

# MỤC LỤC

- 1. TỔNG QUAN VỀ MVC..... 1
  - 1.1 MÔ HÌNH TRUYỀN THÔNG ..... 1
  - 1.2 MÔ HÌNH MVC (MODEL-VIEW-CONTROLLER)..... 2
  - 1.3 CÁCH VẬN HÀNH CỦA MÔ HÌNH ..... 3
  - 1.4 ƯU ĐIỂM CỦA MVC ..... 3
  - 1.5 NHƯỢC ĐIỂM CỦA MVC..... 3

# TÌM HIỂU MÔ HÌNH MVC

## 1. Tổng Quan Về MVC

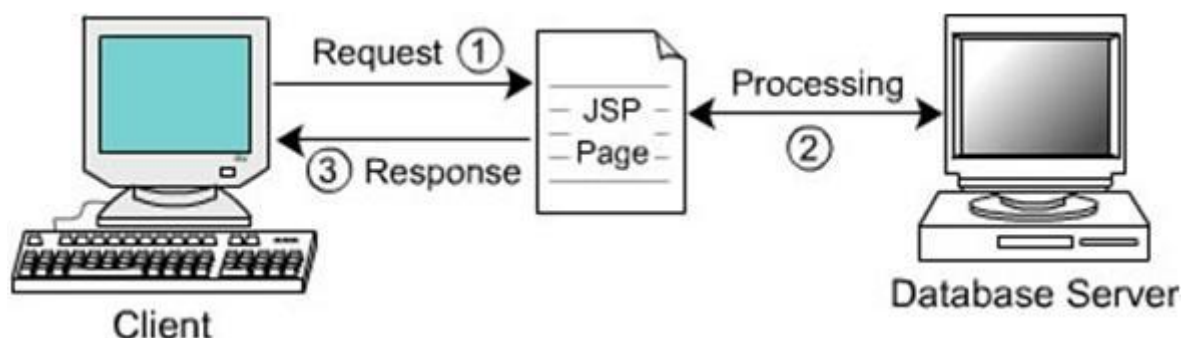
Khi viết một chương trình, bước thiết kế chương trình là một trong những bước quan trọng. Nó quyết định đến sự thành bại của dự án mình làm. Nếu ở bước thiết kế tốt thì chúng ta sẽ dễ dàng phân chia công việc trong nhóm. Dễ dàng nâng cấp hay fix lỗi khi gặp phải... bài viết sẽ tìm hiểu và trình bày về mô hình mvc trong java qua những nội dung sau

### 1.1 Mô Hình Truyền Thống

Trong quá trình lập trình với jsp chúng ta thường đan xen các mã html cùng với mã jsp, đôi khi ta dùng taglib của jsp có khi lại toàn code java trong đó. Do vậy, có nhiều khó khăn ta có thể gặp phải:

- Người thiết kế giao diện cũng cần phải biết các ngôn ngữ lập trình, hoặc bạn phải trực tiếp thiết kế chúng.
- Việc bảo trì chúng thường rất khó khăn, vì một phần các mã chương trình lẫn lộn với mã html.
- Trong một nhóm hay một dự án lớn sẽ rất khó phân công và chia tách công việc
- Khi có lỗi xảy ra, việc tìm ra lỗi và định vị lỗi cũng là một vấn đề. Và còn nhiều những khó khăn khác ...

Mô hình lập trình cổ điển



## 1.2 Mô Hình MVC (Model-View-Controller)

Để khắc phục các khó khăn trên, người ta đưa ra mô hình khác là mô hình mvc (model-view-controller). Tức là: tương ứng với một trang jsp ngày xưa, bây giờ người ta tách nó ra làm ba thành phần: mô hình – khung nhìn – bộ điều khiển.

Phương pháp thiết kế MVC là phương pháp chia nhỏ một ứng dụng thành nhiều lớp hoặc chia nhỏ phần giao diện người dùng (user interface) của một ứng dụng thành 3 phần chính là model, view và controller.

Với phương pháp thiết kế này, các chức năng hiển thị, chức năng logic điều khiển và chức năng truy cập dữ liệu của chương trình được chia thành các phần riêng biệt. Java là một ngôn ngữ lập trình hướng đối tượng thuần túy nên việc áp dụng MVC vào các phần mềm viết bằng java rất dễ dàng và thuận tiện.

Các thành phần trên làm việc như sau:

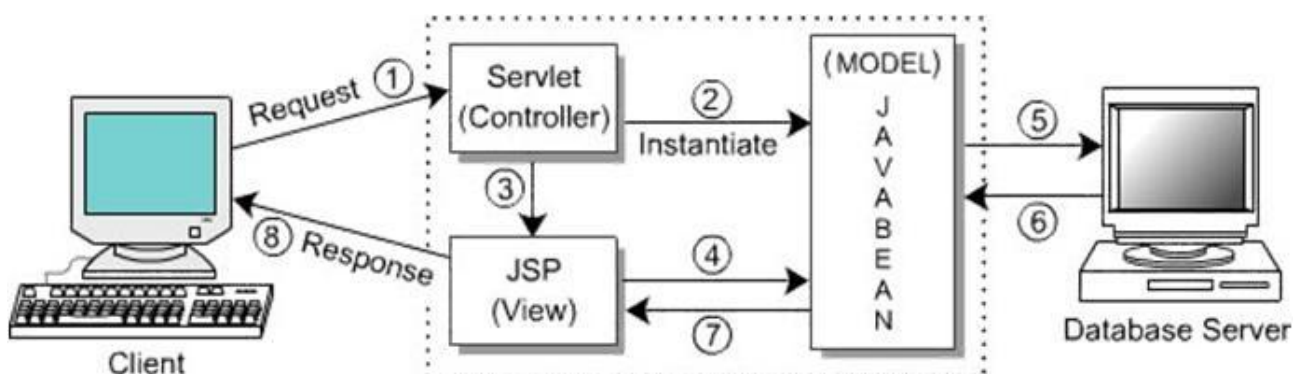
**Model:** mô hình là các lớp java có nhiệm vụ:

- Nhận các yêu cầu từ khung nhìn
- Thi hành các yêu cầu đó (tính toán, kết nối csdl ...)
- Trả về các giá trị tính toán cho view.

**View:** bao gồm các mã tương tự như jsp để hiển thị form nhập liệu, các kết quả trả về từ mô hình...

**Controller:** đồng bộ hoá giữa **view** và **model**. Tức là với một trang jsp này thì sẽ tương ứng với lớp java nào để xử lý nó và ngược lại, kết quả sẽ trả về trang jsp nào.

Như vậy, chúng ta có thể tách biệt được các mã java ra khỏi mã html. Do vậy, nó đã giải quyết được các khó khăn đã nêu ra trong mô hình cổ điển. Người thiết kế giao diện và người lập trình java có thể mang tính chất độc lập tương đối. Việc debug hay bảo trì sẽ dễ dàng hơn, việc thay đổi các theme của trang web cũng dễ dàng hơn.



### 1.3 Cách Vận Hành Của Mô Hình

Đối với 3-Layer, yêu cầu được xử lý tuần tự qua các layer như hình.

- Đầu tiên User giao tiếp với View để gửi đi thông tin và yêu cầu. Tại đây, các thông tin sẽ được kiểm tra, nếu OK chúng sẽ được chuyển xuống Controller.

- Tại Controller, các thông tin sẽ được nhào nặn, tính toán theo đúng yêu cầu đã gửi, nếu không cần đến Database thì Controller sẽ gửi trả kết quả về GUI, ngược lại nó sẽ đẩy dữ liệu (thông tin đã xử lý) xuống Model.

- Model sẽ thao tác với Database và trả kết quả về cho Controller, Controller kiểm tra và gửi nó lên View để hiển thị cho người dùng.

- Một khi gặp lỗi (các trường hợp không đúng dữ liệu) thì đang ở layer nào thì báo lên trên layer cao hơn nó 1 bậc cho tới View thì sẽ báo cho người dùng.

### 1.4 Ưu Điểm Của MVC

Thể hiện tính chuyên nghiệp trong lập trình, phân tích thiết kế. Do được chia thành các thành phần độc lập nên giúp dễ dàng mở rộng, thay đổi quy mô của hệ thống: Khi cần tải lớn, người quản trị có thể dễ dàng thêm các máy chủ vào nhóm, hoặc lấy bớt ra trong trường hợp ngược lại.

### 1.5 Nhược Điểm Của MVC

Đối với dự án nhỏ việc áp dụng mô hình MC gây cồng kềnh, tốn thời gian trong quá trình phát triển. Tốn thời gian trung chuyển dữ liệu của các thành phần.

Việc truyền dữ liệu giữa các tầng sẽ chậm hơn vì phải truyền giữa các tiến trình khác nhau (IPC), dữ liệu cần phải được đóng gói -> truyền đi -> mở gói trước khi có thể dùng được.

Việc phát triển ứng dụng phức tạp hơn.