

Automatically Duplicating Business Logic on Web Client

Asked 16 years, 1 month ago Modified 9 years, 7 months ago

Viewed 516 times



2



Last week I implemented a date validation in our front end, a combination of ASP.NET and heavily dependent on lots of JavaScript code to avoid server round-trips until a form is actually saved. I felt it clumsy that this rule check is not done on the server, but our current architecture and performance requirements prevent this. It's out of my hands for now.

Ideally, this check should be done in both places, but then the server side check would be done with neat, typed, C#, and immediately visible to developers working on that BO, and the client side check is actually done by not even a copy, which is dodgy, but completely different code.

What ways could there be to actually duplicate the server side check on the client side? Using a rule engine, and having an identical rule applied by two trusted rule engines on each side, actually delegating the server check to be done by JavaScript, which is then registered

in the rendered client seems like another option, but seems challenging.

Any ideas on this rather academic versus practical question?

validation

Share

Improve this question

Follow

edited Nov 5, 2008 at 15:29



MusiGenesis

75.2k ● 41 ● 197 ● 338

asked Nov 5, 2008 at 14:48



ProfK

51k ● 126 ● 414 ● 796

6 Answers

Sorted by:

Highest score (default)



2



ASP.NET's Validation Controls can do this.

ASP.NET validation controls also provide two ways of validation: Server-side or Client-side. The nice thing about these Validation controls is that it will preform client-side validation when it detects the browser is able (unless client-side validation has been disabled). Thus reducing roundtrips. And it will preform server-side where necessary. This client-side/server-side detection

and validation is done without extra work by the developer!

http://www.devhood.com/Tutorials/tutorial_details.aspx?tutorial_id=46

Otherwise, you would have to follow the same model. Create set rules, then implement each of them on both sides (in controls, if you can). Then, allow the developer to select which rule to apply to which html controls.

Share Improve this answer

answered Nov 5, 2008 at 14:55

Follow



[EndangeredMassa](#)

17.5k ● 8 ● 56 ● 80

Still the same problem of server-side validation and client-side validation being performed by different code.

– [MusiGenesis](#) Nov 5, 2008 at 15:21

That's not really a problem if you are reasonably sure that they will execute the same logic. And, with ASP.NET's controls, I am. – [EndangeredMassa](#) Nov 5, 2008 at 15:27

My problem is that these controls are too cohesive for our highly coupled hybrid of various home brewed creations. Like deep hierarchies of controls rendered by Repeaters.

– [ProfK](#) Nov 5, 2008 at 16:38

Well, you can either use these or write your own versions of these. Either way, this appears to be the only effective way to get a reliable duplication of client-side and server-side validation logic. Otherwise, you're stuck doing it twice for each validation instance. – [EndangeredMassa](#) Nov 5, 2008 at 19:04

No, I may want to do it twice, one each side, but not 'code' it twice. – [ProfK](#) Nov 5, 2008 at 20:00



Well, you'll never be able to automatically duplicate server side validation code in the client, because

1

1. the languages (javascript vs. C#) are too different



2. the server side code often will have data available that is not present in the client



However, it's perfectly feasible to delete the validation code to the server altogether. Certainly, for exactly the same two reasons, that's the best place for such code. Using Ajax techniques, and the WebRequest object, a short validation stub in the client can quite easily call a server-side validation function with no need for a postback.

Share Improve this answer

answered Nov 5, 2008 at 14:56

Follow



[Tor Haugen](#)

19.6k ● 9 ● 47 ● 64

The validation could possibly be done in JavaScript on the server, and there is a C# to JavaScript compiler that could be used to translate server code to client code. In my case, the client has more information than the server, before the document is saved. – [ProfK](#) Nov 5, 2008 at 16:35

It's not a postback I'm trying to avoid, but any server call, even an AJAX call. – [ProfK](#) Nov 5, 2008 at 16:35

"you'll never be able to" -> That's a strong statement. I think you mean, "it's not possible *now*." It's not outside the realm of possibility to write run a JS interpreter on the server or a C# interpreter on the client. It's also not unheard of to build your own domain specific language (DSL) and to auto-generate parsers in various languages. – [Mark E. Haase](#) Apr 23, 2012 at 17:46



0



I've never done this, but it seems like you could write your validation code in one place on the server, then have your web application access the validation code directly while each client page accesses the same web-service-exposed code using AJAX.



In the .NET world, this could be done simply by writing the validation code as a web service. The AJAX-clients and the server app itself would call methods in this web service to do validation (note: this might be a bad idea from a security perspective - I don't know).

Share Improve this answer

answered Nov 5, 2008 at 15:23

Follow



[MusiGenesis](#)

75.2k ● 41 ● 197 ● 338

As I keep saying, a server round trip is not desirable, i.e. any server round trip, not just a postback, but even an AJAX call. This is an intranet application, with some sites running single, very weak web servers. Every request matters. – [ProfK](#) Nov 5, 2008 at 19:58

@ProfK I don't understand why you accepted an answer that didn't answer your question. – [Mark E. Haase](#) Apr 23, 2012 at 17:49



0

I think a simple rule engine is your best option. Depending on the validation you want to perform it needn't be too complex.



Anything else will involve additional round trips to the server (to perform all validation in C#) or getting the server to execute the JavaScript (which is much harder, I believe, than implementing a rules engine).



The only alternative is duplicating the code (just as Microsoft does).

Share Improve this answer

Follow

answered Nov 6, 2008 at 11:48



[Kramii](#)

8,419 ● 4 ● 34 ● 39



0

See: [MVC Validation - Keep it DRY with a service layer - What is best practice?](#)



The best solution I could find is the one that recommended decorations on your view models, which using MVC, can produce some validation on both the client and server.



Share Improve this answer

Follow

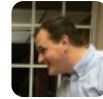
edited May 23, 2017 at 11:58



Community Bot

1 • 1

answered Sep 27, 2012 at 16:06



Tim Ferrell

1,366 • 3 • 18 • 40



0



The good practice is that the system entry should have the validation to make sure consistency is maintained.

So, in your scenario the entry of your system is the web application back-end. So you need to have the validation on that side. Let me give you an example, let's say I create an automation tools that posts data to your web application and no UI is involved then, still if the tool does not provide the firstName, it should fail to record data.

From your explanation it seems your web application contains the service layer as well but this will be much more obvious when your system is divided into a service layer and a web application (as the dumb UI) just to collect users input.

Now you still need to (or at least it is better too) put ui validation for UI as well as you mentioned for a better user experience. UI validation should not be placed for to really prevent users from leaving the field empty in terms of data consistency, because service layer does not (and should not) let users anyway. But it is more for a better user experience (UX) so the users get a faster feedback.

Sometimes, the types of checking that happens on UI and Service layer are similar, in terms of logic. Let say

first name should not be empty. Then it can be possible to share the logic, somehow, but many times, you cannot do the checks on client side, because you need some external data, resources, information... for validation. For example numbers of available stock item cannot be less than 10. In this case you cannot have that information, or if you could, it is irrelevant. Because you must be validated on the saving part, which will probably be changed by the time user submits the form.

So if this is a scenario that you have fixture of these two types of validation, then it is better to use a service side validation, and on client side also, you pass information (in async manner) to service and return the result back. Something like [this service side decision](#). If you have the mix of these validations in one scenario, then I would not even bothering splitting them on UI only and service validations, for the entire validation I would go to service.

Share Improve this answer

edited May 21, 2015 at 0:15

Follow

answered May 21, 2015 at 0:08



Arash Aghlara

350 ● 3 ● 14