

ASP.NET MVC Architecture

Here, you will learn an overview of MVC architecture.

The MVC architectural pattern has existed for a long time in software engineering. All most all the languages use MVC with slight variation, but conceptually it remains the same.

Let's understand the MVC architecture supported in ASP.NET.

MVC stands for Model, View, and Controller. MVC separates an application into three components - Model, View, and Controller.

Model: Model represents the shape of the data. A class in C# is used to describe a model. Model objects store data retrieved from the database.

Model represents the data.

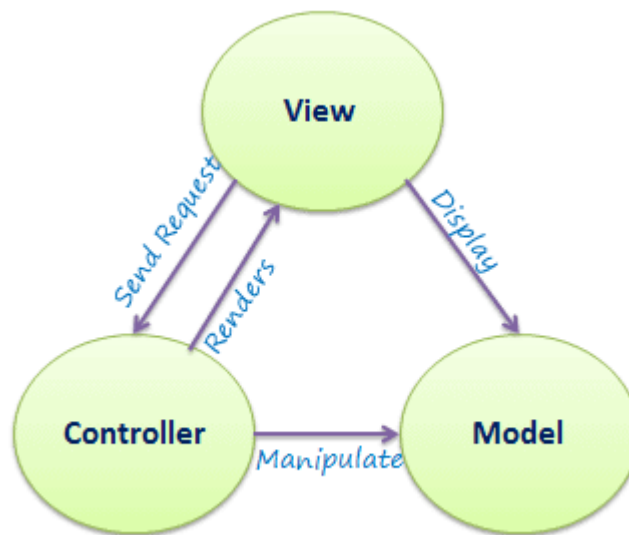
View: View in MVC is a user interface. View display model data to the user and also enables them to modify them. View in ASP.NET MVC is HTML, CSS, and some special syntax (Razor syntax) that makes it easy to communicate with the model and the controller.

View is the User Interface.

Controller: The controller handles the user request. Typically, the user uses the view and raises an HTTP request, which will be handled by the controller. The controller processes the request and returns the appropriate view as a response.

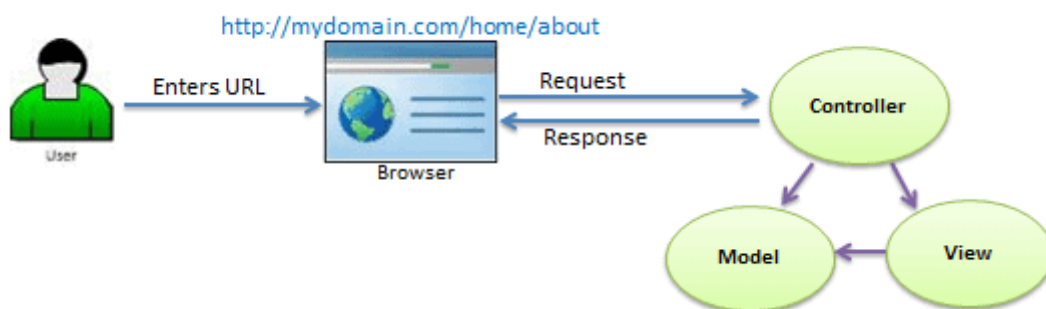
Controller is the request handler.

The following figure illustrates the interaction between Model, View, and Controller.



MVC Architecture

The following figure illustrates the flow of the user's request in ASP.NET MVC.



Request Flow in MVC Architecture

As per the above figure, when a user enters a URL in the browser, it goes to the webserver and routed to a controller. A controller executes related view and models for that request and create the response and sends it back to the browser.



Points to Remember

MVC stands for Model, View and Controller.

Model represents the data

View is the User Interface.

Controller is the request handler.