

# AngularJS Modules

Topics : [AngularJS](#)

Written on [January 09, 2024](#)

In AngularJS, a module is a container for different parts of your application, including controllers, services, directives, filters, and more. Modules help in organizing and structuring your code, making it more maintainable and scalable. They also facilitate the separation of concerns within your application.

1. **Module Creation:** You create a module using the `angular.module` function. The function takes the name of the module and an optional array of dependencies. Dependencies are other modules that the current module depends on.

Example:

```
var myApp = angular.module('myApp', []);
```

In this example, a module named 'myApp' is created with no dependencies.

2. **Dependencies:** You can specify dependencies when creating a module by providing an array of dependency names as the second parameter. These dependencies are other modules that the current module relies on.

Example:

```
var myApp = angular.module('myApp', ['ngRoute', 'ngResource']);
```

In this case, the 'myApp' module depends on the 'ngRoute' and 'ngResource' modules.

3. **Adding Components to Modules:** Once a module is created, you can add various components to it, such as controllers, services, directives, filters, etc. This is typically done using the `controller`, `service`, `directive`, and other functions provided by the module object.

Example:

```
myApp.controller('myController', function($scope) {  
  // Controller logic goes here  
});
```

This adds a controller named 'myController' to the 'myApp' module.

4. **Module Initialization:** After defining the components within a module, you need to bootstrap the module to start the AngularJS application. This is typically done by adding the `ng-app` directive to an HTML element or by manually bootstrapping the application using the `angular.bootstrap` function.

Example (using `ng-app` directive):

```
<html ng-app="myApp">
<!-- Your application content goes here -->
</html>
```

Example (manual bootstrapping):

```
angular.element(document).ready(function() { angular.bootstrap(document,
['myApp']); });
```

5. **Module Configuration and Run Blocks:** Modules can have configuration and run blocks that are executed during the configuration and run phases of the AngularJS application lifecycle, respectively. Configuration blocks are used to configure providers before the application runs, while run blocks are used to perform initialization tasks.

Example:

```
myApp.config(function($routeProvider) {
// Configuration logic goes here
});
myApp.run(function($rootScope) {
// Run-time initialization logic goes here
});
```

In this example, the `config` block configures the route provider, and the `run` block initializes the root scope.