



DI Injection with angular components



- Dependencies are objects or services required for a class to perform its functions
- DI is an application design pattern used to design Angular applications
- Referred to as a coding pattern
 - Where a class requests dependencies from an external source

Windows taskbar: Type here to search, Hitachi Vantara, 29°C, 22:07, 09-04-2024

Dependency Injection Benefits



- Increases the efficiency and modularity of the application
- Allows the supply of data from an injectable class to a component
- Provides declared dependencies when the class is initiated

Video player controls: 2:42 / 5:07

Dependency Providers



- Configures an injector with a DI token to provide runtime value of dependency
- Helps the injector to create instances of dependencies
- Syntax: providers: [<name of provider>]

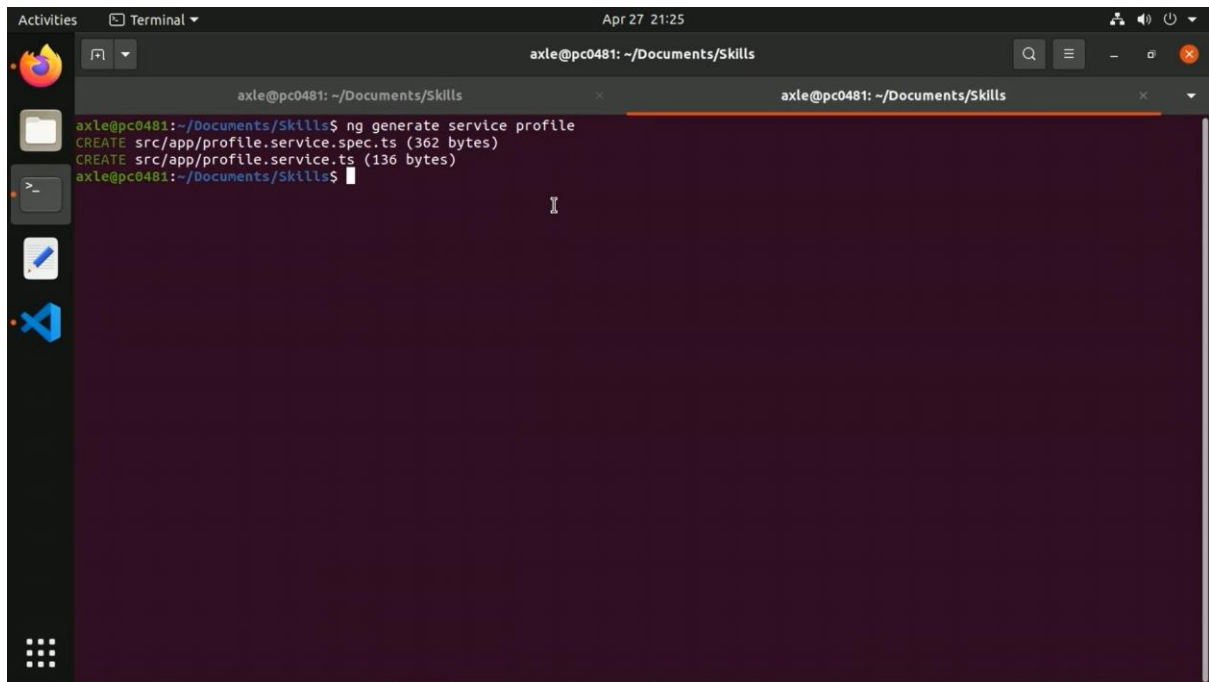
Services in Angular



- Used by components to store and fetch data
- Used to share information amongst unknown classes
- Can be used as provider tokens by injectors to instantiate the class

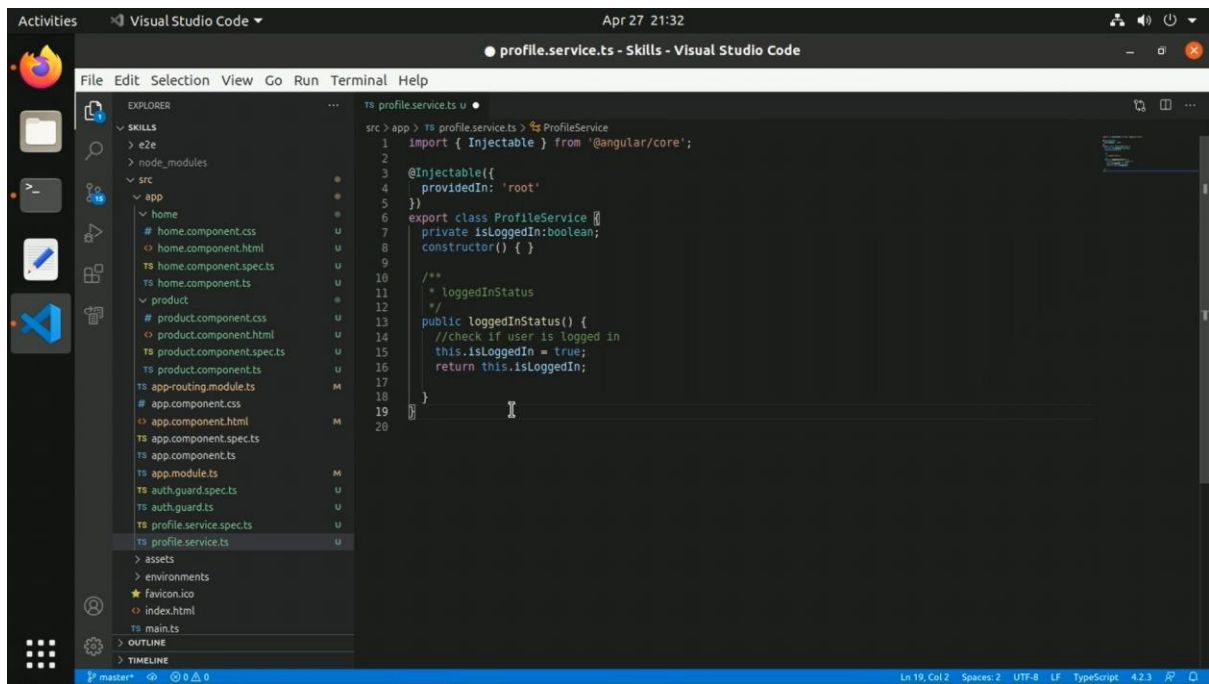
Creating a service in angular

Ng g s profile



A terminal window titled 'axle@pc0481: ~/Documents/Skills' showing the command 'ng generate service profile' and its output. The output indicates that two files were created: 'src/app/profile.service.spec.ts' (362 bytes) and 'src/app/profile.service.ts' (136 bytes). The terminal prompt is 'axle@pc0481:~/Documents/Skills\$'.

```
axle@pc0481:~/Documents/Skills$ ng generate service profile
CREATE src/app/profile.service.spec.ts (362 bytes)
CREATE src/app/profile.service.ts (136 bytes)
axle@pc0481:~/Documents/Skills$
```



A Visual Studio Code editor window titled 'profile.service.ts - Skills - Visual Studio Code' showing the content of the 'profile.service.ts' file. The file is located at 'src > app > TS profile.service.ts'. The code defines a 'ProfileService' class that implements the 'Injectable' interface. It has a private 'isLoggedIn' property and a 'loggedInStatus()' method that returns the 'isLoggedIn' property.

```
src > app > TS profile.service.ts > ProfileService
1  import { Injectable } from '@angular/core';
2
3  @Injectable({
4    providedIn: 'root'
5  })
6  export class ProfileService {
7    private isLoggedIn: boolean;
8    constructor() { }
9
10   /**
11    * isLoggedInStatus
12    */
13   public isLoggedInStatus() {
14     //check if user is logged in
15     this.isLoggedIn = true;
16     return this.isLoggedIn;
17   }
18
19
20
```

Registering a Service in angular

The screenshot shows the Visual Studio Code editor with the file `app.module.ts` open. The Explorer sidebar on the left shows the project structure, including `src/app` and `src/product` folders. The main editor displays the following code:

```
src > app > TS app.module.ts > TS AppModule
1 import { NgModule } from '@angular/core';
2 import { BrowserModule } from '@angular/platform-browser';
3
4 import { AppRoutingModule } from './app-routing.module';
5 import { AppComponent } from './app.component';
6 import { HomeComponent } from './home/home.component';
7 import { ProductComponent } from './product/product.component';
8 import { ProfileService } from './profile.service';
9
10 @NgModule({
11   declarations: [
12     AppComponent,
13     HomeComponent,
14     ProductComponent
15   ],
16   imports: [
17     BrowserModule,
18     AppRoutingModule
19   ],
20   providers: [[ProfileService]],
21   bootstrap: [AppComponent]
22 })
23 export class AppModule { }
24
```

A hover tooltip is visible over the `ProfileService` class in the `providers` array, showing its full path and a list of its methods and properties:

- `ProfileService` (alias) class `ProfileService`
- Methods: `ProcessingInstruction`, `ProgressEvent`, `Promise`, `PromiseRejectionEvent`, `Proxy`
- Properties: `PropertyRead`, `PropertyWrite`, `ProtractorBrowser`, `ProtractorBy`, `ProtractorExpectedConditions`

The screenshot shows the same Visual Studio Code editor with the file `app.module.ts` open. The code is now as follows:

```
src > app > TS app.module.ts > TS AppModule
1 import { NgModule } from '@angular/core';
2 import { BrowserModule } from '@angular/platform-browser';
3
4 import { AppRoutingModule } from './app-routing.module';
5 import { AppComponent } from './app.component';
6 import { HomeComponent } from './home/home.component';
7 import { ProductComponent } from './product/product.component';
8 import { ProfileService } from './profile.service';
9
10 @NgModule({
11   declarations: [
12     AppComponent,
13     HomeComponent,
14     ProductComponent
15   ],
16   imports: [
17     BrowserModule,
18     AppRoutingModule
19   ],
20   providers: [[ProfileService]],
21   bootstrap: [AppComponent]
22 })
23 export class AppModule { }
24
```

The `ProfileService` class has been removed from the `providers` array in the `@NgModule` decorator. The cursor is positioned at the end of line 24.

Service is available for rest of apps

Injecting Services inside an Angular Components

Activities Visual Studio Code Apr 27 22:37

app-routing.module.ts - Skills - Visual Studio Code

File Edit Selection View Go Run Terminal Help

EXPLORER

- SKILLS
- e2e
- node_modules
- src
 - app
 - home
 - home.component.css
 - home.component.html
 - home.component.spec.ts
 - home.component.ts
 - product
 - product.component.css
 - product.component.html
 - product.component.spec.ts
 - product.component.ts
 - app-routing.module.ts
 - app.component.css
 - app.component.html
 - app.component.spec.ts
 - app.component.ts
 - app.module.ts
 - auth.guard.spec.ts
 - auth.guard.ts
 - profile.service.spec.ts
 - profile.service.ts
 - assets
 - environments
 - favicon.ico
 - index.html
 - main.ts
- OUTLINE
- TIMELINE

TS profile.service.ts u TS app-routing.module.ts > ... TS app.module.ts M TS auth.guard.ts u TS app-routing.module.ts M X

```
src > app > TS app-routing.module.ts > ...
1 import { NgModule } from '@angular/core';
2 import { RouterModule, Routes } from '@angular/router';
3 import { HomeComponent } from '../home/home.component';
4 import { ProductComponent } from '../product/product.component';
5 import { AuthGuard } from '../auth.guard';
6
7 const routes: Routes = [
8   { path: '', component: HomeComponent, pathMatch: 'full' },
9   { path: 'home', component: HomeComponent },
10  { path: 'product', canActivate:[AuthGuard] , component: ProductComponent }
11 ];
12
13 @NgModule({
14   imports: [RouterModule.forRoot(routes)],
15   exports: [RouterModule]
16 })
17 export class AppRoutingModule { }
18
```

Ln 18, Col 1 Spaces: 2 UTF-8 LF TypeScript 4.2.3

Activities Visual Studio Code Apr 27 22:39

auth.guard.ts - Skills - Visual Studio Code

File Edit Selection View Go Run Terminal Help

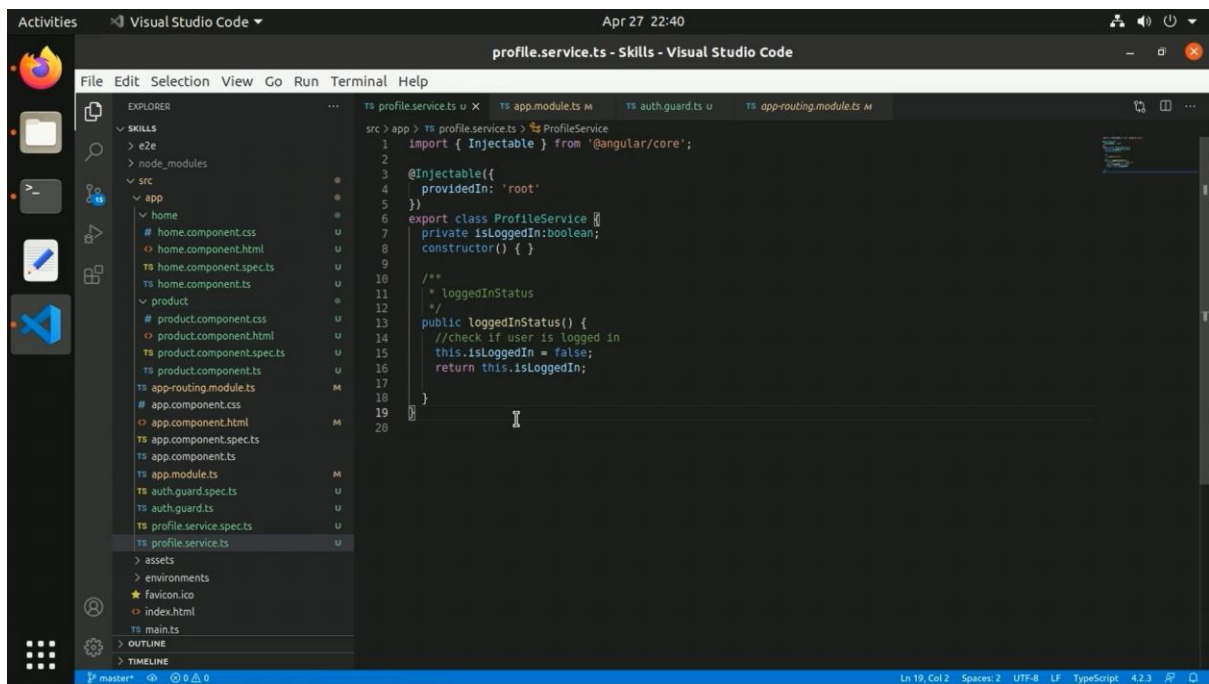
EXPLORER

- SKILLS
- e2e
- node_modules
- src
 - app
 - home
 - home.component.css
 - home.component.html
 - home.component.spec.ts
 - home.component.ts
 - product
 - product.component.css
 - product.component.html
 - product.component.spec.ts
 - product.component.ts
 - app-routing.module.ts
 - app.component.css
 - app.component.html
 - app.component.spec.ts
 - app.component.ts
 - app.module.ts
 - auth.guard.spec.ts
 - auth.guard.ts
 - profile.service.spec.ts
 - profile.service.ts
 - assets
 - environments
 - favicon.ico
 - index.html

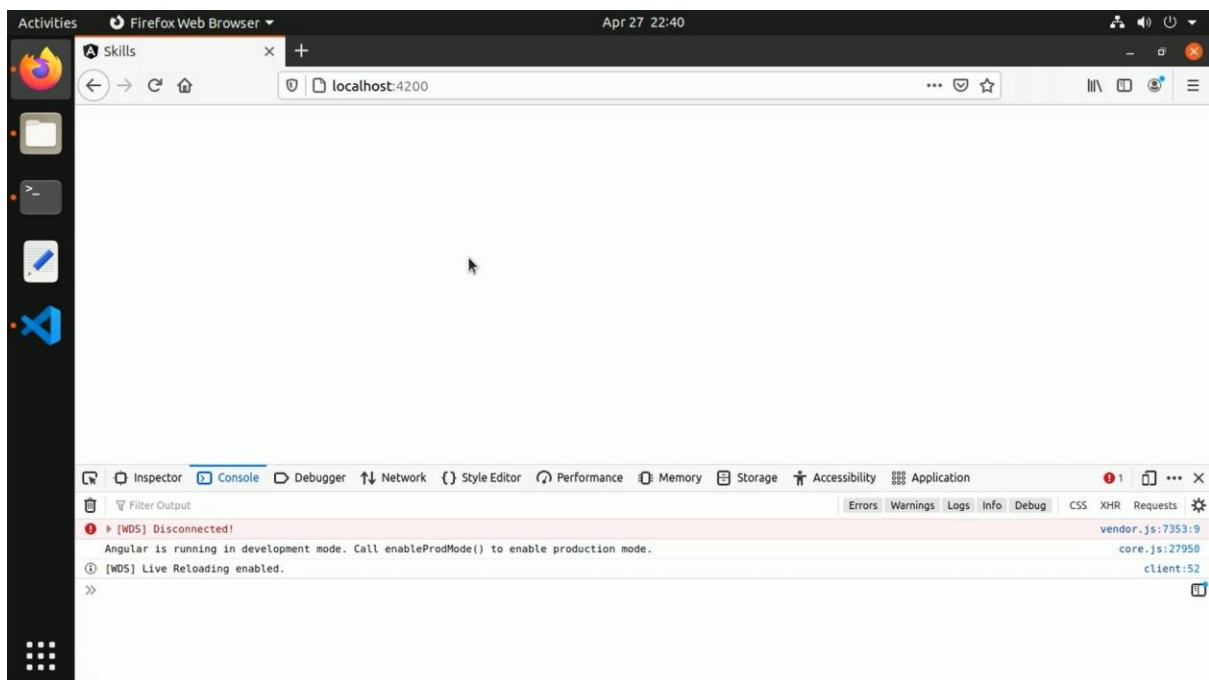
TS profile.service.ts u TS app.module.ts M TS auth.guard.ts u X TS app-routing.module.ts M

```
src > app > TS auth.guard.ts > TS AuthGuard > canActivate
1 import { Injectable } from '@angular/core';
2 import { CanActivate, ActivatedRouteSnapshot, RouterStateSnapshot, UrlTree } from '@angular/router';
3 import { ProfileService } from '../profile.service';
4
5 @Injectable({
6   providedIn: 'root'
7 })
8 export class AuthGuard implements CanActivate {
9   constructor(private profile: ProfileService) {
10
11   }
12
13   canActivate(
14     route: ActivatedRouteSnapshot,
15     state: RouterStateSnapshot): boolean {
16     if(!this.profile.loggedInStatus())
17       return false;
18     else
19       return true;
20   }
21 }
22
```

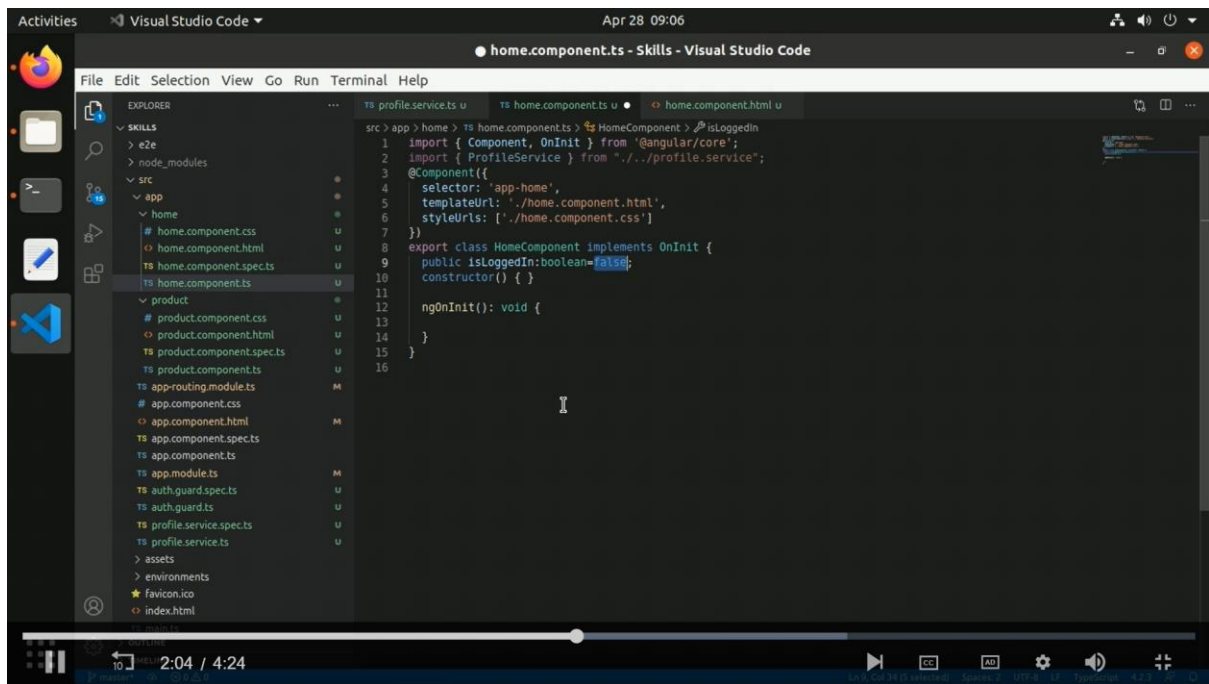
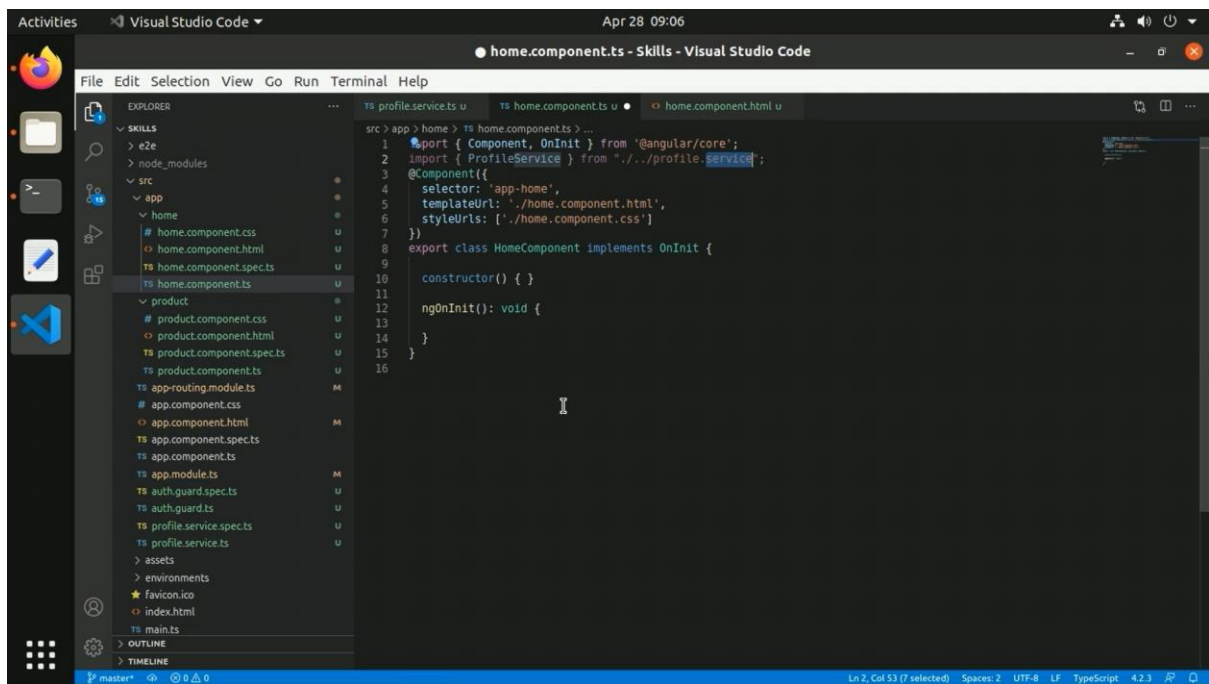
3:50 / 4:39

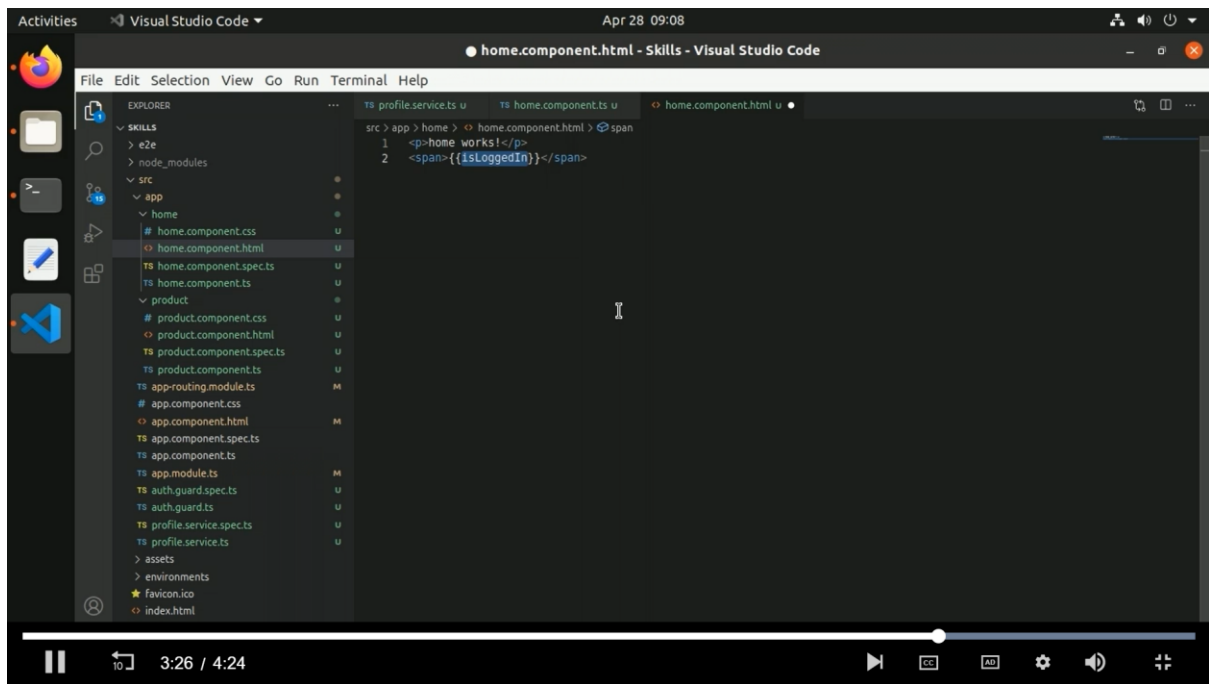
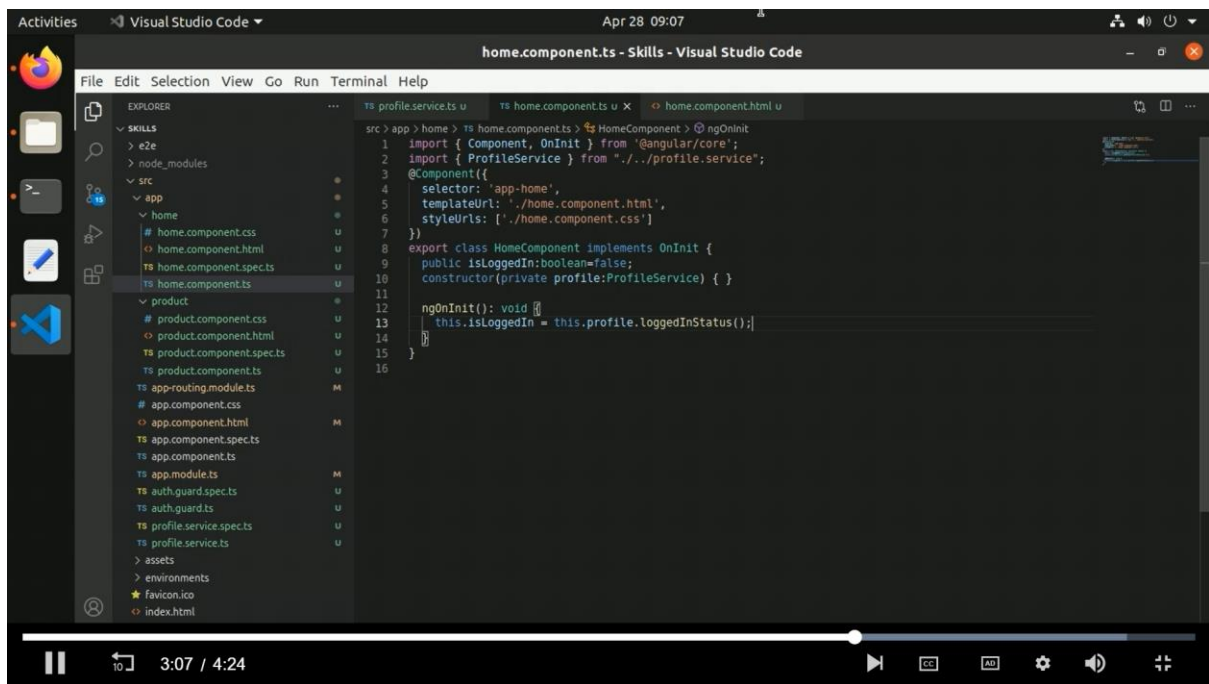


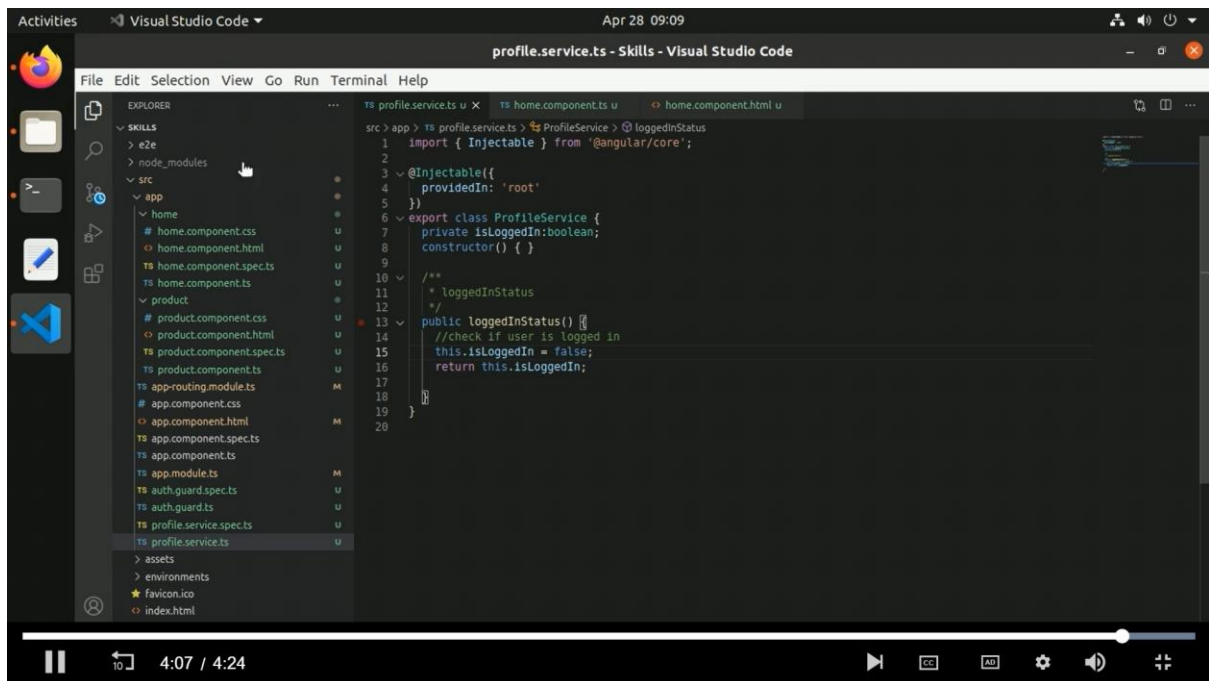
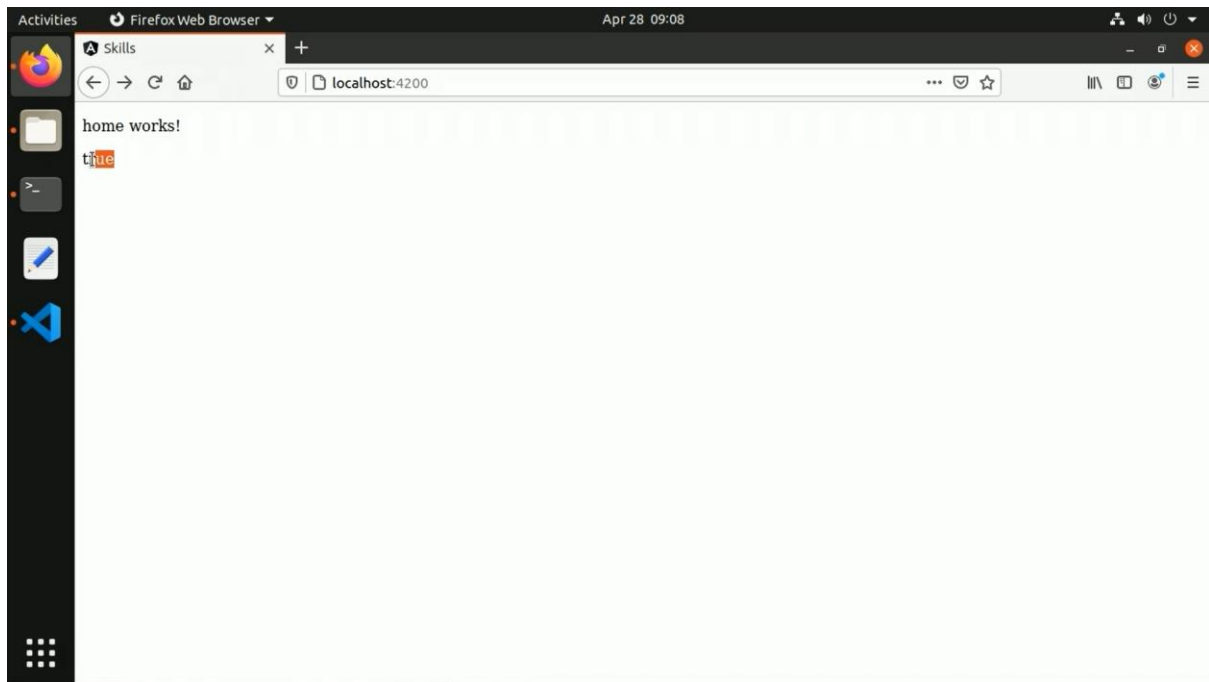
Empty for product

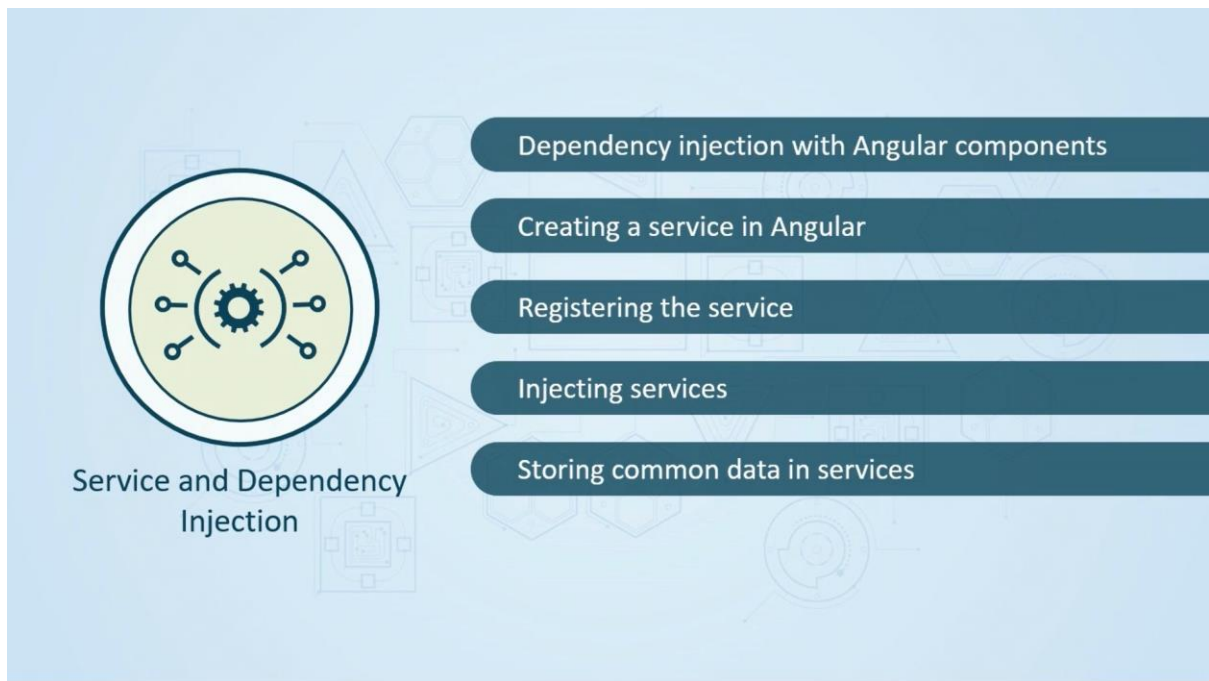
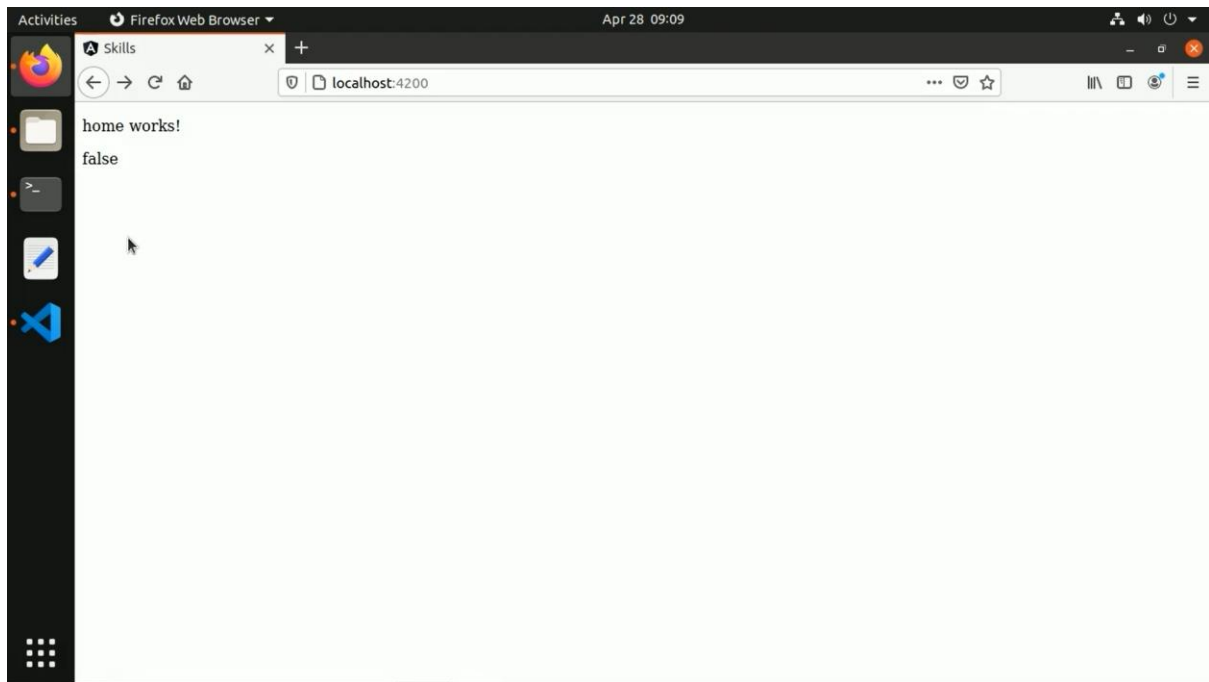


Storing common data in Service









Coming Up Next . . .



Authentication and
authorization



Performance
and deployment



2:00 / 2:09

