



---

# Bài 11

# Internationalization

Module: BOOTCAMP WEB-BACKEND DEVELOPMENT WITH  
LARAVEL 2.1

# Mục tiêu

---



- Trình bày được khái niệm và ý nghĩa của i18n
- Triển khai được i18n trong ứng dụng Laravel
- Sử dụng được session để ghi nhớ Locale của người dùng
- Triển khai được i18n cho Controller
- Triển khai được i18n cho View
- Triển khai được i18n cho các thông báo có sử dụng tham số

---

# Localization

Giới thiệu

Nhận dòng ngôn ngữ

Ghi đè nội dung của file ngôn ngữ

# Giới thiệu

---

- Localization là cách thức cho phép ứng dụng đáp ứng được người dùng ở các địa phương khác nhau
- Localization còn được viết tắt là i10n
- Localization thường được nhắc đến cùng với Internationalization
  - quốc tế hoá - i18n
- Laravel có cơ chế hỗ trợ i10n

# Các file ngôn ngữ

---

- Các file ngôn ngữ được đặt trong thư mục resources/lang

```
/resources
  /lang
    /en
      messages.php
    /es
      messages.php
```

# Nội dung file ngôn ngữ

---

- Trả về một mảng các từ hoặc thông điệp

```
<?php  
  
return [  
    'welcome' => 'Welcome to our application'  
];
```

# Cấu hình Địa phương

---

- Địa phương mặc định được thiết lập trong file config/app.php
- Sử dụng hàm setLocale của App facade để thay đổi địa phương

```
Route::get('welcome/{locale}', function ($locale) {  
    App::setLocale($locale);  
  
    //  
});
```

# Xác định Địa phương hiện tại

---

- Sử dụng `App::getLocale()` để lấy về địa phương hiện tại

```
$locale = App::getLocale();  
  
if (App::isLocale('en')) {  
    //  
}
```



# Lấy về chuỗi được dịch

---

- Sử dụng hàm `__()` để lấy về các chuỗi được dịch

```
echo __('messages.welcome');  
  
echo __('I love programming.');
```

- Có thể sử dụng `__()` hoặc `@lang()` ở trong View

```
{{ __('messages.welcome') }}
```

  

```
@lang('messages.welcome')
```

# Truyền tham số vào chuỗi được dịch



- Có thể truyền tham số vào chuỗi được dịch

```
'welcome' => 'Welcome, :name',
```

```
echo __('messages.welcome', ['name' => 'dayle']);
```

# Upload file lên server

---

- Sử dụng phương thức store() để lưu các file được tải lên
- Tên file được sinh ngẫu nhiên

```
public function update(Request $request)
{
    $path = $request->file('avatar')->store('avatars');

    return $path;
}
```



# Quy định tên của file tải lên

---

- Sử dụng phương thức storeAs() để quy định tên của file tải lên

```
$path = $request->file('avatar')->storeAs(  
    'avatars', $request->user()->id  
);
```



# Nhúng CSS và JavaScript

---

- Laravel sử dụng npm để quản lý và biên dịch CSS và JavaScript
- Có thể sử dụng SASS hoặc Less để viết style
- Mặc định, Laravel sử dụng thư viện Vue cho front-end
- Laravel Mix là tính năng cho phép biên dịch CSS và JavaScript dễ dàng
- Mã nguồn của JavaScript và CSS (SCSS) được đặt trong thư mục `resources/assets/`

# Viết mã CSS

---

- Mặc định, Laravel đính kèm Bootstrap cho phần front-end
- Chạy lệnh ***npm run dev*** để biên dịch các file scss
- File app.css được sinh ra và chứa toàn bộ các mã css của ứng dụng
- Có thể tùy chỉnh file app.css thông qua việc tùy chỉnh file app.scss rồi biên dịch lại

# Viết mã JavaScript

---



- Chạy lệnh ***npm run dev*** để biên dịch các mã JavaScript
- File app.js được sinh ra và chứa toàn bộ mã JavaScript của ứng dụng
- Có thể nhúng các file JavaScript khác nhau vào trong ứng dụng
- Các file JavaScript sau khi được biên dịch thì sẽ được đặt trong thư mục public/js

---

# Hướng dẫn

- Hướng dẫn làm bài thực hành và bài tập
- Chuẩn bị bài tiếp: ***Bảo mật ứng dụng trong Laravel***