How to Upgrade to .NET 8

Upgrading to .NET 8 is a straightforward process. The steps below will explain exactly what you need to do to complete the upgrade.

.NET 8.0 SDK Install

1. Before doing anything else, make sure you have installed the latest .NET 8.0 SDK. You can get it on this page:

   https://dotnet.microsoft.com/en-us/download/dotnet/8.0

   Just make sure you get the SDK installer, not the installer for any of the runtimes.

In Play.Common repository

2. Change the TargetFramework version to net8.0 in Play.Common.csproj:

   ```
   <PropertyGroup>
     <TargetFramework>net8.0</TargetFramework>
     …
   </PropertyGroup>
   ```

3. Upgrade the following package references in Play.Common.csproj to the latest 8.0.x version:

   ```
   <PackageReference Include="Microsoft.AspNetCore.Authentication.JwtBearer" Version="8.0.0" />
   <PackageReference Include="Microsoft.Extensions.Configuration" Version="8.0.0" />
   <PackageReference Include="Microsoft.Extensions.Configuration.Binder" Version="8.0.0" />
   <PackageReference Include="Microsoft.Extensions.DependencyInjection" Version="8.0.0" />
   ```

4. In .github\workflows\ci.yml, find the Setup .NET action and update the .NET version to 8.0.x:

   ```
   - name: Setup .NET
     uses: actions/setup-dotnet@v3
     with:
       dotnet-version: 8.0.x
       source-url: https://nuget.pkg.github.com/${{github.repository_owner}}/index.json
     …
   ```

5. Commit and push changes to remote.

6. Wait for the GitHub workflow to complete and take note of the new Play.Common NuGet package version.

In Play.Identity repository

7. Change the TargetFramework version to net8.0 in all .csproj files:

```
<PropertyGroup>
  <TargetFramework>net8.0</TargetFramework>
  …
</PropertyGroup>
```

8. Upgrade the following package references in Play.Identity.Service.csproj to the latest version:

```
<PackageReference Include="Microsoft.AspNetCore.Identity.UI" Version="8.0.0" />
<PackageReference Include="Microsoft.VisualStudio.Web.CodeGeneration.Design" Version="8.0.0" />
<PackageReference Include="Swashbuckle.AspNetCore" Version="6.4.0" />
```

9. Upgrade the Play.Common package reference in Play.Identity.Service.csproj to the new version just produced by the GitHub workflow. For instance, if the new version was 1.0.16, you would perform this update:

```
<PackageReference Include="Play.Common" Version="1.0.16" />
```

10. In .github\workflows\cicd.yml, find the Setup .NET action and update the .NET version to 8.0.x:

```
- name: Setup .NET
  uses: actions/setup-dotnet@v3
  with:
    dotnet-version: 8.0.x
    source-url: https://nuget.pkg.github.com/${{github.repository_owner}}/index.json
  …
```

11. Update the version of aspnet and the .NET SDK to 8.0 in the Dockerfile:

```
FROM mcr.microsoft.com/dotnet/aspnet:8.0 AS base
…

FROM mcr.microsoft.com/dotnet/sdk:8.0 AS build
…
```

12. Update the path to Play.Identity.Service.dll in launch.json:

```
{
  "name": ".NET Core Launch (web)",
  …
  "program":
"${workspaceFolder}/src/Play.Identity.Service/bin/Debug/net8.0/Play.Identity.Service.dll",
  …
}
```

13. Commit and push changes to remote.

14. Wait for the GitHub workflow to complete and take note of the new Play.Identity.Contracts NuGet package version.

In Play.Catalog repository

15. Change the TargetFramework version to net8.0 in all .csproj files:

    ```
    <PropertyGroup>
      <TargetFramework>net8.0</TargetFramework>
      …
    </PropertyGroup>
    ```

16. Upgrade the following package references in Play.Catalog.Service.csproj to the latest version:

    ```
    <PackageReference Include="Swashbuckle.AspNetCore" Version="6.4.0" />
    ```

17. Upgrade the Play.Common package reference in Play.Catalog.Service.csproj to the new version just produced by the GitHub workflow. For instance, if the new version was 1.0.16, you would perform this update:

    ```
    <PackageReference Include="Play.Common" Version="1.0.16" />
    ```

18. In .github\workflows\cicd.yml, find the Setup .NET action and update the .NET version to 8.0.x:

    ```
    - name: Setup .NET
      uses: actions/setup-dotnet@v3
      with:
        dotnet-version: 8.0.x
        source-url: https://nuget.pkg.github.com/${{github.repository_owner}}/index.json
      …
    ```

19. Update the version of aspnet and the .NET SDK to 8.0 in the Dockerfile:

    ```
    FROM mcr.microsoft.com/dotnet/aspnet:8.0 AS base
    …

    FROM mcr.microsoft.com/dotnet/sdk:8.0 AS build
    …
    ```

20. Update the path to Play.Catalog.Service.dll in launch.json:

    ```
    {
    ```

```
    "name": ".NET Core Launch (web)",
    ...
    "program":
  "${workspaceFolder}/src/Play.Catalog.Service/bin/Debug/net8.0/Play.Catalog.Service.dll",
    ...
  }
```

21. Commit and push changes to remote.

22. Wait for the GitHub workflow to complete and take note of the new Play.Catalog.Contracts NuGet package version.

In Play.Inventory repository

23. Change the TargetFramework version to net8.0 in all .csproj files:

```
<PropertyGroup>
  <TargetFramework>net8.0</TargetFramework>
  ...
</PropertyGroup>
```

24. Upgrade the following package references in Play.Inventory.Service.csproj to the latest version:

```
<PackageReference Include="Microsoft.Extensions.Http.Polly" Version="8.0.0" />
<PackageReference Include="Swashbuckle.AspNetCore" Version="6.4.0" />
```

25. Upgrade the Play.Common package reference in Play.Inventory.Service.csproj to the new version just produced by the GitHub workflow. For instance, if the new version was 1.0.16, you would perform this update:

```
<PackageReference Include="Play.Common" Version="1.0.16" />
```

26. Upgrade the Play.Catalog.Contracts package reference in Play.Inventory.Service.csproj to the new version just produced by the GitHub workflow. For instance, if the new version was 1.0.14, you would perform this update:

```
<PackageReference Include="Play.Catalog.Contracts" Version="1.0.14" />
```

27. In .github\workflows\cicd.yml, find the Setup .NET action and update the .NET version to 8.0.x:

```
- name: Setup .NET
  uses: actions/setup-dotnet@v3
  with:
    dotnet-version: 8.0.x
    source-url: https://nuget.pkg.github.com/${{github.repository_owner}}/index.json
  …
```

28. Update the version of aspnet and the .NET SDK to 8.0 in the Dockerfile:

```
FROM mcr.microsoft.com/dotnet/aspnet:8.0 AS base
…

FROM mcr.microsoft.com/dotnet/sdk:8.0 AS build
…
```

29. Update the path to Play.Inventory.Service.dll in launch.json:

```
{
   "name": ".NET Core Launch (web)",
   …
   "program":
"${workspaceFolder}/src/Play.Inventory.Service/bin/Debug/net8.0/Play.Inventory.Service.dll",
   …
}
```

30. Commit and push changes to remote.

31. Wait for the GitHub workflow to complete and take note of the new Play.Inventory.Contracts NuGet package version.

In Play.Trading repository

32. Change the TargetFramework version to net8.0 in all .csproj files:

```
<PropertyGroup>
  <TargetFramework>net8.0</TargetFramework>
  …
</PropertyGroup>
```

33. Upgrade the following package references in Play.Trading.Service.csproj to the latest version:

```
<PackageReference Include="Swashbuckle.AspNetCore" Version="6.4.0" />
```

34. Upgrade the Play.Common package reference in Play.Trading.Service.csproj to the new version just produced by the GitHub workflow. For instance, if the new version was 1.0.16, you would perform this update:

```
<PackageReference Include="Play.Common" Version="1.0.16" />
```

35. Upgrade the Play.Catalog.Contracts package reference in Play.Trading.Service.csproj to the new version just produced by the GitHub workflow. For instance, if the new version was 1.0.14, you would perform this update:

    ```
    <PackageReference Include="Play.Catalog.Contracts" Version="1.0.14" />
    ```

36. Upgrade the Play.Identity.Contracts package reference in Play.Trading.Service.csproj to the new version just produced by the GitHub workflow. For instance, if the new version was 1.0.25, you would perform this update:

    ```
    <PackageReference Include="Play.Identity.Contracts" Version="1.0.25" />
    ```

37. Update the version of aspnet and the .NET SDK to 8.0 in the Dockerfile:

    ```
    FROM mcr.microsoft.com/dotnet/aspnet:8.0 AS base
    …

    FROM mcr.microsoft.com/dotnet/sdk:8.0 AS build
    …
    ```

38. Update the path to Play.Trading.Service.dll in launch.json:

    ```
    {
      "name": ".NET Core Launch (web)",
      …
      "program": "${workspaceFolder}/src/Play.Trading.Service/bin/Debug/net8.0/Play.Trading.Service.dll",
      …
    }
    ```

39. Commit and push changes to remote.

40. Wait for the GitHub workflow to complete.

41. Congratulations, your migration is complete!