with the HTML template of the current route when the
application is run -->
```

```
    <div ng-view></div>
</body>

</html>
```

Important note here is that the example is not in fact finished. It is more of a starting template for the future examples from the provided list. Hence, there are some TBDs:

country-detail.html:

```
<h1>TODO create country detail view</h1>
```

country-list.html:

```
<ul>
    <li ng-repeat="country in countries">{{country.name}}</li>
</ul>
```

JSON file is the same as in example 33.

Preview:

- China
- India
- United States of America

In summary, routing in AngularJS is an essential part of building single-page applications, allowing users to navigate between different views without reloading the page. It allows developers to define routes for different views, specify templates for each of them, manage data and behaviour of each view.

Conclusions:

After having done this exercise, I familiarized myself with some essential parts of making webpages in AngularJS. Filtering is a powerful tool for manipulating data and displaying it in different ways, http requests are used for fetching and sending data from and to the server (or a local directory in examples provided so far), while routing allows users to navigate between views without reloading the entire page. All of these elements enhance the user experience and provide versatile tools for developers to build powerful and flexible web applications.