# **✅ Document: How We Set "Person Working on Laptop" Lottie Animation in Angular Standalone Project**

This document explains step-by-step how to set up a **Lottie animation** (person working on laptop) in an **Angular standalone component** project. It also includes file locations, code changes, and explanations for each step.

## **🔧 1. Install Required Packages**

Run this in your terminal:

npm install ngx-lottie lottie-web

### **📌 Why:**

* ngx-lottie: Angular wrapper for Lottie animations.
* lottie-web: The core library that renders the animation.

## **📁 2. app.config.ts**

### **✅ File: src/app/app.config.ts**

import { ApplicationConfig } from '@angular/core';  
import { provideRouter } from '@angular/router';  
import { provideAnimations } from '@angular/platform-browser/animations';  
import { provideLottieOptions } from 'ngx-lottie';  
import player from 'lottie-web';  
import { HomeComponent } from './components/home/home.component';  
  
export function playerFactory() {  
 return player;  
}  
  
export const appConfig: ApplicationConfig = {  
 providers: [  
 provideRouter([  
 { path: '', component: HomeComponent }  
 ]),  
 provideAnimations(),  
 provideLottieOptions({ player: playerFactory })  
 ]  
};

### **📌 Why:**

* provideLottieOptions: Registers the Lottie renderer globally.
* playerFactory: Returns the lottie-web player.
* provideRouter: Ensures routing to your HomeComponent.

## **🧩 3. Home Component Setup**

### **✅ File: src/app/components/home/home.component.ts**

import { Component } from '@angular/core';  
import { LottieComponent } from 'ngx-lottie';  
import { AnimationOptions } from 'ngx-lottie';  
  
@Component({  
 selector: 'app-home',  
 standalone: true,  
 imports: [LottieComponent],  
 templateUrl: './home.component.html',  
 styleUrls: ['./home.component.css']  
})  
export class HomeComponent {  
 lottieOptions: AnimationOptions = {  
 path: 'assets/animations/dev-laptop.json',  
 autoplay: true,  
 loop: true  
 };  
}

### **📌 Why:**

* LottieComponent: Required to render the animation tag <ng-lottie>.
* AnimationOptions: Configures the animation's path and behavior.

## **🖼️ 4. Home HTML Template**

### **✅ File: src/app/components/home/home.component.html**

<div class="hero-section">  
 <div class="hero-grid">  
 <div class="hero-content">  
 <h1>Welcome to <span class="brand">LinkedList InfoTech</span></h1>  
 <!-- Description Content -->  
 <div class="cta-buttons">  
 <a routerLink="/projects" class="btn">Our Projects</a>  
 <a routerLink="/contact" class="btn secondary">Contact Us</a>  
 </div>  
 </div>  
 <div class="hero-image">  
 <ng-lottie [options]="lottieOptions" width="400px" height="400px"></ng-lottie>  
 </div>  
 </div>  
</div>

### **📌 Why:**

* <ng-lottie>: Custom Angular element that plays the Lottie JSON.
* [options]: Binds the animation configuration.

## **🎨 5. Styling the Animation**

### **✅ File: src/app/components/home/home.component.css**

.hero-image {  
 flex: 1;  
 min-width: 300px;  
 text-align: center;  
}  
  
.hero-image ng-lottie {  
 max-height: 400px;  
 width: 100%;  
 object-fit: contain;  
}

### **📌 Why:**

* Ensures the animation fits nicely in the layout.
* Adjusts max height/width for responsive design.

## **📂 6. JSON Animation File**

### **✅ File Location:**

src/assets/animations/dev-laptop.json

### **📌 Why:**

* This is the actual animation file exported from LottieFiles.com or similar.
* Must be a valid .json animation.

## **✅ Summary**

|  |  |  |
| --- | --- | --- |
| **Step** | **File** | **Purpose** |
| 1 | package install | Load animation libraries |
| 2 | app.config.ts | Register Lottie renderer |
| 3 | home.component.ts | Define options and bind animation |
| 4 | home.component.html | Place <ng-lottie> tag |
| 5 | home.component.css | Control size/layout |
| 6 | dev-laptop.json | Actual animation content |

Now your standalone Angular app displays the **person working on laptop** Lottie animation! ✅