With Entity Framework Core database first approach how do I separate my entity from infrastructure layer?

Asked 14 days ago Active 8 days ago Viewed 50 times

0

I'm building REST APIs with ASP.NET Core and Entity Framework Core with a database-first approach with clean architecture.

My database is already created and maintained by a DBA team - that's why I need to use Entity Framework Core with a database-first approach, and we need to do reverse engineering using the `Scaffold-DbContext` command in the infrastructure layer.

With clean architecture, entities should be placed in `Core` layer separate from the infrastructure layer so how we will do that and how to resolve your `DbContext` as an interface.

c# entity-framework-core asp.net-core-webapi clean-architecture

edited Feb 21 at 9:21
marc_s
627k 142 1192 1323

asked Feb 21 at 8:58
Amit Kashyap
3 1

I'm not sure I understand your question, but if you ask how to abstract DbContext, repository pattern probably will help. – worldwildwebdev Feb 21 at 10:01

1 Answer

0

I see two options:

1. You accept the dependency to the EF, put the generated entities into your core project and so accept the violation to clean architecture.
2. You create your own entities in Core project without dependencies to EF and use repository pattern to "map" between your entities and those from EF.

Alternative 1 is simple and cheap. Alternative 2 gives you independence from EF and some abstraction to your dba team. Choose based on what is more important for your project.

answered Feb 27 at 6:40

plainionist

1,292 1 7 16

I would select 1 option. Is it best practice to move auto generated db entities from Persistence Layer to Domain Layer (generated by Scaffold-DbContext command)? is there any command to generate db entities in separate layer and db context in separate layer? – Amit Kashyap Mar 2 at 13:05

It all depends on what you want to optimize for: do you need rapid prototyping and is your code base small? Option 1 might be best. Do you need independence from other teams, specific frameworks and technologies and your code base is big and needs to be maintained for years: go for option 2. (I am not aware of such command) – plainionist Mar 2 at 21:13