MVC

Model-Views-Controller

MVC is an acronym for Model-View-Controller. It is a design pattern for software projects. It is used majorly by Node developers and by C#, Ruby, PHP framework users too. In MVC pattern, application and its development are divided into three interconnected parts. The advantage of this is it helps in focusing on a specific part of the application name, the ways information is presented to and accepted from, the user. It helps in allowing for efficient code reuse and the parallel development of the application. Even if the project structure might look a little different than an ideal MVC structure, the basic program flow in and out the application remains the same.

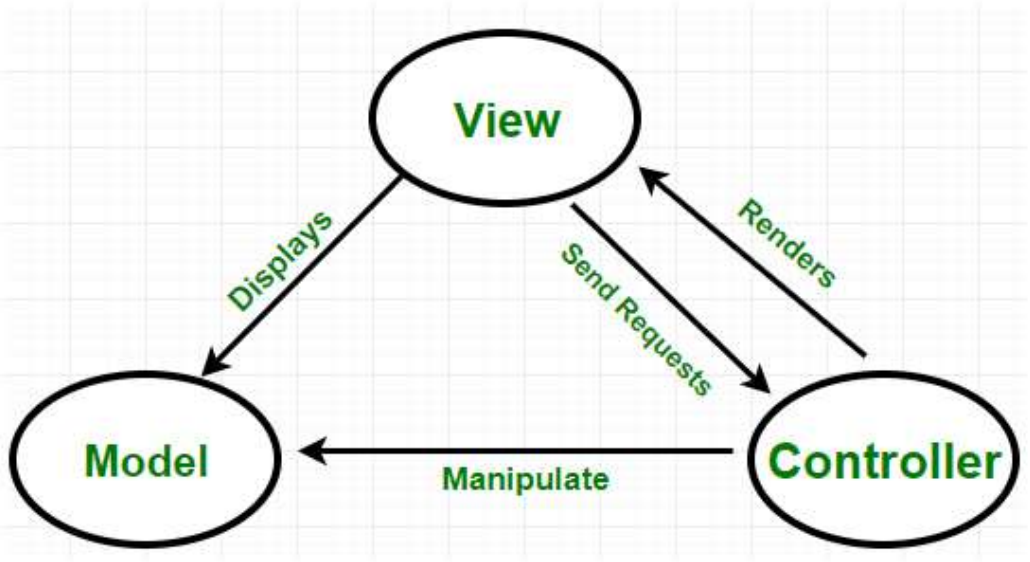
The application and its development are divided into three interconnected parts.

- Model: Model represents the structure of data, the format and the constraints with which it is stored. It maintains the data of the application. Essentially, it is the database part of the application.

- View: View is what is presented to the user. Views utilize the Model and present data in a form in which the user wants. A user can also be allowed to make changes to the data presented to the user. They consist of static and dynamic pages which are rendered or sent to the user when the user requests them.

- Controller: Controller controls the requests of the user and then generates appropriate response which is fed to the viewer. Typically, the user interacts with the View, which in turn generates the appropriate request, this request will be handled by a controller. The controller renders the appropriate view with the model data as a response.

MVC Architecture



MVC architecture

Advantages

- Faster Development Process
- Ability To Provide Multiple Views
- Support For Asynchronous Technique
- Modification Does Not Affect The Entire Model
- MVC Model Returns The Data Without Formatting
- SEO Friendly Development Platform

Debugging in NodeJS

Node.js Debugging: Debugging is a concept to identify and remove errors from software applications. In this article, we will learn about the technique to debug a Node.js application.

Debugging in JS Debugging is not easy. But fortunately, all modern browsers have a built-in JavaScript debugger. Built-in debuggers can be turned on and off, forcing errors to be reported to the user. There are mainly 3 methods in debugging JS:

1. The console.log() method
2. Setting breakpoints
3. The debugger keyword

1.Console.log() method:

The console.log() is a function in JavaScript which is used to print any kind of variables defined before in it or to just print any message that needs to be displayed to the user.

Syntax:

console.log(A);

2.Setting Breakpoints

In the debugger window, you can set breakpoints in the JavaScript code. At each breakpoint, JavaScript will stop executing, and let you examine JavaScript values. After examining values, you can resume the execution of code (typically with a play button).

3.The debugger Keyword

The debugger keyword stops the execution of JavaScript, and calls (if available) the debugging function. This has the same function as setting a breakpoint in the debugger. If no debugging is available, the debugger statement has no effect. With the debugger turned on, this code will stop executing before it executes the third line.

Syntax:

debugger;