Terms Demo

Wednesday, August 2, 2023 6:00 PM

Important terms

TypeScript

Extends JavaScript, turning it into a real programming language. Object oriented, support interfaces, tighter integration with editors, and a lot of cool stuff. If you've used class structure in OpenEdge and used JavaScript, you have the knowledge to understand TypeScript.

Angular CLI

The Angular Command Line Interface allows you to create and manage your Angular project and elements. All the confusing "stuff" is generated automatically within your project when you generate elements.
Example: ng generate component loginscreen

Component

Components are the main building blocks for Angular applications. Each component consists of 3 (OK 4) parts:

- An HTML template that declares what renders on the page
- A TypeScript class that "Holds the Code"
- A CSS style sheet
- A spec .ts file that contains unit test for the component

Service

Like a helper class for logic shared across components. For example, you might create an auth.service.ts file to hold all auth methods.

Dependency Injection

Services are typically created as a singleton and Angular handles injecting it into your module. You define the service as a dependency in the constructor of a Component.

Directive

You can create a directive to govern how an element behaves.

Module:

An NgModule mechanism to group components, directives, and services that are related, in such a way that can be combined with other modules to create an application. Every project has a root module (AppModule).

Lazy Loading

Angular's way of enhances user experience by loading non-essential content in the background and presenting it when the user interacts with the page in some way.

Observables

A lazy-loading promise on steroids great for working with async data requests.

NPM

Both a tool and a huge software registry