

Repository en Unit-of-Work Design Patterns

Het repository-design patroon:

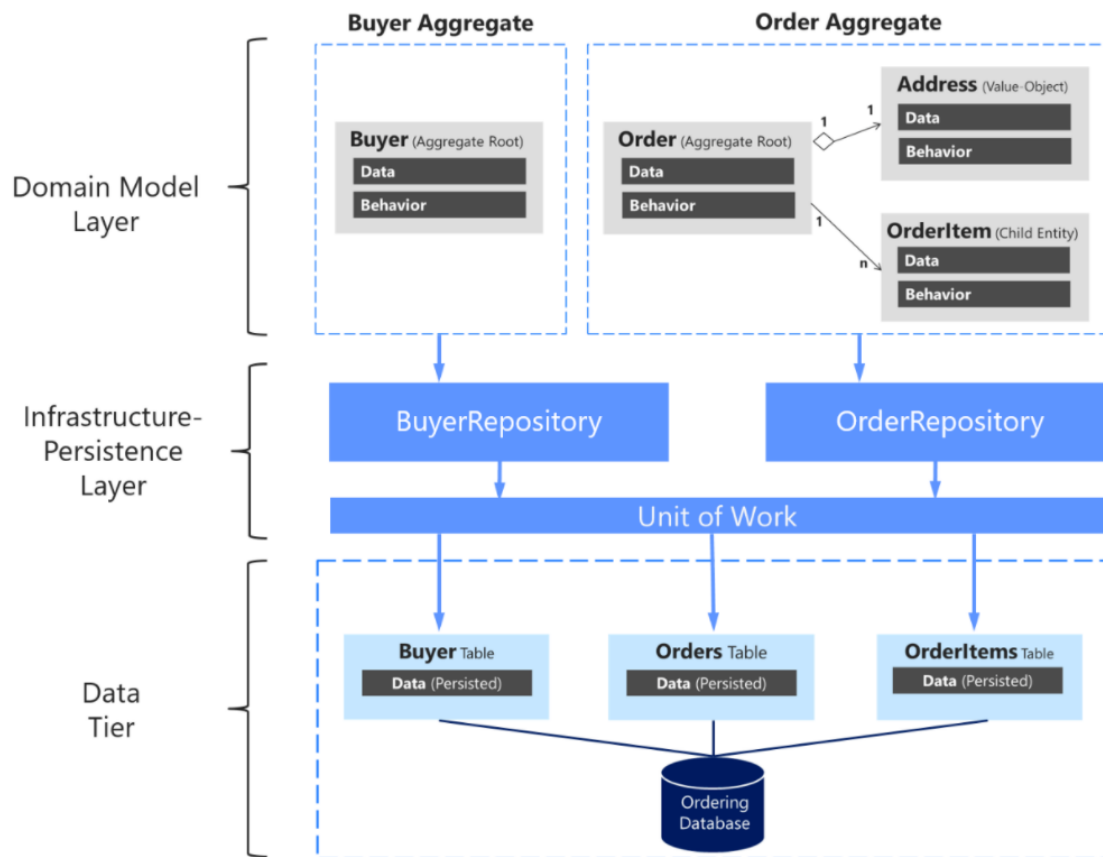
Repositories zijn klassen of componenten die de logica bevatten die nodig zijn om toegang te krijgen tot data sources. Ze centraliseren de gemeenschappelijke data-functionaliiteit, zorgen voor een betere onderhoudbaarheid en ontkoppelen de infrastructuur of technologie die wordt gebruikt om toegang te krijgen tot databases van de domein-modellaag. Als u een Object-Relational Mapper (ORM) zoals Entity Framework gebruikt, wordt de code die geïmplementeerd moet worden vereenvoudigd dankzij LINQ.

Conceptueel bevat een repository een verzameling van objecten die in de database zijn opgeslagen en operaties die erop kunnen worden uitgevoerd, die dicht bij de persistentielaag liggen. Repositories ondersteunen ook het doel om, duidelijk en in één richting de dependencies (afhankelijkheden) tussen work domain en de data allocation of (ORM- mapping).

Op implementatieniveau is een repository gewoon een klasse met gegevenspersistentiecode die wordt gecoördineerd door een unit-of-work klasse (DbContext in EF Core)

Definieer één repository per aggregaat

Voor elke geaggregeerde of geaggregeerde root moet je één repositoryklasse maken. Bij Domain-Driven Design (DDD) -patronen, gaat het enige kanaal dat je moet gebruiken om de database aan te spreken via de repository-klassen. Kort gezegd, stelt een repository je in staat om gegevens in het geheugen te plaatsen die afkomstig zijn uit de database in de vorm van de domein-entiteiten. Zodra de entiteiten zich in het geheugen bevinden, kunnen ze worden gewijzigd en vervolgens weer in de database worden bewaard.



<https://docs.microsoft.com/en-us/dotnet/architecture/microservices/microservice-ddd-cqrs-patterns/infrastructure-persistence-layer-design>

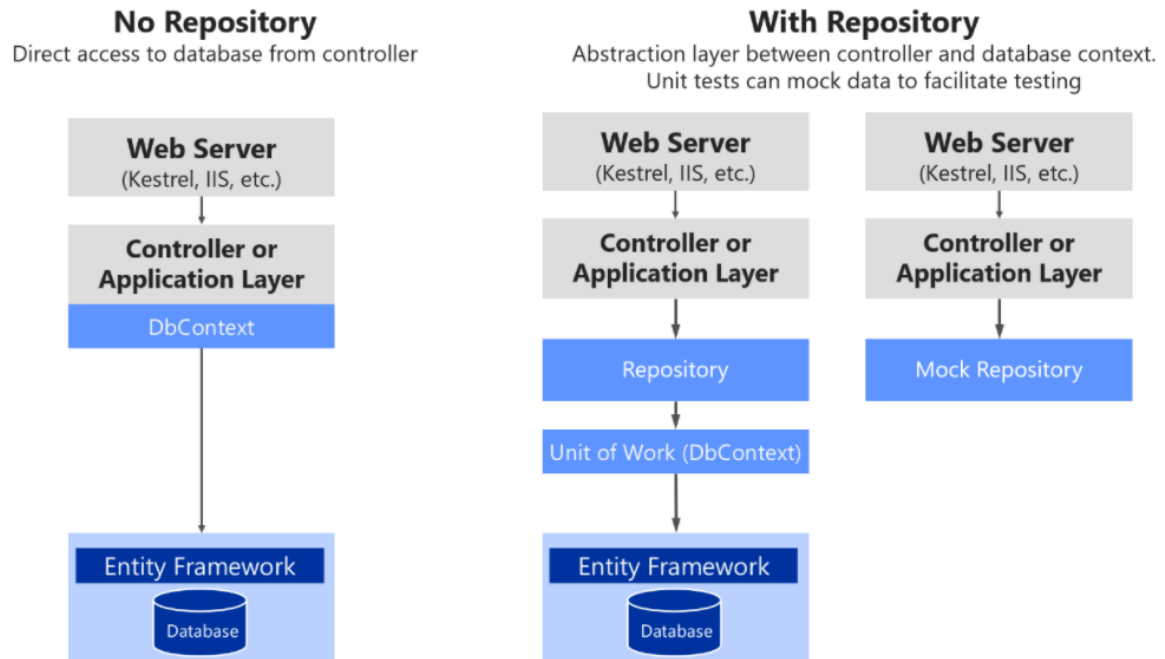


Figure 7-18. Using custom repositories versus a plain DbContext

Bron:

<https://docs.microsoft.com/en-us/dotnet/architecture/microservices/microservice-ddd-cqrs-patterns/infrastructure-persistence-layer-implementation-entity-framework-core>

<https://www.amazon.com/Patterns-Enterprise-Application-Architecture-Martin/dp/0321127420/>

<https://medium.com/net-core/repository-pattern-implementation-in-asp-net-core-21e01c6664d7>