# When should I create a new controller class in ASP.NET MVC?

Asked 10 years, 8 months ago Active 10 years, 8 months ago Viewed 4k times

7

1

I'm learning MVC and am having trouble deciding when I should create a new controller versus just adding an action and view associated with an existing controller. On the one hand, Single Responsibility would seem to say a controller should be limited to a few actions. When I try this, though, the number of classes grows exponentially (model, views and controller for each)- to the point I wonder if I'm going overboard.

For example, the default AccountController has Login, ChangePassword, and Register actions. I would tend to instead create a LoginController, PasswordController, and ProfileController, and related model classes. So where there was 1 class, there would be 3-6.

Is there any good rule of thumb on this?

asp.net-mvc design-patterns

asked Jan 7 '09 at 23:31
Daniel
**1,106** 1 10 23

Maybe this is helpful (from Ruby): stackoverflow.com/a/8050513/1627888 – Yo Ludke May 21 '14 at 7:22

## 3 Answers

7

I think you need to be pragmatic about it. I'm working on a project that consists of a StatsController. The number of actions is continuously growing (RandomStat, MostPopular, MostViewed, MostVoted, etc...) the list goes on and on. These actions are simple to satisfy since the StatsController's dependencies don't change. I'm using an IoC to satisfy what my Controllers need and when I start to see my Controllers needing references to new objects, this is a signal that they need to be broken apart.

**4,684** 2 22 42

10

You should dedicate a controller for each model type you're manipulating. The controller acts as a collection of actions that act upon those models. This is generally the rule of thumb, but sometimes a controller's scope transcends a single model.

The AccountController deals with all things authentication related. This is an example of going beyond a single model's scope to encompass authentication in general. What are the key parts of authentication? Retrieving users, changing passwords, etc.

answered Jan 7 '09 at 23:36
Soviut
**61k** 37 150 217

I am working on a Rails application which does not have any model since all the info is being fetched by external API. How can I decide when to create a new controller in this case ? There is no storage required in the application. The main entity in my application is flight with API support for searching, checking availability, booking etc. TIA – furiabhavesh Jun 24 '15 at 13:11

1 @furiabhavesh That should be a new question, not a comment. You can map your controllers to your API controllers. – Soviut Jun 24 '15 at 16:47

0

My current AccountController has 12 methods which to me is completely manageable.

I have another controller which currently has 34 methods but those are all tied to a single view and they each have around 8-10 lines of code max (check required parameters, update the model, and redirect as necessary).

They key is to encapsulate your *business logic* in an entirely separate module. That will allow your action handlers to remain extremely light weight and can make testing your business logic easier.

answered Jan 8 '09 at 6:03
Todd Smith
**12.3k** 11 49 74