

AngularJS Routing

Topics : [AngularJS](#)

Written on [January 09, 2024](#)

In AngularJS, routing allows you to build single-page applications (SPAs) where the content is dynamically loaded and changed based on the URL. AngularJS provides a built-in module called `ngRoute` that facilitates client-side routing.

Step 1: Include AngularJS and ngRoute:

First, include the AngularJS and `ngRoute` scripts in your HTML file. You can use CDN links or download the scripts locally:

```
<!DOCTYPE html>
<html lang="en" ng-app="myApp">
<head>
<meta charset="UTF-8">
<title>AngularJS Routing</title>
<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular-route.min.js"></script>
</head>
<body>
<!-- Content will be loaded dynamically here -->
<div ng-view></div>

<script>
// Define your AngularJS app and include the ngRoute module
var app = angular.module('myApp', ['ngRoute']);
</script>
</body>
</html>
```

Step 2: Configure Routes:

Configure the routes for your application using the `config` method of your AngularJS app. Define a route for each view and specify the controller and template to be used:

```
app.config(function($routeProvider) {
$routeProvider
.when('/', {
templateUrl: 'views/home.html',
```

```

controller: 'HomeController'
})
.when('/about', {
  templateUrl: 'views/about.html',
  controller: 'AboutController'
})
.when('/contact', {
  templateUrl: 'views/contact.html',
  controller: 'ContactController'
})
.otherwise({
  redirectTo: '/'
});
});

```

In this example, routes for the home, about, and contact pages are defined. The `otherwise` method sets the default route to redirect to the home page.

Step 3: Create Views and Controllers:

Create separate HTML views and controllers for each route:

```

<!-- home.html -->
<h2>Home Page</h2>
<p>Welcome to the home page!</p>

```

```

<!-- about.html -->
<h2>About Page</h2>
<p>Learn more about us.</p>

```

```

<!-- contact.html -->
<h2>Contact Page</h2>
<p>Get in touch with us.</p>

```

```

// Define controllers for each view
app.controller('HomeController', function($scope) {
  // Controller logic for the home page
});

```

```

app.controller('AboutController', function($scope) {
  // Controller logic for the about page
});

```

```

app.controller('ContactController', function($scope) {
  // Controller logic for the contact page
});

```

Step 4: Set up Links for Navigation:

Use the `ng-href` or `ng-click` directives to create links for navigating between views:

```
<!-- navigation.html -->
<nav>
<a ng-href="#/">Home</a>
<a ng-href="#/about">About</a>
<a ng-href="#/contact">Contact</a>
</nav>
```

Step 5: Include Navigation in Your Main HTML:

Include the navigation in your main HTML file:

```
<body>
<!-- Content will be loaded dynamically here -->
<div ng-view></div>

<!-- Navigation links -->
<div ng-include="views/navigation.html"></div>

<script>
// Define your AngularJS app and include the ngRoute module
var app = angular.module('myApp', ['ngRoute']);

// Configure routes (as shown in Step 2)

// Define controllers for each view (as shown in Step 3)
</script>
</body>
```

Step 6: Run Your Application:

Finally, start your AngularJS application. Open your HTML file in a web browser, and you should see the dynamic loading of content as you navigate between routes.

© Copyright **Aryatechno**. All Rights Reserved. Written tutorials and materials by [Aryatechno](#)