

# What is the purpose of the Model in MVC? [closed]

Asked 4 years, 8 months ago   Active 4 years, 8 months ago   Viewed 2k times

**Closed.** This question needs to be more [focused](#). It is not currently accepting answers.

 **Want to improve this question?** Update the question so it focuses on one problem only by [editing this post](#).

Closed 4 years ago.

I have read a lot about MVC, and I have even written a modestly sophisticated data-driven MVC mobile web app in .Net, but (since I am still learning), I did not see what to put in the model. The app had a dozen or so views, a primary controller, and helper files for SQL interactions and custom data types. I was able to data bind just fine without using the model, and custom data types worked as anticipated. Please be kind, what am I missing? What is the purpose of the model, and why should I be compelled use it (other than it being a standard)?

c#

asp.net-mvc

model-view-controller

edited May 22 '15 at 16:14



Robert Harvey


160k 41 299 447

asked May 22 '15 at 16:07




user4855057

165 1 1 10

The "helper files for SQL interactions and custom data types" might have been models, without you realising. Difficult to say without seeing what you came up with – [Rowland Shaw](#) May 22 '15 at 16:09 

2 I would grab a MVC book, learn the basics before asking. Cheers mate! – [TGarrett](#) May 22 '15 at 16:10

Wow, you guys definitely aren't kind. Thank goodness people were mean to you when you were learning so you could pass it along to the next generation of "I never had to learn anything, I just knew it because I'm a programming god" generation. I have read many books, and MVC still doesn't make sense. I wanted practical reasoning from experienced programmers, and I got people downvoting because they're too good for a newb question (other than Rowland, who has a good mentoring mindset). Now I have to delete my account so my rep isn't destroyed. TGarret - you've never been stuck on something? – [user4855057](#) May 22 '15 at 16:15 

If you ever see something like `@model IEnumerable<MvcMovie.Models.Movie>` at the top of your View, it's likely you're using the `Model` and you just

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



## 4 Answers



Simply put - "The MVC Model contains all application logic (business logic, validation logic, and data access logic), except pure view and controller logic.

1

With MVC, models both hold and manipulate application data."



You are not following the standard MVC pattern if you ignore the Model all together. However, as you found out - you don't need to use it. You can call into the database directly from the controllers and return types you've defined in the DAL (Data Access Layer) - which indirectly you are using as the Model in your design. This isn't good practice.



You should be compelled to follow the pattern so you achieve 'Separation of Concerns' in your application's architecture. You can find a lot of information online about this topic as it relates to the MVC pattern.

[Model Definition](#)

answered May 22 '15 at 16:42



[William Xifaras](#)

4,371 2 13 19



MVC is primarily a *User Interface design pattern*. The Model is a container for "everything else," everything that doesn't have to do with UI.

4

In general, you push as much logic as you can back from the View into the Controller or a ViewModel, and you make your controllers as thin as possible by pushing as much logic as you can back into the Model. So the short answer to "what goes in the Model" is "*everything that doesn't go into a View, ViewModel or Controller.*"



Specifically:

1. Domain objects, like `customer`
2. Services

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



5. Object-Relational mappers

6. Database

7. Repositories

edited May 22 '15 at 16:19

answered May 22 '15 at 16:12



[Robert Harvey](#)

**160k** 41 299 447



The View is a template that can be filled by passing a model containing data into it.

0

It allows the view to be customized depending on what the model contains.



How else are you filling data into your view?



Maybe jQuery gets?

answered May 22 '15 at 16:11



[OZZ](#)

**3,887** 44 69



In your case, the "helper files for SQL interactions and custom data types" is your model. The model is an abstraction over the true data source, eg the database.

0

For small projects, what you have done works just fine. As projects get larger though, binding the view directly to the database/model and not fully abstracting the model causes more and more problems.



answered May 22 '15 at 16:11



[David Arno](#)

**38.3k** 15 75 121