Báo cáo đồ án môn học

**Kiến trúc phần mềm – TH2014**

*TP. HCM, tháng 1/2018*

Nội dung

[1. Giới thiệu và mục đích của tài liệu 1](#_Toc503361865)

[2. Thành viên nhóm 1](#_Toc503361866)

[3. Các kỹ thuật sử dụng trong ứng dụng 1](#_Toc503361867)

[3.1. Các kỹ thuật sử dụng trong Model 1](#_Toc503361868)

[3.1.1. Validations 1](#_Toc503361869)

[3.1.2. Callbacks 1](#_Toc503361870)

[3.1.3. Migrations 1](#_Toc503361871)

[3.2. Các kĩ thuật sử dụng trong Controller 1](#_Toc503361872)

[3.2.1. Sessions 1](#_Toc503361873)

[3.2.2. Cookies 1](#_Toc503361874)

[3.2.3. HTTP Authentications 2](#_Toc503361875)

# Giới thiệu và mục đích của tài liệu

# Thành viên nhóm

1. Đinh Đức Thông – 1412530
2. Phan Quốc Trung – 1412590

# Các kỹ thuật sử dụng trong ứng dụng

## Các kỹ thuật sử dụng trong Model

### Validations

#### Giới thiệu

**Active Record** cho phép bạn phê chuẩn (**validate**) trạng thái của **model** trước khi nó được ghi vào cơ sở dữ liệu. Có một vài phương thức mà bạn có thể sử dụng để kiểm tra các model của bạn và phê chuẩn rằng một giá trị thuộc tính không rỗng, là duy nhất và chưa tồn tại trong cơ sở dữ liệu, tuân theo một định dạng cụ thể và nhiều hơn nữa.

**Validation** là một vấn đề rất quan trọng để xem xét khi lưu vào cơ sở dữ liệu, nên các phương thức save và update đưa nó vào tầm quan trọng khi chạy: Chúng trả về false khi validation thất bại và chúng đã không thực sự thực hiện bất kỳ thao tác nào trên cơ sở dữ liệu.

#### Sử dụng trong ứng dụng

### Callbacks

#### Giới thiệu

Các **Active Record callback** cho phép bạn gắn mã vào các sự kiện nhất định trong chu kì sống của các model của bạn. Điều này cho phép bạn thêm hành vi (behavior) vào các model của bạn bằng sự thực thi mã một cách rõ ràng khi các sự kiện đó xảy ra, như khi bạn tạo một bản ghi mới, cập nhật nó, hủy nó…

#### Sử dụng trọng ứng dụng

### Migrations

#### Giới thiệu

Rails cung cấp một ngôn ngữ domain-specific cho việc quản lí một lược đồ cơ sở dữ liệu được gọi là các **migration**. Các migration được lưu trong các tập tin được thực thi với bất kì cơ sở dữ liệu mà Active Record hỗ trợ sử dụng rake.

#### Sử dụng trong ứng dụng

## Các kĩ thuật sử dụng trong Controller

### Sessions

#### Giới thiệu

Ứng dụng của bạn có một session cho mỗi người dùng trong đó bạn có thể lưu một số lượng nhỏ của dữ liệu mà sẽ được lưu trữ giữa các request. Session chỉ có hiệu lực trong controller và view và có thể dùng một trong các cơ chế lưu trữ khác nhau:

* ActionDispatch::Session::CookieStore – Lưu mọi thứ trên client.
* ActionDispatch::Session::CacheStore – Lưu dữ liệu trên Rails cache.
* ActionDispatch::Session::ActiveRecordStore – Lưu dữ liệu vào cơ sở dữ liệu sử dụng Active Record. (yêu cầu activerecord-session\_store gem).
* ActionDispatch::Session::MemCacheStore – Lưu dữ liệu vào một memcached cluster.

Mọi session store sử dụng một cookie để lưu một ID duy nhất cho mỗi session (bạn phải sử dụng một cookie, Rails sẽ không cho phép bạn đưa session ID vào URL khi điều này bảo mật kém hơn).

#### Sử dụng trong ứng dụng

### Cookies

#### Giới thiệu

Ứng dụng của bạn có thể lưu trữ các số lượng nhỏ của dữ liệu trên client – được gọi là các cookie – mà sẽ được lưu qua các request và thậm chí các session. Rails cung cấp sự truy cập dễ dàng tới các cookie thông qua các phương thức cookie, mà, giống với session, làm việc như một hash.

#### Sử dụng trong ứng dụng

### HTTP Authentications