AngularJS Includes

- With AngularJS, you can include HTML from an external file using the **ng-include** directive:

- <div ng-include="'myFile.htm'"></div>

What is Angular JS Service

- It provides us method to keep data across the lifetime of the angular app
- It provides us method to communicate data across the controllers in a consistent way
- This is **a singleton object** and it gets **instantiated only once per application**
- It is used to organize and share data and functions across the application
- ***"Service instance gets created when applications components need it"***

AngularJS Services

- **Services** are JavaScript functions, which are responsible to perform only specific tasks.
- It is **a function, or object, that is available for, and limited to, your AngularJS application.**
- They are individual entities which are maintainable and testable.
- The controllers and filters can call them on requirement basis.
- Services are normally injected using the dependency injection mechanism of AngularJS.
- AngularJS has about 30 built-in services.
- For example →$http, $route, $window, $location, etc.
- The inbuilt services are always prefixed with **$ symbol**.

$location service

- The **$location** service has methods which return information about the location of the current web page:

```
var app = angular.module('myApp', []);
app.controller('customersCtrl', function($scope, $location) {
    $scope.myUrl = $location.absUrl();
});
```

- $location service is passed in to the controller as an argument. In order to use the service in the controller, it must be defined as a dependency.
- absUrl(); url([url]); protocol(); host(); port(); path([path]); search(search, [paramValue]); hash([hash]); replace(); state([state]);

Why to use service?

- $location service, it seems like you could use objects that are already in the DOM, like the window.location object, and
- you could, but it would have some limitations, at least for your AngularJS application.

- AngularJS constantly supervises your application, and for it to handle changes and events properly.
- AngularJS prefers that you use the $location service instead of the window.location object.

The $timeout Service

- The $timeout service is AngularJS' version of the window.setTimeout function.

$timeout([fn], [delay], [invokeApply], [Pass]);

```
var app = angular.module('myApp', []);
app.controller('myCtrl', function($scope, $timeout) {
  $scope.myHeader = "Hello World!";
  $timeout(function () {
    $scope.myHeader = "How are you today?";
  }, 2000);
});
```

The $interval Service

- The $interval service is AngularJS' version of the window.setInterval function.

$interval(fn, delay, [count], [invokeApply], [Pass]);

```
var app = angular.module('myApp', []);
app.controller('myCtrl', function($scope, $interval) {
  $scope.theTime = new Date().toLocaleTimeString();
  $interval(function () {
      $scope.theTime = new Date().toLocaleTimeString();
  }, 1000);
});
```

Creating custom angular service

There are two ways to create a service –**Factory** and **Service**

- **Factory method**
  - The most common way to create a service is by using the Module’s Factory API.
  - Use the factory method to create an object, add properties to it and return the same object.
  - Later it can be injected to the components like controller, service, filter or directive

```
var mainApp = angular.module("mainApp", []);
mainApp.factory('MathService', function() {
 var factory = {};
factory.multiply = function(a, b) { return a * b } return factory; });
```

Creating custom angular service

- **Service method**
  - This is instantiated with the new keyword.
  - It can be provided with an instance of the function passed to the service.
  - This object instance becomes the service object that AngularJS registers and is injected to the required components.
  - Use this keyword to add properties and functions to this service object.
  - Unlike factory method, this does not return anything.

```
var mainApp = angular.module("mainApp", []);
mainApp.service('CalcService', function(MathService) {
this.square = function(a) {
 return MathService.multiply(a,a); } });
```

The $http Service

- The $http service is one of the most common used services in AngularJS applications. The service makes a request to the server, and lets your application handle the response.

- Use the $http service to request data from the server:

Methods of $http service

There are several shortcut methods:

- .get()
- .delete()
- .get()
- .head()
- .jsonp()
- .patch()
- .post()
- .put()

The $http Service -Properties

- The response from the server is an object with these properties:
  - **.config** the object used to generate the request.
  - **.data** a string, or an object, carrying the response from the server.
  - **.headers** a function to use to get header information.
  - **.status** a number defining the HTTP status.
  - **.statusText** a string defining the HTTP status.

Ng-view

- Your application needs a container to put the content provided by the routing.
- This container is the ng-view directive.
- There are three different ways to include the ng-view directive in your application:

- 1. <div ng-view></div>
- 2. <ng-view></ng-view>
- 3. <div class="ng-view"></div>

- Applications can only have one ng-view directive, and this will be the placeholder for all views provided by the route.

AngularJS Routing

- The **ngRoute** module helps your application to become a Single Page Application(SPA)

- If you want to navigate to different pages in your application, but you also want the application to be a SPA (Single Page Application), with no page reloading, you can use the **ngRoute** module.

- The ngRoute module *routes* your application to different pages without reloading the entire application.

- It will load the relevant data and HTML snippet instead of fetching the entire HTML again and again.
- When we are using ngRoute of AngularJS the browser **does not make any additional requests**

SPA using ngRoute module steps

1. To make your applications ready for routing, you must include the AngularJS Route module:

<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.0/**angular-route.js**"></script>

2. Then you must add the ngRoute as a dependency in the application module:

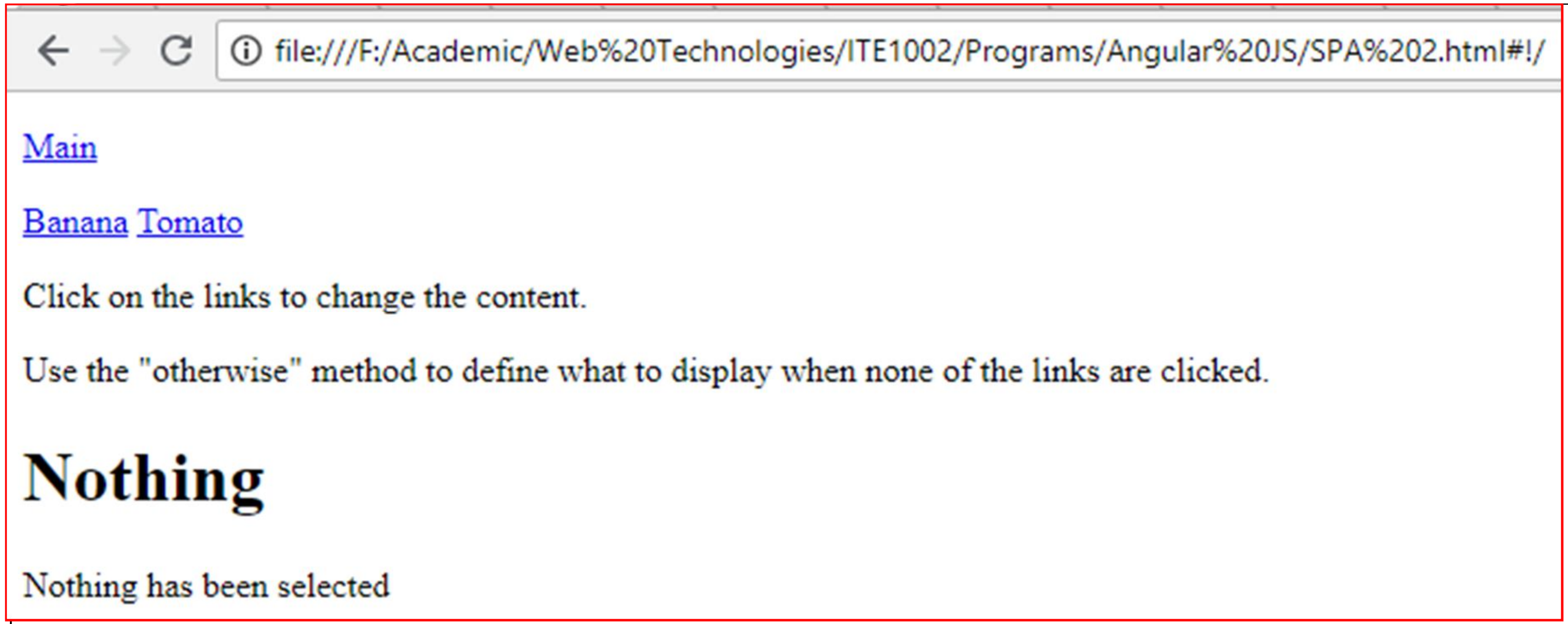
    var app = angular.module("myApp", ["ngRoute"]);

3. Now your application has access to the route module, which provides the $routeProvider.
4. Use the $routeProvider to configure **different routes** in your application.
5. Define the $routeProvider using the **config** method of your application. Work registered in the config method will be performed when the application is loading.
6. Your application needs a container to put the content provided by the routing. This container is the **ng-view** directive.

Single Page Application example-

```
<!DOCTYPE html><html><script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.0/angular.min.js"
> </script>
<script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.0/angular-
route.js"></script>
<body ng-app="myApp">
<p><a href="#/!">Main</a></p>
<a href="#!banana">Banana</a>
<a href="#!tomato">Tomato</a>
<p>Click on the links to change the content.</p>
<p>Use the "otherwise" method to define what to display when none of the
links are clicked.</p>
```

```
<div ng-view></div><script>
var app = angular.module("myApp", ["ngRoute"]);
app.config(function($routeProvider) {
    $routeProvider
    .when("/banana", {
        template: "<h1>Banana</h1><p>Bananas contain around
75% water.</p>"    })
    .when("/tomato", {
        template : "<h1>Tomato</h1><p>Tomatoes contain around
95% water.</p>"    })
    .otherwise({
        template : "<h1>Nothing</h1><p>Nothing has been
selected</p>"
    });});</script></body></html>
```



```
file:///F:/Academic/Web%20Technologies/ITE1002/Programs/Angular%20JS/SPA%202.html#!/
```

Main

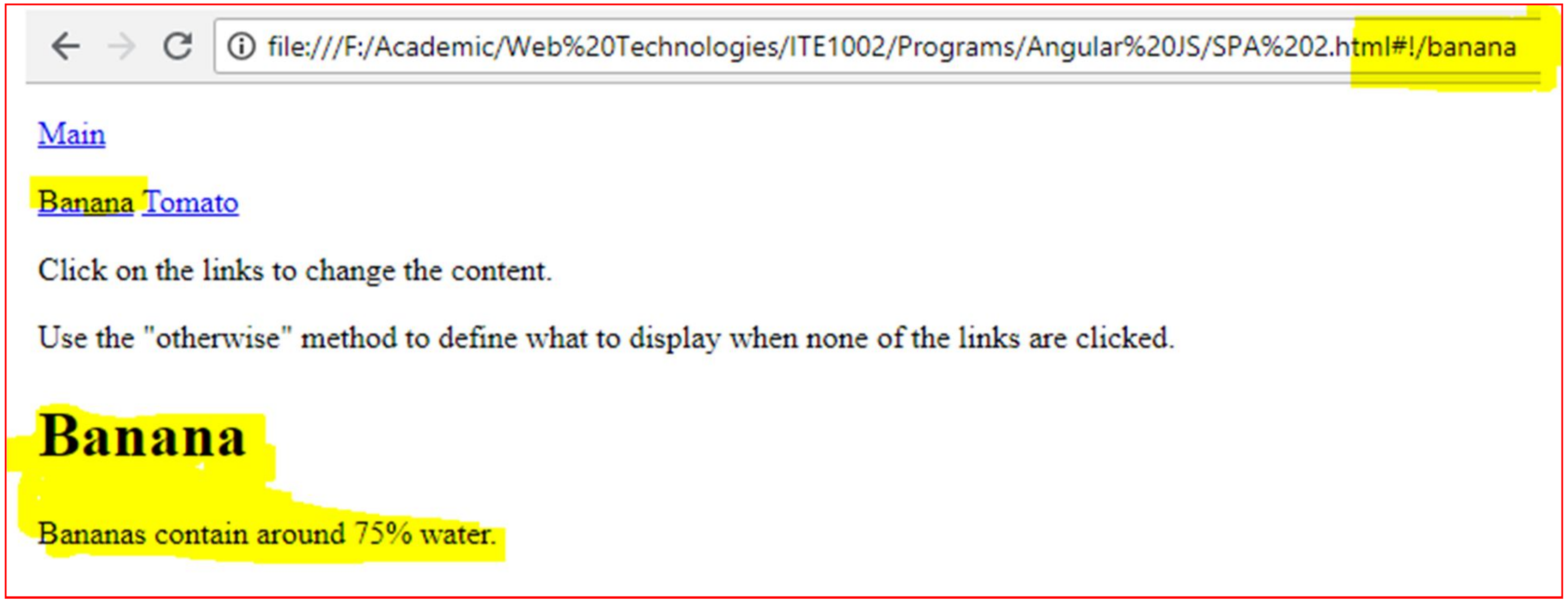
Banana Tomato

Click on the links to change the content.

Use the "otherwise" method to define what to display when none of the links are clicked.

Nothing

Nothing has been selected



```
file:///F:/Academic/Web%20Technologies/ITE1002/Programs/Angular%20JS/SPA%202.html#!/banana
```

Main

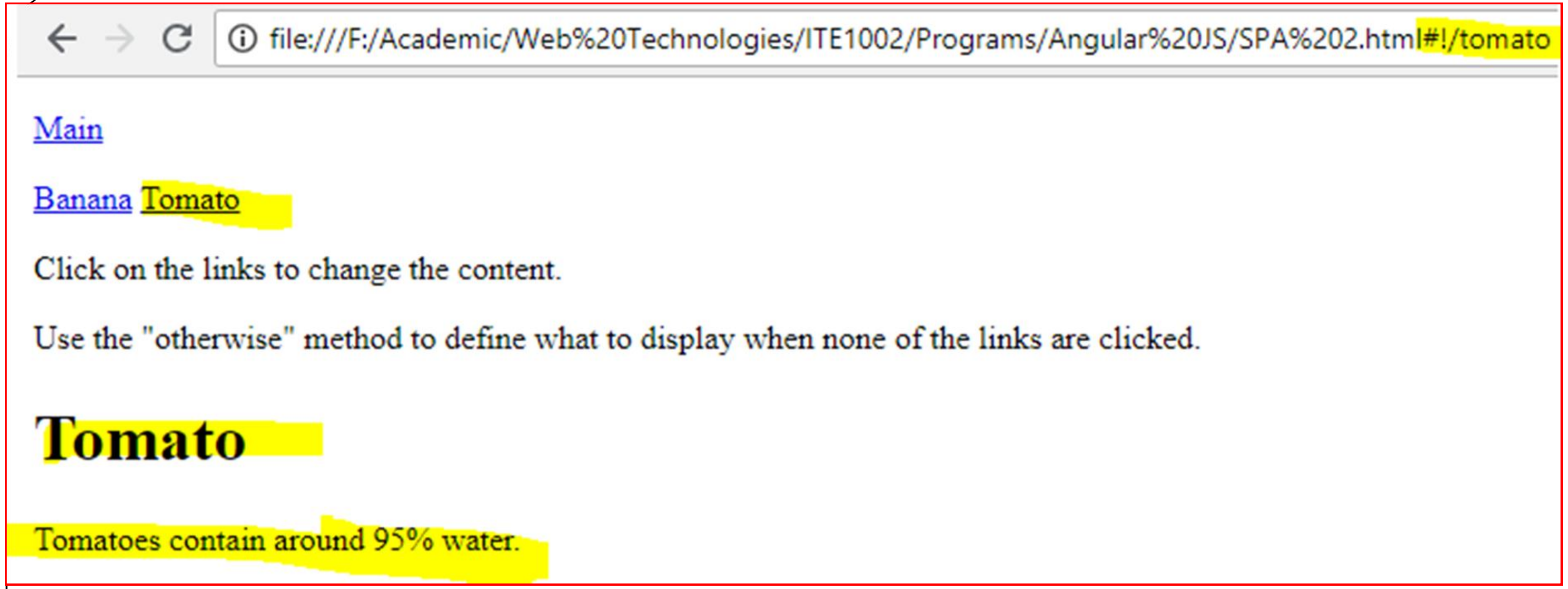
Banana Tomato

Click on the links to change the content.

Use the "otherwise" method to define what to display when none of the links are clicked.

Banana

Bananas contain around 75% water.


file:///F:/Academic/Web%20Technologies/ITE1002/Programs/Angular%20JS/SPA%202.html#!/tomato

Main

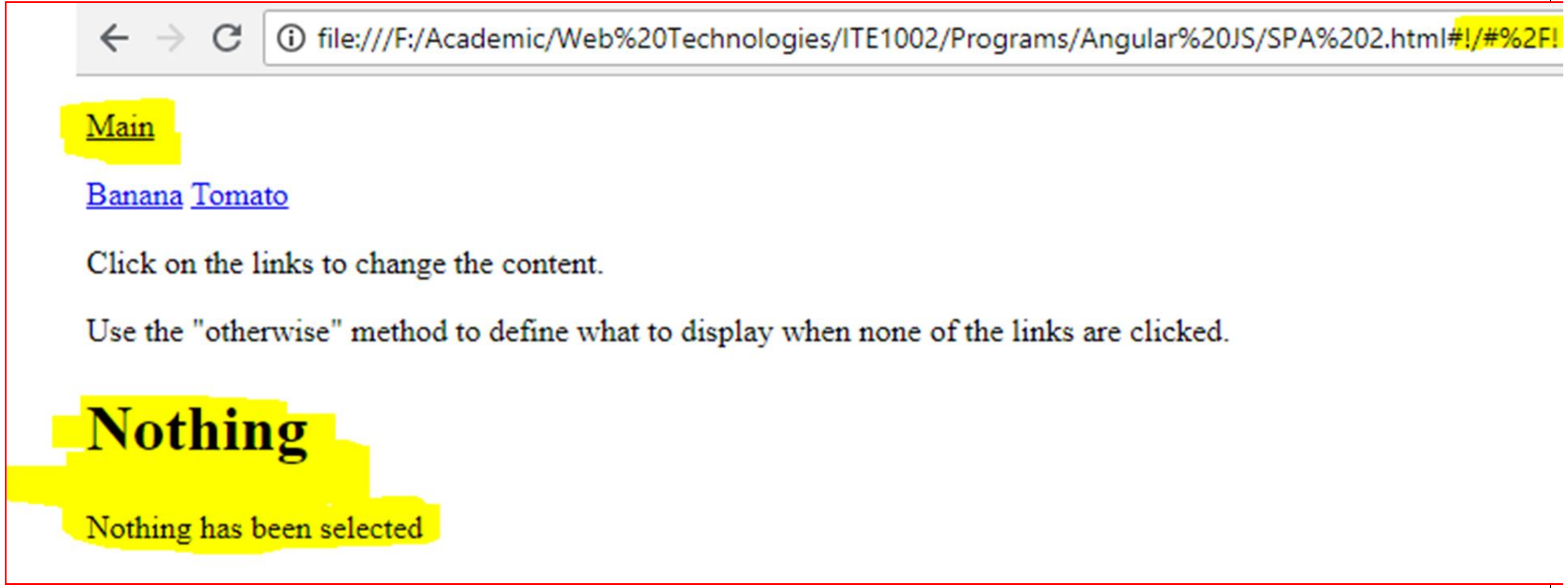
Banana Tomato

Click on the links to change the content.

Use the "otherwise" method to define what to display when none of the links are clicked.

Tomato

Tomatoes contain around 95% water.


file:///F:/Academic/Web%20Technologies/ITE1002/Programs/Angular%20JS/SPA%202.html#!/#%2F!

Main

Banana Tomato

Click on the links to change the content.

Use the "otherwise" method to define what to display when none of the links are clicked.

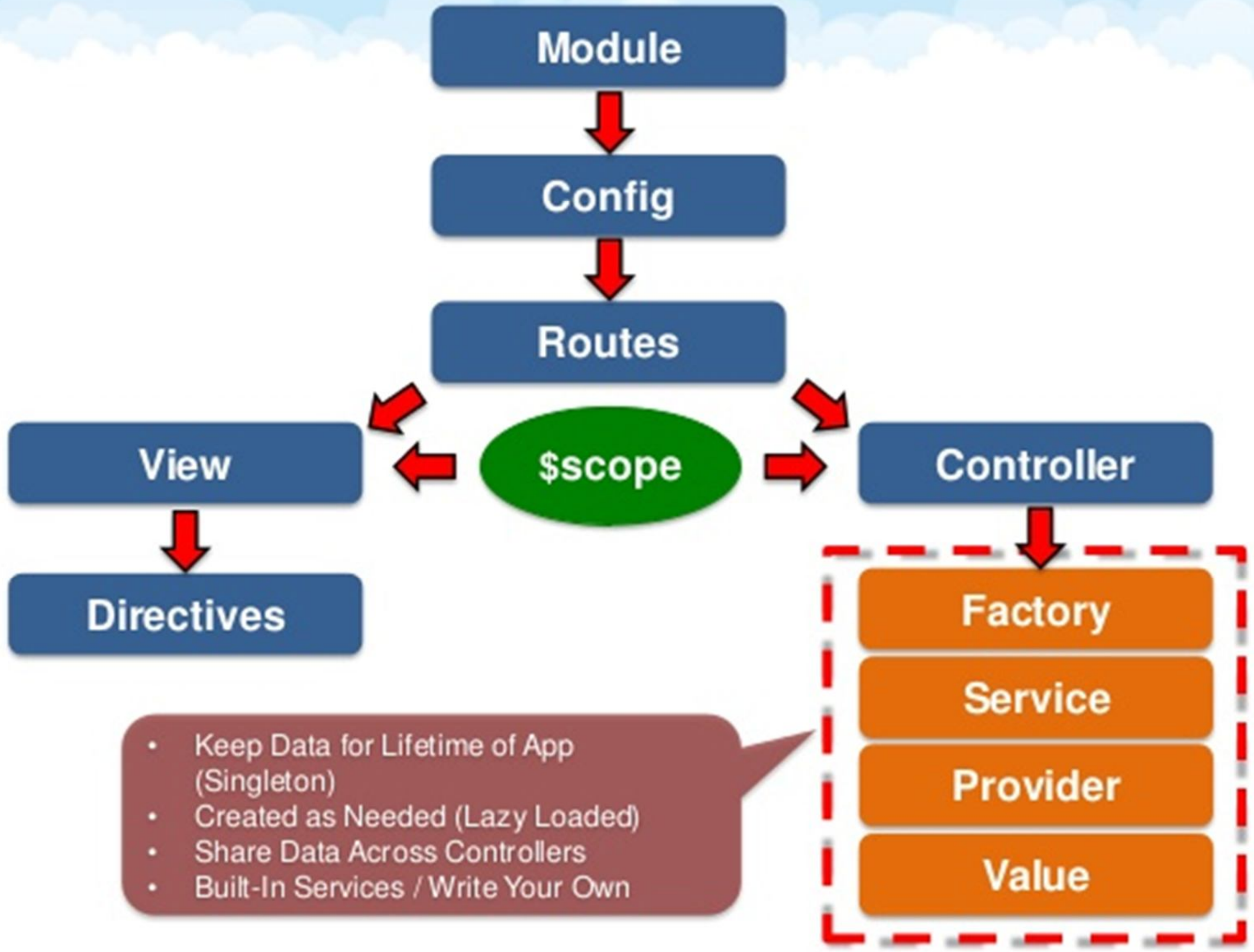
Nothing

Nothing has been selected

```
<!DOCTYPE html><html><script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/ang
ular.min.js"></script>
<script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/ang
ular-route.js"></script>
<body ng-app="myApp">
<p><a href="#/!">Main</a></p>
<a href="#!Home">Home</a>
<a href="#!Blog">Blog</a>
<a href="#!About">About</a>
<div ng-view></div>
```

```
<script>var app = angular.module("myApp", ["ngRoute"]);
app.config(function($routeProvider) {    $routeProvider
    .when("/", {
        templateUrl : "SPA1.html"    })
    .when("/Home", {
        templateUrl : "SPA1_Home.html"    })
    .when("/Blog", {
        templateUrl : "SPA1_Blog.html"    })
    .when("/About", {
        templateUrl : "SPA1_About.html"
    });});</script><p>Click on the links to navigate to "Home
page", "Blog page", "About", or back to "main
page"</p></body></html>
```

Anatomy of an Angular App



- Keep Data for Lifetime of App (Singleton)
- Created as Needed (Lazy Loaded)
- Share Data Across Controllers
- Built-In Services / Write Your Own