Warsaw University of Technology	Advanced Internet Programming
Project 9 AngularJS - The basics	Maciej Iwańczyk 311258
Date: 26.04.2024	Grade:

1. Introduction:

In today's project I adapted three AngularJS examples to display data from Wikipedia about the largest power stations. The examples showcase different AngularJS features such as data binding, routing, and dynamic views.

2. Link to the website (later it will be uploaded to UNIX server):

The website: **https://zstarwarss.github.io/AIPprojectMI/**
The files and the codes: **https://github.com/ZstarwarsS/AIPprojectMI**

3. Example 32:

This example demonstrates how to display data in a table format using AngularJS. It fetches data from a JSON file and binds it to the view.
HTML:

```
<table>
  <tr>
    <th>Name</th>
    <th>Country</th>
    <th>Capacity (MW)</th>
    <th>Type</th>
    <th>Image</th>
  </tr>
  <tr ng-repeat="station in powerStations">
    <td>{{station.name}}</td>
    <td>{{station.country}}</td>
    <td>{{station.capacity | number}}</td>
    <td>{{station.type}}</td>
    <td><img ng-src="{{station.imageURL}}" width="100"></td>
  </tr>
</table>
```

Explanation:

- ng-repeat="station in powerStations": Iterates over each power station in the powerStations array.

- {{station.name}}, {{station.country}}, etc.: Bind the station's properties to the respective table cells.
- <img ng-src="{{station.imageURL}}" width="100">: Dynamically sets the image source for each station.

Data Source:

```
[
  {
    "name": "Three Gorges Dam",
    "country": "China",
    "capacity": 22500,
    "type": "Hydroelectric",
    "imageURL": "//upload.wikimedia.org/wikipedia/commons/thumb/a/ab/ThreeGorgesDam-China2009.jpg/1920px-ThreeGorgesDam-China2009.jpg"
  },
  {
    "name": "Itaipu Dam",
    "country": "Brazil/Paraguay",
    "capacity": 14000,
    "type": "Hydroelectric",
    "imageURL": "//upload.wikimedia.org/wikipedia/commons/thumb/d/d2/Itaipu_geral.jpg/1920px-Itaipu_geral.jpg"
  },
  {
    "name": "Xiluodu Dam",
    "country": "China",
    "capacity": 13860,
    "type": "Hydroelectric",
    "imageURL":
    "//upload.wikimedia.org/wikipedia/commons/thumb/1/19/%E6%BA%AA%E6%B4%9B%E6%B8%A1%E6%B0%B4%E7%94%B5%E7%AB%992011%E4%B8%8A%E6%B8%B8%
    1%E6%B0%B4%E7%94%B5%E7%AB%992011%E4%B8%8A%E6%B8%B8%E9%B8%9F%E7%9E%B0.jpg"
  }
]
```

4. Example 34:

This example lists the names of power stations using AngularJS's ng-repeat.

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

Explanation:

- ng-repeat="station in powerStations": Iterates over each power station in the powerStations array.
- {{station.name}}: Binds the name of each power station to a list item.

5. Example 36:

This example extends the list of power stations to include links to detailed views of each station.

```
<ul>
  <li ng-repeat="station in powerStations"><a href="#/example36/{{station.name}}">{{station.name}}</a></li>
</ul>
```

detail:

```
<h1>{{name}}</h1>
```

Explanation:

- ng-repeat="station in powerStations": Iterates over each power station.
- {{station.name}}: Creates a link for each station that navigates to a detailed view of that station.

6. Menu (Project 9 combined.html)

This is the main application file that combines the examples and sets up routing.

HTML and AngularJS Setup:

```
<!DOCTYPE html>
<html ng-app="combinedApp">
<head>
  <meta charset="utf-8">
  <title>Angular.js Combined Examples</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 combinedApp = angular.module('combinedApp', ['ngRoute']);

    combinedApp.config(function($routeProvider) {
      $routeProvider
        .when('/', {
          template: '<h1>Welcome to AngularJS Examples</h1><p>Select an example from the menu to view it.</p>'
        })
        .when('/example32', {
          templateUrl: 'example32.html',
          controller: 'Example32Ctrl'
        })
        .when('/example34', {
          templateUrl: 'example34.html',
          controller: 'Example34Ctrl'
        })
        .when('/example36', {
          templateUrl: 'example36.html',
          controller: 'Example36Ctrl'
        })
        .when('/example36/:stationName', {
          templateUrl: 'power-station-detail.html',
          controller: 'Example36DetailCtrl'
        })
        .otherwise({
          redirectTo: '/'
        });
    });
```

```
    combinedApp.controller('Example32Ctrl', function ($scope, $http){
      $http.get('power_stations.json').success(function(data) {
        $scope.powerStations = data;
      });
    });

    combinedApp.controller('Example34Ctrl', function ($scope, $http){
      $http.get('power_stations.json').success(function(data) {
        $scope.powerStations = data;
      });
    });

    combinedApp.controller('Example36Ctrl', function ($scope, $http){
      $http.get('power_stations.json').success(function(data) {
        $scope.powerStations = data;
      });
    });

    combinedApp.controller('Example36DetailCtrl', function ($scope, $routeParams){
      $scope.name = $routeParams.stationName;
    });

    combinedApp.controller('MenuCtrl', function($scope, $location) {
      $scope.isActive = function(viewLocation) {
        return viewLocation === $location.path();
      };
    });
  </script>
</head>
<body>
  <ul class="menu" ng-controller="MenuCtrl">
    <li><a href="#/" ng-class="{active: isActive('/')}">Home</a></li>
    <li><a href="#/example32" ng-class="{active: isActive('/example32')}">Example 32</a></li>
    <li><a href="#/example34" ng-class="{active: isActive('/example34')}">Example 34</a></li>
    <li><a href="#/example36" ng-class="{active: isActive('/example36')}">Example 36</a></li>
  </ul>
  <div ng-view></div>
</body>
</html>
```

Explanation:

- ng-app="combinedApp": Initializes the AngularJS application.
- Routing Configuration: Defines routes for the different examples.
- Controllers: Manage data fetching and binding for each example.
- Menu: Allows navigation between examples with active state indication.

7. Topic 2 View encapsulation

View encapsulation in Angular is a mechanism to restrict styles defined in one component from affecting other components. Angular provides three encapsulation modes:

1. Emulated (default): Emulates native shadow DOM behavior by preprocessing the CSS.
2. Native: Uses the browser's native shadow DOM implementation.
3. None: No encapsulation; styles are applied globally.

Inspecting Generated CSS:

- In Emulated mode, Angular adds unique attribute selectors to encapsulate styles.
- In Native mode, styles are scoped to the shadow DOM.
- In None mode, styles are applied without scoping.

Mixing Encapsulation Modes:

- Components can use different encapsulation modes within the same application.
- Mixing modes can be useful for applying global styles while keeping component-specific styles isolated.

8. Final project:

Home | Example 32 | Example 34 | Example 36

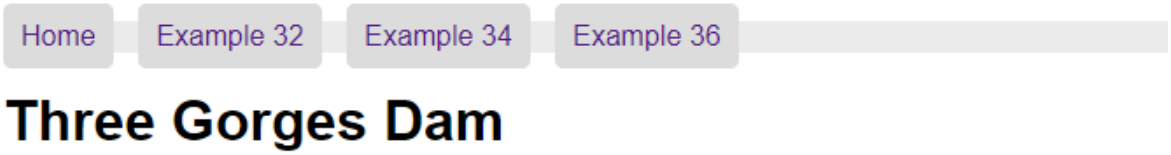
Name	Country	Capacity (MW)	Type	Image
Three Gorges Dam	China	22,500	Hydroelectric	
Itaipu Dam	Brazil/Paraguay	14,000	Hydroelectric	
Xiluodu Dam	China	13,860	Hydroelectric	

Home | Example 32 | Example 34 | Example 36

- Three Gorges Dam
- Itaipu Dam
- Xiluodu Dam

Home | Example 32 | Example 34 | Example 36

- Three Gorges Dam
- Itaipu Dam
- Xiluodu Dam



9. Conclusion:

AngularJS provides powerful features for building dynamic web applications, including data binding, dependency injection, and routing. The provided examples illustrate how to use AngularJS to display and interact with data, and the combined project shows how to integrate multiple examples into a single application with navigation. Understanding view encapsulation helps manage styles effectively across different components.