

Adding a Dockerfile

Let's create a dockerfile to containerize or dockerize our Identity microservice.

In Play.Identity repo

1. Move the Clients section from appsettings.Development.json to appsettings.json
2. Add the Docker VS Code extension
3. Open command palette and type:

Docker: Add Docker files to workspace

4. "Most of the time you will select Linux here, because that allows your container to run on any Linux based operating system. Also, docker images based on Linux are much smaller and fast to start than Windows ones and most Docker features are first available for Linux containers. You would only create a Windows based image if you absolutely need a feature only available on Windows."
5. Follow wizard using port 5002
6. Move Dockerfile to root
7. Notice .dockerignore
8. Notice updates to tasks.json and launch.json
9. Simplify Dockerfile:

```
FROM mcr.microsoft.com/dotnet/aspnet:5.0-focal AS base
WORKDIR /app
EXPOSE 5002
```

```
ENV ASPNETCORE_URLS=http://+:5002
```

```
# Creates a non-root user with an explicit UID and adds permission to access the /app folder
# For more info, please refer to https://aka.ms/vscode-docker-dotnet-configure-containers
RUN adduser -u 5678 --disabled-password --gecos "" appuser && chown -R appuser /app
USER appuser
```

```
FROM mcr.microsoft.com/dotnet/sdk:5.0-focal AS build
```

```
COPY ["src/Play.Identity.Contracts/Play.Identity.Contracts.csproj", "src/Play.Identity.Contracts/"]
```

```
COPY ["src/Play.Identity.Service/Play.Identity.Service.csproj", "src/Play.Identity.Service/"]
```

```
RUN dotnet restore "src/Play.Identity.Service/Play.Identity.Service.csproj"
```

```
COPY ./src ./src  
WORKDIR "/src/Play.Identity.Service"  
RUN dotnet publish "Play.Identity.Service.csproj" -c Release --no-restore -o /app/publish
```

```
FROM base AS final  
WORKDIR /app  
COPY --from=build /app/publish .  
ENTRYPOINT ["dotnet", "Play.Identity.Service.dll"]
```

10. Explain dockerignore file
11. Commit and push changes

In the next lesson you will modify this dockerfile to be able to consume NuGet packages from GitHub packages.