

Beyond Syllabus

Introduction of MVC framework.

Introduction To ASP.NET MVC

In this article you will learn about Introduction to ASP.NET MVC. Model-View-Controller (MVC) is a framework for developing applications which has become so popular these days. Gone are the days when developers would go for ASP.NET to create applications from scratch. MVC is a Model Controller View Pattern and using it in ASP.NET is why we term it as ASP.NET MVC.

In this article, I will give a brief introduction on ASP.NET MVC. Model-View-Controller (MVC) is a framework for developing applications which have become so popular these days. Gone are the days when developers would go for ASP.NET to create applications from scratch. MVC is a Model Controller View Pattern and using it in ASP.NET is why we term it as ASP.NET MVC.

The basic need of introducing MVC was to make complex application development easy. MVC is a lightweight highly testable framework as compared to traditional ASP.NET Web Forms. MVC focusses on Separation of Concerns. The purpose of MVC is to separate the content from the presentation and data processing from content. One thing that MVC has done is to separate the view from the code i.e., unlike Web Forms where “*.aspx” is attached to “*.aspx.cs” here in MVC View is a separate entity entirely.

ASP.NET MVC is stateless. In this we do not pass requests to the page like Web Forms. But we talk to the controller. And controller handles the request and fetches the desired data from the model and transfers the data to the view. And this View is used to display the data to the end user.

There are three basic components in ASP.NET MVC:

- Model
- View
- Controller

Model

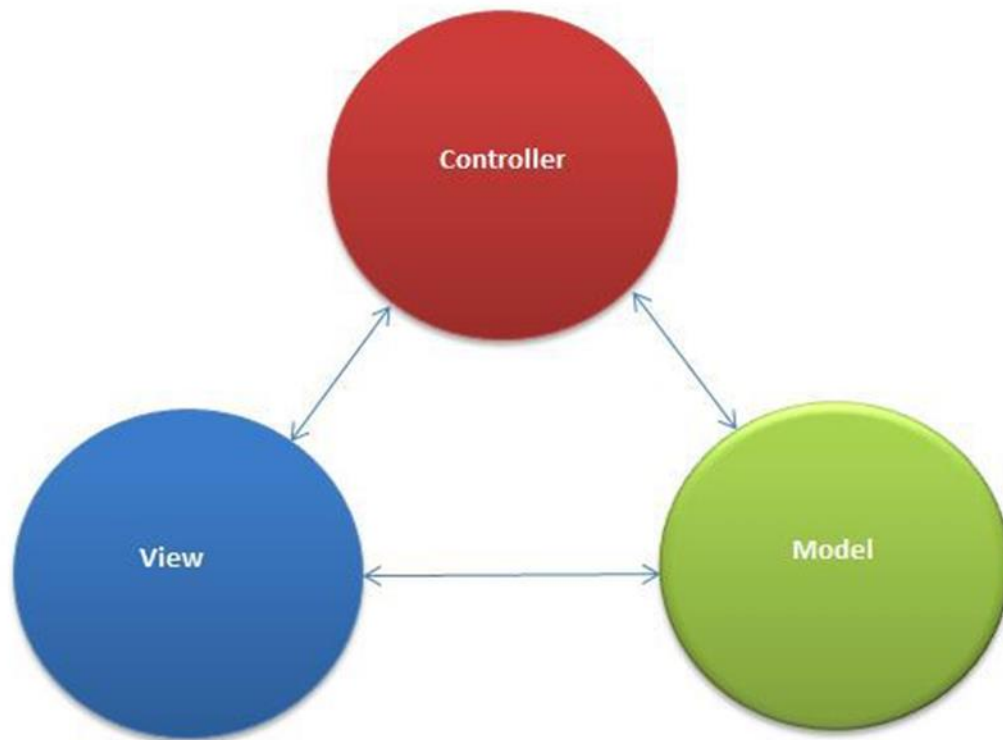
This is the business layer. It helps retrieve data from the database. These are simple class files that contain the properties.

View

This component is responsible for displaying data on the screen. In MVC we use Razor Syntax. The extension of the view has “*.cshtml” instead of “*.aspx” which we have used in ASP.NET in the past.

Controller

It handles input control. All the user interaction is done through Controller. A Controller is nothing but a class file that contains the methods.



Introduction to ASP.NET MVC

ASP.NET MVC URL Structure

MVC helps to maintain a pretty good URL that is readable to the user. In MVC the URL has specific meaning which we will understand now. The following is the structure that explains how `http://www.sitename.com/ControllerName/ActionName/Id` URLs are in MVC. Controller Name is passed after the site name followed by the Action Name present in the controller. And if there is an Id we have to specify the Id as well, though it is optional.

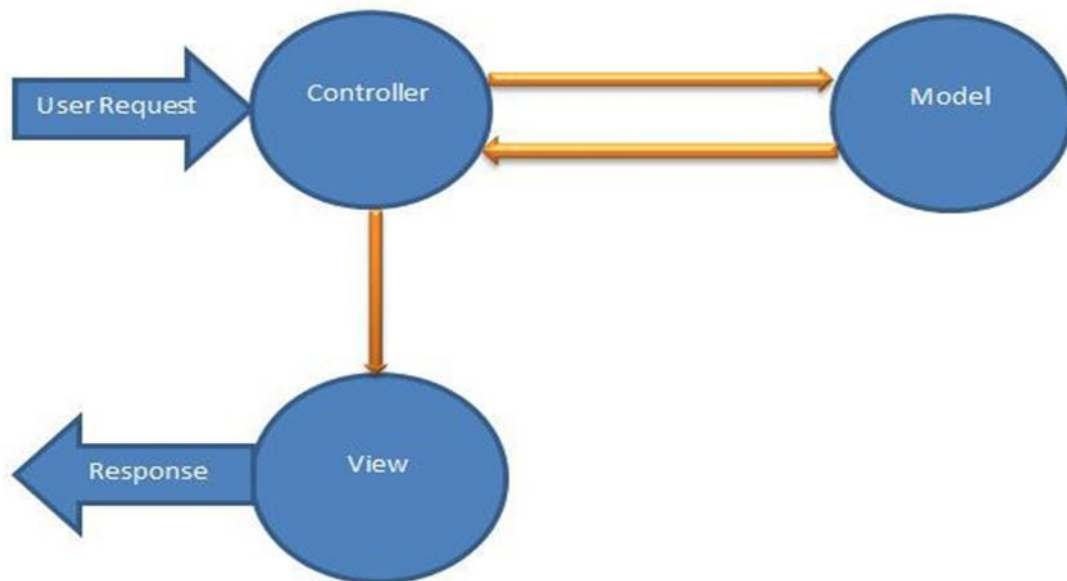
So, what this `http://www.dummywebsite.com/Products/View/1` URL would be like in some application let us check now.

In the preceding URL “Products” is the controller and View is the name of the ActionResult method that accepts an ID that has a value “1” passed in the URL. We will understand URL routing in detail in the later posts. Here I just want to brief about the URL convention that ASP.NET MVC follows.

Comparing the `http://www.dummysite.com/Products.aspx?id=1` URL to the traditional Web Forms had it been made in the older version would be something like it.

Clearly we can see ASP.NET MVC URLs are much more user friendly. Microsoft has released many versions of ASP.NET MVC of which MVC 6 is the latest one. Some or the other features have been introduced in the subsequent versions of this framework.

How MVC works



Introduction to ASP.NET MVC

The request goes to controller. Controller fetches the data from the model and passes this data to the View. The view displays the data to the end user.

This is the basic flow of ASP.NET MVC. In further tutorials we will study each component of MVC in depth and their functions.