Domain Driven Design - External Data API as Respository or Service

Asked 10 years, 4 months ago Active 10 years, 4 months ago Viewed 3k times

9

4

In a blog application developed using domain driven design a 'Post' entity has a related collection of tag entities.

On creating the post (e.g populating the object from the UI) i would like to call a third party API via REST which takes the content of the post and extracts semantic tags (link text) for association.

Main Question : What is the best way to design this...

Is it best designed so that the Post entity would call a domain service such as PostServices.GetTags(Postcontent) passing its content and retrieving back a list of tags.?

** PostServices.GetTags would then inteface with the REST API via a further wrapper class.

Or should the third party API be wrapped as a repository?

Should the function Post.GenerateTags(), should not exist within the domain entity at all?

Further questions :

1 : I have also read that it is not good practice to have a domain entity converse with a domain service. Is this true?

2 : Is it ok to get a reference to the PostServices domain service via a factory creation method. e.g...

IPostService PostService = ServiceUtil.GetPostService(); return PostService.GetTags(Post.content);

3 : Is it acceptable to have the domain service coupled to a third party api?

4 : Should the domain entity simply just know how to deal with the tags received via the application layer which has called the REST API.

Slowly trying to get my head around DDD, however I can't seem to find any examples of how to implement this sort of thing.

repository domain-driven-design service infrastructure

asked Feb 16 '10 at 11:49

sjb101

111 1 5

1 Answer

Active Oldest Votes

3

should have:

- PostsRepository
- Post (Entity)
- Tag (value object)

A Post has got a list of tags.

Questions:

1 : I have also read that it is not good practice to have a domain entity converse with a domain service. Is this true?

Yes, itsn't a good practice. You entity hasn't want to be coupled with a domain service. If you do that, then you couldn't reuse them later. Did you have consider to fire a domain event. You can tell your service domain do something fire domain events.

2. : Is it ok to get a reference to the PostServices domain service via a factory creation method. e.g..IPostService PostService = ServiceUtil.GetPostService(); return PostService.GetTags(Post.content);

Yes, it's possible. A factory method could return an abstract class or an interface. This is a good Software Design Principle "code to an interface not to an implementation". If you do this, you'll be able to change your implementation later and you won't have to change yout client code.

3 : Is it acceptable to have the domain service coupled to a third party api?

I don't recommend you this, but it is acceptable.

Sorry, I don't understand question 4.

Look this link. I hope it help you.

https://stackoverflow.com/questions/901462/ddd-where-is-it-most-appropriate-to-get-a-list-of-value-objects/901562#901562

edited May 23 '17 at 11:53
Community ♦
1 1

answered Feb 16 '10 at 17:41
yeraycaballero
1,515 3 18 29

Thanks for the answer, however I am still a little unclear. Are you suggesting that the Post entity uses a factory to obtain a reference to a domain service which supplies tags. Also Tags are unique in my system and are identified via a primary key does this mean they are an entity ? – sjb101 Feb 16 '10 at 19:05

I have just found the following link which offers a good explanation of Value objects... tech.groups.yahoo.com/group/domaindrivendesign/message/6290 So Tag should be value object. Anyway back to the main question above...... – sjb101 Feb 16 '10 at 19:20

No, Post has got a List of tags. And you could retrieve tags from PostsRespository. A blog is a easy domain. You only have posts, tags, and comments. I don't undestand why you need a domain service. What is the responsability of yout domain service? – yeraycaballero Feb 16 '10 at 23:14

yeraycaballero - The main question was where would I handle the call to a third party API web service which returns a list of tags based on the content of a post. Should it be directly from in the Post domain entity? Function Post.BuildTag() tags = externalapiservice.gettags(me.content) end – sjb101 Feb 17 '10 at 18:03