International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 01 | Jan 2020 www.irjet.net p-ISSN: 2395-0072

MVC Framework: A Modern Web Application Development Approach and Working

Mr. Siddharth Singh

Student, Semester-I, MSC(I.T.)
Keraleeya Samajam's Model College, Thakurli East, Thane, Maharashtra, India

---

Abstract - *Model, view and controller (MVC) is a well-known three layer development architecture used for web-based applications developments. This paper gives the some details related to the MVC layers, its uses, advantages, disadvantages. We have stated the three layers of MVC in detail and their functionalities. The main objective of the study is to give overview on all the layer of the MVC and main functionalities.*

Key Words: MVC Architecture; Advantages and disadvantages; features; IDE and Languages; Example based on ASP.NET;

1. INTRODUCTION

Model view controller (MVC) is an architectural pattern usually used in web-based applications. It provides 3 main layers; model, view, and controller. Most of the developers use MVC as a standard web-design pattern. It is a complete framework. Most of the languages like Java, PHP, Python, C#, etc. use this pattern to develop the applications. In Java, it is known as Spring-MVC framework, in PHP it is known as cake PHP, Microsoft introduced a framework called ASP.Net MVC and so on. MVC provide three types of classes:

1.1 Model- Model is an important layer of MVC application. It manages the information in the form of data which is used to represent the output with the help of views. It depicts the database records. It manages the application data. It basically contains the application data, logic definition, function specification, business rule involvement. A model possibly is a single object or it is some composition of objects. This layer can manage the data and also allow the database communication i.e. insert, retrieve and update data in the database.

The domain model in application should follow certain rules:

- Contain the domain data.
- Contain the logic for creating, modifying the domain data.
- Provides a proper API that exposes the model data and operations on it.

Following rules should not be followed by domain data:

- Expose details of how model data is obtained and managed.
- Contain logic that controls the request to call a model based on user interaction.
- Contain logic for displaying data to the user.

1.2 View- Views are used to create the user-interface of our application. While using the user interface, users interact with our web pages. View shows the results of the data that is contained in model. A view has the responsibility to display all data of the model. It only shows the required attributes and hides the unnecessary attributes. It thus provides us the advantage of presentation encapsulation. It is script-based tags like JSP, ASP and PHP and very easily integrated with AJAX technology.

The following rules should be followed by a view:

- Contain logic and markup required to present data to the user.

The following rules should not be used by a view:

- Contain complex logic.
- Contain logic that creates, stores, or manipulates the model.

1.3 Controller- Controller layer are used to respond to the user's requests. Controller perform the users requested actions. A controller is the connection between the user and the system. Controller handles both Model and View. It controls how the data flow in model and updates the view as soon as the data get altered. It is the separation between the Model and the View. The controller receives the data; it works on the data and then carries out the execution which alters the state of the data model. These component works with model layer and select the proper view that is to be displayed to the user according to user requests.

Rules followed by controller are:

- Contain the logic required to initialize model data.

- Contain the logic required by the view to present data from the model.
- Contain the logic required to update the model based on user interactions.

Following rules should not be used controller:

- Manipulate data outside the scope.
- Contain logic that manages persistence of data.
- Contain logic that displays data to the user.

MVC pattern architecture is basically a three-layered architecture. It separates the characteristics of application. Its 1st layer is related to the user input logic, 2nd layer is related to the business logic and 3rd layer is used to implement user interface logic. MVC offers extremely loose coupling among these three layers. MVC pattern are used to define the location of all logic in application. MVC patterns provide the facility of parallel development. It means that every layer of the application are not dependent on each other i.e. three developer will work on the one layer of application. One developer will be working on user input logic (controller logic), other developer will be working on the user interface logic (view) and third developer will be working on the business logic (model) at the same time.

2. TIME TO USE MVC PATTERN

MVC pattern architecture help us to follow the concept of separation of concern, it helps to implement the codes individually in the model, views and controller layer in the applications.

MVC makes it simple to test our application as relation among different parts of application is clearer and coherent.

MVC benefits us to implement a test-driven development approach, in which we implement automated test cases before we write the code. These unit test cases help us predefine and verify requirements of new code before writing it.

If we are developing an web project with enough serious stimulating on the user side to refuse to go along with JavaScript alone. The some of the characteristics that will guide us whether to use MVC architecture in our application or not:

- An asynchronous communication is needed.
- The functionality of our web-application is not to reload a full page just like AJAX for example viewing products in an online ecommerce platform.
- Manipulation of data is mostly on the client side (browser) rather than server side.
- Same type of data is being delivered in different ways on a single page (navigation).
- When our application has many insignificant connections that are used to modify data(button, switches).

3. ADVANTAGES AND DISADVANTAGES OF MVC ARCHITECTURE

3.1 Advantages:

- MVC architecture helps us to regulate the complexness of application by dividing it into 3 parts i.e. model, view and controller.
- MVC does not use server-based forms, that's why it is ideal for those developers who want full control over their application behavior.
- MVC use front controller pattern. Front controller pattern handles the multiple incoming requests using controller. Front controller provides centralized control. We need to configure only one controller in web server instead of many.
- Front controller provides support in routing communications to create our web-based application.

3.2 Disadvantages:

- Understanding flow of application is very hard one.
- It is little bit complicated to implement and unsuitable for small level applications.
- It is hard to understand the MVC architecture.
- It's deployment is little bit hard one.
- Must have strict rules on methods.
- There is so much use of jQuery, AJAX, javascript which makes the program complicated.
- MVC is difficult to learn, as programmer should have a deep knowledge of pattern.
- There is no built-in user control for web pages in MVC.

4. MVC FEATURES

As we divide the logic of our application into three tasks (input logic, business logic, interface logic), testing of these components would become very easy. Testability is very fast and flexible, as we can use any unit testing framework

compatible with MVC framework. It is an extensible and pluggable framework. We can design the components of our application in such a way that these are easily replaceable or can be modified easily. We can plug our own view engine, URL routing strategy, action method constraint serialization. Instead of depending on class to create objects we use a technique dependency injection (DI) which enable us to inject object into classes. MVC provide URL mapping component that helps us to build using understandable and searchable URLs. Instead of using file name extensions MVC support URL naming patterns that are very useful for search engine optimization (SEO) and representational state transfer (REST) addressing. Some frameworks of MVC such as ASP.NET MVC framework provide us some built in features such as form authentication, session management, transactional business logic, web application security, object relational mapping, etc.

5. IDE'S AND PROGRAMMIMG LANGUAGE USED WITH MVC

There are various IDE's and programming languages which can be used to developed web-based application with the help of MVC architecture. Depending upon the interest of developers, they can use any of the IDE's and programming languages to develop web-based application. Below is the list of IDE's and programming languages which can be used to develop web-based application using MVC architecture:

5.1 Integrated Development Environment (IDE)

- Visual Studio: Visual studio is a complete development environment which provide us facility to create different kinds of application. When we want to develop application using ASP.NET MVC framework then visual studio is very helpful for us.
- MYSQL Server- relational database management server to maintain the database.
- SQL Server- a database engine to create and maintain database.
- MYSQL Workbench- a database design tool.
- Net Beans- IDE provide complete environment to develop java applications or different applications.
- Glassfish Server: Java EE application server.

5.2 Programming or Web Languages

- HTML, CSS, JQUERY, AJAX for designing.
- Servlet and Java server pages (JSP) used with Net beans.
- EJB (Enterprise Java beans) technologies.
- JSTL (Java server pages standard tag libraries).
- JPA (Java persistence API).
- JDBC (Java database connectivity).
- ASP.NET MVC used with Visual studio.

There are also some other IDE's and languages which can be used with MVC architecture.

6. TRADITIONAL WEB APPROACH

In traditional web page approaches, all the codes are written in single file i.e, either in JSP and Servlet, ASP, PHP. Most of this languages have their inbuilt tags which are used to process the request and fetch data according to request. Since this inbuilt tags are heavyweight, the processing time taken by this approach is some how longer compared to MVC approach for heavy traffic webpages. For example ASP.NET web form works on event driven approach and also has the concept of viewstate and postback. As both viewstate and postback has certain limitations which ampacts the web application performance (In ASP.NET web form, viewstate stores 100KB of data which impacts web-page performance).

7. DESCRIBING THE ARCHITECTURE

We should verify the functional elements of our application and determine the manner through which they they're going to communicate with one other. There are many design patterns are available which can be used to develop an application. Here we aim to use MVC design pattern.

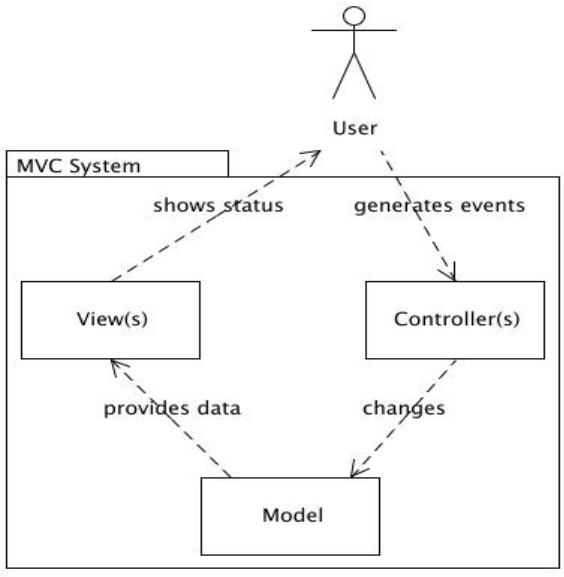

Figure 1: Functionality of each layer in MVC architecture.

8. DISCUSSIONS

MVC design pattern has massive influence in the world of web-based application. It is very helpful for the developers. Developers can easily make their web-based applications using MVC architecture. It reduces the complexity of the application and divides the application into three major components i.e. model, view and controller. It provides full control over the web application to the developers. Using MVC design pattern testing of various elements of application becomes very much simple. MVC architecture provides higher level of abstraction to their web applications.

9. STEPS TO CREATE A BASIC WEB APPLICATION IN ASP.NET MVC

9.1 Creating a model class name Student

```
namespace MVC_BasicTutorials.Models
{
 public class Student
 {
      public int StudentID{ get; set; }
      public string StudentName{ get; set; }
      public int Age{ get; set; }
 }
}
```

9.2 Creating a controller name StudentController using System;

```
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.MVC;
using MVC_BasicTutorials.Models;
namespace MVC_BasicTutorials.Controllers
{
 Public class StudentController : Controller
 {
      Public ActionResult Index()
      {
          Return View();
      }
 }
}
```

9.3 Creating a view name Index.cshtml

```
@model
IEnumerable<MVC_BasicTutorials.Models.Student>
@{
  ViewBag.Title = "Index";
  Layout = "~/Views/Shared/_Layout.cshtml";
}
<h2>Index</h2>
<p>
  @Html.ActionLink("Create New", "Create")</p><table class="table">
  <tr>
    <th>
      @Html.DisplayNameFor(model => model.StudentName)
    </th>
    <th>
      @Html.DisplayNameFor(model => model.Age)
    </th>
    <th></th>
  </tr>
@foreach (var item in Model) {
  <tr>
    <td>
      @Html.DisplayFor(modelItem => item.StudentName)
    </td>
    <td>
      @Html.DisplayFor(modelItem => item.Age)
    </td>
```

```
    <td>

        @Html.ActionLink("Edit", "Edit", new
{ id=item.StudentId }) |

        @Html.ActionLink("Details", "Details", new
{ id=item.StudentId  }) |

        @Html.ActionLink("Delete", "Delete", new { id =
item.StudentId })

    </td>

  </tr>

}

</table>
```

10. LIMITATION OF OUR STUDY

In our research paper we have discussed about MVC components and talked about its three major elements model, view and controller and explained about communication of these components with each other. We don't give enough detail about these elements individually. There are several other frameworks of MVC which can be used for the development of web-based application.We have not provided a proper codes which are used in ASP.NET MVC and not provided a detailed and complete steps to create web-based application in ASP.NET MVC.

11. CONCLUSION

The scalable, transparent and portable web-based application can be created with the help of MVC architecture. MVC architecture help us to know the concept of parallel development as it divides the logic of application into three layers, so each different developer can work simultaneously on these three layers of the same web-based application. The cost of setting up the MVC architecture are usually high. The processing time for small MVC project vs non-MVC projects are normally same but, large project build in MVC architecture will be processed faster than non-MVC project.

12. REFERENCES

[1] Gupta P, Govil MC (2010) MVC Design pattern for the multi framework distributed applications using XML, spring and struts framework.

[2] Pro ASP.Net MVC 5 book by adam freeman.

[3] www.interserver.net.

[4] www.tutorialspoint.com.

[5] www.tutorialteacher.com.