

Phát triển ứng dụng web

NodeJs
part 2

Nội dung

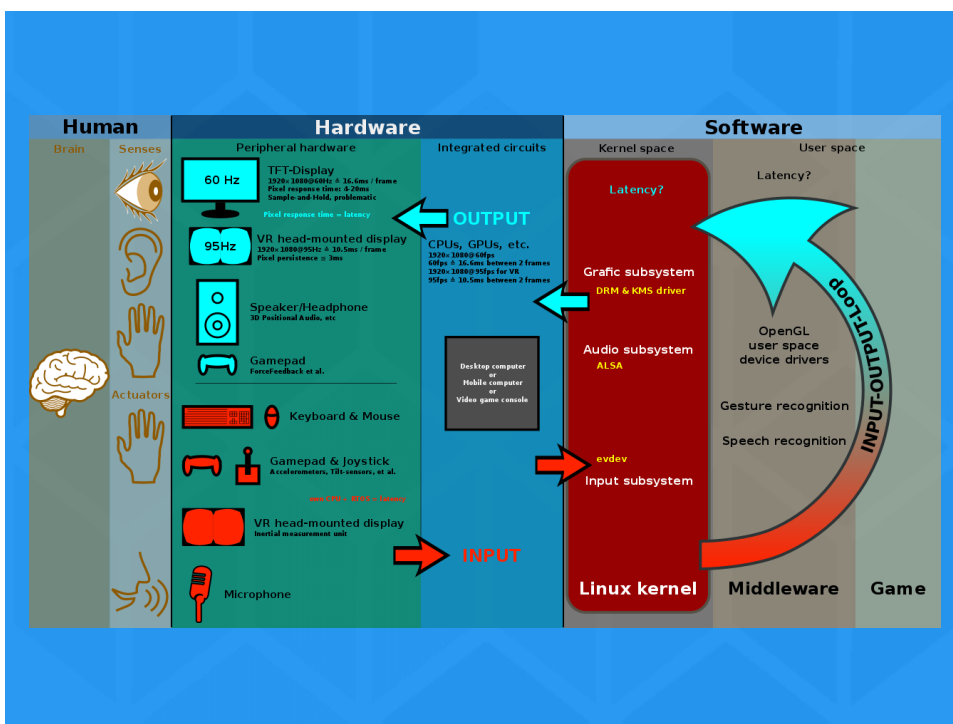
- ☐ Middleware
- ☐ MVC Pattern

Nội dung

- ❑ **Middleware**
- ❑ MVC Pattern

Khái niệm Middleware

- ❑ **Middleware** trong ngành công nghệ phần mềm được định nghĩa là một phần mềm có nhiệm vụ làm cầu nối (bridge), cung cấp các dịch vụ từ phía hệ điều hành đến các ứng dụng, giúp các ứng dụng có thể tương tác với các thành phần được hệ điều hành cho phép.
Middleware được coi là chất kết dính giữa các phần mềm với nhau (*wikipedia*)



Middleware trong Web

- ❑ Trong web server, Middleware sẽ đóng vai trò trung gian giữa **request/response** (tương tác với người dùng) và các xử lý logic bên trong web server.
- ❑ Về bản chất, Middleware là các hàm được dùng để tiền xử lý, lọc các **request** trước khi đưa vào xử lý logic hoặc điều chỉnh các **response** trước khi gửi về cho người dùng.

Middleware trong ExpressJS

- ❑ Cơ bản khi **ExpressJS** hoạt động là các **Middleware** được thực hiện liên tiếp nhau từ quá trình nhận **request** cho đến **response**.
- ❑ Tham số cơ bản cho **Middleware** trong **ExpressJS** là:
 - ❑ **req**: Request
 - ❑ **res**: Response
 - ❑ **next**: để chuyển sang **Middleware** tiếp theo.
 - ❑ **err**: quản lý lỗi.

Chức năng middleware trong expressjs

- ❑ Thực hiện bất cứ đoạn code nào.
- ❑ Làm thay đổi các đối tượng request và response.
- ❑ Kết thúc một quá trình request - response.
- ❑ Gọi hàm middleware tiếp theo trong stack.
- ☞ *Nếu hàm middleware hiện tại không kết thúc quá trình request – response thì nó phải gọi hàm next() để chuyển sang middleware tiếp theo.*

Các loại Middleware trong ExpressJS

- ☐ Application-level middleware (middleware cấp ứng dụng).
- ☐ Router-level middleware (middleware cấp điều hướng - router).
- ☐ Error-handling middleware (middleware xử lý lỗi).
- ☐ Built-in middleware (middleware sẵn có).
- ☐ Third-party middleware (middleware của bên thứ ba).

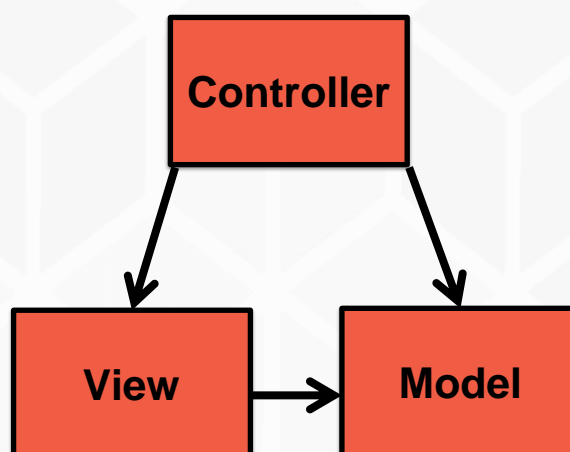
Nội dung

- ☐ *Middleware*
- ☐ **MVC Pattern**

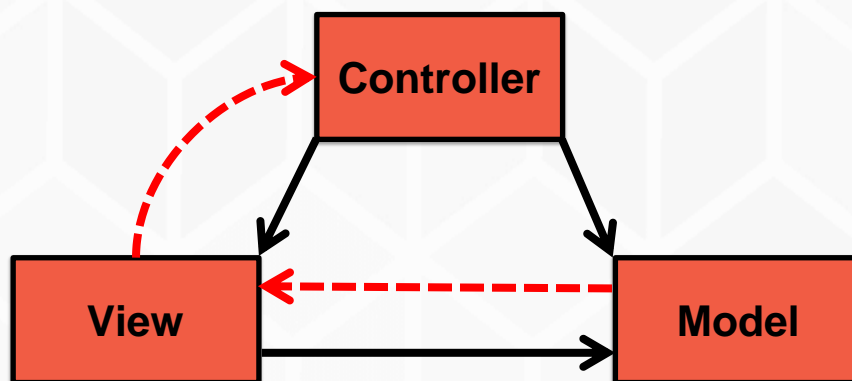
MVC pattern

- ❑ **Model-View-Controller (MVC)** pattern chia nhỏ các thành phần dữ liệu, thể hiện (output) và dữ liệu nhập từ người dùng (input) thành những thành phần riêng biệt.
 - ❑ **Model** chứa dữ liệu và các tính toán xử lý logic để giải quyết vấn đề mà phần mềm hướng tới (business logic).
 - ❑ **View** là thành phần đảm nhận việc thể hiện những dữ liệu của Model. View bao gồm những gì thể hiện trên màn hình như các control, form... Trên cùng một Model, có thể có nhiều View.
 - ❑ **Controller** là thành phần đảm nhận việc xử lý đáp trả lại các dữ liệu được đưa vào từ người dùng như các sự kiện chuột, bàn phím, các tương tác lên các control... Controller là cầu nối giữa người dùng và ứng dụng.

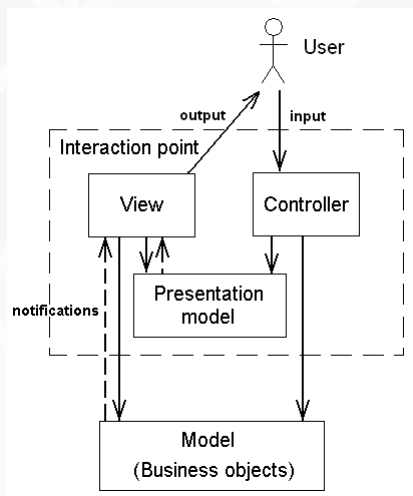
Mô hình MVC cơ bản



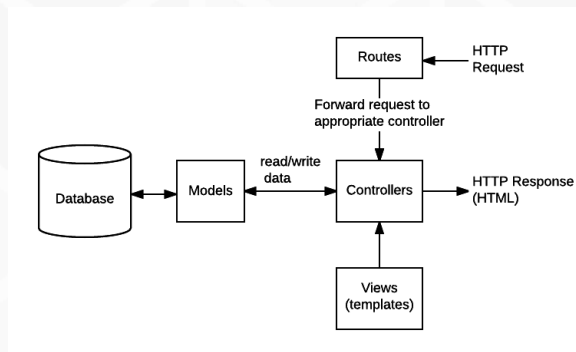
MVC Observer Pattern



Mô hình tương tác



Mô hình MVC với ExpressJS

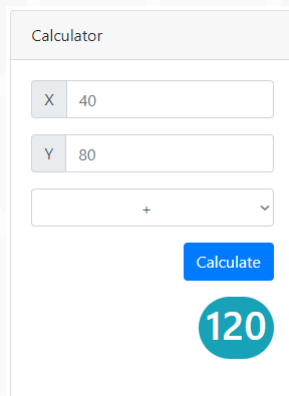


Nội dung

- ☐ MVC Pattern
- ☐ Middleware
- ☒ **Bài tập**

Bài tập

- ❑ Xây dựng ứng dụng Web “Bé tập tính” với express.js cho phép nhập vào 2 số x và y, lựa chọn phép tính và hiển thị kết quả (sử dụng **MVC pattern**).



- ❑ Yêu cầu sử dụng **MVC pattern** để xây dựng ứng dụng web giúp ghi nhận và phản hồi về việc đăng ký tham gia Hội nghị với các chức năng:
 - ❑ Trang chủ giới thiệu thông tin về Hội nghị
 - ❑ Trang xác nhận tham gia Hội nghị với các thông tin: Họ tên, Email, Điện thoại và xác nhận tham gia hay không.
 - ❑ Trang phản hồi kết quả.

☐ Sử dụng MVC pattern để xây dựng ứng dụng hiển thị dữ liệu từ address: '<https://reqres.in>' với các chức năng

☐ Hiển thị danh sách user (có phân trang)

☐ Xem chi tiết user

☐ Tìm kiếm

☐ Thêm user

☐ Cập nhật / Xóa user