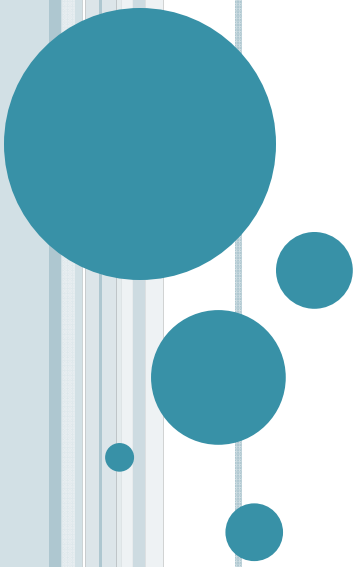
Understanding the Model-View-Controller



OBJECTIVES

- What is MVC?
- Model
- View
- Controller
- MVC with PHP
- Advantages of using MVC

Introduction

- Different layers can be implemented in different languages or distributed on different machines.
- AJAX applications can implement the View layer directly in Javascript in the browser, invoking JSON services.
- The controller can be partially implemented on client, partially on server.s

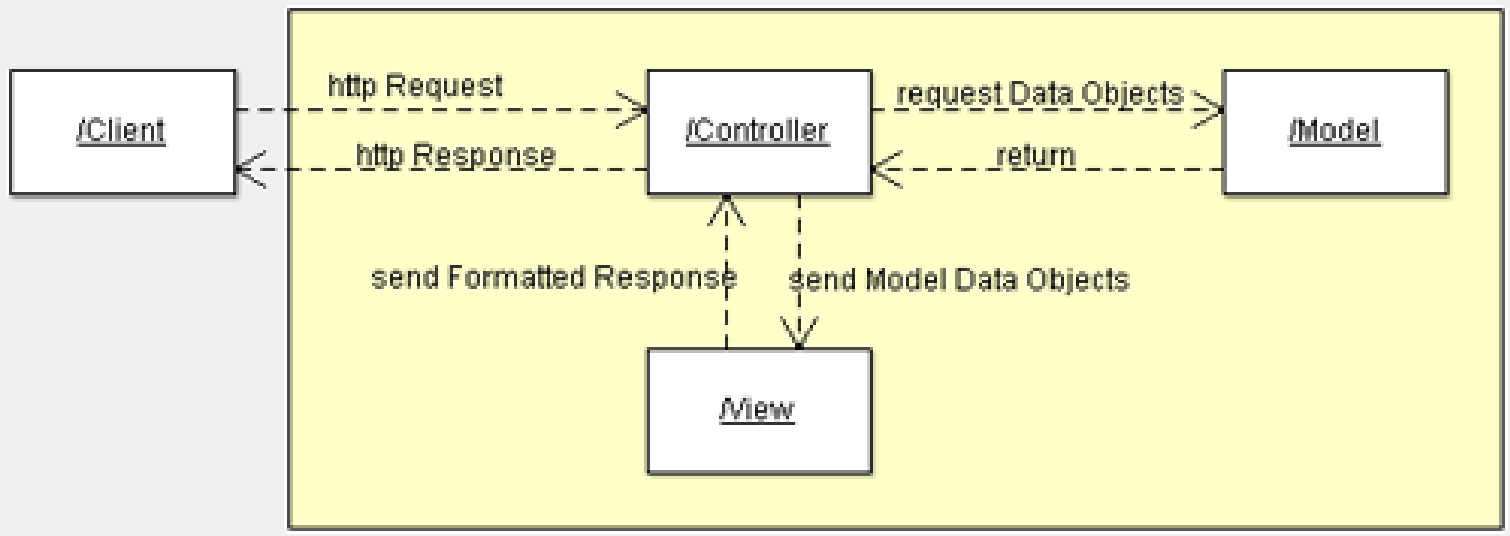
Introduction

- The model view controller pattern is the most used pattern for today's world web applications.
- It has been used for the first time in Smalltalk and then adopted and popularized by Java.
- At present there are more than a dozen PHP web frameworks based on MVC pattern.
- The MVC pattern separates an application in 3 modules: Model, View and Controller.

M-V-C

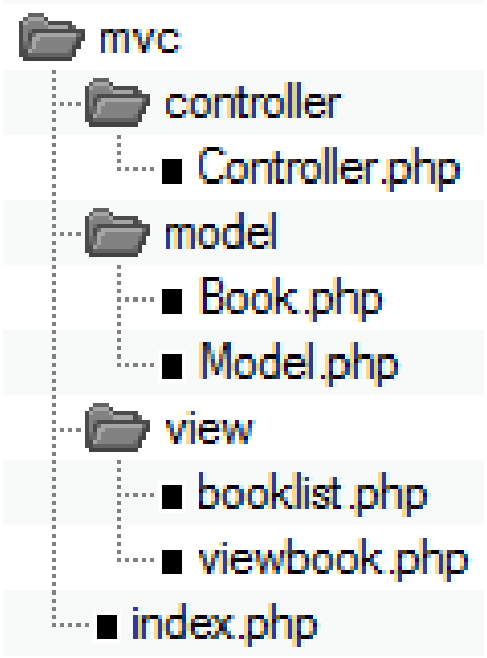
- The **model** is responsible to manage the data; it stores and retrieves entities used by an application, usually from a database, and contains the logic implemented by the application.

- The **view (presentation)** is responsible to display the data provided by the model in a specific format.
- It has a similar usage with the template modules present in some popular web applications, like wordpress, joomla, …



M-V-C

- The **controller** handles the model and view layers to work together.
- The controller receives a request from the client, invokes the model to perform the requested operations and sends the data to the View.
- The view formats the data to be presented to the user, in a web application as an html output.



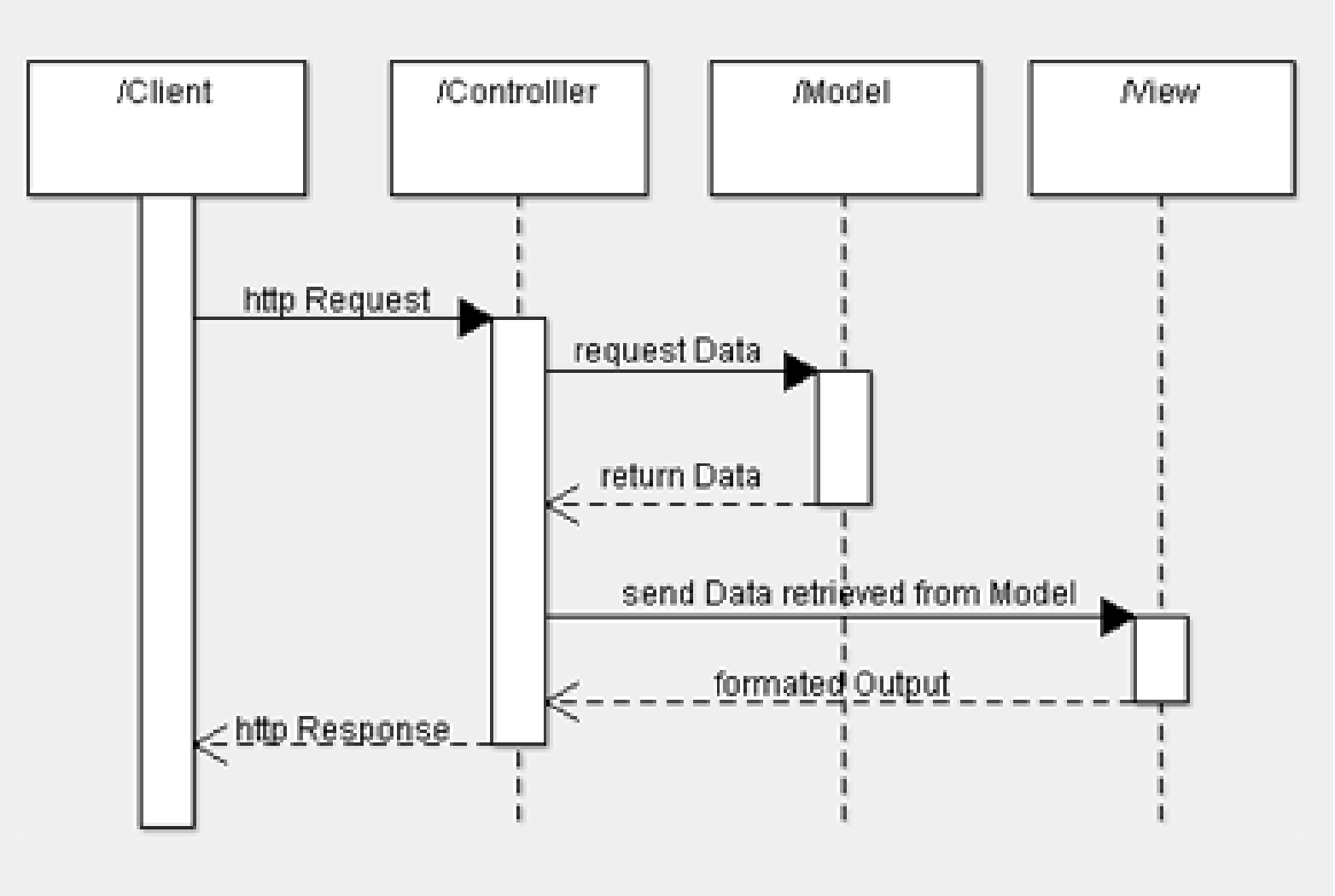
A CRUDE ANALOGY

- The muscles, bones and organs in your body are the ‘model’ that holds everything vital.
- The skin, hair, etc. are the ‘view’ that controls the outward representation of the muscles and bones (the model).
- Your senses are the ‘controller’ that helps the ‘model’ interact with the world.

Controller

- The controller is the first thing which takes a request, parses it, initializes and invoke the model and takes the model response and sends it to the presentation layer.
- It's practically the liant between the Model and the View, a small framework where Model and View are plugged in.
- The application entry point will be index.php. The index.php file will delegate all the requests to the controller.
- The controller decides which data is required from the model. Then it calls the model class to retrieve the data.
- Note that the controller does not know anything about the database or about how the page is generated. (controller/Controller.php)

MVC Sequence Diagram: the flow during a HTTP request



A controller is the environment that bridges the divide between the user and the system.

Model and Entity Classes

- The Model represents the data and the logic of an application, what many calls business logic.
- The model contains the actual data that has to be shown to the user.
- Usually, it's responsible for:
  - storing, deleting, updating the application data.
  - Generally it includes the database operations.
  - Encapsulating the application logic.
  - This is the layer that should implement all the logic of the application.
  - The most common mistakes are to implement application logic operations inside the controller or the view(presentation) layer.
  - Model.php and Book.php (An Entity class)

Model and Entity Classes

- In a good implementation of the MVC pattern only entity classes should be exposed by the model and they should not encapsulate any business logic.
- Their solely purpose is to keep data.
- Depending on implementation Entity objects can be replaced by xml or Json chunk of data.
- The model will include all the entities and the classes to persist data into the database, and the classes encapsulating the business logic.

View (Presentation)

- The view is a *visual* representation of the model.
- The view(presentation layer)is responsible for formatting the data received from the model in a form accessible to the user.
- The data can come in different formats from the model: simple objects(sometimes called Value Objects), xml structures, json, etc.
- The controller delegates the data from the model to a specific view element, usually associated to the main entity in the model.
- The view layer can use a template system to render the html pages.
- The template mechanism can reuse specific parts of the page: header, menus, footer, lists and tables, etc.

View (Presentation)

- The view informs the way data structured in the model will be made visible to the user.
- For example, the HTML may hold the text, but the colors, font size and font style information is actually held by independent CSS stylesheets – i.e. the view.

ADVANTAGES OF MODEL VIEW CONTROLLER PATTERN

- The Model and View are separated, making the application more flexible.
- The Model and view can be changed separately, or replaced.
- For example a web application can be transformed in a smart client application just by writing a new View module, or an application can use web services in the backend instead of a database, just replacing the model module.
- Each module can be tested and debugged separately.