C#Corner

Learn Flutter Programmin

What is Model and ViewModel in MVC

Become a member   Login

Post   Ask Question

[Download Free .NET & JAVA Files API](#)
[Try Free File Format APIs for Word/Excel/PDF](#)

Model and ViewModel are two things we always hear about in MVC. In this article I will show you the differences between them.

Let's begin with its common definition.

What is Model or Domain Model?

Actually, the word "model" has hundreds of meanings in software development, but here we will talk about "model" in the context of a MVC design pattern. I would define a model as an object that we use to send information to the database, to perform business calculations and to render a view. In other words, "model" represents the domain of the application that helps us to save, create, update and delete records. Usually we put all our model classes in the Model folder.

What is ViewModel?

ViewModel in the MVC design pattern is very similar to a "model". The major difference between "Model" and "ViewModel" is that we use a ViewModel only in

rendering views. We put all our ViewModel classes in a "ViewModels" named folder, we create this folder.

C#Corner   Learn Flutter Programmin   ecome a member   sig.   Login

Post   Ask Question

Let's assume we want to implement a view page that will have three textboxes for Username, Password and Re-enter Pass
as given below:

```
{
    public String Username { get; set; }
    public String Password { get; set; }
    public String RePassword { get; set; }
}
```

For the sake of the view this model works fine. But this is actually an incorrect approach because we are trying to overcrowd the database. I can't see any use of the "RePassword" property in the database.

Now, if we take the advantage of a ViewModel, we can safeguard the database from redundant data. Here's how, design the following "Model" that will be our Domain Model:

```
//this will represent domain of the application
public class Login
{
    public String Username { get; set; }
    public String Password { get; set; }
}
```
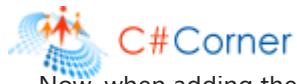
And then following "ViewModel"
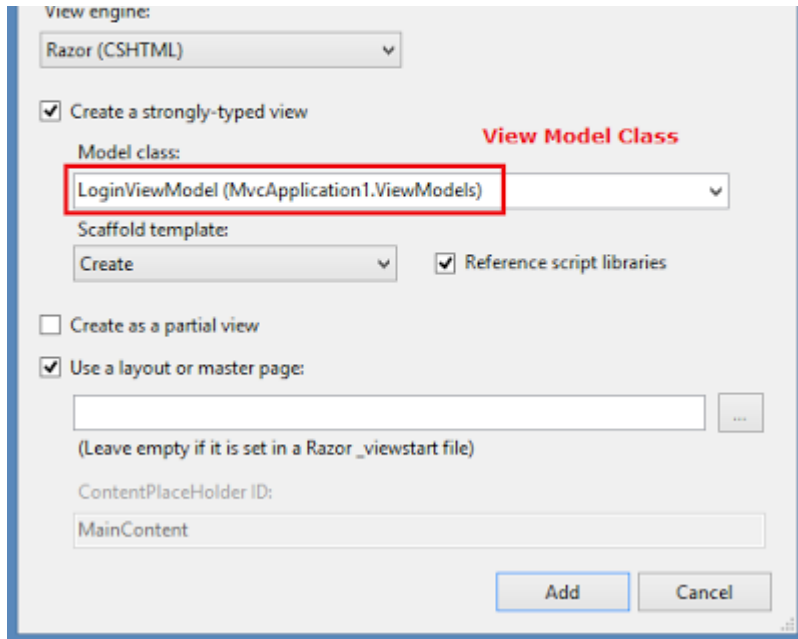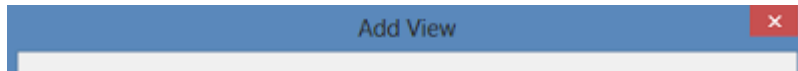
```
//this will help in rendering great views
public class LoginViewModel
{
    public String Username { get; set; }
```

```
public String Password { get; set; }
public String RePassword { get; set; }
```

Now, when adding the view, pick the ViewModel class to get the strongly-typed benefits.

Now the question is, how do we transform the Model or Domain Model from the "ViewModel"? Let's see how.

Transforming Model from ViewModel

There are various ways to do this. In the form POST action we can create a new object of type Login model and then assign the properties one by one and leave the unwanted properties.

```
[HttpPost]
public ActionResult Login(LoginViewModel viewModel)
{
```

```
//validate the ViewModel
```

C#Corner   Learn Flutter Programmin

```
var login = new Login()
{
    Username = viewModel.Username,



//save the login var



    return View();
}
```

In the code above, after validating the ViewModel I'm transforming the Model or Domain Model. So, by using this way you can stop overcrowding the database with unnecessary fields.

Let's look at one more use of ViewModel

I've a blog where I want to display the list of the latest posts, latest comments, post categories in a single view. How I can do that? I can strongly type my view to any one of the models, right? Here comes the view model.

I created a view model called BlogViewModel that contains the latest posts, latest comments and other stuff as properties and I bind my view with this model. The posts and comments are domain models while the BlogViewModel is the view model I created specially for the view.

Tomorrow I'll show my blog in a mobile version and at that time I may create a simple view model that contains only fewer properties. Finally, view models are for views and usually they acts as wrappers over the real domain models.

Hope this helps.

C#Corner   Learn Flutter Programmin

C# Corner   Become a member   Login

Next Recommended Article   Post   Ask Question

MVC4 Mobile Application Using View Model Architecture

(Model in MVC Pattern) (MVC Pattern) (ViewModel in MVC Pattern)

Abhimanyu K Vatsa *TOP 50*

Senior Software Developer | Microsoft MVP | Author | Speaker | Blogger at ITORIAN.COM

http://www.itorian.com

4   **15.8m**   **2**   **5**

View Previous Comments
     10   **14**

Type your comment here and press Enter Key (Minimum 10 characters)

Your points in regard to the difference between a "Domain Model" and "ViewModel" are good. However your definition of a Model in the context of MVC is not good, and very confusing for people who don't already know. The nature of asp.net MVC makes this an easy mistake to make unfortunately. The Model portion of MVC refers to the layer that "directly manages the data, logic and rules of the application" (as per Wikipedia). Most people mistake the ViewModel as being the Model in Asp.Net MVC, but really it is only a part of it as the Model encompasses much more than that.

Brandon Moore   Jan 06, 2020
     1883 8 0   0   0   Reply

Yeah, sure it's help.

Brijesh Mavani

Dec 10, 2018

Thanks ..Helpful explained

morani farhad

Sep 17, 2018

1869   22   0

Become a member   Login

Post   Ask Question   Reply

1727   164   51.3k   1   0   Reply

Tq very much sir

viswanath p

Aug 03, 2018

1866   25   146   1   0   Reply

One of the finest example sir, thank Q.

Satish B

Jul 25, 2018

1764   127   0   1   0   Reply

A helpful article explained in simple words!

Dennis Thomas

Mar 01, 2018

752   1.8k   611.7k   2   0   Reply

Nice one .. Thanks for sharing

Amit Singh

Dec 12, 2017

519   3.5k   101.7k   1   0   Reply

By doing this, are we not breaking the MVC pattern ? Is this a best practice ?

gowshik raj

Jul 19, 2017

1882   9   0   1   0   Reply

Clearly explained and easy to understand.Thanks

Karthik Elumalai

Jan 07, 2017

374   5.2k   996.4k   1   0   Reply

C#Corner   Learn Flutter Programmin

C# Corner   Become a member   Login

Post   Ask Question

FEATURED ARTICLES

Getting Started with the Windows Azure Function App

Plugin Architecture using MEF Framework

My C# Implementation Of Basic Linear Algebra Concepts

C#Corner   Learn Flutter Programmin

Creating An Optical Character Reader Using Blazor And Azure Computer Vision

C#Corner

Post   Ask Question   ⋁ All ◯

TRENDING UP

01   Intro To Pattern Matching - Covers C# 9

02   Getting Started With Bot Framework Composer

03   Delegates In C#

04   Multiple Dynamic Greeting Message Using Bot Framework Composer

05   Advanced Features In C#

06   Type of Filters in MVC Application and Why They're Important

07   C#.NET - Access Authorized REST Web API Method

C#Corner   Learn Flutter Programmin

Become a member   Login

C#Corner

View All

Post   Ask Question
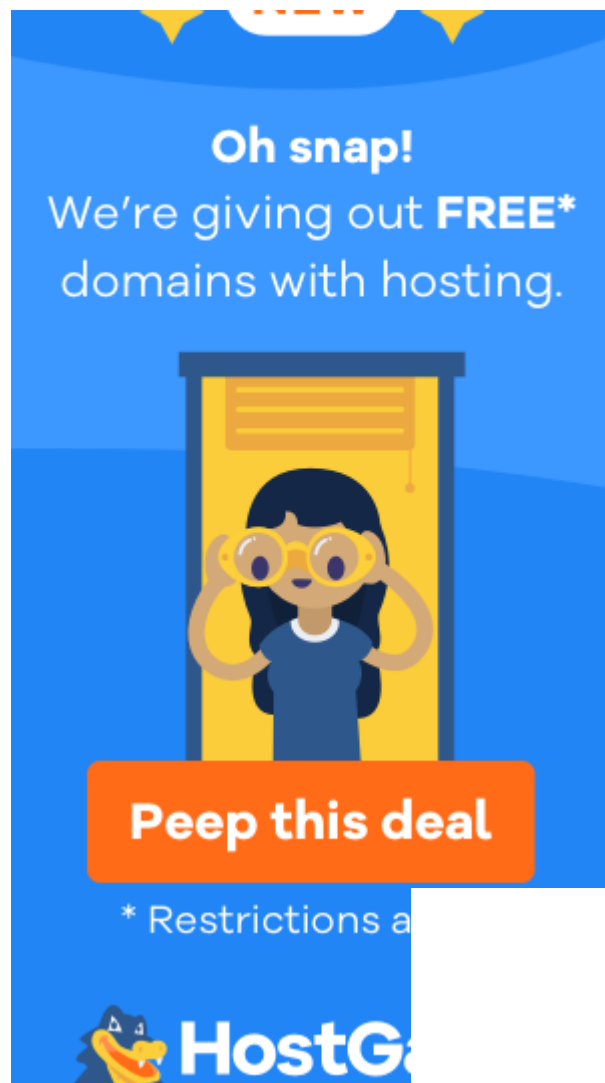
C# Corner

Learn Flutter Programmin

C# Tutorials   Common Interview Questions   Stories   Consultants   Ideas   Ce   tions

Become a member   Login

©2020 C# Corner. All contents are copyright of their authors.

Post   Ask Question