?????????? .NET ?? ????? ??????

В данном примере .NET обновляется до версии 10 под WSL:Ubuntu. В проектах используется база PostgreS, для нее также будут обновлены пакеты поддержки.

Нужно обновить **SDK**, **TargetFramework** и все основные пакеты EF/Npgsql.

????????? ????????????? SDK

В WSL:

```
dotnet --list-sdks
```

Если .NET 10 отсутствует, установить его.

Для Ubuntu/WSL:

```
sudo apt update
sudo apt install dotnet-sdk-10.0
```

Потом проверить:

```
dotnet --list-sdks
```

Должно появиться что-то вроде:

```
8.0.xxx
10.0.xxx
```

???????? global.json (???? ????)

Проверьте:

```
find . -name global.json
```

Если файл есть и содержит:

```
{
  "sdk": {
    "version": "8.0.127"
  }
}
```

то либо удалить его, либо изменить на:

```
{
  "sdk": {
    "version": "10.0.100"
  }
}
```

(точную версию смотрите в `dotnet --list-sdks`)

Очень полезная команда:

```
dotnet --info
```

В выводе будет что-то вроде:

```
.NET SDK:
 Version:   8.0.127

global.json file:
  /home/devadmin/projects/global.json
```

или

```
global.json file:
  Not found
```

Тогда сразу видно, какой файл фиксирует версию SDK.

???????? ??? csproj

Заменить:

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

на

```
<TargetFramework>net10.0</TargetFramework>
```

Можно массово:

```
grep -R "TargetFramework" .
```

в файлах csproj нужно обновить только тег <TargetFramework>, версии пакетов в <PackageReference> обновятся при обновлении EF Core.

???????? EF Core

После перехода на net10:

```
dotnet add package Microsoft.EntityFrameworkCore --version 10.0.0
dotnet add package Microsoft.EntityFrameworkCore.Design --version 10.0.0
dotnet add package Microsoft.EntityFrameworkCore.Relational --version 10.0.0
dotnet add package Npgsql.EntityFrameworkCore.PostgreSQL --version 10.0.0
dotnet add package EFCore.CheckConstraints --version 10.0.0
```

???????? ??????????? EF

Проверить:

```
dotnet tool list -g
```

Если EF старый:

```
dotnet tool update --global dotnet-ef
```

Проверить:

```
dotnet ef --version
```

???????? ? ???????????

Из корня решения:

```
dotnet clean
find . -type d -name bin -exec rm -rf {} +
find . -type d -name obj -exec rm -rf {} +
```

```
dotnet restore
dotnet build
```

Для каждого проекта можно проверить пакеты, которые требуют обновления:
Удобная команда:

```
dotnet list package --outdated
```

Она покажет что можно обновить. Например есть такой вывод:

```
The following sources were used:
   https://api.nuget.org/v3/index.json

Project `ReactPostgresWsl.Api` has the following updates to its packages
   [net10.0]:
   Top-level Package                          Requested   Resolved   Latest
   > Microsoft.AspNetCore.OpenApi             8.0.27      8.0.27     10.0.8
   > Microsoft.EntityFrameworkCore            10.0.0      10.0.0     10.0.8
   > Microsoft.EntityFrameworkCore.Design     10.0.0      10.0.0     10.0.8
   > Npgsql.EntityFrameworkCore.PostgreSQL    10.0.0      10.0.0     10.0.2
   > Swashbuckle.AspNetCore                   6.6.2       6.6.2      10.2.1
```

тогда обновление для проекта будет таким:

```
dotnet add package Microsoft.AspNetCore.OpenApi --version 10.0.8
dotnet add package Microsoft.EntityFrameworkCore --version 10.0.8
dotnet add package Microsoft.EntityFrameworkCore.Design --version 10.0.8
dotnet add package Npgsql.EntityFrameworkCore.PostgreSQL --version 10.0.2
dotnet add package Swashbuckle.AspNetCore --version 10.2.1
```

После этого:

```
dotnet restore
dotnet clean
dotnet build
```

Ещё проверьте версию `dotnet-ef`:

```
dotnet ef --version
```

Если там не 10.x:

```
dotnet tool update --global dotnet-ef
```

Revision #3
Created 2026-06-08 04:50:31 UTC by Nikolay
Updated 2026-06-08 07:41:14 UTC by Nikolay