

AngularJS Includes

Topics : [AngularJS](#)

Written on [January 09, 2024](#)

In AngularJS, the `ngInclude` directive is commonly used to include external HTML templates into your application. This can be useful for modularizing your application and reusing code across different views. Here's a brief guide on how to use `ngInclude`:

Basic Usage:

Suppose you have an HTML template file named "template.html" that you want to include in another HTML file:

```
<!-- template.html -->
<div>
<h2>This is the included template</h2>
<!-- Content of the included template goes here -->
</div>
```

Now, you can use `ngInclude` in your main HTML file to include this template:

```
<!DOCTYPE html>
<html lang="en" ng-app="myApp">
<head>
<meta charset="UTF-8">
<title>AngularJS Includes</title>
<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
</head>
<body ng-controller="IncludeController">
<div ng-include="template.html"></div>

<script>
angular.module('myApp', []).controller('IncludeController', function($scope) {
// Controller logic goes here
});
</script>
</body>
</html>
```

In this example:

- The `ng-include` directive is used to include the "template.html" file.
- The path to the template is specified as a string (e.g., 'template.html').

Dynamic Includes:

You can use a variable in the `ng-include` directive to dynamically include different templates based on some condition:

```
<div ng-include="templateUrl"></div>
```

In the controller:

```
angular.module('myApp').controller('IncludeController', function($scope) {  
  // Conditionally set the templateUrl based on some logic  
  $scope.templateUrl = 'template1.html';  
});
```

Multiple Includes:

You can use multiple `ng-include` directives to include multiple templates:

```
<div ng-include="template1.html"></div>  
<div ng-include="template2.html"></div>
```

Include with a Scope:

You can use the `ng-include` directive with a specified scope using the `ng-controller` directive:

```
<div ng-include="'template.html'" ng-controller="IncludeController"></div>
```

In this case, the `IncludeController` will be associated with the included template.

Include with Variables:

You can pass variables to the included template by using the `ng-include`'s `ng-init` attribute:

```
<div ng-include="'template.html'" ng-init="variable='Hello from  
parent'"></div>
```

In the included template:

```
<div>  
<p>{{ variable }}</p>  
</div>
```

Using ng-include with ng-repeat:

You can use `ng-include` inside an `ng-repeat` to include templates dynamically for each item:

```
<div ng-repeat="item in items" ng-include="'template.html'"></div>
```

In this way, you can dynamically include templates based on the items in your data array.