ARYATECHNO

AngularJS MVC Architecture

Topics : AngularJS
Written on January 08, 2024

AngularJS follows the Model-View-Controller (MVC) architectural pattern, although in AngularJS terminology, the components are often referred to as the Model-View-ViewModel (MVVM) pattern. The core idea behind MVC/MVVM is to separate concerns and organize the code into distinct components for better maintainability, testability, and scalability.

Here's an overview of the MVC/MVVM architecture in AngularJS:

1. **Model:**
   - **Definition:** The model represents the application's data and business logic. It is responsible for managing and storing data and responding to requests for information.
   - **In AngularJS:** In AngularJS, the model is typically represented by JavaScript objects or data structures. This could include data retrieved from APIs, user inputs, or any other form of application state.
2. **View:**
   - **Definition:** The view is responsible for displaying the user interface and presenting data to the user. It observes the model for changes and updates the UI accordingly.
   - **In AngularJS:** The view in AngularJS is represented by HTML templates. These templates contain directives that bind to the model and display data dynamically. AngularJS uses two-way data binding to keep the view and model in sync automatically.
3. **Controller/ViewModel:**
   - **Definition:** The controller (or ViewModel in AngularJS) acts as an intermediary between the model and the view. It handles user inputs, updates the model, and manipulates the view.
   - **In AngularJS:** AngularJS controllers (or Angular components in modern Angular versions) are responsible for managing the application's behavior. They define functions and properties that are bound to the view. Controllers interact with services to handle business logic and retrieve or update data.

Here's a simple example in AngularJS to illustrate these concepts:

```
<!DOCTYPE html>
<html lang="en" ng-app="myApp">
<head>
<meta charset="UTF-8">
```

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>AngularJS MVC Example</title>
<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
</head>
<body ng-controller="myCtrl">

<h1>{{ greeting }}</h1>
<input type="text" ng-model="name" placeholder="Enter your name">
<button ng-click="updateGreeting()">Update Greeting</button>

<script>
var app = angular.module('myApp', []);

app.controller('myCtrl', function($scope) {
$scope.name = 'User';
$scope.greeting = 'Hello, ' + $scope.name + '!';

$scope.updateGreeting = function() {
$scope.greeting = 'Hello, ' + $scope.name + '!';
};
});
</script>

</body>
</html>
```

In this example:

- **Model:** `$scope.name` represents the data or state.
- **View:** HTML elements with `ng-model` and `{{}}` (expression binding) represent the UI.
- **Controller/ViewModel:** The AngularJS controller defines functions like `updateGreeting` to handle user interactions.