

Sampath Lokuge Tech Blog

If you can't explain it simply, you don't understand it well enough - Albert Einstein

[Home](#)
[Site Map](#)
[Disclaimer](#)
[Articles of The Day](#)
[Communities](#)
[About Me](#)
[Contact Me](#)
[Freelance Work](#)

[Share](#)
[Share](#)
[Tweet](#)

Monday, October 29, 2012

How to Use ViewModel with ASP.NET MVC

The Most Viewed Article On My Tech Blog



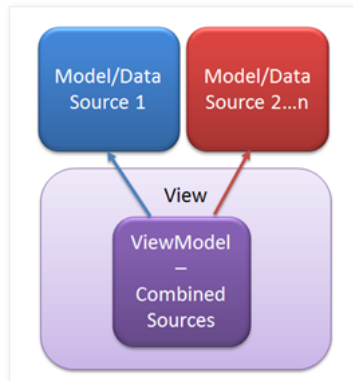
What is a Model

- Parts of the application that implement the domain logic.
- also known as business logic.
- Domain logic handles the data that is passed between the database and the UI.
- As an example, in an inventory system, the model keeps track of the items in storage and the logic to determine whether an item is in stock.
- In addition, the model would be the part of the application that updates the database, when an item is sold and shipped out of the warehouse.
- Often, the model also stores and retrieves model state in a database.

What is a ViewModel

- Allow you to shape multiple entities from one or more data models or sources into a single object.
- Optimized for consumption and rendering by the view.

It shows below image



Why We Use ViewModel

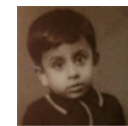
1. If you need to pass more than one thing to a strongly-typed view (which is best practice), you will generally want to create a separate class to do so.
2. This allows you to validate your ViewModel differently than your domain model for attribute-based validation scenarios
3. Can be used to help shape and format data.
e.g: need a date or money value formatted a particular way?
ViewModel is the best place to do it.
4. The use of a ViewModel can make the interaction between model and view more simple



Search This Blog

Subscribe To

- ☒ Posts
 ☐ Comments



Sampath
Lokuge
@sampathlok
Software
Developer (Senior)

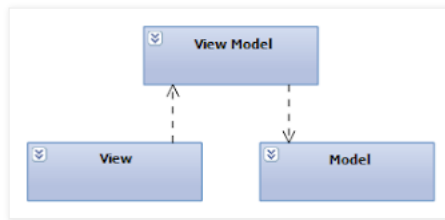
Authority | Scholar
[Articles](#)
[Messages](#)
[Reputation](#)



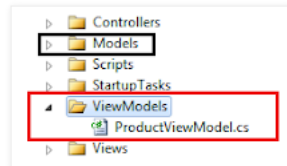
Sampath Lokuge T...
 88 likes

Be the first of your friends to like this

Follow Me

ViewModel Interaction with Model and View**Where Should We Create ViewModel (physically)**

1. In a folder called ViewModels that resides in the root of the project. (small applications)



2. As a .dll referenced from the MVC project (any size applications)
3. In a separate project(s) as a service layer, for large applications that generate view/content specific data. (enterprise applications)

Best Practices When Using ViewModels

1. Put only data that you'll render (use in View) in the ViewModel
2. The View should direct the properties of the ViewModel, this way it fits better for rendering and maintenance
3. Use a Mapper when ViewModels become Complex (i.e. [How to Use ValueInjector ?](#))

Let's Try with Simple Example

- Asp.net MVC 5, C# and Visual Studio 2013 express has been used.
- Please follow an In-line Comments on Code.
- This is a simple application with product category drop down list, product name text box and Save button.

Sample Project's Solution Tree**Tweets** by @Sampath_Lokuge

sampath lokuge
@Sampath_Lokuge

Rerukane Chandawimala Maha Thero -
Books 21 collection - bit.ly/2QSTFX9

Aluth.com: Rerukane ...
අනිසුරු රේරුකා...
aluth.com

Sep 29, 2019

sampath lokuge
@Sampath_Lokuge

Firebase Cloud Functions - Resizing
Images after Upload
youtu.be/YGsmWKMMiYs via @YouTube

YouTube @YouTube

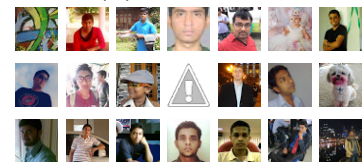

[Embed](#)
[View on Twitter](#)

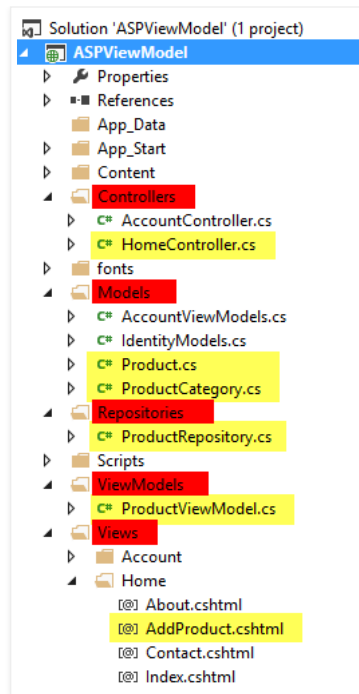
Visitors

US 92,945	AU 9,130
IN 84,771	DE 8,992
GB 22,595	RU 8,080
BR 12,665	FR 7,980
CA 11,352	VN 7,389
TR 11,133	PL 6,866

Pageviews: 549,439
Flags Collected: 212

FLAG counter

Followers
Followers (43) [Next](#)

[Follow](#)
Total Page Views
855,074



Domain Models

Product.cs

```
public class Product
{
    public Product() { Id = Guid.NewGuid(); }
    public Guid Id { get; set; }
    public string ProductName { get; set; }

    public virtual ProductCategory ProductCategory { get; set; }
}
```

ProductCategory.cs

```
public class ProductCategory
{
    public int Id { get; set; }
    public string CategoryName { get; set; }

    public virtual ICollection<Product> Products { get; set; }
}
```

ViewModel

ProductViewModel.cs

```
public class ProductViewModel
{
    public Guid Id { get; set; }

    [Required(ErrorMessage = "required")]
    public string ProductName { get; set; }

    public int SelectedValue { get; set; }

    public virtual ProductCategory ProductCategory { get; set; }

    [DisplayName("Product Category")]
    public virtual ICollection<ProductCategory> ProductCategories { get; set; }
}
```

Controller Action Methods

```
/// <summary>
```

```

/// to generate view with categories for entering product data
/// </summary>
[HttpGet]
public ActionResult AddProduct()
{
    //instantiate the product repository
    var productRepository = new ProductRepository();

    //for get product categories from database
    var productCategories = productRepository.GetAllProductCategories();

    //for initialize viewModel
    var productViewModel = new ProductViewModel();

    //assign values for viewModel
    productViewModel.ProductCategories = productCategories;

    //send viewModel into UI (View)
    return View("AddProduct", productViewModel);
}

/// <summary>
/// to save entered data
/// </summary>
[HttpPost]
public ActionResult AddProduct(ProductViewModel productViewModel) //save entered data
{
    //instantiate the product repository
    var productRepository = new ProductRepository();

    //get product category for selected drop down list value
    var productCategory=productRepository.GetProductCategory(productViewModel.SelectedValue);

    //for get all product categories
    var productCategories = productRepository.GetAllProductCategories();

    //for fill the drop down list when validation fails
    productViewModel.ProductCategories = productCategories;

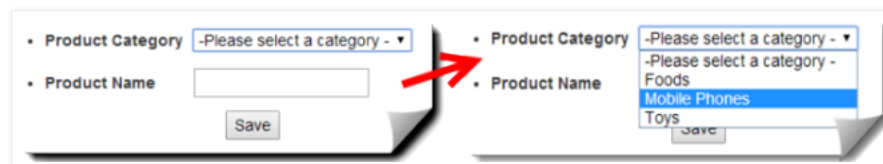
    //for initialize Product domain model
    var productObj = new Product
    {
        ProductName = productViewModel.ProductName,
        ProductCategory = productCategory,
    };

    if (ModelState.IsValid) //check for any validation errors
    {
        //code to save valid data into database

        return RedirectToAction("AddProduct");
    }
    else
    {
        //when validation failed return viewModel back to UI (View)
        return View(productViewModel);
    }
}

```

View



AddProduct.cshtml

```
@model ASPViewModel.ViewModels.ProductViewModel //set your ViewModel here
```

```
<div class="boxedForm">
```

```
@using (Html.BeginForm())
```

```

{
    Html.EnableClientValidation(true);

    <ul style="margin-top:50px;">
        <li style="width: 370px">
            @Html.LabelFor(m => m.ProductCategories)
            @Html.DropDownListFor(m => m.SelectedValue,
new SelectList(Model.ProductCategories, "Id", "CategoryName"), "-Please select a category-")
            @Html.ValidationMessageFor(m => m.ProductCategory.Id)
        </li>
        <li style="width: 370px;margin-top:10px;">
            @Html.LabelFor(m => m.ProductName)
            @Html.EditorFor(m => m.ProductName)
            @Html.ValidationMessageFor(m => m.ProductName)
        </li>
    </ul>
    <div class="action">
        <button class="actionButton" type="submit">
            <span>Save</span>
        </button>
    </div>
}
</div>

```

Repository Methods by using the Entity Framework

ProductRepository.cs

```

public class ProductRepository
{
    /// <summary>
    /// to get product category
    /// </summary>
    public ProductCategory GetProductCategory(int categoryId)
    {
        var productCategoryList = GetAllProductCategoriesMockDatas();

        return (from p in productCategoryList
            where (p.Id == categoryId)
            select p).FirstOrDefault();
    }

    /// <summary>
    /// to get all product categories
    /// </summary>
    /// <returns></returns>
    public List<ProductCategory> GetAllProductCategories()
    {
        var productCategoryList = GetAllProductCategoriesMockDatas();

        return (from p in productCategoryList
            select p)
            .OrderBy(p => p.CategoryName)
            .ToList();
    }

    /// <summary>
    /// to Get All Product Categories mock datas
    /// </summary>
    private List<ProductCategory> GetAllProductCategoriesMockDatas()
    {
        var productCategoryList = new List<ProductCategory>();

        productCategoryList.Add(new ProductCategory
        {
            Id = 1,
            CategoryName = "Foods",
        });

        productCategoryList.Add(new ProductCategory
        {
            Id = 2,
            CategoryName = "Toys",
        });

        productCategoryList.Add(new ProductCategory
        {
            Id = 3,
            CategoryName = "Mobile Phones",
        });
    }
}

```

```

        return productCategoryList;
    }

}

```

Conclusion

- ViewModels help you organize and manage data in MVC applications when you need to work with more complex data than the other objects allow.
- Using ViewModels gives you the flexibility to use data as you see fit.
- ViewModels are generally a more flexible way to access multiple data sources than domain models.
- Even for simple scenarios always try to use ViewModel approach to maintain consistency in Your coding practices.

I hope this helps to You. Comments and feedback greatly appreciated.

If you feel it was a good article, Give me a +1. Thank You.

You Might Also Like

- [How to Use ValueInjector with Asp.net MVC ViewModel](#)

Posted by **Sampath Lokuge** at 11:50 PM



90 comments:

Anonymous October 30, 2012 at 12:49 AM

Good info!

- Aleksandr Golovatuy -

[Reply](#)

[Replies](#)



Sampath Lokuge October 30, 2012 at 12:51 AM

Hi Aleksandr,

Thanks for your quick response.
Do Keep in Touch.

[Reply](#)



Danilo Cecilia October 30, 2012 at 5:29 AM

good stuff!

[Reply](#)

[Replies](#)



Sampath Lokuge October 30, 2012 at 5:37 AM

Hi Danilo,

Thanks for your response.
Do Keep in Touch.

[Reply](#)

Anonymous October 30, 2012 at 5:32 AM

Maybe you can recommend some good books about ASP.NET MVC

- Aleksandr Golovatuy -

[Reply](#)

[Replies](#)



Sampath Lokuge October 30, 2012 at 5:33 AM

Hi Aleksandr,

Yes.If you can follow "ASP.NET MVC in Action" books from "MANNING" Publications.Its really fine.

Latest book is "ASP.NET MVC 4 in Action".
Good Luck.

[Reply](#)



Hasitha Ariyaratna October 30, 2012 at 11:34 AM

Nice , superb clean explanation with guide .
Keep up the great job !
Cheers.

[Reply](#)

[Replies](#)



Sampath Lokuge October 30, 2012 at 12:04 PM

Hi Hasitha,

Thanks for your appreciation.
Do Keep in Touch.

[Reply](#)

Anonymous October 30, 2012 at 12:13 PM

Thanks Sampath.

Already we are working in a MVC Razor project.

-Aref Samad-

[Reply](#)

[Replies](#)



Sampath Lokuge October 30, 2012 at 12:16 PM

Hi Aref,

Nice to hear you.If you having any problem let me know.
Do Keep in Touch.

[Reply](#)

Anonymous October 30, 2012 at 12:22 PM

likes this.

- Akila De Alwis -

[Reply](#)

[Replies](#)



Sampath Lokuge October 30, 2012 at 12:23 PM

Hi Akila,

Thanks for your response.
Do Keep in Touch.

[Reply](#)

Anonymous October 30, 2012 at 4:28 PM

Very good article on ViewModel.

-Visharad Dhavle-

[Reply](#)

[Replies](#)



Sampath Lokuge October 30, 2012 at 4:29 PM

Hi Visharad,

Thanks for your appreciation.
Do Keep in Touch.

[Reply](#)

Anonymous October 30, 2012 at 6:55 PM

likes this.

-shyam chawda-

[Reply](#)

[Replies](#)



Sampath Lokuge October 30, 2012 at 6:55 PM

Hi shyam,

Thanks for your response.
Do Keep in Touch.

[Reply](#)

Anonymous October 31, 2012 at 3:36 PM

like this.

-Andrea Calvario-

[Reply](#)

[Replies](#)



Sampath Lokuge October 31, 2012 at 3:37 PM

Hi Andrea,

Thanks for your response.
Do Keep in Touch.

[Reply](#)

Anonymous October 31, 2012 at 7:49 PM

Nice. Congratulations!

[Reply](#)

[Replies](#)



Sampath Lokuge October 31, 2012 at 8:00 PM

Hi ,

Thanks for your response.
Do Keep in Touch.

[Reply](#)

Anonymous November 2, 2012 at 6:11 PM

like this.

-Esta Tsutsa-

[Reply](#)

[Replies](#)



Sampath Lokuge November 2, 2012 at 6:11 PM

Hi Esta,

Thanks for your response.
Do Keep in Touch.

[Reply](#)

Anonymous November 7, 2012 at 12:29 PM

good article sampath

-Buddhi Weragoda-

[Reply](#)

[Replies](#)



Sampath Lokuge November 7, 2012 at 12:29 PM

Hi Buddhi,

Thanks for your appreciation.

Do Keep in Touch.

[Reply](#)

Anonymous November 7, 2012 at 7:18 PM

Good stuff. Very clean explanation.

-Luís Barbosa-

[Reply](#)

[Replies](#)



Sampath Lokuge November 7, 2012 at 7:19 PM

Hi Luís,

Thanks for your feedback.
Do Keep in Touch.

[Reply](#)

Anonymous November 8, 2012 at 8:16 PM

Clear, Short and direct to the issue.

Great!!

-Emilio Plaza Garcia-

[Reply](#)

[Replies](#)



Sampath Lokuge November 8, 2012 at 8:17 PM

Hi Emilio,

Thanks for your appreciation.
Do Keep in Touch.

[Reply](#)

Anonymous November 9, 2012 at 7:15 PM

like this.

-Kalpani Hettige-

[Reply](#)

[Replies](#)



Sampath Lokuge November 9, 2012 at 7:16 PM

Hi Kalpani,

Thanks for your feedback.
Do Keep in Touch.

[Reply](#)

Anonymous November 14, 2012 at 8:58 PM

like this.

-Maria Manenti-

[Reply](#)

[Replies](#)



Sampath Lokuge November 14, 2012 at 8:59 PM

Hi Maria,

Thanks for your response.
Do Keep in Touch.

[Reply](#)

Mukesh Selvaraj November 19, 2012 at 6:51 PM



Hi,

Nice article.

In my current project also we used view models. which is more useful when we want to bind more than one model in our view means..

I am having some queries related to this article please clarify.

1. what is `Html.BeginAbsoluteRouteForm`? instead of using this we can use `Html.BeginForm` ?
2. `@Html.CompleteEditorFor` is this your custom html extension method right? If yes means please share the code it will give more clarity to understand the purpose of this control.
- 3.

```
//when validation failed return viewmodel back to UI (View)
```

```
return View(productViewModel);
```

In post method directly you are returning the model what you are getting while doing like this we will get exception in this line in view page

```
@Html.DropDownListFor(m => m.SelectedValue, new SelectList(Model.ProductCategories, "Id", "CategoryName"), "Please select a category -")
```

Because `Model.ProductCategories` will vanished. Before returning the model values again we have to re populate the items otherwise we have to use `ViewBag`.

Please excuse if above mentioned informations anything wrong means.

Thanks in advance.

Regards,
Mukesh Selvaraj

[Reply](#)

Replies



Sampath Lokuge November 20, 2012 at 1:29 AM

Hi Mukesh,

Thanks for your feedback.

A 1 :

Yes. Actually `@Html.BeginAbsoluteRouteForm` is a html helper extension method which we have written. It's not coming with default html helper methods of MVC.

You can find `@Html.BeginRouteForm()` instead of that which is coming with MVC.

If you need to look sample of above extension method. It's like below.

```
public static MvcForm BeginAbsoluteRouteForm(this HtmlHelper htmlHelper, string routeName,
object routeValues, FormMethod method, object htmlAttributes)
{
    return htmlHelper.BeginAbsoluteRouteForm(routeName, new RouteValueDictionary(routeValues),
method,
new RouteValueDictionary(htmlAttributes));
}
```

A 2:

Yes. Your right. It's an extension method for custom template. Sample code as below.

```
public static MvcHtmlString CompleteEditorFor(this HtmlHelper html, Expression> expression, string
labelOverride = null, object[] labelParameters = null)
{
    var components = new MvcHtmlString[] {
html.LabelFor(expression, labelParameters, labelOverride),
html.EditorFor(expression),
html.ValidationMessageFor(expression)
};

var outputs = components.Where(item => item != null).Select(item => item.ToString()).ToArray();
return MvcHtmlString.Create(string.Join("", outputs));
}
```

A 3 : Yes your right. I have missed some code there and updated it on original article. Please check that.

One Important thing I have to mentioned here. That is never use `ViewBag` or `ViewData` with your View. It's really ugly and difficult to maintain. Always try to use `ViewModel` as best practices (even for simple scenarios).

Good luck.



Mukesh Selvaraj November 20, 2012 at 7:14 AM

Hi Sampath,

Thanks for your reply. Please see my comments below.

A1: Will check and let you know.

A2: Will check and let you know.

A3: I am having little bit deviation here.

In my project i have one page it contains more than 20 drop downs.

For that i used ViewBags only. I agree with you it is some what goosy, It will slow down view page performance. But I think we can't fully avoid this ViewBag.

Do you have any good article about pros and cons and Best practices about ViewBag.

Thanks in advance.

Regards,
Mukesh Selvaraj



Sampath Lokuge November 21, 2012 at 1:18 PM

Hi Mukesh,

A3 :

Yes you can fully avoid ViewBag and you must to.

These are some reasons for that

The problem comes in from the fact that ViewBag is a dynamic object

There are no compile time type checks

If you misspell something you won't be notified from the compiler

Please check below link for more details

<http://completeddevelopment.blogspot.com/2011/12/stop-using-viewbag-in-most-places.html>

For Your performance issue:

You can use output cache or data cache method for boost your page performance heavily.



Mukesh Selvaraj November 21, 2012 at 4:00 PM

Hello Sampath,

Thanks for your reply. Surely I will taken care in my upcoming development tasks.

Reply



eshan February 10, 2013 at 9:19 AM

Hi Sampath,

Good article.

Reply

Replies



Sampath Lokuge February 10, 2013 at 12:50 PM

Hi eshan,

Thanks for your feedback.

Do Keep in Touch.

Reply



Haritha April 19, 2013 at 1:22 AM

Great post...!!!

But I have some problems with view models.

Can we move the logic inside the controller to the view model? (so that we can keep the controller thin; model fat and view dumb)

Ex: Let's add the logic inside the constructor of the ProductViewModel

```
public class ProductViewModel
{
    public ProductViewModel()
    {
        this.ProductCategories = Repository.GetAllProductCategories();
    }
}
```

```
[DisplayName("Product Category")]
public virtual ICollection ProductCategories { get; set; }
```

```
//Add other properties
```

}

Now the controller

```
[HttpGet]
public ActionResult AddProduct()
{
    var productViewModel = new ProductViewModel();
    return View("AddProduct", productViewModel);
}
```

OK. Let's concern how to handle the POST action

```
[HttpPost]
public ActionResult AddProduct(ProductViewModel productViewModel)
{
    if (ModelState.IsValid)
    {
        productViewModel.Add();
        return RedirectToAction("AddProduct");
    }

    return View(productViewModel);
}
```

Now let's implement the Add method in ProductViewModel

```
public class ProductViewModel
{
    public void Add()
    {
        this.ProductCategory = Repository.GetProductCategory(this.SelectedValue);
        Repository.AddProduct(this);
    }
}
```

[Reply](#)

[Replies](#)



Sampath Lokuge April 22, 2013 at 12:24 AM

Hi Haritha,

Thanks for your feedback.

About Your question:

You're totally wrong here when we're considering the MVC architectural pattern (separation of concerns). You must never call repository methods inside the ViewModel or Model. That thing must handle inside the Controller.

ViewModel only interacts with Models and View. Not with Repository.

If you need to slim the controller, then you can have number of ways to do that. Some of them are as below.

1. You can remove business related logic into separate service layer.
2. You can reduce lots of mapping related code inside the controller by using "ValueInjector Mapper". Check here for more info : <http://sampathloku.blogspot.com/2012/11/how-to-use-valueinjecter-with-aspnet.html>

I hope you got the point which I told. If you're still having issues about this I would like to hear it from you.

[Reply](#)

Anonymous June 20, 2013 at 2:18 PM

Great article, finally a complete example. However, I have a newbie question that I'm hoping you can clarify that has nothing to do with View Models. How do these work:

```
GetProductCategory...
GetAllProductCategories()
```

It's like you're calling something that doesn't exist (in the code) at least not in your example. Or does it? Microsoft magic? Can you just add "Get" and "GetAll" to the front like that to retrieve the ProductCategory and ProductCategories? Does it only work with Repository(ies)? Do you have a link maybe outlining the details of that? Hopefully this makes sense what I'm asking and I appreciate any help. I tried googling it with no luck. Thanks!

[Reply](#)

[Replies](#)



Sampath Lokuge June 20, 2013 at 10:12 PM

Hi sumnone,

Yes. You're right. Here I have used repositories for call above methods. Those are specific methods

for my App.I didn't put those repository methods on my code above.B'cos my idea was just explain how to use ViewModel with MVC.But anyway for your reference I have put those methods are as below.If you need more help Plz let me know.

```

--- GetProductCategory() ---

//for get product category
public ProductCategory GetProductCategory(int categoryId)
{
    return (from productCategory in Catalog.ProductCategories
    where (productCategory.Id == categoryId)
    select productCategory).FirstOrDefault();
}

---GetAllProductCategories()---

//for get all product categories
public List GetAllProductCategories()
{
    return (from productCategory in Catalog.ProductCategories
    select productCategory)
    .OrderBy(p => p.CategoryName)
    .ToList();
}

```

Anonymous June 21, 2013 at 2:48 AM

Excellent, I'm at the stage right now where if I don't see every piece of the puzzle, I'm left wondering. Anyway, I thought I might be missing out on a cool trick or something, but it makes perfect sense now. Thanks so much for your reply and as well, the article. Regards.

Reply



ANIL BABU Mandla July 4, 2013 at 5:21 PM

hi i can update data from database using Mvc,
This is my insert update code

At the time of inserting insert data successfully but at the time of Update not updateing the data Please See my code below

```

[HttpPost]

public ActionResult Create(HttpPostedFileBase imageFile, Actor model, string cmd)
{
    try
    {
        if (imageFile != null && imageFile.ContentLength > 0)
        {
            var fileName = Path.GetFileName(imageFile.FileName);
            if (cmd == "Create")
            {
                model.Image = fileName;
                db.Actors.Add(model);
                db.SaveChanges();
            }
            else
            {
                var actor = db.Actors.Where(m => m.ActorID == model.ActorID).FirstOrDefault();
                if (actor != null)
                {
                    actor.Image = fileName;
                    db.SaveChanges();
                }
            }
        }

        var path = Path.Combine(Server.MapPath("~/Content/Actors/"), model.ActorID.ToString());
        if (!Directory.Exists(path))
            Directory.CreateDirectory(path);
        imageFile.SaveAs(path + "/" + fileName);
    }
    catch { }

    return RedirectToAction("Index");
}

UPDATE:

[HttpGet]
public ActionResult Edit(int id)
{
    var actor = db.Actors.Where(m => m.ActorID == id).FirstOrDefault();
    ViewBag.IsUpdate = true;
    return PartialView("_Actor", actor);
}

[HttpPost]

```

```

public ActionResult Edit(HttpPostedFileBase imageFile, Actor model, string cmd)
{
    if (ModelState.IsValid)
    {
        try
        {
            if (imageFile != null && imageFile.ContentLength > 0)
            {
                var fileName = Path.GetFileName(imageFile.FileName);
                if (cmd == "Update")
                {
                    model.Image = fileName;
                    db.Actors.Add(model);
                    db.SaveChanges();
                }
                else
                {
                    var actor = db.Actors.Where(m => m.ActorID == model.ActorID).FirstOrDefault();
                    if (actor != null)
                    {
                        actor.Image = fileName;
                        db.SaveChanges();
                    }
                }
            }
        }
        catch { }
        return RedirectToAction("Index");
    }
}

var path = Path.Combine(Server.MapPath("~/Content/Actors/"), model.ActorID.ToString());
if (!Directory.Exists(path))
    Directory.CreateDirectory(path);
imageFile.SaveAs(path + "/" + fileName);
}
}
}
catch { }
return RedirectToAction("Index");
}
}

```

Reply

Replies



Sampath Lokuge July 4, 2013 at 5:35 PM

Hi ANIL,

Your UPDATE code is wrong.

Wrong Code :

```

if (cmd == "Update")
{
    model.Image = fileName;
    db.Actors.Add(model);
    db.SaveChanges();
}

```

Correct Code :

```

if (cmd == "Update")
{
    model.Image = fileName;
    db.SaveChanges();
}

```

Check this Article : <http://www.integralwebsolutions.co.za/Blog/EntryId/895/Entity-Framework-5-CRUD-Operations.aspx>

If you would like you can join with us on linkedin MVC 4 Community. Check this : http://www.linkedin.com/groups?home=&gid=5019062&trk=anet_ug_hm

I hope this will help to you.

Reply



ANIL BABU Mandla July 4, 2013 at 6:13 PM

thanks for replying,
modifying your code also not working.
Please correct my below code

I am new to mvc

```

[HttpPost]
public ActionResult Edit(HttpPostedFileBase imageFile, Actor model, string cmd)
{
    if (ModelState.IsValid)
    {
        try
        {
            if (imageFile != null && imageFile.ContentLength > 0)
            {
                var fileName = Path.GetFileName(imageFile.FileName);
                if (cmd == "Update")

```

```

{
    model.Image = fileName;
    db.SaveChanges();
}
else
{
    var actor = db.actors.Where(m => m.ActorID == model.ActorID).FirstOrDefault();
    if (actor != null)
    {
        actor.Name = model.Name;
        actor.Description = model.Description;

        actor.Image = fileName;
        db.SaveChanges();
    }
}

var path = Path.Combine(Server.MapPath("~/Content/Actors/"), model.ActorID.ToString());
if (!Directory.Exists(path))
    Directory.CreateDirectory(path);
imageFile.SaveAs(path + "/" + fileName);
}
}

catch { }
}
return RedirectToAction("Index");
}

```

[Reply](#)[Replies](#)**Sampath Lokuge** July 4, 2013 at 6:33 PM

Hi ANIL,

```

if (cmd == "Update")
{
    model.Image = fileName;
    db.SaveChanges();
}

```

Try below scenarios :

1. put a brake point inside the above method and check whether is it fire
2. If so then remove "model.Image = fileName;" line and put "model.Description="ANIL";" and then check whether your db gets updated.
3. If db updated then your problem is with image file check that.

[Reply](#)**ANIL BABU Mandla** July 5, 2013 at 11:33 AM

Thank you,
model.Description="ANIL"; Not updated in database,
I am putting break point Edit not fired At the update also fired "Create Control" is fired

[Reply](#)[Replies](#)**Sampath Lokuge** July 5, 2013 at 12:18 PM

Hi ANIL,

I am not clear what you're telling. Put more information about your findings.

When you click Update button, isn't come (or fire) "public ActionResult Edit(HttpPostedFileBase imageFile, Actor model, string cmd)" post Action method ? If so then your problem may be with routing ? You have to put your forms html code with related to this for check whether is there any issue. Plz provide better explanation about your findings.

[Reply](#)**Marvin.K.Serunjogi** October 18, 2013 at 1:36 PM

Thanks for the post. I have been searching for this.....

[Reply](#)[Replies](#)**Sampath Lokuge** October 18, 2013 at 6:16 PM

Hi Marvin,

Thanks, I'm glad you enjoyed it :)

Reply



Santosh Bhandari October 24, 2013 at 4:13 PM

one user post a detail information in a asp.net mvc4...
How to make secure this form ...other user cannot see,edit and delete the post the details of first user

Reply

Replies



Sampath Lokuge October 24, 2013 at 4:30 PM

Hi Santosh,

If you use ASP.net Membership provider for user management then according to the "User Role" of the user where you can enable or disable the Buttons which can use for edit or delete.

<http://www.asp.net/mvc/tutorials/mvc-music-store/mvc-music-store-part-7>

Reply



Enes Albay October 29, 2013 at 4:10 AM

Thanks for your great post.

I have problem with dropdownlist

In Create.cshtml file how can i get category using dropdownlistfor

Like this:

```
@Html.DropDownListFor(model => model.SelectedValue, new SelectList(Model.ProductCategories, "Value", "Text"))
```

or this:

```
@Html.DropDownListFor(model => model.ProductCategory, new SelectList(Model.ProductCategories, "Value", "Text"))
```

Reply

Replies



Sampath Lokuge October 29, 2013 at 11:43 AM

Hi Enes,

You're warmly welcome.

About your question : It depends the way you declare your VM. So if you follow the way which I showed above,you can use it as below (with default value).

```
@Html.DropDownListFor(m => m.SelectedValue,new SelectList(Model.ProductCategories, "Id", "CategoryName"),"-Please select a category -")
```

If you need more information,Please let me know.



Enes Albay October 30, 2013 at 3:15 AM

I'm grateful for your help. thanks

Reply



Rahul Murari November 19, 2013 at 12:50 AM

Please explain this given in Controller..

```
Repository.GetAllProductCategories(); // where that came
```

Reply

Replies



Sampath Lokuge November 19, 2013 at 12:54 PM

Hi Rahul,

Those are repository methods which I used for retrieve data.I have updated my post above for those details under 'Repository Methods'.Plz check that.

Reply



Andy Mai December 22, 2013 at 10:24 PM

Great article,

I do have question on the Model. If I am applying this ViewModel and use entity framework.

1. Should the product class has the ProductCategoryId?

2. In the product view model, can I use "public Product Product {get;set;} " instead of individual fields?

Thanks

Reply

Replies



Sampath Lokuge December 23, 2013 at 2:05 PM

Hi Andy,

Thanks :)

A 1: It depends on according to the table structure of your app. In my case it has 1:M relationship with the Product and ProductCategory tables. So I maintained the ProductCategory as a Traversal property on above example.

A 2: Yes you can. But the problem is then you unable to validate Product fields by using Field validate attributes (such as Required). B'cos the ViewModel is the best place to do those kind of Field validations.

Reply



Thiru Raja February 7, 2014 at 9:29 PM

Informative code flow

Reply

Replies



Sampath Lokuge February 8, 2014 at 10:36 AM

Hi Thiru,

Thanks, I'm glad you enjoyed it! :)

Reply



Moisés Alexander Salazar Vila March 19, 2014 at 11:55 PM

Wow man, this post is great.

Reply

Replies



Sampath Lokuge March 20, 2014 at 11:22 AM

Hi Moisés,

You're warmly welcome. :)

Reply



sanksk March 26, 2014 at 12:55 AM

Nice post. When you return RedirectToAction, it goes to get method where it creates a new viewmodel object. What if I want to use attributes/properties set to Viewmodel in post method and pass that viewmodel to get method?

I mean, if I say [HttpGet] public ActionResult Index(XYZViewmodel xyzViewModel) { } and [HttpPost] public ActionResult Index(XYZViewmodel xyzViewModel) { xyzViewModel.property1 = someValue; return RedirectToAction("Index", xyzViewModel); }

Will I get the same property coming back to get method? I tried doing in that way, but the property contents have been changed.

In the above example, I set property1 to someValue in post method. When I redirect to get method, the property has been changed to some other value other than someValue. Can you please let me know the solution of my problem. Sorry if I am not clear.

Reply

Replies



Sampath Lokuge April 2, 2014 at 6:03 PM

Hi Sanksk,

Thanks. :)

Answer For Your Question :

Hence you're redirecting to the Action method, You can use "TempData" with it. Please read this article for more info. <http://sampathloku.blogspot.com/2012/09/how-to-use-aspnet-mvc-tempdata-properly.html>

Reply**Anonymous** April 24, 2014 at 9:30 PM

Hi I have a problem with DropDownListFor on the Edit view.

Basically I'm using a partial view which contains my form and in my Edit and Create view I call this partial view.

I have around 5 similar DropDownListFor and these work well on create action but in edit doesn't, mainly I'm not getting (unable) to set the selected value.

In my Edit Action (GET), I fill my property ViewModel if the true object has the property filled.

```
if(icb.BAOfficer != null)
editICB.BAOfficer = icb.BAOfficer;
```

```
List staffs = _fireService.GetAllStaffs().ToList();
staffs.Insert(0, new Staff { StaffId = -1, Name = "" });
editICB.BAOfficers = staffs;
```

```
return View(editICB);
This is how I'm filling my drop down and how I'm trying to set the selected value.
```

```
@Html.DropDownListFor(model => model.BAOfficerSelected, new SelectList(Model.BAOfficers, "StaffId",
"Name", (Model.BAOfficer != null ? Model.BAOfficer.StaffId:-1)), new { @class = "rounded indent" })
@Html.ValidationMessageFor(model => model.BAOfficer.StaffId)
```

Reply**Rommel David** June 2, 2014 at 6:34 PM

Very useful article!
Thank you and God bless!

Reply**Replies****Sampath Lokuge** June 2, 2014 at 6:39 PM

Hi Rommel,

Thanks, I'm glad you found it useful ! :)

Reply**Mohanraj** June 3, 2015 at 10:57 AM

Thanks

Reply**Ayaz M** September 4, 2015 at 3:33 PM

Very Good Article! Simple and understandable !

Reply**niall300** October 23, 2015 at 1:41 AM

Excellent tutorial thanks, clear and concise, thanks

Reply**P.Seshathri** October 28, 2015 at 11:00 AM

Great article....

Reply**Diya Patel** November 3, 2015 at 5:35 PM

Nice article i was really impressed by seeing this article, it was very interesting and it is very useful for me.I get a lot of great information from this blog. Thank you for your sharing this informative blog.

Informatica Training in chennai
QTP Training in Chennai
Selenium Training in Chennai
Best Big Data & Hadoop Training in Chennai

Reply**Dinju Thomas** November 5, 2015 at 5:23 PM



Oracle Training in chennai | Oracle D2K Training In chennai

This information is impressive; I am inspired with your post writing style & how continuously you describe this topic. After reading your post, thanks for taking the time to discuss this, I feel happy about it and I love learning more about this topic..

[Reply](#)



Naviya Nair November 27, 2015 at 12:49 PM

Great Explanation...

[ASP.Net MVC Training](#)

[Online MVC Training](#)

[Online MVC Training from India](#)

[MVC Training in Chennai](#)

[Reply](#)



Nouman ullah baig January 20, 2016 at 12:09 PM

Simply want to say your article is as surprising. The clarity on your put up is simply cool and i could suppose you're an expert in this subject.

[Download CShare App Apk for Android](#)

[Reply](#)



Reshad Ahmadi March 26, 2016 at 5:44 PM

nice content [Techpave](#)

[Reply](#)



WT Labs April 5, 2016 at 12:08 PM

Wow, Excellent post. This article is really very interesting and effective. I think its must be helpful for us. thanks for sharing your informative. [asp.net training in jalandhar](#)

[Reply](#)



Usama Fassy April 5, 2016 at 11:29 PM

Awsome stuff..

[Reply](#)



subbu raj May 17, 2016 at 3:20 PM

Good detail about view model in asp.net mvc.

[ASP.Net Migration](#)

[Visual FoxPro to .Net](#)

[Reply](#)



Rohith June 25, 2016 at 12:48 PM

Yes, you are absolutely correct...And it is very informative and very clear and easy to understand.. [seo training in chennai](#)

[Reply](#)



Manoj Kumar July 5, 2016 at 12:27 PM

nic post..

<http://mkniit.blogspot.in>

[Reply](#)



Nikita Mittal July 12, 2016 at 3:08 PM

Wow what a cool blog you have here! I am impressed. You really put a lot of time and effort into this. I wish I had your creative writing skills, progressive talent and self discipline to produce a blog like you did.

Great information provided about [MVC Training](#) . I appreciate your work.

[Reply](#)



Iqbal Dik September 24, 2016 at 6:40 PM

Thanks for sharing this valuable information to our vision. You have posted a trust worthy blog keep sharing

[dot net training in chennai](#)

[php training in chennai](#)

[java training in chennai](#)

[Reply](#)



amandaayers September 29, 2016 at 5:52 PM

Nice post! Thank you for the info! You can also check out ShieldUI MVC, it has really versatile capabilities <https://demos.shieldui.com/mvc>

[Reply](#)



Unknown January 30, 2017 at 5:35 PM

Saved my day ! thanks ! :D

[Reply](#)



Pavithra M March 16, 2017 at 12:20 PM

It is really a great work and the way in which u r sharing the knowledge is excellent.Thanks for helping me to understand basic concepts. As a beginner in Dot Net programming your post help me a lot.Thanks for your informative article. [dot net training institute in velachery](#) | [top10 dot net training institutes in chennai](#)

[Reply](#)



Bayo Adegbola March 23, 2017 at 9:31 PM

Simple and unique work

[Reply](#)

Enter your comment...



Comment as:

[ahm7dkhalifa@](#) ▼

[Sign out](#)

[Publish](#)

[Preview](#)

☐ [Notify me](#)

Thanks for your Feedback.

[Newer Post](#)

[Home](#)

[Older Post](#)

Subscribe to: [Post Comments \(Atom\)](#)

Simple theme. Powered by [Blogger](#).