

Is this domain or application service

Asked 3 years, 6 months ago Active 3 years, 6 months ago Viewed 703 times



I am building authentication micro-service/domain by using DDD and I am still having trouble with identifying where does each service belong. At this point I am not sure does Authentication service belongs to the domain services or application services.

2



Should I wrap this behavior in domain service, and expose response object via application service, or this should stay as it is - as application service.



1



```
public class AuthenticationService : IAuthenticationService
{
    IAuthUnitOfWork _uow;
    IUserRepository _userRepository;
    IUserTokenFactory _userTokenFactory;

    public AuthenticationService(IUserTokenFactory userTokenFactory, IUserRepository
userRepository,
        IAuthUnitOfWork uow)
    {
        _userTokenFactory = userTokenFactory;
        _userRepository = userRepository;
        _uow = uow;
    }

    public async Task<UserTokenResponse> AuthenticateAsync(string email, string
password)
    {
        var user = await _userRepository.GetByEmailAndPasswordAsync(email,
password);
        //TODO: Add null check for user
        var userToken = await _userTokenFactory.CreateWithAsync(user);

        await _uow.SaveChangesAsync();

        return new UserTokenResponse
        {
            ExpiressOn = userToken.ExpiressOn,
            Token = userToken.Token
        };
    }
}
```

domain-driven-design

ddd-service

asked Aug 16 '16 at 18:19



Robert

2,374

4

18

35

1 Answer



9



- **Application Services** coordinate application flow and infrastructure, but do not execute business logic rules or invariants. It is common to see calls to repositories, units of work, and to accept and return service contract objects or request/response objects. They generally do not accept or return domain entities or valueobjects.
- **Domain services** are unaware of infrastructure or overall application flow - they exclusively encapsulate business logic rules. They accept domain entities or value objects, carry out conditional operations on those entities or objects, or perform business rule calculations, and then return primitives or domain entities or value objects.

Based on these concepts, your sample service is definitely an application service, as it is interacting with your repository and unit of work, and returning a "UserResponse" type (a 'response' type does not sound like a domain entity).

Your application service `AuthenticationService` is delegating to a service called `UserTokenFactory`. `UserTokenFactory` accepts a domain entity (user) and returns a domain valueobject (usertoken). Presumably it encapsulates in an infrastructure-agnostic way the business rules associated with creating the user token. As such, this looks like more like a domain service. A factory which is responsible for the creation of domain concepts such as entities and value objects is just a special type of domain service (in my opinion) although you will most commonly see 'domain services' referring to services that perform some business logic that requires coordinating between multiple types of entities.

So - I think your structure here is appropriate - you have an application service coordinating infrastructure and flow, which delegates to a special service to execute the business logic.

answered Aug 16 '16 at 20:29



Chris Simon

4,492

1

9

22

Hey Chris! I really liked your succinct summary here. When you say the domain services: "then return primitives or domain entities or value objects" - can you please clarify in which cases? It'd be great to have a specific rule to follow. – Jimbo Aug 27 '18 at 20:32

For example, my current understanding is: If the domain service is going to interact with other domain services, then it's okay to return domain entities or value objects. If the domain service is returning to an application service, (because as you explained application services "do not accept or return entities or value objects"), then you would keep this separation by scalars. Does this sound right? – [Jimbo](#) Aug 27 '18 at 20:33 
