### **Заметки к занятию 3. 06.11.2024**

#### **Проверка загрузки изображения и добавление водяного знака**

При загрузке изображения в Laravel можно проверять его валидность и добавлять водяной знак.

**Проверка загрузки файла**:

Проверяем, был ли загружен файл и является ли он валидным.  
  
if ($request->hasFile('image') && $request->file('image')->isValid()) {

// Получаем путь к загруженному изображению

$imagePath = $request->file('image')->getPathname();

**Добавление водяного знака на изображение**:

Устанавливаем текст водяного знака, цвет и позицию на изображении.  
// Текст водяного знака

$text = "Watermark Text"; // Замените на нужный текст

return response()->stream(function () use ($imagePath, $text) {

$image = imagecreatefromjpeg($imagePath);

// Определяем цвет текста (например, белый)

$textColor = imagecolorallocate($image, 255, 255, 255);

// Устанавливаем параметры шрифта

$fontSize = 5; // Размер шрифта (1-5 для встроенных шрифтов PHP)

$margin = 10; // Отступ от края изображения

// Получаем ширину и высоту текста

$textWidth = imagefontwidth($fontSize) \* strlen($text);

$textHeight = imagefontheight($fontSize);

// Рассчитываем позицию текста в правом нижнем углу

$x = imagesx($image) - $textWidth - $margin;

$y = imagesy($image) - $textHeight - $margin;

// Добавляем текст на изображение

imagestring($image, $fontSize, $x, $y, $text, $textColor);

// Устанавливаем заголовок для вывода изображения

header('Content-Type: image/jpeg');

// Отправляем изображение пользователю

imagejpeg($image);

// Освобождаем память

imagedestroy($image);

}, 200, ['Content-Type' => 'image/jpeg']);

Если изображение недействительно, возвращаем ошибку:  
  
return response()->json(['error' => 'Invalid image upload'], 400);

#### **2. Использование регулярных выражений (regex) для проверки данных**

**Проверка, что строка начинается с заглавной буквы** (поддерживает кириллицу и латиницу):  
  
regex:/^[А-ЯA-Z]/u

**Проверка пароля**:

Должен содержать хотя бы одну строчную букву, одну заглавную букву и одну цифру, минимум 3 символа.  
  
regex:/^(?=.\*[a-z])(?=.\*[A-Z])(?=.\*\d).{3,}$/u

**Публикация конфигурации CORS**

Чтобы изменить настройки CORS (Cross-Origin Resource Sharing), можно опубликовать конфигурацию:

php artisan config:publish cors

Это позволяет настроить политику доступа для API и других ресурсов.

#### **Создание правила валидации**

В Laravel можно создать кастомные правила валидации для проверки данных.

**Создание правила**:

Команда для создания правила FirstLetterRule, которое проверяет первую букву строки:  
  
php artisan make:rule FirstLetterRule

**Пример реализации правила**:

После создания правила его можно использовать в валидаторах для проверки данных, например, чтобы строка начиналась с определенной буквы.

Эти материалы помогут углубить знания в обработке изображений, валидации данных и настройке политики доступа в Laravel.