

Keeping validation logic in sync between server and client sides [closed]

Asked 16 years, 3 months ago Modified 2 years, 7 months ago

Viewed 1k times



9



Closed. This question is [opinion-based](#). It is not currently accepting answers.



Want to improve this question? Update the question so it can be answered with facts and citations by [editing this post](#).

Closed 6 years ago.

[Improve this question](#)

In my [previous question](#), most commenters agreed that having validation logic both at client & server sides is a good thing.

However there is a problem - you need to keep your validation rules in sync between database and client code.

So the question is, how can we deal with it?

One approach is to use ORM techniques, modern ORM tools can produce code that can take care of data validation prior sending it to the server.

I'm interested in hearing your opinions.

Do you have some kind of standard process to deal with this problem? Or maybe you think that this is not a problem at all?

EDIT

Guys, first of all thank you for your answers.

Tomorrow I will sum up you answers and update question's text like in [this case](#).

database

architecture

Share

Improve this question

Follow

edited May 9, 2022 at 6:26



Viradex

3,728 ● 3 ● 13 ● 39

asked Sep 2, 2008 at 14:16



aku

124k ● 33 ● 176 ● 203

4 Answers

Sorted by:

Highest score (default)





4



As mentioned in one of the answers to the other post, if you are going to keep your layers separated, there is no good way to avoid duplicating the validation logic in each layer. If you use something to automatically tie them together, you have introduced a sort of coupling between the layers that might hinder you down the road. This might be one of those cases where you just have to keep track of things manually.

However you go about it, you have to make sure each layer is doing its own validation, because you never know how that layer is going to be accessed. There's no guarantee that all the layers you implemented will always stay together.

Share Improve this answer

answered Sep 2, 2008 at 14:24

Follow



Eric Z Beard

38.4k ● 27 ● 101 ● 147



2



I like to use a validation service, which doesn't necessarily care about the origin of the data to be validated. This can work in a few different ways when you get to the part about transmitting validation rules to a client (i.e. web page), but I feel the most important aspect of this is to have a single authority for the actual validation rules.

For example, if you have validation logic on your data core entities, like a collection of `ValidationRule` objects that are checked via a `Validate` method - a very typical

scenario, then I would promote those same rules to the client (javascript) via a transformation.

In the ASP.NET world (the only one I can speak to) there are a couple of ways to do this. My preferred method involves creating custom validators that tie in to your UI widgets to fields (and all their validation rules) on your entities. The advantage of this is that all your validation logic can be bundled into a single validator. The downside is that your validation messages will become dense, since the validation rules are all tested at once. This can, of course, be mitigated by having your validation logic return only a mention of the first failure, etc.

This answer probably sounds sort of nebulous and unspecific, but the two points that I'd like to make are:

1. Validation should occur as close as possible to the points where data is entered and where it's committed.
2. The same validation rules should be used wherever validation occurs - if client-side validation passes, then it should never fail validation later on (pre-save business rules, foreign key violation, etc.)

[Share](#) [Improve this answer](#)

answered Sep 2, 2008 at 14:38

[Follow](#)



[user2870](#)

31 ● 1



Some framework provides a validation support the may keep your client and server validation in sync. Take a look

2

at this [Seam validation tutorial](#) using annotations. It's a good implementation and very easy to understand.



Anyway, if you don't want to rely on frameworks, I think it is easy to implement something similar.

Share Improve this answer

answered Sep 2, 2008 at 14:48

Follow



Marcio Aguiar

14.5k ● 6 ● 40 ● 42



1



If you're using ASP.Net there are a number of validation controls you can use. These controls are written in a very generic way, such that most of them automatically duplicate your validation logic between the client and server, even though you only set options for the control in one place.

You are also free to inherit from them to create additional domain specific validators, and there are third-party control packs on the web you can get that add to the base controls.

Even if you're not using ASP.Net it's worth taking a look at how this is done. It will give you ideas for how to do something similar in your own platform.

Share Improve this answer

answered Sep 2, 2008 at 14:51

Follow



Joel Coehoorn

415k ● 114 ● 577 ● 813