

Overview

With anything you build, whether it's a house or a computer, there's an underlying structure that keeps the project organized. For a house, that structure may be a floor plan, designating the organization of a kitchen and a few bedrooms. Web applications are no different. One popular structure for web applications is called MVC, or model view controller. We use this model, or architectural pattern, to organize our code into parts that all have their own functions.

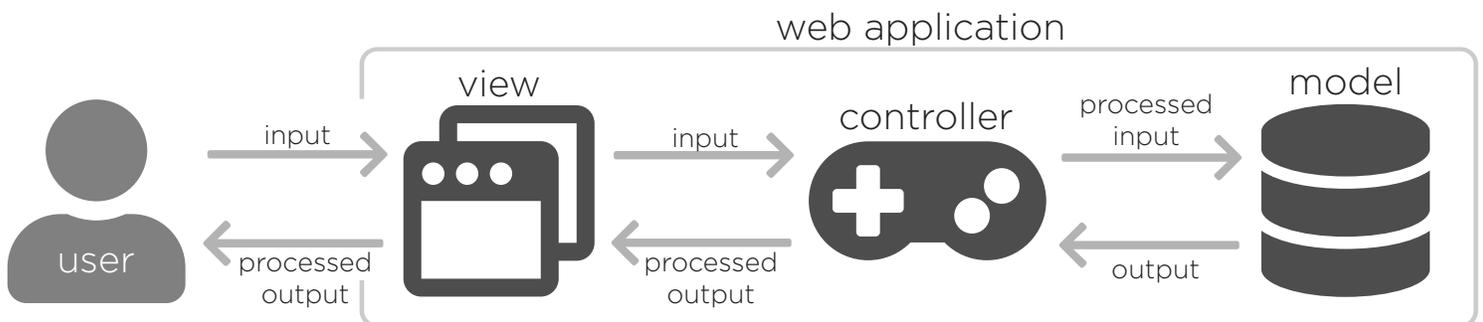
Key Terms

- MVC
- model
- view
- controller
- encapsulation

MVC

MVC is an architectural pattern that splits a web application into 3 logical components: model, view, and controller. The **model** stores and manages the application's data; this could consist of a database or some other file containing data. The **view** concerns the user-facing presentation of the application; it is application's output. The view might include files written in HTML, CSS, and Javascript. The **controller** is the logic that connects the model and the view. It is in charge of moving information between the user and the model as well as processing the given information. The controller might consist of programs written in Python and/or SQL.

For example, imagine an application that takes in a location and outputs nearby restaurants. The view would have an interface prompting the user to input a location. The location would then be passed to the controller and the controller would use the location to select the desired information from the model. In this case, the model could be a database containing a list of restaurants and their addresses. The controller would be in charge performing the necessary calculations to tell the model which restaurants it wants. Once the controller has retrieved a list of nearby restaurants from the model, the controller can send that information to the view to present to the user.



The view is responsible for enabling communication between the user and the application. May include HTML, CSS, and Javascript files.

The controller is the mastermind; it takes input from the view, processes it, uses it to send requests to the model, takes output from the model, processes it, and then passes it back to the view. May include programs written in Python and/or SQL.

The model stores data. It responds to requests from the controller to return particular data or update itself. May include a database.

Why MVC?

MVC is popularly used in industry because of its **encapsulation** of different parts of a web application. Although the model, view, and controller work together, their functions are independent of each other. Therefore, the model, view, and controller can be developed separately and simultaneously, making the MVC scalable and extensible. It also allows for the delegation of tasks among a large team of people, leading to a faster development process. Within these teams, the model, view, and controller can be independently tested. After a web application is built, MVC allows updates to be made without needing to update all parts of the application; any one part can be changed without changing the other, as long as the interactions between them remain the same. This compartmentalization also enables code reuse between different web applications.