HAVELSAN

Creating an Nx Dynamic Module Federation Workspace with Webpack Enhanced Runtime in Angular

Erman Enginler Follow 4 min read · Feb 23, 2025

6 2

Medium Search Write 5

Home

Library

Profile

Stories

Stats

In this article we are going to learn how to create an workspace with nx dynamic module federation using Module Fedration plugin of enhanced runtime plugin with webpack.

nx angular workspace using dynamic module federation with enhanced runtime

If you are familiar with **dynamic module federation** in nx you used to use `loadRemoteModule` for routing a remote module in your route paths. And to use module federation configs you used to call `setRemoteDefinitions` in ***main.ts*** of host and pass the definitions which are exported configs from **module-federation-config** files.

In Nx Docs page it is mentioned as below that

> *Nx uses @module-federation/enhanced*
>
> *As of Nx 19.5, our Module Federation support is provided by the @module-federation/enhanced package. This package is owned and maintained by Zack Jackson, the creator of Module Federation, and the ByteDance team.*
>
> *Using this package for Nx's Module Federation support keeps our support aligned with the latest*

> *improvements, features and bug fixes for Module Federation.*
>
> *You can learn more about Module Federation Enhanced on their docs.*

So let's learn how to use this new module federation enhanced package by creating a new nx workspace.

> At the time I wrote this article I use:
> node v20.10.0
> npm v10.2.3
> nx@20.4.6
> angular@19.1.0

First, we create an nx workspace using

```
npx create-nx-workspace@20.4.6
```

It will prompt you with a couple of questions. I choose "web" as a workspace name and "none" for stack .

Select None for stack

> ***Important:*** *Do not choose “**Angular**” as the prompt for **stack** above. It would add a host which is not set for dynamic federation. We will continue with “N**one**” and add applications manually.*

all choices for prompts

After the creation is done you will see a message **“Welcome to the nx community”**

In order to create applications first we need to make some modifications on package.json

1. Open *package.json* at root and add the following ***devDependencies***

```
"devDependencies": {
...
  "@nx/angular": "20.4.6",
  "@schematics/angular": "^19.1.8"
...
}
```

2. remove ***workspaces*** node from ***package.json***

3. run `npm install`

After `npm install` completes run this command on terminal to create a **host** and a **remote** application with **dynamic module federation.** Assume we have a project with host as **products** and remote as **cart.**

```
npx nx g @nx/angular:host apps/products --remotes=
```

After generation completed cli would create a bunch of files and 2 applications at **apps** folder.

When we examine the **main.ts** at the host application (products) we clearly see that it uses new `@module-federation/enhanced/runtime` with its `init` method.

But there is a problem with

```
..import('./bootstrap')..
```

If we try to compile and serve host at this point we will get some errors.
We need some further modifications in order to make this project build and compile.

Open *tsconfig.base.json* at the root and swap properties with this

```
{
  "compilerOptions": {
    "isolatedModules": true,
    "noEmitOnError": true,
    "noFallthroughCasesInSwitch": true,
    "noImplicitOverride": true,
    "noImplicitReturns": true,
    "noUnusedLocals": true,
    "strict": true,
    "useDefineForClassFields": false,
    "moduleResolution": "node",
    "target": "es2022",
    "module": "esnext",
    "skipLibCheck": true,
    "baseUrl": ".",
    "paths": {
      "cart/Routes": ["apps/cart/src/app/remote-en
    }
  }
}
```

now serve your host application using

```
npx nx run products:serve:development --devRemotes
```

When we navigate to */cart* we can clearly see it is served from ***localhost:4201/default-apps_cart_src_app_remote-entry_entry_component_ts.js***

So the dynamic module federation works.

As we examine ***app.routes.ts*** at the host application we see that the lazy loading (**loadChildren**) is loading module via **loadRemote** (not loadRemoteModule) which is imported from `@module-federation/enhanced/runtime`

app.routes.ts of host app

As we examine ***module-federation.manifest.json*** at *host/public folder* we can see it is pointing **localhost:4201/mf-manifest.json.** which is mentioned in manifest — Module federation

For further readings

- Module Federation and Nx | Nx
- runtimePlugins — Module federation

That’s all. Please like.

You may download the sample project on GitHub ermaneng/medium_nx_dynamic_module_federation

Good readings
ermaneng

Angular Nx Module Federation Webpack Havelsan

Published in HAVELSAN

Follow

89 followers · Last published Dec 29, 2025

Trend Stories on Technology and Culture from HAVELSAN Writers

Written by Erman Enginler

Follow

52 followers · 25 following

Frontend Developer

Responses (2)

László Kővári

What are your thoughts?

Nicolas Naso
Aug 20, 2025

Excellent! I have two questions:

1. The **init** method is marked as *deprecated* and it suggest to use **getInstance** or **createInstance**. How would you configure in the main.ts?

2. The setRemoteDefinitions of Nx ('@nx/angular/mf') load each defintion on… more

Reply

Okan Çilingiroğlu
Apr 9, 2025

Yet another great article Erman, thanks mate.

Reply

More from Erman Enginler and HAVELSAN

How to Use Assets Properly in Module…

If you are using a module federated workspace at…

Jun 3, 2023 77

Using Protocol Buffers Between…

In this article, I want to demonstrate how to use…

Feb 16, 2024 37 1

A Practical Guide to Functional…

Explore functional programming: immutabilit…

Aug 28, 2023 39

How to Pass Url Query Parameters (Routin…

In this article we are going to learn how to pass router…

Nov 10, 2023 42 1

See all from Erman Enginler

See all from HAVELSAN

Recommended from Medium

Build Ultra-Fast Dev Loops with Vite &…

A practical playbook to slash wait time with Vite HMR,…

Oct 11, 2025 111 2

From Figma to Code: Mastering the Figm…

If you're a developer who's ever spent hours manually…

Sep 8, 2025 2

Stop Memorizing Design Patterns: Us…

Choose design patterns based on pain points: apply…

Jan 29 1.6K 9

Angular 21 Monorepo Micro Frontends wit…

Learn how to build a micro frontend architecture in an…

Jan 24 149

Essential Considerations for…

When building micro frontends for a multi-tenan…

Aug 26, 2025

How to Build and Publish a Reusable…

If you've ever wanted to create a modular, reusable…

Sep 20, 2025

See more recommendations

Help Status About Careers Press Blog Privacy Rules Terms Text to speech