migration is of 2 types 1)hybird 2)rewrite

If your project is large and business-critical, use the Hybrid Approach (ngUpgrade) for a smooth transition.

move from controllers to components, migrate services, and replace AngularJS routing.

1.Install Angular and the upgrade module using npm:

npm install @angular/upgrade @angular/core @angular/common @angular/compiler @angular/platform-browser

@angular/platform-browser-dynamic rxjs zone.js --save

2.ensure it is using components instead of controllers

3

Start migrating services (.service() instead of .factory()).

Replace AngularJS routing ($routeProvider) with Angular Router:Rewrite routing using the Angular Router (@angular/router).

Convert directives and pipes.

Eventually, remove AngularJS dependencies and bootstrap with Angular only

4.<!-- AngularJS (Two-Way Binding) -->

<input ng-model="name">

<!-- Angular 2+ (One-Way Binding) -->

<input [value]="name" (input)="name = $event.target.value">

5.🔹 Step 6: Optimize Performance

Replace $scope.$watch with RxJS Observables for state management.

Use lazy loading (loadChildren) for modules

install the ionic

Install Ionic CLI

npm install -g @ionic/cli

Create a New Ionic Project

Choose a Framework (Angular, React, Vue)

Step 4: Navigate to the Project Directory

Step 5: Run the Ionic App Locally- cmd ionic serve

Understanding the Folder Structure in Ionic

Creating Pages, Components, and Other Feature

Add Navigation and Routing`

Step 6: Add Platforms (Android/iOS)-ionic capacitor add android

Step 7: Build and Run the App on a Device

to Build the application we use cmd like

ionic build

to Run on an Android Emulator

ionic capacitor run android

step 8:Install Ionic Native Plugins

npm install @capacitor/camera

**9.Build for Production**: ionic build --prod

--------------------------------------------------

Scaffolding in Ionic refers to the automatic generation of files and components to speed up development

. The Ionic CLI provides commands **to create pages,**

Commad: **ionic generate page PageName**

**2. Generating Components**

Commad: **ionic generate component ComponentName**

**3. Generating Services**

Commad: **ionic generate service ServiceName**

**4. Generating Directives**

**ionic g directive DirectiveName**

**5. Generating Pipes**

**ionic generate pipe PipeName**

**6. Generating Modules**

**ionic generate module ModuleName**

**Lazy load the module in routing:**

**const routes: Routes = [ { path: 'user', loadChildren: () => import('./user/user.module').then(m => m.UserModule) } ];**

What is ion-router-outlet?

it is same as router outlet in angular -it is directive which renders activated route component.

In ionic it is container .It acts as a dynamic area that is replaced by different pages based on the route

In your root component (usually **app.component.html**), you'll add the <**ion-router-outlet></**ion-router-outlet> element where you want the routed pages to be displayed.

**Emulator**

**1.Ionic + Capacitor allows you to emulate and test Angular-based apps in mobile environments (Android/iOS).**

**2.1**. Install Ionic CLI

**npm install -g @ionic/cli**

**2.** Create an Ionic App with Angular

**ionic start myApp blank --type=angular**

**3.** Add Mobile Platforms (**Android**/iOS)

**ionic capacitor add android**

**4.** Run on Emulator

**ionic capacitor run android**

**5. Using Ionic DevApp for Quick Testing**

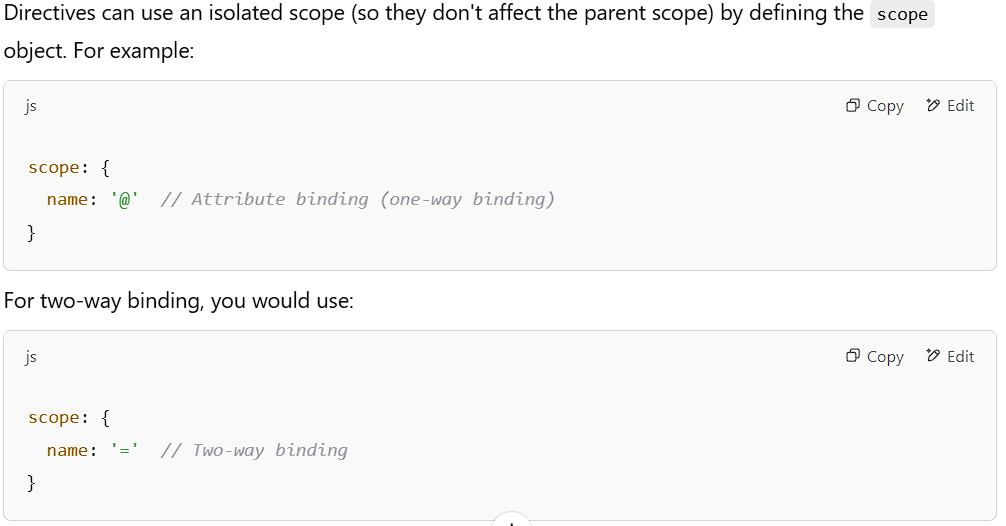
**ionic serve**

**Creating a Custom Directive in AngularJS**

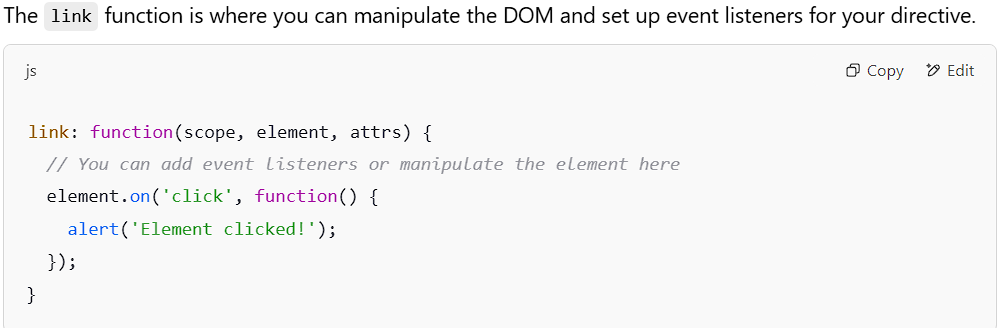
****

****

**4. Using Isolated Scope**

****

**5. Link Function**

****