



Bộ môn Công nghệ Phần mềm
Viện CNTT & TT
Trường Đại học Bách Khoa Hà Nội

LẬP TRÌNH WEB HƯỚNG JAVA

Bài 07: Tổng quan MVC

Giảng viên: ThS. Trịnh Tuấn Đạt
Bộ môn CNPM

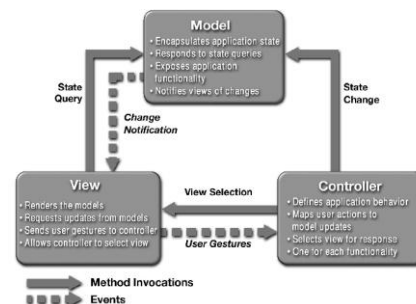
Email: tringhtuandat.bk@gmail.com/dattt@soict.hut.edu.vn

Nội dung

- 1. Giới thiệu về mẫu thiết kế MVC
- 2. Quá trình phát triển của kiến trúc ứng dụng Web
 - 2.1. Model 1
 - 2.2. Model 2
 - 2.3. Application frameworks

1. Giới thiệu về mẫu thiết kế MVC

MVC Pattern



Model

- Model (**Business process layer**)
 - Mô hình hóa dữ liệu và hành vi (**data & behavior**) trong xử lý nghiệp vụ (**business process**)
 - Chịu trách nhiệm
 - Thực hiện các truy vấn DB
 - Tính toán trong các nghiệp vụ
 - VD: Xử lý các **orders**
 - Đóng gói dữ liệu và hành vi, độc lập với tầng presentation

View

- View (**Presentation layer**)
 - Hiển thị thông tin tùy thuộc vào loại client
 - Biểu diễn kết quả của tầng business logic (**Model**)
 - Không cần quan tâm làm thế nào có được thông tin, hoặc thông tin ở đâu (Model chịu trách nhiệm)

Controller

- Controller (**Control layer**)
 - Kết nối tương tác của người dùng với các nghiệp vụ cung cấp phía sau
 - Chọn ra cách biểu diễn phù hợp
 - Ví dụ: ngôn ngữ, biến đổi định dạng thông tin theo vùng, quyền hạn người dùng
 - Một request tới ứng dụng sẽ chuyển cho tầng Control.
 - Tầng này quyết định request được xử lý như thế nào, và thông tin sẽ được trả lại như thế nào

7

2. Quá trình phát triển của kiến trúc ứng dụng Web

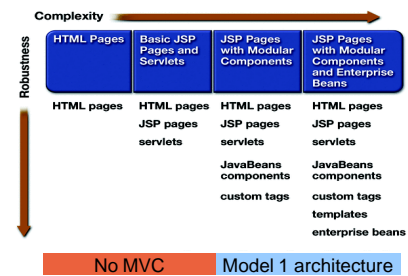
8

2. Quá trình phát triển của kiến trúc MVC

- No MVC
- MVC Model 1 (**Page-centric**)
- MVC Model 2 (**Servlet-centric**)
- Web application frameworks
 - Struts
- Standard-based** Web application framework
 - JavaServer Faces (JSR-127)

9

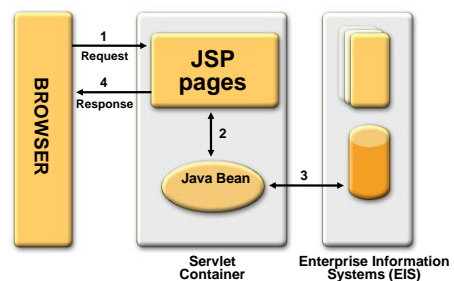
2. Quá trình phát triển của kiến trúc ứng dụng Web cho đến khi Model 1 ra đời



10

2.1. Model 1 Architecture (Page-centric)

2.1. Model 1 (Page-Centric Architecture)



11

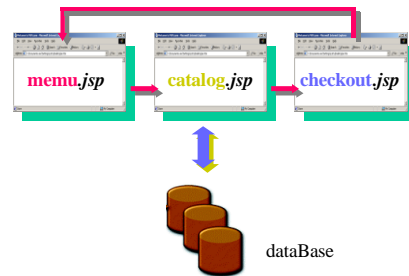
12

Kiến trúc Page-centric

- Bao gồm 1 loạt các trang JSP có liên hệ chặt chẽ với nhau
 - Các trang JSP xử lý tất cả: **presentation**, **control**, và **business process**
- Business process logic** và **control** được **CODE CỨNG** trong các trang JSP
 - Dưới dạng JavaBeans, scriptlets, expression
- Chuyển trang được thực hiện
 - Khi user click vào 1 liên kết. Ví dụ: ``
 - Qua hành động submit form. Ví dụ: `<FORM ACTION="search.jsp">`

13

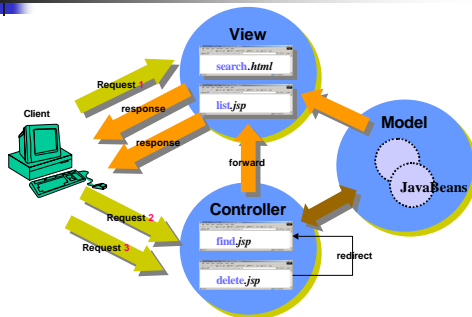
Kiến trúc Page-centric



page-centric catalog application

14

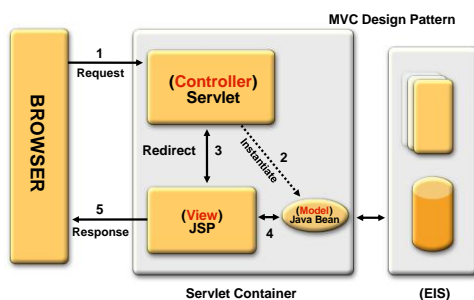
Kịch bản sử dụng trong kiến trúc Page-centric



15

2.2. Model 2 (Servlet-Centric Architecture)

Model 2 Architecture (Servlet-centric)



17

Tại sao sử dụng kiến trúc Model 2?

- Nếu muốn biểu diễn các trang JSP khác nhau, tùy theo dữ liệu nhận được?
 - Riêng JSP với JavaBeans và custom tags (Model 1) chưa xử lý tốt được
- Giải pháp
 - Sử dụng đồng thời Servlet và JSP (Model 2)
 - Servlet xử lý request gửi tới, xử lý 1 phần dữ liệu, thiết lập các beans, forward kết quả cho 1 trong nhiều trang JSP nào đó

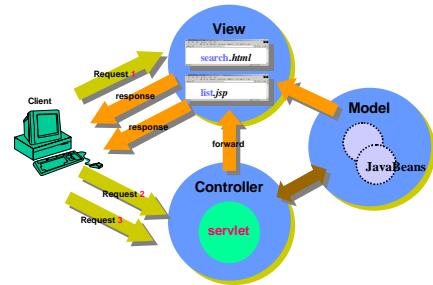
18

Servlet-centric Architecture

- JSP chỉ được sử dụng để biểu diễn kết quả (**presentation**)
 - Xử lý điều khiển (Control) thực hiện bởi servlets
- Servlet hoạt động như một **gatekeeper**
 - Cung cấp các services thông dụng, như **authentication**, **authorization**, **login**, **error handling**, ...
- Servlet hoạt động như một **central controller**
 - Quyết định logic phù hợp để xử lý các request, sẽ gửi request đến những nơi nào, ...
 - Thực hiện việc điều hướng (redirecting)

19

Servlet-centric Scenario



20

2.3. Web Application Frameworks

Web Application Frameworks

- Dựa trên kiến trúc MVC Model 2
- Hầu hết các ứng dụng Web phải cung cấp các chức năng
 - Nhận (**receive**) và gửi tiếp (**Dispatching**) HTTP requests
 - Gọi các phương thức từ tầng model
 - Tổng hợp và chọn ra các views trả về cho client
- Cung cấp các classes và interfaces cho lập trình viên sử dụng/mở rộng

21

22

Tại sao sử dụng các Web Application Framework?

- Phân tách tầng presentation và các business logic thành các components
- Cung cấp 1 điểm điều khiển trung tâm
- Cung cấp các tính năng mở rộng
- Dễ dàng kiểm thử unit (**unit-testing**) và bảo trì
- Nhiều công cụ hỗ trợ
- Ổn định
- Có cộng đồng hỗ trợ mạnh mẽ
- Đơn giản hóa chế độ đa ngôn ngữ (internationalization)
- Đơn giản hóa việc validate đầu vào

23

Tại sao sử dụng các Web Application Framework?

- Frameworks đang phát triển mạnh mẽ
- JSP/Servlets vẫn còn khó sử dụng
- Frameworks định nghĩa các components chuẩn, cho phép tái sử dụng.
- Frameworks còn chỉ rõ cách thức phối hợp các components trong 1 ứng dụng

24



Một số Web Application Frameworks

- Apache Struts I and II
- Spring Framework MVC
- JavaServer Faces (JSR-127)
- Echo
- Tapestry
- Wicket
- ...