

Introduction to AngularJS

AngularJS is a popular open-source framework that simplifies web development by creating interactive single-page applications (SPAs). Unlike traditional websites that load new pages for each click, SPAs offer a smoother user experience by updating content on the same page.

AngularJS makes this possible by transforming static HTML into dynamic content that adapts to user interactions.

Features like data binding and dependency injection streamline development, saving you time and effort. With regular updates and a large community, AngularJS ensures your web applications stay modern and efficient.

AngularJS is a JavaScript Framework

AngularJS is a JavaScript framework that can be used in 2 ways given below:

- . Using Angular CLI and NPM**
- . Using CDN Link in script tag**

```
<script  
src="https://cdnjs.cloudflare.com/ajax/libs/angular.js/1.8.3/angular.min.js"  
    integrity="sha512-  
KZmyTq3PLx9EZI0RHSbHQuXtrvdJ+m35tuOi  
wlcZfs/rE7NZv29ygNA8SFCkMXTnYZQK2OX0
```

```
Gm2qKGfvWEtRXA=="  
    crossorigin="anonymous"  
referrerpolicy="no-referrer">  
</script>
```

Why Choose AngularJS?

- 1. Easy to Work With:** AngularJS requires only basic knowledge of HTML, CSS, and JavaScript. You don't need to be an expert; just bring your curiosity and creativity.
- 2. Time-Saving Components:** AngularJS allows you to work with reusable components, saving time and reducing unnecessary code. Components are the building blocks of your application.
- 3. Ready-to-Use Templates:** AngularJS leverages plain HTML templates, which it compiles and makes ready for use. No complex setup—just write your HTML and let AngularJS do the heavy lifting.
- 4. Directives:** AngularJS extends HTML with custom elements and attributes called directives. These enable you to create reusable components and define custom behaviors for your application. Directives make it easier to manipulate the DOM, handle events, and encapsulate complex UI logic within a single component.

Steps to Install Angular JS using Angular CLI

Step 1: Install the Angular CLI

```
npm install -g @angular/cli
```

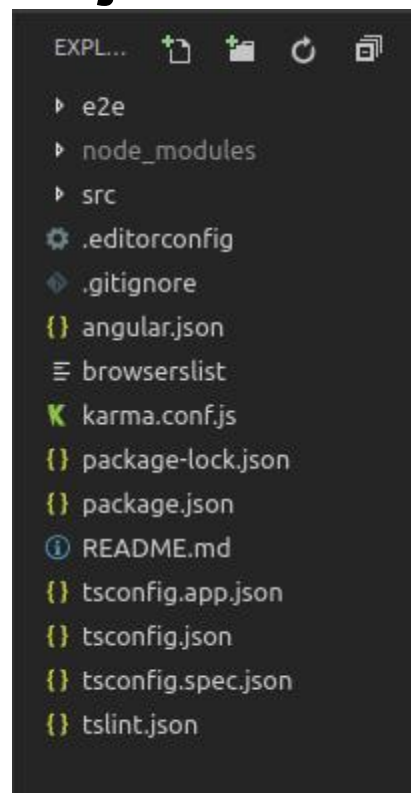
Step-2: Initialize new project using the below command

```
ng new my-app
```

Step-3: Go to your project directory

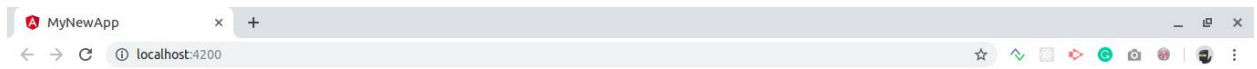
```
cd my-app
```

Project Structure:



Step-4: Run the application using the following command in the terminal

```
ng serve
```



Welcome to myNewApp!



Here are some links to help you start:

- [Tour of Heroes](#)
- [CLI Documentation](#)
- [Angular blog](#)

Steps to Create First Angular App

Example: This example illustrates the basic Hello World app using Angular JS. JavaScript

// Filename - app.component.ts

```
import { Component } from '@angular/core';
```

```
@Component({  
  selector: 'my-app',  
  template: '<h1>Hello World!</h1>',  
  styleUrls: ['./app.component.css']  
})
```

```
export class AppComponent {  
}
```

Output:

Hello World!

History of AngularJS

AngularJS was first developed in 2008-2009 by Miško Hevery and Adam Abrons at Brat Tech LLC. Its original purpose was to provide software for an online JSON storage service and simplify the development of enterprise applications that were valued by the megabyte. Currently, AngularJS is maintained by Google. Version 1.6, which introduced the concept of component-based application architecture, was the first release. This version removed the Sandbox, which was responsible for security, despite the fact that it had several vulnerabilities that could bypass it.

Key Features

Model-View-Controller (MVC) Architecture

- **Model:** Manages data and logic, responding to requests from the view and instructions from the controller.
- **View:** Displays application data to users.

- **Controller: Orchestrates communication between the model and view, updating the model based on user interactions.**
- 2. Filters and Data Transformation:**
 - **AngularJS provides powerful filters for transforming data before displaying it in templates. Common filters include filter, orderBy, and currency.**
 - **You can also create custom filters tailored to your application's needs.**
- 3. Routing and Single-Page Applications (SPAs):**
 - **Implement client-side routing to create SPAs.**
 - **Handle route parameters and navigation seamlessly.**
- 4. Services and Dependency Injection:**
 - **Services are reusable components that provide specific functionality.**
 - **Dependency injection ensures loose coupling between components, making your code more maintainable.**
- 5. Custom Directives and Components:**
 - **Build reusable components using custom directives.**
 - **Isolate scope and communicate between components.**

6. Testing AngularJS Applications:

- . Write unit tests using Jasmine and Karma.**
- . Explore end-to-end testing with Protractor.**

7. Advanced Topics:

- . Dive into promises and asynchronous programming.**
- . Interact with RESTful APIs.**
- . Optimize performance and follow best practices.**

Applications of AngularJS

1. Dynamic Forms:

- . AngularJS excels at creating dynamic forms with real-time validation and feedback.**

2. Real-Time Dashboards:

- . Build interactive dashboards that update in real time based on user interactions.**

3. Collaborative Apps:

- . AngularJS is perfect for chat applications and collaborative document editing.**

4. E-Commerce Platforms:

- . Create seamless shopping experiences with AngularJS-powered product catalogs and checkout processes.**

Advantages of AngularJS

- It facilitates the Two-way Binding that helps to render correspondingly the changes made to the view or the model.**
- It helps to create a responsive web application, along with providing the prototyping that can be utilized to load the application faster.**
- It uses the concept of directive that helps to add functionality to the application. For this, the overall length of the code reduces along with discarding the repetition of the code that is specific to perform the particular task.**