

Routing in Standalone Component Applications



Lara Newsom

Software Engineer | Speaker | Angular GDE

@laranerdsm

Ways to Bootstrap an Angular Application

Module Based

The default bootstrapping in version 16 and below for the `ng new` command

Standalone Component Based

The default bootstrapping in future versions for the `ng new` command

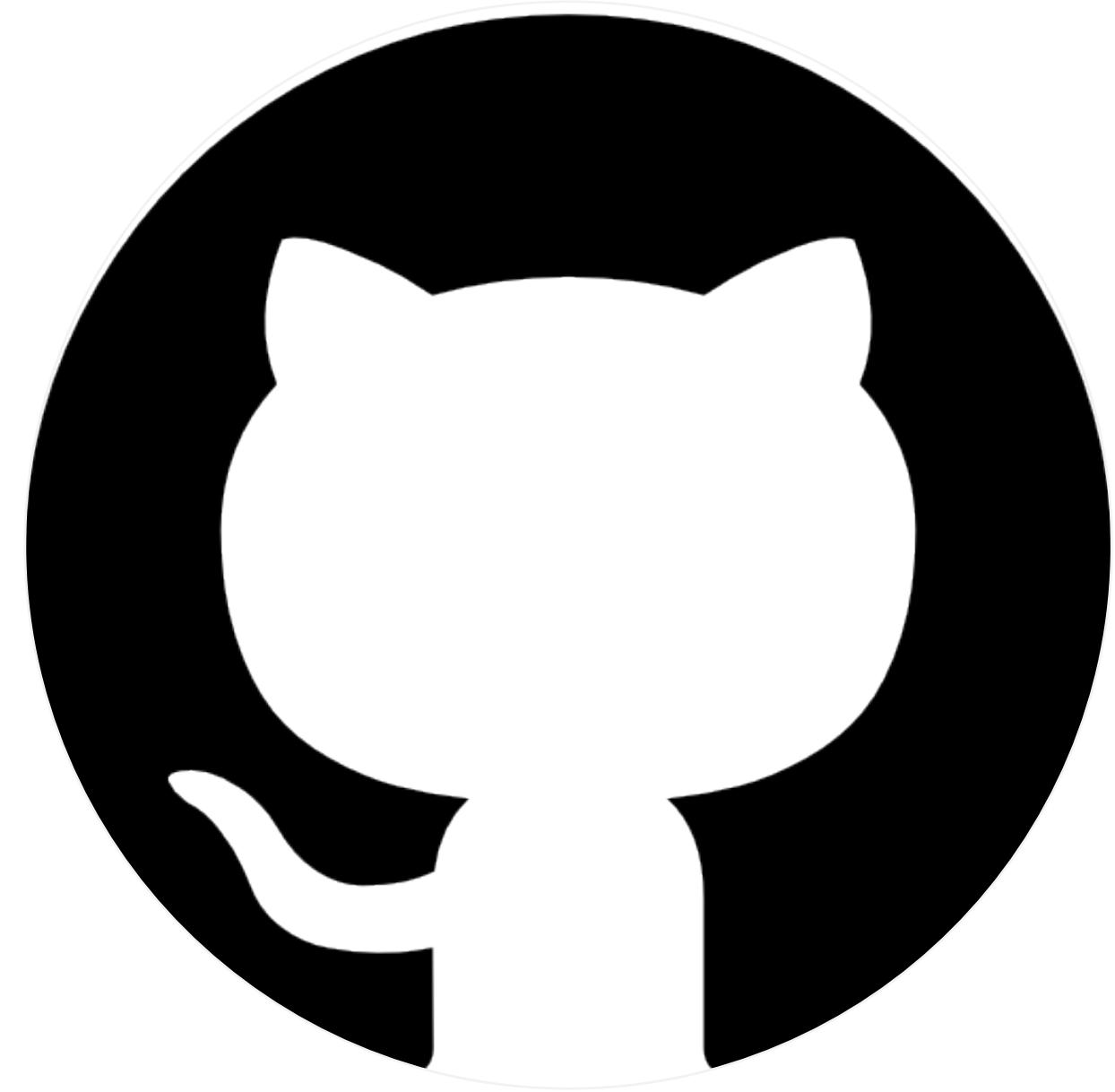


In this module, we will add routing to an existing application bootstrapped with a standalone component

Why Standalone Components?



- Do not require declaration in a module**
- Lower barrier of entry**
- Allows routing directly to route declarations and components**
- Can exist alongside modules**
- Already heavily used in the Angular framework itself**



Repository for Module Four

github.com/lara-newsom/angular-routing-course

- module-four-start

Bootstrapping the Router with Standalone Components

Bootstrapped in main.ts file

Declared inside bootstrapApplication

Routing is provided in a providers configuration array

Routing is bootstrapped in the provideRouter function



The provideRouter function
requires an array of routes
and can accept optional
configuration functions

What Are the Main Differences?

- Bootstrapping the router
- Configuring the router
- Implementing lazy loading

Placeholder slide: Enable tracing



Topics Previously Covered in Module Three

Declaring routes

Adding a router outlet

Enabling router tracing to debug issues

Demo

**Bootstrap routing in a standalone
component based application**

Summary

Initialized routing for an application bootstrapped with standalone components

Introduced the provideRouter function

Learned how to configure the router

Enabled router tracing

The Main Differences Between Standalone and Modules

Router configuration

**Components must import
RouterModule exports**

**Subsequent module sample
applications are bootstrapped
with a standalone component**

Up Next:

User Navigation between Views
