

# **BÀI 7:**

## **Phát triển ứng dụng JSP-Servlet theo mô hình MVC**

# Mục tiêu bài học

- ◆ Giới thiệu về mô hình: Kiến trúc tổng quan, cơ chế hoạt động
- ◆ Vai trò và ý nghĩa của các thành phần trong mô hình MVC
  - ◆ Model
  - ◆ View
  - ◆ Controller
- ◆ Các bước xây dựng ứng dụng JSP-Servlet theo mô hình MVC

## Giới thiệu về design pattern

- ◆ Trong quá trình thiết kế ứng dụng Web, ta cần đảm bảo rằng ứng dụng Web sẽ đáp ứng được những mục tiêu sau:
  - ◆ Khả năng bảo trì và mở rộng (extensibility and maintainability)
  - ◆ Phân chia công việc (division of work)
  - ◆ Khả năng mở rộng (Scalability)
  - ◆ Khả năng tái sử dụng lại mã nguồn (code reusability)
- ◆ Những mục tiêu này sẽ giống nhau ở các loại ứng dụng, ngoài ra các ứng dụng thuộc các lĩnh vực khác nhau sẽ có cùng các vấn đề thông dụng, vì vậy ta cần sử dụng các design pattern.

## Giới thiệu về design pattern

- ◆ Design pattern: Là các giải pháp đúng đắn, đã được kiểm chứng cho những vấn đề phổ biến của các ứng dụng phần mềm.
- ◆ Một số lợi ích của việc sử dụng design patterns:
  - ◆ Là các giải pháp đã được kiểm chứng (Proven design)
  - ◆ Có thể tái sử dụng (Reusable)
  - ◆ Có thể diễn đạt, giải thích (Expressive)
- ◆ MVC là một design pattern phổ biến, được sử dụng trong quá trình phát triển các ứng dụng phần mềm.
- ◆ MVC design cho phép phát triển các ứng dụng, với việc tách biệt mã nguồn của ứng dụng thành các phần data access, presentation, và business logic của ứng dụng thành các thành phần sau:
  - ◆ Model
  - ◆ View
  - ◆ Controller

## Thành phần Model

### ◆ Model:

- ◆ Là thành phần chứa dữ liệu và tương tác với dữ liệu của ứng dụng.
- ◆ Là các non-visual objects và cung cấp dữ liệu cho ứng dụng. Dữ liệu hiển thị trên giao diện người dùng sẽ được cung cấp bởi model.
- ◆ Được tạo bằng JavaBeans hoặc các công nghệ khác.

## Thành phần View

### ◆ Thành phần view:

- ◆ Là thành phần thể hiện các màn hình giao diện của ứng dụng
- ◆ View thể hiện thông tin được cung cấp bởi model.
- ◆ View có thể hiển thị dữ liệu được cung cấp bởi model theo nhiều dạng khác nhau: windows form, web form, console.
- ◆ View có 2 tác dụng:
  - ◆ Hiển thị dữ liệu
  - ◆ Tương tác với người dùng
- ◆ Trong ứng dụng Java Web, thành phần View được phát triển bằng các trang JSP.



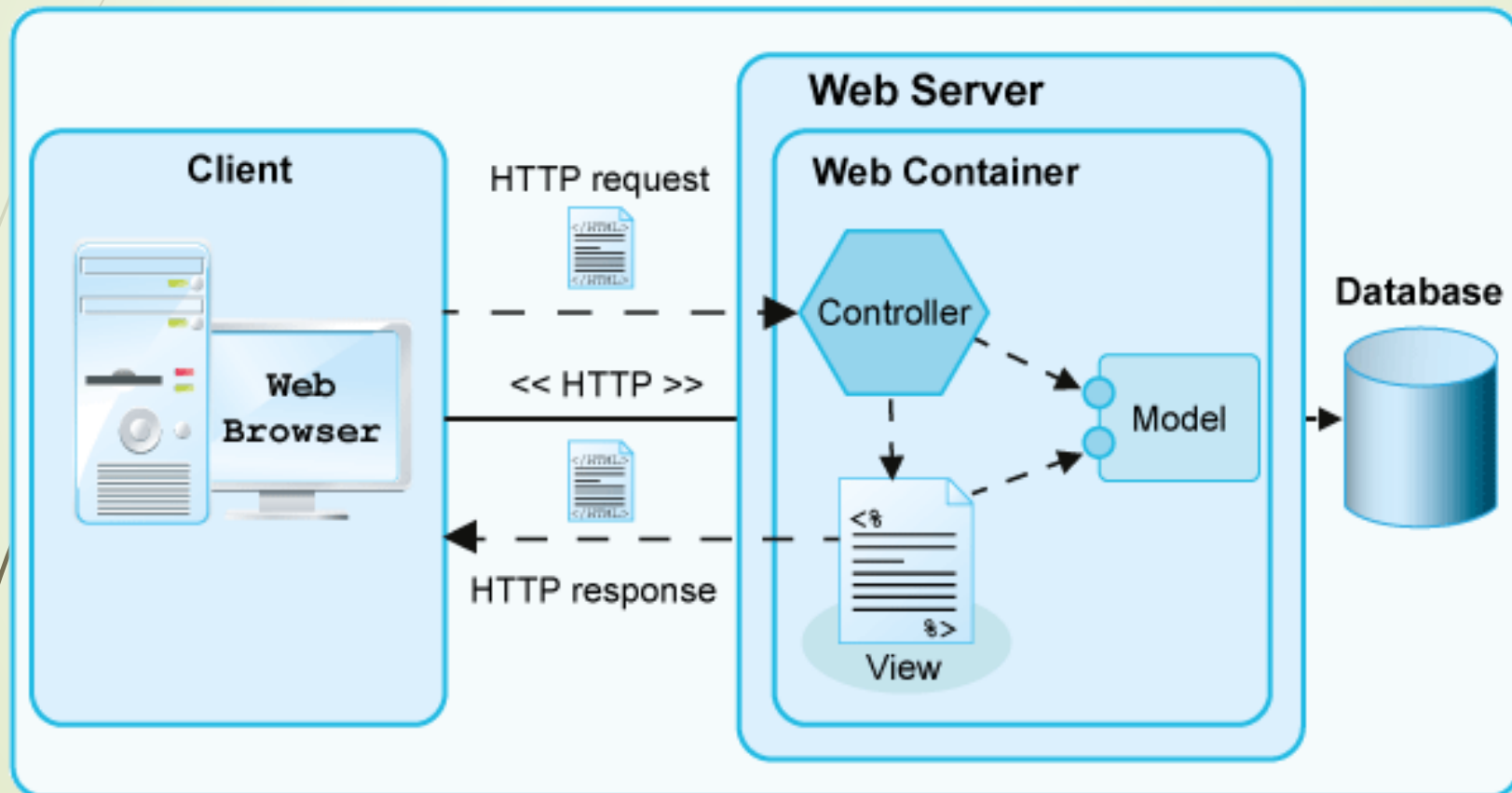
## Thành phần Controller

### ◆ Thành phần controller:

- ◆ Là thành phần trung gian giữa model và view, chịu trách nhiệm xử lý các request được nhận từ view, và xử lý dữ liệu được truyền từ model.
- ◆ Chịu trách nhiệm kiểm soát toàn bộ luồng thực thi của ứng dụng.
- ◆ Được cài đặt bằng cách sử dụng các servlets.
- ◆ Controller cung cấp việc xử lý cho các view, chịu trách nhiệm trong việc xử lý, điều hướng, lựa chọn, điều khiển các view.

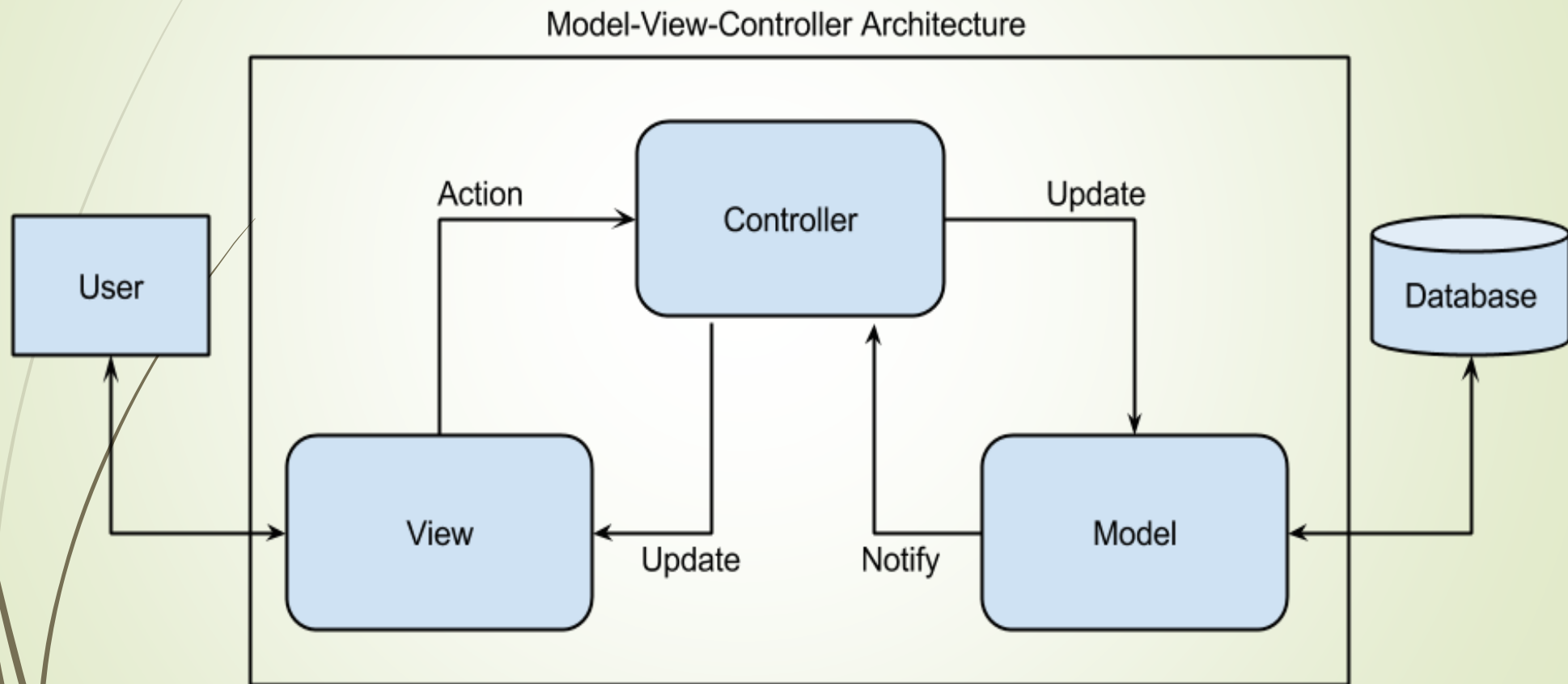
## Thành phần Controller

- Hình bên dưới minh họa về quá trình tương tác giữa ba thành phần model, view, và controller trong ứng dụng.





## ◆ Tương tác giữa 3 thành phần Model-View-Controller



## Lưu ý

- ◆ Một số điều cần lưu ý trong mô hình MVC:
  - ◆ Khi áp dụng mô hình MVC, các view sẽ không trực tiếp nhận request từ phía client, mà request sẽ được gửi tới controller. Controller sẽ tiếp nhận request, xử lý request rồi triệu gọi view tương ứng để hiển thị kết quả.
  - ◆ Trong mô hình MVC, view không chứa dữ liệu mà chỉ hiển thị dữ liệu. Dữ liệu sẽ được cung cấp bởi model.
  - ◆ Các view không trực tiếp tương tác và forward đến nhau, mà sẽ phải thông qua controller.

## Minh họa

- ◆ Giáo viên demo xây dựng ứng dụng MVC
- ◆ Học viên thực hành

- ◆ Giới thiệu về mô hình: Kiến trúc tổng quan, cơ chế hoạt động
- ◆ Vai trò và ý nghĩa của các thành phần trong mô hình MVC
  - ◆ Model
  - ◆ View
  - ◆ Controller
- ◆ Các bước xây dựng ứng dụng JSP-Servlet theo mô hình MVC