Organization of services in service layer?

Asked 8 years, 4 months ago Active 8 years, 4 months ago Viewed 156 times

0

I have a Java server application with a ton of different entities. So far, each entity top level entity has its own CRUD service. By top level, I mean the root of a tree of entities that can stand alone.

Now I am getting into the heart of my Flex client and find I am needing/writing many different queries. But where best to put these queries?

Say for example, I have a query to find all "foos" based on their associate with a certain "bar". Currently that query is on the "foo" service (findAllByBar), but I am finding that it would be very convenient to have it (also?) in the "bar" service (findFoos). On the other hand, I could also create a query service and lump all the queries in there.

Whats a good practice to do here?

java apache-flex api service

asked Oct 11 '11 at 14:18

HDave
18.8k 24 127 213

2 Answers

1

Try to layer your application in these perspectives:

- Domain: design your class as entities like "Customer", value objects like "Address" or "Color", and aggregate roots (like "Order" which includes a list of "LineItem")
- Repositories: these are the data access for the entities, create a repository for each aggregat root (CustomerRepository, OrderRepository, ...)
- Services: create a coarse grained services spitted by logical business abstractions or bounded context not by entities, it is not logical to create a service for order and a service for items and a service for customers when all these entities are representing one atomic business value of order processing, then your service will use all required repositories to handle the data access.

example:

```
public class OrderRepository {
  public Foo getById(int id) {
     //
  }

  public Foo getByCustomer(Customer customer) {
    //
  }
}

public class CustomerRepository {
  public Foo getById(int id) {
     //
  }

  public Foo getByUserName(string userName) {
    //
  }
}

public class TradingService {
  private OrderRepository _orderRepository;
  private CustomerRepository _customerRepository;

  public TradingService(OrderRepositoryInterface orderRep, CustomerRepositoryInterface
cusRep) {
    _orderRepository = orderRep;
    _customerRepository = custRep;
  }

  public void placeOrder(string customerUserName, Order order) {
    Customer customer = _customerRepository.getByUserName(customerUserName);
    order.setCustomer(customer);
    _orderRepository.add(order);
    // ....
  }
}
```

answered Oct 11 '11 at 15:30

Mohamed Abed

4,485 16 29

This is what I have today, but consider the case where the client as a Customer entity in hand and wants to get an order. Should that be a new finder on

the customer service or should the client invoke the query you already have on the order service? – HDave Oct 11 '11 at 16:17

If the View that has a Customer entity generates an event that requests an order by customer, what difference does it really make which Class fulfills that request? The View shouldn't need to know how that happens, and the Service shouldn't need to know where the request originated or what else the requesting object knows about the situation. – Amy Blankenship Oct 11 '11 at 17:18

1 HDave: you would add a finder in the order service to find order by customer .. as the entity u need is the order, so the order service is the proper place to provide this info .. also you may consider separating query service from business operations service sort of CQRS theory .. so you can group finders in OrderQueryService and business operations in OrderService – Mohamed Abed Oct 11 '11 at 20:39

OK, so you are saying that the entity type returned dictates the service it should go in? – HDave Oct 11 '11 at 22:17

1 Yes exactly, you request an order no matter what parameter needed then it is an order request, when you request a customer you are interested in returning a customer no matter what query inputs needed. – Mohamed Abed Oct 12 '11 at 9:05

1

I would put queries in their respective classes instead of creating one (bloatable)query service

answered Oct 11 '11 at 15:12

Just Me
244 1 5

OK -- But which class? or both? – HDave Oct 11 '11 at 16:17

Both foo and bar.. Bar service will have GetAllBarsByFoo() and foo service will have GetAllFoosByBar() method. It depends on which of those methods are required – Just Me Oct 11 '11 at 22:14

Just to clarify, in your opinion, must the GetAllBarsByFoo be in the Bars service because it returns Bars? – HDave Oct 11 '11 at 22:17

1 correct. As each class in our project should have high cohesion. This is one of rules of OO design. – Just Me Oct 12 '11 at 10:12