

Software Engineering Stack Exchange is a question and answer site for professionals, academics, and students working within the systems development life cycle. It only takes a minute to sign up.

Join this community

Anybody can ask a question



Anybody can answer

The best answers are voted up and rise to the top



Mapping Domain models to ViewModel

Asked 4 years, 2 months ago Active 4 years, 2 months ago Viewed 9k times



I have an Asp.net MVC5 application structured like this:

5

Core



1. Domain project



Infrastructure



2. DAL project

3. Utilities

UI

4. UI project (contains ViewModel at the moment, will probably put that into separate project as I expect that application will grow through time)

It works through UnitOfWork and a generic repository. Injected with Unity.

By using our site, you acknowledge that you have read and understand our Cookie Policy, Privacy Policy, and our Terms of Service.



I'm strictly trying to use ViewModels for my UI Views, where should I put the logic to map Domain Model \Leftrightarrow ViewModel?

Should I do it in the Controller or perhaps put the mapping logic into the ViewModel itself and create methods `ViewModel.FromDomainObject(DomainObject domainObject)` or `ViewModel.ToDomainObject(ViewModel viewModel)` etc.?

asp.net-mvc

entity-framework

mvvm

asked Dec 29 '15 at 20:22



Iztoksson

187 2 11

1 Answer

Keep mapping implementation out of your controllers. Instead, *call* the thing responsible for mapping in your controller.

6

```
public ActionResult FetchSomething()
{
    var thingMapper = new ThingMapper();
    var viewModel = thingMapper.ToViewModel(service.FetchSomething());
    return View(viewModel);
}
```



Of course, if your ViewModel is just a property or two, you can skip the mapper.

```
public ActionResult FetchSomething()
{
    var thing = service.FetchSomething();
    var viewModel = new ThingViewModel { Prop1 = thing.Prop1 };
    return View(viewModel);
}
```

With a little luck, [AutoMapper](#) might help you map objects without writing custom mappers.

I wouldn't store mapping implementation in the ViewModels themselves. This gets awkward when one ViewModel can represent data from more than one source. It also mixes responsibilities a little too much for my tastes.

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).





Scant Roger

8,868

2

26

46

I made a simple mock yesterday and put mapping to the ViewModel itself, it works but as you say it not really SRP friendly. I think I will just go ahead and make a ViewModel project in UI level folder and separate mapping from ViewModel itself. Much like I have with Domain models and configuration/mapping classes. – [Iztoksson](#) Dec 30 '15 at 7:23

