

X. CÚ PHÁP INCLUDE TRONG ANGULARJS

- HTML không hỗ trợ nhúng trang web HTML trong trang HTML. Để đạt được chức năng này có thể sử dụng các phương thức sau:
- Sử dụng Ajax - Tạo một server call để nhận trang HTML tương ứng và thiết lập nó trong thành phần innerHTML của phần điều khiển HTML.
- Sử dụng Server Side Includes - JSP, PHP và các công nghệ web server khác có thể include HTML page trong một trang web động.
- Với AngularJS, chúng ta có thể nhúng trang HTML sử dụng ng-include directive.

```
<div ng-app="" ng-controller="sinhvienController">
  <div ng-include="main.html"></div>
  <div ng-include="tenMonHoc.html"></div>
</div>
```

```
<html>
<head>
<title>Cu Phap Include trong AngularJS</title>
<script
src="http://ajax.googleapis.com/ajax/libs/angularjs/1.3.14/angular.min.js"></script>
<style>
table, th, td {
  border: 1px solid grey;
  border-collapse: collapse;
  padding: 5px;
}
table tr:nth-child(odd) {
  background-color: #f2f2f2;
}
table tr:nth-child(even) {
  background-color: #ffffff;
```

```
}
</style>
</head>
<body>
<h2>Ung dung AngularJS</h2>
<div ng-app="ungdungAngularjs" ng-controller="sinhvienController">
<div ng-include="main.html"></div>
<div ng-include="tenMonHoc.html"></div>
</div>
<script>
var ungdungAngularjs = angular.module("ungdungAngularjs", []);

ungdungAngularjs.controller('sinhvienController', function($scope) {
    $scope.sinhvien = {
        ho: "Tran Bat",
        ten: "Tai",
        hocphi:400,
        tenMonHoc:[
            {ten:'Vat Ly Dai Cuong',diemthi:9.0},
            {ten:'Triet Hoc',diemthi:9.5},
            {ten:'Toan',diemthi:9.5},
            {ten:'Tieng Anh',diemthi:9.5},
            {ten:'Kinh Te',diemthi:9.0}
        ],
        hoten: function() {
            var doituongsinhvien;
            doituongsinhvien = $scope.sinhvien;
            return doituongsinhvien.ho + " " + doituongsinhvien.ten;
        }
    };
};
```

```
});  
</script>  
</body>  
</html>
```

← → ↻ 🏠 ⓘ file:///E:/FPT_FPT%20Poly/JavaScript%201/Buoi%2016/cuphapInclude.html#/

Ung dung AngularJS

Nhap ho:	<input type="text" value="Tran Bat"/>
Nhap ten:	<input type="text" value="Tai"/>
Ho va Ten:	Tran Bat Tai

Mon Hoc :

Ten	Diem thi
Vat Ly Dai Cuong	9
Triet Hoc	9.5
Toan	9.5
Tieng Anh	9.5
Kinh Te	9

XI. AJAX TRONG ANGULARJS

- AngularJS cung cấp điều khiển \$http mà làm như là service để đọc dữ liệu từ server. Server có thể tạo các lời gọi tới cơ sở dữ liệu để nhận các bản ghi. AngularJS cần dữ liệu dưới định dạng JSON. Khi dữ liệu sẵn sàng, \$http có thể được sử dụng để nhận dữ liệu từ server theo cách sau đây:

```
function sinhvienController($scope,$http) {  
var url="dulieuSV.txt";
```

```
$http.get(url).success( function(response) {  
    $scope.sinhvienk60 = response;  
    });  
}
```

Ở đây tệp dulieuSV.txt chứa các bản ghi về sinh viên. \$http service tạo một ajax call và lấy kết quả trả về cho đối tượng sinh viên. "sinhvien" model có thể dùng để vẽ bảng với HTML.

dulieuSV.txt

```
[  
{  
  "HovaTen" : "Trần Bất Tài",  
  "MSSV" : 20150456,  
  "Diemthi" : "8.0"  
},  
{  
  "HovaTen" : "Nguyen Thi Minh Linh",  
  "MSSV" : 20150457,  
  "Diemthi" : "9.0"  
},  
{  
  "HovaTen" : "Ha Kieu Linh",  
  "MSSV" : 20150458,  
  "Diemthi" : "8.5"  
},  
{  
  "HovaTen" : "Ho Ngoc Ha",  
  "MSSV" : 20150459,
```

```
"Diemthi" : "9.5"
```

```
}
```

```
]
```

viduAJAX.html

```
<html>
```

```
<head>
```

```
<title>Vi du AJAX trong AngularJS</title>
```

```
<style>
```

```
table, th, td {
```

```
    border: 1px solid grey;
```

```
    border-collapse: collapse;
```

```
    padding: 5px;
```

```
}
```

```
table tr:nth-child(odd) {
```

```
    background-color: #f2f2f2;
```

```
}
```

```
table tr:nth-child(even) {
```

```
    background-color: #ffffff;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h2>Ung dung AngularJS</h2>
```

```
<div ng-app="" ng-controller="sinhvienController">
```

```
<table>
```

```
    <tr>
```

```
        <th>Ho va Ten</th>
```

```
        <th>MSSV</th>
```

```

        <th>Diem thi</th>
    </tr>
    <tr ng-repeat="sinhvien in sinhvienk60">
        <td>{{ sinhvien.HovaTen }}</td>
        <td>{{ sinhvien.MSSV }}</td>
        <td>{{ sinhvien.Diemthi }}</td>
    </tr>
</table>
</div>
<script>
function sinhvienController($scope,$http) {
var url="dulieuSV.txt";
    $http.get(url).success( function(response) {
        $scope.sinhvienk60 = response;
    });
}
</script>
<script
src="http://ajax.googleapis.com/ajax/libs/angularjs/1.2.15/angular.min.js"></script>
</body>
</html>

```

← → ↺ 🏠 ⓘ file:///E:/FPT_FPT%20Poly/JavaScript%201/Buoi%2016/viduAJAX.html

Ung dung AngularJS

Ho va Ten	MSSV	Diem thi
Dao Duy Ngoc	20150456	8.0
Nguyen Thi Minh Linh	20150457	9.0
Ha Kieu Linh	20150458	8.5
Ho Ngoc Ha	20150459	9.5

XII. THÀNH PHẦN VIEW TRONG ANGULARJS

- AngularJS hỗ trợ Single Page Application thông qua multiple view trên một trang đơn.
- Để làm được điều này, AngularJS cung cấp ng-view và ng-template directive và \$routeProvider service.
- **Giới thiệu ng-view trong AngularJS:**

- Thẻ ng-view đơn giản là tạo nơi giữ các màn hình view tương ứng có thể được đặt trong nó dựa vào cấu hình.
- Cách sử dụng

Định nghĩa thẻ div với ng-view trong module chính.

```
<div ng-app="ungdungAngularJS">
...
<div ng-view></div>
</div>
```

- **Giới thiệu ng-template trong AngularJS:**

- ng-template directive được sử dụng để tạo ra các HTML view sử dụng thẻ script. Nó chứa thuộc tính "id" được sử dụng bởi \$routeProvider để liên kết view và controller.
- Cách sử dụng

Định nghĩa một khối script với kiểu như ng-template trong module chính.

```
<div ng-app="ungdungAngularJS">
...
</div>
```

```
<script type="text/ng-template" id="themSV.html">
  <h2> Add Student </h2>
  {{message}}
</script>

</div>
```

- Giới thiệu \$routeProvider trong AngularJS:

- Là dịch vụ chính trong việc tạo các cấu hình cho địa chỉ URL, liên kết chúng với trang HTML tương ứng hoặc ng-template và gắn controller với chúng.
- Cách sử dụng

Định nghĩa một khối script trong module chính và thiết lập cấu hình định tuyến.

```
var ứng dụngAngularJS = angular.module("ứng dụngAngularJS", ['ngRoute']);

ứng dụngAngularJS.config(['$routeProvider',
function($routeProvider) {
  $routeProvider.
    when('/themSV', {
      templateUrl: 'themSV.html',
      controller: 'themSVController'
    }).
    when('/quansatSV', {
      templateUrl: 'quansatSV.html',
      controller: 'quansatSVController'
    }).
    otherwise({
      redirectTo: '/themSV'
    });
}]);
```


- Dưới đây là những điểm quan trọng cần xem xét từ ví dụ trên.
 - \$routeProvider định nghĩa là một hàm dưới config của ứng dụng AngularJS module sử dụng khóa là "\$routeProvider".
 - \$routeProvider.when định nghĩa một địa chỉ URL "/themSV" được sử dụng để liên kết đến trang "themSV.html", trong đó themSV.html nên đặt chung thư mục đường dẫn với trang HTML. Nếu trang HTML không được định nghĩa, ng-template sẽ sử dụng id="themSV.html". Chúng ta sử dụng ng-template.
 - "otherwise" được sử dụng để thiết lập view mặc định.
 - "controller" được để thiết lập controller tương ứng với từng view.
- Ví dụ Dưới đây là ví dụ minh họa cho những directive mô tả bên trên.

thanhphanView.html

```
<html>
<head>
  <title>Vi du View trong Angular JS</title>
  <script
src="http://ajax.googleapis.com/ajax/libs/angularjs/1.3.14/angular.min.js"></script>
  <script          src="http://ajax.googleapis.com/ajax/libs/angularjs/1.3.14/angular -
route.min.js"></script>
</head>
<body>
  <h2>Ung dung AngularJS</h2>
  <div ng-app="ungdungAngularJS">
    <p><a href="#themSV">Them sinh vien</a></p>
    <p><a href="#quansatSV">Quan sat sinh vien</a></p>
    <div ng-view></div>
    <script type="text/ng-template" id="themSV.html">
      <h2> Them sinh vien </h2>
      {{message}}
```

```

</script>
<script type="text/ng-template" id="quansatSV.html">
  <h2> Quan sat sinh vien </h2>
  {{message}}
</script>
</div>

<script>
var ungdungAngularJS = angular.module("ungdungAngularJS", ['ngRoute']);

ungdungAngularJS.config(['$routeProvider',
  function($routeProvider) {
    $routeProvider.
      when('/themSV', {
        templateUrl: 'themSV.html',
        controller: 'themSVController'
      }).
      when('/quansatSV', {
        templateUrl: 'quansatSV.html',
        controller: 'quansatSVController'
      }).
      otherwise({
        redirectTo: '/themSV'
      });
  }]);

ungdungAngularJS.controller('themSVController', function($scope) {
  $scope.message = "Trang nay se duoc su dung de hien thi mot form de them sinh
vien";
});

```

```

    ungdungAngularJS.controller('quansatSVController', function($scope) {
        $scope.message = "Trang nay se duoc su dung de quan sat tat ca sinh vien";
    });
</script>
</body>
</html>

```



XIII. SCOPE TRONG ANGULARJS

- Scope là đối tượng JavaScript đặc biệt có vai trò liên kết controller và view.
- Scope chứa thông tin là các dữ liệu model. Trong controller, dữ liệu model có thể được truy cập qua đối tượng \$scope.

```

<script>
    var ungdungAngularjs = angular.module("ungdungAngularjs", []);

    ungdungAngularjs.controller("nhanvienController", function($scope) {
        $scope.message = "Ben trong nhan vien controller";
        $scope.type = "Nhan vien";
    });

```

```
</script>
```

- Dưới đây là những điểm quan trọng của ví dụ trên.
 - o \$scope được truyền như là tham số đầu tiên của controller trong hàm khởi tạo của nó.
 - o \$scope.message và \$scope.type là các model được sử dụng trong trang HTML.
 - o Chúng ta thiết lập các giá trị cho model và tác động lên Module ứng dụng với controller và nhanvienController.
 - o Chúng ta có thể định nghĩa các hàm với \$scope.
- Tính kế thừa của Scope trong AngularJS
 - o Scope là controller riêng biệt. Chúng ta định nghĩa nested controller (các controller lồng nhau) để các controller con sẽ kế thừa từ các controller cha..

```
<script>
    var ungdungAngularjs = angular.module("ungdungAngularjs", []);

    ungdungAngularjs.controller("nhanvienController", function($scope) {
        $scope.message = "Ben trong nhan vien controller";
        $scope.type = "Nhan vien";
    });

    ungdungAngularjs.controller("nhanvienITController", function($scope) {
        $scope.message = "Ben trong nhan vien IT controller";
    });
</script>
```

- Dưới đây là những điểm chính qua ví dụ trên.
 - o Chúng ta tạo giá trị biến model cho nhanvienController.

- Chúng ta ghi đè thông báo của controller con là nhanvienITController. Khi "message" được sử dụng trong các module của nhanvienITController, giá trị message ghi đè sẽ được sử dụng.
- Dưới đây là phần ví dụ minh họa cho phần hướng dẫn bên trên.

viduScope.html

```
<html>
<head>
  <title>Vi du Scope trong AngularJS</title>
</head>
<body>
  <h2>Ung dung AngularJS</h2>
  <div ng-app="ungdungAngularjs" ng-controller="nhanvienController">
    <p>{{message}} <br/> {{type}} </p>
    <div ng-controller="nhanvienITController">
      <p>{{message}} <br/> {{type}} </p>
    </div>
    <div ng-controller="nhanvienBHController">
      <p>{{message}} <br/> {{type}} </p>
    </div>
  </div>
  <script
src="http://ajax.googleapis.com/ajax/libs/angularjs/1.3.14/angular.min.js"></script>
  <script>
    var ungdungAngularjs = angular.module("ungdungAngularjs", []);


    ungdungAngularjs.controller("nhanvienController", function($scope) {
      $scope.message = "Ben trong nhan vien controller";
      $scope.type = "Nhan vien";
```

```
});

ungdungAngularjs.controller("nhanvienITController", function($scope) {
    $scope.message = "Ben trong nhan vien IT controller";
});

ungdungAngularjs.controller("nhanvienBHController", function($scope) {
    $scope.message = "Ben trong nhan vien ban hang controller";
    $scope.type = "Nhan vien BH";
});

</script>
</body>
</html>
```

 Tập | E:/FPT_FPT%20Poly/JavaScript%201/Buoi%2016/viduScope.html

Ung dung AngularJS

Ben trong nhan vien controller
Nhan vien

Ben trong nhan vien IT controller
Nhan vien

Ben trong nhan vien ban hang controller
Nhan vien BH

XIV. CÁC SERVICE TRONG ANGULARJS

- AngularJS hỗ trợ các khái niệm "Seperation of Concerns - Chia để trị" sử dụng cấu trúc service. Service là các hàm JavaScript và có nhiệm vụ trên những task nhất định. Nó làm cho chúng thành những thực thể riêng rẽ dễ dàng trong việc bảo trì và kiểm thử. Controller,

filter có thể gọi chúng một cách đơn giản. Service thường được inject sử dụng cơ chế dependency injection của AngularJS.

- AngularJS cung cấp rất nhiều những service được định nghĩa cho trước: \$http,\$scope,\$route,\$window,\$location... Mỗi một service có những nhiệm vụ nhất định.
- Ví dụ, \$http được sử dụng để tạo ra các ajax request lên server để lấy dữ liệu về. \$route được sử dụng để định nghĩa thông tin routing Những service mặc định của AngularJS bắt đầu bởi biểu tượng \$.
- **Có 2 cách để tạo một service trong AngularJS là:**

factory

service

- **Sử dụng phương thức factory trong AngularJS**
 - o Khi sử dụng factory method, đầu tiên chúng ta định nghĩa factory và gán method cho nó.

```
var ứng dụngAngularJS = angular.module("ứng dụngAngularJS", []);
    ứng dụngAngularJS.factory('toanhocService', function() {
        var factory = {};
        factory.phépnhân = function(a, b) {
            return a * b
        }
        return factory;
    });
```

- **Sử dụng phương thức service trong AngularJS:**
 - o Sử dụng service method, chúng ta sẽ định nghĩa service sau đó gán method với nó. Chúng ta cũng inject những service có sẵn cho nó.

```
ứng dụngAngularJS.service('tínhBPService', function(toanhocService){
    this.bìnhphuong = function(a) {
```

```
return toanhocService.phepnhan(a,a);  
}  
});
```

- Ví dụ Dưới đây là ví dụ minh họa cho chỉ dẫn bên trên.

testAngularJS.jsp

```
<html>  
<head>  
  <title>Vi du Service trong AngularJS</title>  
  <script  
src="http://ajax.googleapis.com/ajax/libs/angularjs/1.3.14/angular.min.js"></script>  
</head>  
<body>  
  <h2>Ung dung AngularJS</h2>  
  <div ng-app="ungdungAngularJS" ng-controller="tinhBPController">  
    <p>Nhap mot so: <input type="number" ng-model="number" />  
    <button ng-click="binhphuong()">X<sup>2</sup></button>  
    <p>Ket qua: {{ketqua}}</p>  
  </div>  
  <script>  
    var ungdungAngularJS = angular.module("ungdungAngularJS", []);  
    ungdungAngularJS.factory('toanhocService', function() {  
      var factory = {};  
      factory.phepnhan = function(a, b) {  
        return a * b  
      }  
    })
```



```
        return factory;
    });

    ungdungAngularJS.service('tinhBPService', function(toanhocService){
        this.binhphuong = function(a) {
            return toanhocService.phepnhan(a,a);
        }
    });

    ungdungAngularJS.controller('tinhBPController', function($scope, tinhBPService) {
        $scope.binhphuong = function() {
            $scope.ketqua = tinhBPService.binhphuong($scope.number);
        }
    });
</script>
</body>
</html>
```

XV. DEPENDENCY INJECTION TRONG ANGULARJS

- Dependency Injection là một mô hình thiết kế phần mềm mà trong đó các thành phần được đưa ra từ những phần phụ thuộc nó - dependencies thay cho việc hard coding chúng trong các thành phần. Điều này làm cho cách thành phần phụ thuộc nhau trong phần cấu hình. Nó giúp việc làm có các thành phần có tính tái sử dụng cao, dễ bảo dưỡng và kiểm thử.
- AngularJS cung cấp kỹ thuật Dependency Injection, cho phép các thành phần lõi của AngularJS có thể được inject tới các thành phần phụ thuộc khác.

value

factory

service

provider

constant

- **Đối tượng value trong AngularJS**

- là đối tượng JavaScript đơn giản và được sử dụng để thiết lập các giá trị tới controller trong các bước cấu hình.

```
// Định nghĩa một module

var ứng dụngAngularjs = angular.module("ứng dụngAngularjs", []);

// tạo một đối tượng value dưới dạng "giatrimac dinh" và truyền dữ liệu cho nó.

ứng dụngAngularjs.value("giatrimac dinh", 5);
...

// inject giá trị này trong controller bởi sử dụng tên của nó là "giatrimac dinh"

ứng dụngAngularjs.controller('tinhBPController', function($scope, tinhBPService,
giatrimac dinh) {
    $scope.number = giatrimac dinh;
    $scope.ketqua = tinhBPService.binhphuong($scope.number);

    $scope.binhphuong = function() {
        $scope.ketqua = tinhBPService.binhphuong($scope.number);
    }
});
```

- Hàm factory trong AngularJS

- factory là một hàm để sử dụng trả về giá trị. Nó tạo ra giá trị theo yêu cầu mỗi khi service hoặc controller yêu cầu. Ta thường dùng các hàm factory để tính và trả về giá trị.

```
// Định nghĩa một module

var ứng dụngAngularjs = angular.module("ứng dụngAngularjs", []);

// Tạo một factory là "toanhocService" mà cung cấp một phương thức là phép nhân để trả về tích của hai số
ứng dụngAngularjs.factory('toanhocService', function() {
    var factory = {};
    factory.phép nhân = function(a, b) {
        return a * b
    }
    return factory;
});

//inject "toanhocService" trong một service để lợi dụng phương thức phép nhân của factory.

ứng dụngAngularjs.service('tínhBPService', function(toanhocService){
    this.bình phương = function(a) {
        return toanhocService.phép nhân(a,a);
    }
});
...
```

- Đối tượng service trong AngularJS

- service là một đối tượng singleton javascript chứa tập các hàm cho các mục đích cụ thể. Service được định nghĩa sử dụng hàm service() và sau đó inject nó đến controller.

```

// Dinh nghia mot module

var ungdungAngularjs = angular.module("ungdungAngularjs", []);
...

// Tao mot service ma dinh nghia mot phuong thuc binhphuong de tra ve binh phuong cua mot
so.

ungdungAngularjs.service('tinhBPService', function(toanhocService){
    this.binhphuong = function(a) {
        return toanhocService.phepnhan(a,a);
    }
});

//inject "tinhBPService" vao trong controller

ungdungAngularjs.controller('tinhBPController', function($scope, tinhBPService,
giatrimacdinhh) {
    $scope.number = giatrimacdinhh;
    $scope.ketqua = tinhBPService.binhphuong($scope.number);

    $scope.binhphuong = function() {
        $scope.ketqua = tinhBPService.binhphuong($scope.number);
    }
});

```

- Hàm provider trong AngularJS

- provider được sử dụng bởi trong nội bộ AngularJS để tạo service, factory ... trong quá trình cài đặt (quá trình mà AngularJS khởi tạo chính nó). Dưới đây mô tả script có thể tạo toanhocService trong đó chúng ta tạo trước đó. Provider là một phương thức factory đặc biệt với phương thức get() trả về giá trị là value/service/factory.

```
// Định nghĩa một module
var ứng dụngAngularjs = angular.module("ứng dụngAngularjs", []);
...

// Tạo một service bởi sử dụng provider mà định nghĩa phương thức bình phương để trả về bình
phương của một số.
ứng dụngAngularjs.config(function($provide) {
    $provide.provider('toanhocService', function() {
        this.$get = function() {
            var factory = {};
            factory.phép nhân = function(a, b) {
                return a * b;
            }
            return factory;
        };
    });
});
```

- **constant trong AngularJS**

- constant được sử dụng để truyền các giá trị trong tại giai đoạn cấu hình.

```
ứng dụngAngularjs.constant("tham số Cấu Hình", "giá trị hằng");
```

- Dưới đây là ví dụ minh họa các khái niệm trên:

```
<html>
<head>
  <title>Vi du Dependency Injection trong AngularJS</title>
</head>
<body>
  <h2>Ung dung AngularJS</h2>
  <div ng-app="ungdungAngularjs" ng-controller="tinhBPController">
    <p>Nhap mot so: <input type="number" ng-model="number" />
    <button ng-click="binhphuong()">X<sup>2</sup></button>
    <p>Ket qua: {{ketqua}}</p>
  </div>
  <script
src="http://ajax.googleapis.com/ajax/libs/angularjs/1.3.14/angular.min.js"></script>
  <script>
    var ungdungAngularjs = angular.module("ungdungAngularjs", []);

    ungdungAngularjs.config(function($provide) {
      $provide.provider('toanhocService', function() {
        this.$get = function() {
          var factory = {};
          factory.phepnhan = function(a, b) {
            return a * b;
          }
          return factory;
        };
      });
    });

    ungdungAngularjs.value("giatrimacdinhh", 5);
```

```

    undungAngularjs.factory('toanhocService', function() {
        var factory = {};
        factory.phepnhan = function(a, b) {
            return a * b;
        }
        return factory;
    });

    undungAngularjs.service('tinhBPService', function(toanhocService){
        this.binhphuong = function(a) {
            return toanhocService.phepnhan(a,a);
        }
    });

    undungAngularjs.controller('tinhBPController', function($scope, tinhBPService,
giatrimacdinhh) {
        $scope.number = giatrimacdinhh;
        $scope.ketqua = tinhBPService.binhphuong($scope.number);

        $scope.binhphuong = function() {
            $scope.ketqua = tinhBPService.binhphuong($scope.number);
        }
    });
</script>
</body>
</html>

```

XVI. TẠO CUSTOM DIRECTIVE TRONG ANGULARJS

- Custom directive được sử dụng trong AngularJS để kế thừa chức năng của HTML. Custom directive được định nghĩa sử dụng hàm "directive". Một custom directive đơn giản là thay thế những thành phần cho nó được kích hoạt. Ứng dụng AngularJS trong quá trình khởi tạo tìm kiếm những thành phần phù hợp và hoạt động một lần sử dụng phương thức compile() trong custom directive sau đó sử dụng phương thức link() để tạo custom directive dựa vào scope của các directive. AngularJS cung cấp cho chúng ta cách tạo custom directive theo những loại sau:

Element directive - Directive kích hoạt khi một thành phần trùng tên với nó gặp phải.

Attribute - Directive kích hoạt khi gặp phải một thuộc tính trùng với nó.

CSS - Directive kích hoạt khi gặp phải một css style trùng tên với nó.

Comment - Directive kích hoạt khi gặp phải một comment trùng với nó.

- Tìm hiểu Custom Directive trong AngularJS
 - o Định nghĩa thẻ html tùy biến.

```
<sinhvien ten="Chinh"></sinhvien><br/>  
<sinhvien ten="Manh"></sinhvien>
```

- Tiếp theo, chúng ta định nghĩa custom directive để xử lý thẻ custom HTML trên.

```
var ungdungAngularjs = angular.module("ungdungAngularjs", []);  
  
// Tao mot directive, trong do tham so dau tien la phan tu html de duoc dinh kem.  
// Chung ta dang dinh kem the html la sinhvien.  
// Directive se duoc dinh kem ngay khi bat ky phan tu sinhvien nao duoc nhap vao trong html
```



```

ungdungAngularjs.directive('sinhvien', function() {

    // Dinh nghĩa đối tượng directive

    var directive = {};

    // restrict = E, chỉ ràng directive là <i>Element</i> directive

    directive.restrict = 'E';

    //template thay thế hoàn toàn phần tử với phần text của nó.

    directive.template = "Sinh vien: <b>{{sinhvien.ten}}</b> , MSSV: <b>{{sinhvien.mssv}}</b>";

    // scope được sử dụng để phân biệt các phần tử sinh viên dựa trên tiêu chí là <i>ten</i>.

    directive.scope = {
        sinhvien : "=ten"
    }

    // compile được gọi trong khi khởi tạo ứng dụng. AngularJS gọi nó một lần khi html page được tải.

    directive.compile = function(element, attributes) {
        element.css("border", "1px solid #cccccc");

        // Hàm hamLienKet được liên kết với mọi phần tử với scope để lấy dữ liệu cụ thể.

```

```

var hamLienKet = function($scope, element, attributes) {
    element.html("Sinh    vien:    <b>"+$scope.sinhvien.ten    + "</b>    ,    MSSV:
<b>"+$scope.sinhvien.mssv+"</b><br/>");
    element.css("background-color", "#ff00ff");
}
return hamLienKet;
}
return directive;
});

```

- Định nghĩa controller để cập nhật scope cho directive. Ở đây là các đặt tên các thuộc tính của biến scope.

```

ungdungAngularjs.controller('sinhvienController', function($scope) {
    $scope.Chinh = {};
    $scope.Chinh.ten = "Dao Duy Canh";
    $scope.Chinh.mssv = 123456789;

    $scope.Manh = {};
    $scope.Manh.ten = "Nguyen Thi Bich";
    $scope.Manh.mssv = 123321123;
});

```

Ví dụ 1:

```
<html>
```

```

<head>
  <title>Tao Custom Directive trong AngularJS </title>
</head>
<body>
  <h2>Ung dung AngularJS</h2>
  <div ng-app="ungdungAngularjs" ng-controller="sinhvienController">
    <sinhvien ten="Chinh"></sinhvien><br/>
    <sinhvien ten="Manh"></sinhvien>
  </div>
  <script
src="http://ajax.googleapis.com/ajax/libs/angularjs/1.3.14/angular.min.js"></script>
  <script>
    var ungdungAngularjs = angular.module("ungdungAngularjs", []);

    ungdungAngularjs.directive('sinhvien', function() {
      var directive = {};
      directive.restrict = 'E';
      directive.template = "Sinh vien: <b>{{sinhvien.ten}}</b> , MSSV:
<b>{{sinhvien.mssv}}</b>";

      directive.scope = {
        sinhvien : "=ten"
      }

      directive.compile = function(element, attributes) {
        element.css("border", "1px solid #cccccc");

        var hamLienKet = function($scope, element, attributes) {
          element.html("Sinh vien: <b>"+$scope.sinhvien.ten +"</b> , MSSV:
<b>"+$scope.sinhvien.mssv+"</b><br/>");

```

```
        element.css("background-color", "#ff00ff");
    }

    return hamLienKet;
}

return directive;
});

ungdungAngularjs.controller('sinhvienController', function($scope) {
    $scope.Chinh = {};
    $scope.Chinh.ten = "Tran Bat Tai";
    $scope.Chinh.mssv = 123456789;

    $scope.Manh = {};
    $scope.Manh.ten = "Nguyen Thi Bich";
    $scope.Manh.mssv = 123321123;
});

</script>
</body>
</html>
```

file:///E:/FPT_FPT%20Poly/JavaScript%201/Buoi%2016/Directive_AngularJS/Vidu1.html

Ung dung AngularJS

Sinh vien: **Tran Bat Tai** , MSSV: **123456789**

Sinh vien: **Nguyen Thi Bich** , MSSV: **123321123**

XVII. ĐA NGÔN NGỮ (i18n) TRONG ANGULARJS

- AngularJS cung cấp sẵn các chức năng Đa ngôn ngữ (i18n) với 3 kiểu filter là currency, date và number.
- Chúng ta chỉ cần kết hợp đoạn js tương ứng với theo vị trí tùy Quốc gia. Mặc định nó sẽ gắn với vị trí trong trình duyệt web. Ví dụ, khi sử dụng vị trí ở Đan Mạch, bạn sử dụng đoạn script sau đây:

```
<script src="https://code.angularjs.org/1.2.5/i18n/angular-locale_da-dk.js"></script>
```

Ví dụ cho vị trí Đan Mạch trong AngularJS

dangonngu.html

```
<html>
<head>
  <title>Vi du Da ngon ngu (i18n) trong AngularJS</title>
</head>
<body>
  <h2>Ung dung AngularJS</h2>
  <div ng-app="ungdungAngularjs" ng-controller="nhanvienController">
    {{luong | currency }} <br/><br/>
    {{ngaytinhluong | date }} <br/><br/>
    {{nhanvienid | number }}
  </div>
  <script
src="http://ajax.googleapis.com/ajax/libs/angularjs/1.3.14/angular.min.js"></script>
  <script
src="https://code.angularjs.org/1.3.14/i18n/angular-locale_da-
dk.js"></script>
  <script>
    var ungdungAngularjs = angular.module("ungdungAngularjs", []);
```

```
    ungdungAngularjs.controller('nhanvienController', function($scope) {  
        $scope.luong = 2000;  
        $scope.ngaytinhluong = new Date();  
        $scope.nhanvienid = 3301;  
    });  
  
</script>  
</body>  
</html>
```

Ví dụ sử dụng vị trí của Trình duyệt

```
<html>  
<head>  
    <title>Vi du Da ngon ngu (i18n) trong AngularJS</title>  
</head>  
<body>  
    <h2>Ung dung AngularJS</h2>  
    <div ng-app="ungdungAngularjs" ng-controller="nhanvienController">  
        {{luong | currency }} <br/><br/>  
        {{ngaytinhluong | date }} <br/><br/>  
        {{nhanvienid | number }}  
    </div>  
    <script  
src="http://ajax.googleapis.com/ajax/libs/angularjs/1.3.14/angular.min.js"></script>  
    <!-- <script src="https://code.angularjs.org/1.3.14/i18n/angular-locale_da-dk.js"></script>  
-->  
    <script>
```

```
var ungdungAngularjs = angular.module("ungdungAngularjs", []);

ungdungAngularjs.controller('nhanvienController', function($scope) {
    $scope.luong = 2000;
    $scope.ngaytinhluong = new Date();
    $scope.nhanvienid = 3301;
});

</script>
</body>
</html>
```

← → ↻ 🏠 ⓘ file:///E:/FPT_FPT%20Poly/JavaScript%201/Buoi%2016/Đa%20ngôn%20ngữ%20(i18n)_AngularJS/Vidu3.html

Ung dung AngularJS

\$2,000.00

Apr 20, 2020

3,301