PART – 1
Model-View-Controller (MVC)

Contents

- What is MVC Framework?
- Features of MVC
- MVC Architecture
- Example

What is MVC Framework?

- The **Model-View-Controller (MVC)** framework is an architectural pattern that separates an application into three main logical components Model, View, and Controller. Hence the abbreviation MVC
- Each architecture component is built to handle specific development aspect of an application. MVC separates the business logic and presentation layer from each other

What is MVC Framework?

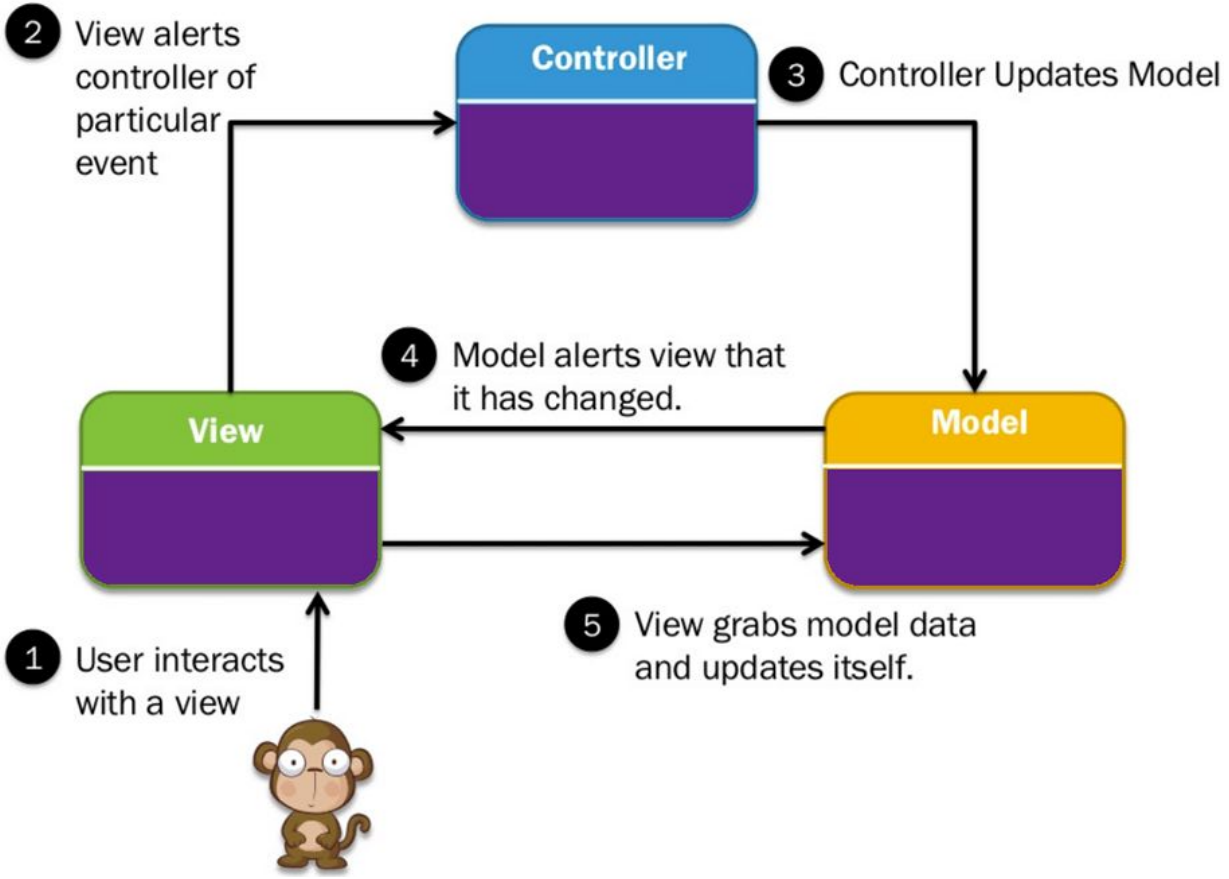
- It was traditionally used for desktop graphical user interfaces (GUIs)

- Nowadays, MVC architecture has become popular for designing web applications as well as mobile apps

Features of MVC

- Easy and frictionless testability. Highly testable, extensible and pluggable framework
- Offers full control over your HTML as well as your URLs
- Leverage existing features provided by ASP.NET, JSP, Django, etc.
- Clear separation of logic: Model, View, Controller. Separation of application tasks viz. business logic, Ul logic, and input logic
- URL Routing for Search Engine Optimization (SEO) Friendly URLs. Powerful URL- mapping for comprehensible and searchable URLs
- Supports for Test Driven Development (TDD)

MVC Architecture



MVC Architecture

Three important MVC the components are:

- Model: It includes all the data and its related logic
- View: Present data to the user or handles user interaction
- Controller: An interface between Model and View components

MVC Architecture

Model

- The model component stores data and its related logic. It represents data that is being transferred between controller components or any other related business logic. For example, a Controller object will retrieve the customer info from the database. It manipulates data and send back to the database or use it to render the same data

- It responds to the request from the views and also responds to instructions from the controller to update itself. It is also the lowest level of the pattern which is responsible for maintaining data

MVC Architecture

View

- A View is that part of the application that represents the presentation of data
- Views are created by the data collected from the model data. A view requests the model to give information so that it resents the output presentation to the user
- The view also represents the data from chats, diagrams, and table. For example, any customer view will include all the UI components like text boxes, drop downs, etc

MVC Architecture

Controller

- The Controller is that part of the application that handles the user interaction. The controller interprets the mouse and keyboard inputs from the user, informing model and the view to change as appropriate
- A Controller send's commands to the model to update its state(E.g., Saving a specific document). The controller also sends commands to its associated view to change the view's presentation (For example scrolling a particular document)

Example

- Let's assume you go to a restaurant. You will not go to the kitchen and prepare food which you can surely do at your home. Instead, you just go there and wait for the waiter to come on
- Now the waiter comes to you, and you just order the food. The waiter doesn't know who you are and what you want he just written down the detail of your food order

Example

- Then, the waiter moves to the kitchen. In the kitchen waiter not prepare your food
- The cook prepares your food. The waiter is given your order to him along with your table number
- Cook then prepared food for you. He uses ingredients to cooks the food. Let's assume that your order a vegetable sandwich. Then he needs bread, tomato, potato, capsicum, onion, bit, cheese, etc. which he sources from the refrigerator

Example

- Cook final hands over the food to the waiter. Now it is the job of the waiter to moves this food outside the kitchen
- Now waiter knows which food you have ordered and how they are served

View= You (customer)
Waiter= Controller
Cook= Model
Refrigerator= Data

Advantages of MVC

- Easy code maintenance
- MVC Model component can be tested separately from the user
- Easier support for new type of clients
- Development of the various components can be performed parallelly

Disadvantages of MVC

- Difficult to reuse this model
- The framework navigation can some time complex as it introduces new layers of abstraction which requires users to adapt to the decomposition criteria of MVC
- There is a need for multiple programmers to conduct parallel programming

PART – 2
Command Pattern

Contents

- Introduction
- Implementation
- Advantages & Disadvantages

Introduction

- Command pattern is a data driven design pattern and falls under behavioral pattern category
- A request is wrapped under an object as command and passed to invoker object
- Invoker object looks for the appropriate object which can handle this command and passes the command to the corresponding object which executes the command

Implementation

- Four terms always associated with the command pattern are *command*, *receiver*, *invoker* and *client*
- A *command* object knows about *receiver* and *invokes* a method of the receiver
- Values for parameters of the receiver method are stored in the command

Implementation

- The receiver object to execute these methods is also stored in the command object
- The receiver then does the work when the *execute()* method in *command* is called
- An *invoker* object knows how to execute a command

Advantages & Disadvantages

Advantages

- Makes our code extensible as we can add new commands without changing existing code
- Reduces coupling the invoker and receiver of a command

Disadvantages

- Increase in the number of classes