Early .NET 6 Observations

Sunday November 14, 2021

Not meant as a comprehensive review of .NET 6. Just a few items I've encountered so far. I'll update this article as needed.

Single-File Publishing

A while back, I tested single-file publishing in .NET 5. It was improved from earlier versions, but still forced you to deploy more than one file when targeting Windows or macOS.

I just retested with .NET 6, and it now produces a single file for deployment for all four targets I checked (Windows 10 64-bit, Linux 64-bit, Linux ARM, and macOS 10.14 Mojave).

(Testing with a console application)

Hot Reload

You don't have to use Visual Studio 2022 to take advantage of hot reload. It's supported by the CLI as well. So if, for example, you're working on a web app in Visual Studio Code, don't run the project from the IDE. Instead, run the project with 'dotnet watch' from the command line, then just make your edits in the IDE, and they'll show up in the running app in real time.

It's probably also worth mentioning that Hot Reload apparently barely made the cut for the SDK.[1)] Sounds like it might be indicative of bigger problems brewing inside Microsoft.

Nullable Reference Types

New projects created in .NET 6 now turn on nullable reference types by default. Checking your .csproj file, you'll see this:

```
<Nullable>enable</Nullable>
```

This means you might see some warnings you weren't seeing before. When the compiler sees declaration of a non-nullable variable, like this:

```
public string companyName;
```

It checks usage of the variable, and warns you of two things:

- Declaration of the variable without indicating that it's nullable. (Should be public string? companyName;)
- Instances of using the variable contents where you might be getting a null value.

Issues with React-Redux Project

Testing with .NET SDK version 6.0.100.

Badly Formed JSON

The reactredux template generates a project with badly-formed JSON in ClientApp/package.json. Luckily, it’s easy to fix. When you open package.json, the error is in this block:

```
"resolutions": {
  "url-parse": ">=1.5.0",
  "lodash": ">=4.17.21",
},
```

Remove the extra comma at the end of the “lodash” line, and the project will build.

Wrong Target Framework

I also noticed that the template targets net5.0 instead of net6.0:

TestProject.csproj

```
<TargetFramework>net5.0</TargetFramework>
```

It does build correctly both when I leave it at net5.0, and when I change it to net6.0.

[1] https://dusted.codes/can-we-trust-microsoft-with-open-source

From:
https://blog.devtoprd.com/ - **Jim's Blog**

Permanent link:
https://blog.devtoprd.com/doku.php?id=posts:2021:2021.11.14_-_early_.net_6_observations

Last update: **2025/03/31 14:11**