TuxRide 2019

Components of website : MVC Architecture and Intro to node js

Organized by- Development sig,Web club Nitk

Model View Controller

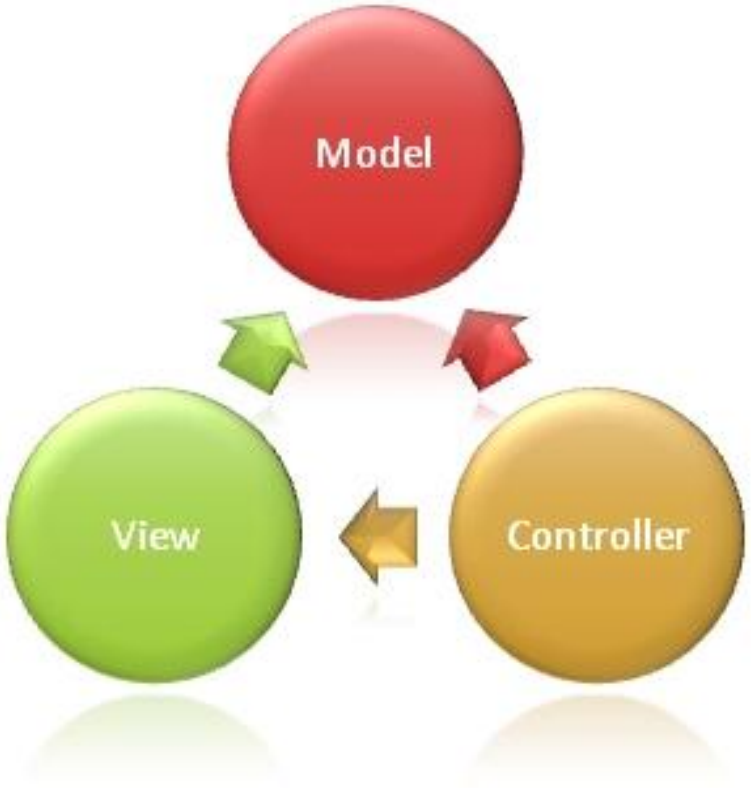
Introduction

- **M**odel **V**iew **C**ontroller or MVC as it is popularly called, is a software design pattern for developing web applications.

- **Model–view–controller** (**MVC**) is a software architecture pattern which separates the representation of information from the user's interaction with it .

Parts of MVC

- A Model View Controller pattern is made up of the following three parts:
- **Model**
- **View**
- **Controller**

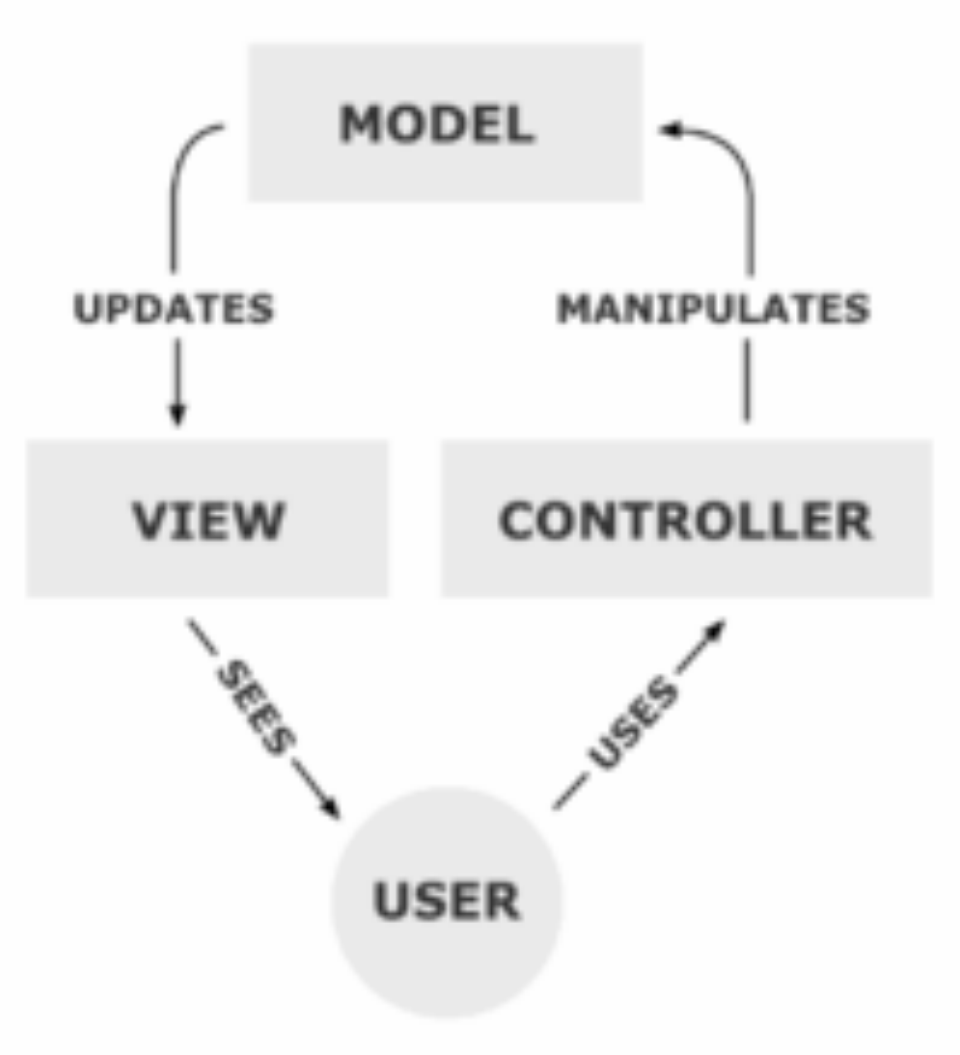


The MVC model defines web applications with 3 logic layers:

The business layer (Model logic)

The display layer (View logic)

The input control (Controller logic)


MODEL
UPDATES
MANIPULATES
VIEW
CONTROLLER
SEES
USES
USER

Model

- The model is responsible for managing the data of the application.
- It responds to the request from the view and it also responds to instructions from the controller to update itself
- It is the lowest level of the pattern which is responsible for maintaining data.
- The Model represents the application core (for instance a list of database records).
- It is also called the domain layer

View

- The View displays the data (the database records).
- A **view** requests information from the model, that it needs to generate an output representation.
- MVC is often seen in web applications, where the view is the HTML page.

Controller

- **The Controller** is the part of the application that handles user interaction.
- Typically controllers read data from a view, control user input, and send input data to the model.
- It handles the input, typically user actions and may invoke changes on the model and view.

Workflow in MVC - Example

Though MVC comes in different flavours, the control flow generally works as follows:

1. The user interacts with the user interface in some way (e.g., user presses a button)
2. A controller handles the input event from the user interface, often via a registered handler or callback.
3. The controller accesses the model, possibly updating it in a way appropriate to the user's action (e.g., controller updates user's shopping cart).

4. A view uses the model to generate an appropriate user interface (e.g., view produces a screen listing the shopping cart contents).

   The view gets its own data from the model. The model has no direct knowledge of the view.

Dependence hierarchy

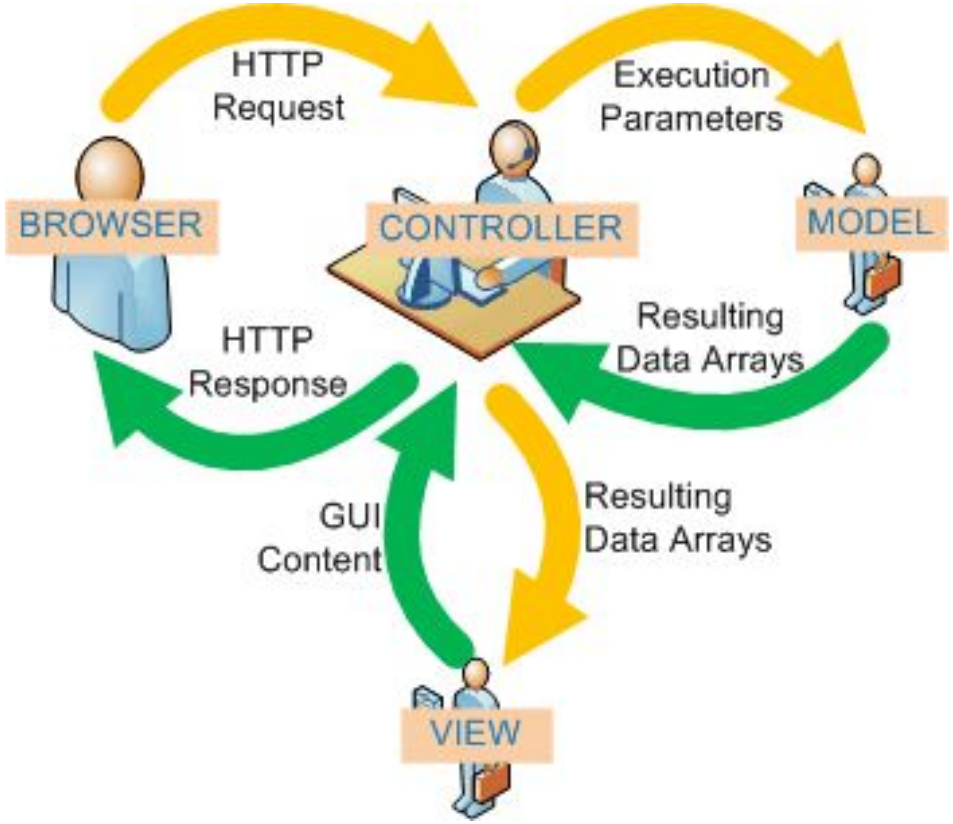
- There is usually a kind of hierarchy in the MVC pattern.
- The Model knows only about itself.
- That is, the source code of the Model has no references to either the View or Controller.

- The View however, knows about the Model. It will poll the Model about the state, to know what to display.
- That way, the View can display something that is based on what the Model has done.
- But the View knows nothing about the Controller.
- The Controller knows about both the Model and the View.

Why dependence hierarchy is used?

- The reason to keep it this way is to minimize dependencies.
- No matter how the View class is modified, the Model will still work.
- Even if the system is moved from a desktop operating system to a smart phone, the Model can be moved with no changes.
- But the View probably needs to be updated, as will the Controller.

Working of MVC in web application



Normal Web Page vs. MVC

- The MVC programming model is a lighter alternative to traditional Web Page/Forms.
- It is a lightweight, highly testable framework, integrated with all existing features, such as Security, and Authentication.

Advantages

- Clear separation between presentation logic and business logic.
- Each object in mvc have distinct responsibilities.
- parallel development
- easy to maintain and future enhancements
- All objects and classes are independent of each other.

NODE JS

Introduction Node.js

Node.js was created by **Ryan Dahl** starting in 2009, and its growth is sponsored by Joyent, his employer.

What is node.js ?

- Evented I/O for JavaScript
- Server Side JavaScript
- Runs on Google's V8 JavaScript Engine

Why Use Node.js ?

- Node's goal is to provide an easy way to build scalable network programs.

What is unique about Node.js?

1. JavaScript used in client-side but node.js puts the JavaScript on server-side thus making communication between client and server will happen in same language

2. Servers are normally thread based but Node.JS is “Event” based. Node.JS serves each request in a Evented loop that can able to handle simultaneous requests.

What is Unique about Node.Js (Con't)?

3. Node.JS programs are executed by V8 Javascript engine the same used by Google chrome browser.

What can you do with Node ?

- It is a command line tool.
- It lets you Layered on top of the TCP library is a HTTP and HTTPS client/server.

- The JS executed by the V8 javascript engine (the ting that makes Google Chrome so fast)

- Node provides a JavaScript API to access the network and file system.

What can’t do with Node?

- Node is a platform for writing JavaScript applications outside web browsers. This is not the JavaScript we are familiar with in web browsers. There is no DOM built into Node, nor any other browser capability.

- Node can’t run on GUI, but run on terminal

Threads VS Event-driven

Threads	Asynchronous Event-driven
Lock application / request with listener-workers threads	only one thread, which repeatedly fetches an event
Using incoming-request model	Using queue and then processes it
multithreaded server might block the request which might involve multiple events	manually saves state and then goes on to process the next event
Using context switching	no contention and no context switches
Using multithreading environments where listener and workers threads are used frequently to take an incoming-request lock	Using asynchronous I/O facilities (callbacks, not poll/select or O_NONBLOCK) environments

Why node.js use event-based?

In a normal process cycle the webserver while processing the request will have to wait for the IO operations and thus blocking the next request to be processed.

Node.JS process each request as events, The server doesn’t wait for the IO operation to complete while it can handle other request at the same time.

When the IO operation of first request is completed it will call-back the server to complete the request.

npm

- **npm** (**node.js package manager**) is a package manager for the JavaScript programming language. It is the default package manager for the JavaScript runtime environment Node.js. It consists of a command line client, also called npm, and an online database of public and paid-for private packages, called the npm registry.

HTTP Methods

- GET Method
- A GET request retrieves data from a web server by specifying parameters in the URL portion of the request.

- POST Method
- The POST method is used when you want to send some data to the server, for example, file update, form data, etc.
- DELETE Method
- The DELETE method is used to request the server to delete a file at a location specified by the given URL.
- PUT Method
- The PUT method is used to request the server to store the included entity-body at a location specified by the given URL.

Making a todo app in node js

- Clone this repo-
- **https://github.com/mananpoddar/TuxrideDev**