ASP.Net MVC should you create a model for each view?

Asked 9 years ago Active 1 year, 1 month ago Viewed 2k times

8

I'm fairly new to ASP.Net MVC and I find myself making a single model class for every view. E.g. SignInModel, SignUpModel, EditProfileModel and so forth.

Many of these are somewhat similar, having the same DB fiels, and then a few custom properties.

Is this really how MVC is supposed to be done, or did I miss some aspect of it ?

I realize this may be subjective, but there ought to be a "best-practice" around this.

4

asp.net-mvc

asked Feb 22 '11 at 11:17

Steffen

12.1k 7 47 64

1 Answer

15

> Is this really how MVC is supposed to be done, or did I miss some aspect of it ?

That's absolutely how it MVC is supposed to be done and you aren't missing any aspect of it.

A View Model per View.

✓

You could still have base view model classes and use inheritance of your view models but be careful especially with the validation rules which might be different between the different views. So for example if on one view a property was required it might no longer be required on another view so if you have used a base view model class and data annotations you would be pretty much toasted.

Do not for one second hesitate to create view models even if you repeat some properties. That's what view models are meant for => respond to the specific needs of a view and bear in mind that those needs could change quite often, so having separate view models might look difficult at the beginning but at the long term it is the best solution both in terms of maintenance and good practices IMHO.

answered Feb 22 '11 at 11:18
Darin Dimitrov
903k 237 3106
2813

1 I agree with Darin. Use View Models as required to meet your needs within the view. Don't be afraid of having lots of View Models. – abarr Feb 22 '11 at 11:28

1 Would you create a VieModel even if you already had enough data in a Model? I find the ViewModel useful for when you have data you need to pull data from disparate Models, but if I have all the data the view needs in one Model I pass this down. Where is the benefit in maintaining a set of ViewModels that are all only a subset of a single Model? – DannyLane Feb 22 '11 at 13:07

Excellent, thanks a lot for elaborating on this :-) I'll stick with the many models. Particularly I find the part about validation rules very helpful - I hadn't thought about that. – Steffen Feb 22 '11 at 13:15

4 @DannyLane, don't forget that a view model could also be used to store view-specific flags / state / etc. Stuff to support the view that isn't necessarily related to the models themselves. – Mark Freedman Feb 22 '11 at 14:22