AngularJS Services

Topics : AngularJS
Written on January 09, 2024

In AngularJS, services are reusable, singleton objects that perform specific tasks or provide functionality across different parts of an application. Services are a fundamental building block in the AngularJS architecture and are designed to promote code organization, modularity, and separation of concerns. Here's an overview of AngularJS services:

Types of Services:

AngularJS provides several types of services out of the box:

1. **Built-in Services:**
    - AngularJS comes with a set of built-in services, such as `$http` for making HTTP requests, `$rootScope` for the top-level scope, and `$routeParams` for accessing route parameters.
2. **Custom Services:**
    - Developers can create custom services by registering them with the AngularJS module. These services can encapsulate business logic, data manipulation, or any other functionality that needs to be shared across controllers, directives, or other components.

Creating Custom Services:

```
// Define a custom service named 'myDataService'
angular.module('myApp').service('myDataService', function() {
this.getData = function() {
// Service logic to fetch or provide data
return ['Item 1', 'Item 2', 'Item 3'];
};
});
```

Using Services in Controllers:

```
angular.module('myApp').controller('MyController', function($scope, myDataService) {
$scope.items = myDataService.getData();
});
```

In this example, the `myDataService` is injected into the `MyController` controller, allowing the controller to use the functionality provided by the service.

Singleton Nature:

Services in AngularJS are singletons, meaning that there is only one instance of each service in the application. When a service is injected into multiple controllers or components, they all share the same instance of that service. This ensures that data and state are consistent across different parts of the application.

Dependency Injection:

AngularJS uses dependency injection to provide services to controllers, directives, and other components. When a component needs access to a service, AngularJS automatically injects the required service as a parameter.

```
// Dependency injection example in a controller
angular.module('myApp').controller('MyController', function($scope, myDataService) {
// Controller logic using the myDataService
});
```

Common Built-in Services:

1. **`$http`:**
    - Used for making HTTP requests. It provides methods like `get`, `post`, `put`, and `delete`.
2. **`$rootScope`:**
    - The top-level scope that is shared among all AngularJS components. While it's available, it's generally recommended to avoid excessive use of `$rootScope` to prevent potential scope-related issues.
3. **`$location`:**
    - Provides information about the current URL and allows navigation.
4. **`$routeParams`:**
    - Allows access to parameters in the URL route.

Custom Services vs. Factories:

In addition to the `service` method, AngularJS also provides the `factory` method for creating custom services. While both `service` and `factory` achieve similar goals, they have some differences in their implementation details. Generally, `factory` allows more flexibility in creating services with complex initialization logic.

Example of a Factory:

```
// Define a custom service using the factory method
angular.module('myApp').factory('myDataService', function() {
var data = ['Item 1', 'Item 2', 'Item 3'];

return {
```

```
getData: function() {
// Service logic to fetch or provide data
return data;
}
};
});
```