

Домашняя работа
по предмету:

Продвинутое программирование на PHP — Laravel

Выполнил: Байборodin Александр

Урок 12. Интеграция с внешними сервисами

Цели практической работы:

Научиться:

- интегрировать отправку писем через почтовый клиент;
- настраивать отправку сообщений в мессенджер.

Что нужно сделать:

В этой практической работе вы реализуете уведомления через внешние сервисы.

1. Создайте новый проект Laravel или откройте уже существующий.
2. Создайте новую ветку вашего репозитория от корневой (main или master).
3. Настройте регистрацию и аутентификацию пользователей.
4. Настройте почтовый клиент любого сервиса.
5. Впишите в файл .env нужные значения для почтового сервиса.
6. Создайте письмо **Welcome.php** командой `php artisan make:mail Welcome`.
7. В конструкторе класса присвойте свойству класса `$user` параметр конструктора класса.

```
public User $user;  
public function __construct(User $user)  
{  
    $this->user = $user;  
}
```

8. Создайте шаблон мейлинга **welcome.blade.php** в директории **resources/views/emails** с кодом внутри

```
Добрый день,  
{  
{ $user -> name }  
},  
спасибо за регистрацию.
```

9. Добавьте код отправки вашего письма в функцию **store** класса **RegisteredUserController**.

10. Подключите клиент мессенджера Telegram командой `composer require irazasyed/telegram-bot-sdk`

11. Создайте бота и канал, добавьте бота в телеграм-канал.

12. Укажите в файле `.env` значения, необходимые для работы бота.

13. Проверьте работу бота с помощью тестового маршрута.

```
Route::get('test-telegram', function () {  
    Telegram::sendMessage([  
        'chat_id' => env('TELEGRAM_CHANNEL_ID', ''),  
        'parse_mode' => 'html',  
        'text' => 'Произошло тестовое событие'  
    ]);  
    return response()->json([  
        'status' => 'success'  
    ]);  
});
```

14. Добавьте код уведомления в мессенджер о новом пользователе вашей системы в функцию **store** класса **RegisteredUserController**.

15. Зарегистрируйтесь на сайте.

16. Проверьте, что сообщение отправлено на электронную почту (рекомендуется использовать для регистрации тот почтовый ящик, с которого отправляется сообщение, чтобы избежать блокировки адреса за спам).

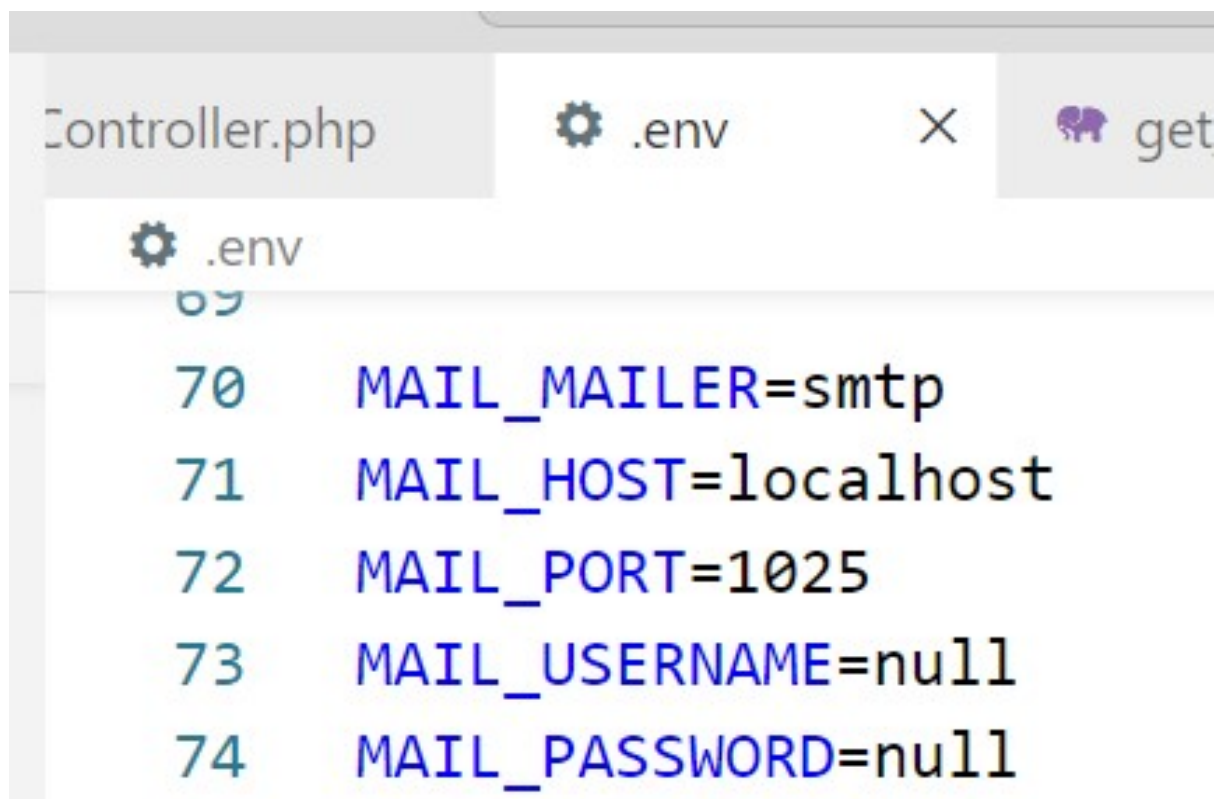
17. Проверьте, что в Telegram пришло уведомление о регистрации нового пользователя.

Решение:

Для выполнения задания возьмем проект урока 11, в котором настроены регистрация и аутентификация пользователей.

Теперь настроим почтовый сервис (на примере Mailtrap) для чего добавим в файл .env строки:

```
MAIL_MAILER=smtp
MAIL_HOST=localhost
MAIL_PORT=1025
MAIL_USERNAME=null
MAIL_PASSWORD=null
MAIL_ENCRYPTION=null
MAIL_FROM_ADDRESS="test@example.com"
MAIL_FROM_NAME="${APP_NAME}"
```



Создадим файл Welcome.php командой make:

```
A@LAPTOP-RHI13GV6 MINGW64 /c:/laravelapp/lesson12/proje
$ php artisan make:mail welcome
```

```
INFO Mailable [C:\laravelapp\lesson12\project-name
ated successfully
Отредактируем файл:
```

.env

Welcome.php ×

app > Mail > Welcome.php

```
2
3 namespace App\Mail;
4
5 use Illuminate\Bus\Queueable;
6 use Illuminate\Contracts\Queue\ShouldQueue;
7 use Illuminate\Mail\Mailable;
8 use Illuminate\Mail\Mailables\Content;
9 use Illuminate\Mail\Mailables\Envelope;
10 use Illuminate\Queue\SerializesModels;
11
12 class Welcome extends Mailable
13 {
14     use Queueable, SerializesModels;
```

Создадим шаблон resources/views/emails/welcome.blade.php:

welcome.blade.php ×

resources > views > emails > welcome.blade.php

```
1 @component('mail::message')
2     # Добрый день, {{ $user->name }}
3
4     Спасибо за регистрацию в нашем
5
6     @component('mail::button', ['url' => route('home')])
7     Перейти на сайт
8 @endcomponent
```

Редактируем app/Http/Controllers/Auth/RegisteredUserController.php:

```
<?php
```

```
namespace App\Http\Controllers\Auth;

use App\Http\Controllers\Controller;
use App\Models\User;
use App\Providers\RouteServiceProvider;
use Illuminate\Auth\Events\Registered;
use Illuminate\Http\RedirectResponse;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;
use Illuminate\Support\Facades\Hash;
use Illuminate\Validation\Rules;
use Illuminate\View\View;
use App\Mail>Welcome;
use Illuminate\Support\Facades\Mail;

class RegisteredUserController extends Controller
{
    /**
     * Display the registration view.
     */
    public function create(): View
    {
        return view('auth.register');
    }

    /**
     * Handle an incoming registration request.
     *
     * @throws \Illuminate\Validation\ValidationException
     */
    public function store(Request $request): RedirectResponse
    {
        $request->validate([
            'name' => ['required', 'string', 'max:255'],
            'email' => 'required|string|email|max:255|unique:users',
            'password' => 'required|string|confirmed|min:8',
        ]);

        $user = User::create([
            'name' => $request->name,
            'email' => $request->email,
            'password' => Hash::make($request->password),
        ]);

        event(new Registered($user));
```

```

        Auth::login($user);

        Mail::to($user->email)->send(new Welcome($user));

        return redirect(RouteServiceProvider::HOME);
    }
}

```

Устанавливаем пакет telegram-bot-sdk:

```

$ composer require irazasyed/telegram-bot-sdk
./composer.json has been updated
Running composer update irazasyed/telegram-bot-sdk
Loading composer repositories with package information
Updating dependencies
Lock file operations: 2 installs, 0 updates, 0 removals
  - Locking irazasyed/telegram-bot-sdk (v3.15.0)
  - Locking league/event (3.0.3)
Writing lock file
Installing dependencies from lock file (including require-dev)
Package operations: 2 installs, 0 updates, 0 removals
  - Downloading league/event (3.0.3)
  - Downloading irazasyed/telegram-bot-sdk (v3.15.0)
  - Installing league/event (3.0.3): Extracting archive
  - Installing irazasyed/telegram-bot-sdk (v3.15.0): Extracting archive
1 package suggestions were added by new dependencies, use `composer
suggest` to see details.
Generating optimized autoload files
> Illuminate\Foundation\ComposerScripts::postAutoLoadDump
> @php artisan package:discover --ansi

```

INFO Discovering packages.

```

irazasyed/telegram-bot-sdk
.....
laravel/breeze
..... DONE
laravel/sail
..... DONE
laravel/sanctum
..... DONE
laravel/tinker
..... DONE
nesbot/carbon
..... DONE
nunomaduro/collision
..... DONE
nunomaduro/termwind
..... DONE
spatie/laravel-ignition
..... DONE
..... DONE

```

81 packages you are using are looking for funding.

Use the `composer fund` command to find out more!

```
> @php artisan vendor:publish --tag=laravel-assets --ansi --force
```

INFO No publishable resources for tag [laravel-assets].

No security vulnerability advisories found.
Using version ^3.15 for irazasyed/telegram-bot-sdk

Добавим в файл .env данные о токене и чате Telegram:

```
TELEGRAM_BOT_TOKEN= bot_token  
TELEGRAM_CHANNEL_ID= channel_id
```

Публикуем конфиг:

```
$ php artisan vendor:publish --  
provider="Telegram\Bot\Laravel\TelegramServiceProvider"
```

INFO Publishing assets.

Copying file [C:\laravelapp\lesson12\project-name\vendor\irazasyed\telegram-bot-sdk\src\Laravel\config\telegram.php] to [C:\laravelapp\lesson12\project-name\config\telegram.php] **DONE**

Добавляем в routes/web.php тестовый маршрут Telegram:

```
<?php
```

```
use App\Http\Controllers\ProfileController;  
use Illuminate\Support\Facades\Route;  
use App\Http\Controllers\UsersController;  
use Telegram\Bot\Laravel\Facades\Telegram;
```

```
Route::get('/', function () {  
    return view('welcome');  
});
```

```
Route::get('/dashboard', function () {  
    return view('dashboard');  
})->middleware(['auth', 'verified'])->name('dashboard');
```

```
Route::middleware('auth')->group(function () {  
    Route::get('/profile', [ProfileController::class, 'edit'])->  
        >name('profile.edit');  
    Route::patch('/profile', [ProfileController::class, 'update'])->  
        >name('profile.update');  
    Route::delete('/profile', [ProfileController::class,  
        'destroy'])->name('profile.destroy');  
});
```

```
Route::get('/users', [UserController::class, 'index'])->  
    >middleware('auth');
```

```
Route::get('test-telegram', function () {
```

```

Telegram::sendMessage([
    'chat_id' => env('TELEGRAM_CHANNEL_ID'),
    'parse_mode' => 'html',
    'text' => 'Произошло тестовое событие'
]);

return response()->json([
    'status' => 'success'
]);
});

Route::get('/test-email', function() {
    try {
        Mail::raw('Test email content', function($message) {
            $message->to('test@example.com')->subject('Test Email');
        });
        return 'Email sent successfully! Check Mailpit at
http://localhost:8025';
    } catch (\Exception $e) {
        return 'Error: ' . $e->getMessage();
    }
});

require __DIR__.'/auth.php';

```

Обновляем файл RegisteredUserController.php:

```
<?php
```

```

namespace App\Http\Controllers\Auth;

use App\Http\Controllers\Controller;
use App\Models\User;
use App\Providers\RouteServiceProvider;
use Illuminate\Auth\Events\Registered;
use Illuminate\Http\RedirectResponse;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;
use Illuminate\Support\Facades\Hash;
use Illuminate\Validation\Rules;
use Illuminate\View\View;
use App\Mail>Welcome;
use Illuminate\Support\Facades\Mail;
use Telegram\Bot\Laravel\Facades\Telegram;

class RegisteredUserController extends Controller
{
    /**
     * Display the registration view.

```



```

    */
    public function create(): View
    {
        return view('auth.register');
    }

    /**
     * Handle an incoming registration request.
     *
     * @throws \Illuminate\Validation\ValidationException
     */
    public function store(Request $request): RedirectResponse
    {
        $request->validate([
            'name' => ['required', 'string', 'max:255'],
            'email' => 'required|string|email|max:255|unique:users',
            'password' => 'required|string|confirmed|min:8',
        ]);

        $user = User::create([
            'name' => $request->name,
            'email' => $request->email,
            'password' => Hash::make($request->password),
        ]);

        event(new Registered($user));

        Auth::login($user);

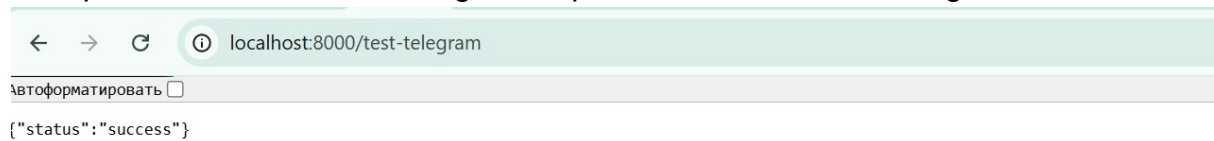
        Mail::to($user->email)->send(new Welcome($user));

        // Отправка уведомления в Telegram
        try {
            Telegram::sendMessage([
                'chat_id' => env('TELEGRAM_CHANNEL_ID'),
                'parse_mode' => 'html',
                'text' => "<b>Новый пользователь!</b>\n"
                    . "Имя: {$user->name}\n"
                    . "Email: {$user->email}\n"
                    . "Дата регистрации: {$user->created_at}"
            ]);
        } catch (\Exception $e) {
            \Log::error('Telegram notification failed: ' . $e-
                >getMessage());
        }

        return redirect(RouteServiceProvider::HOME);
    }
}

```

Проведем тест получения сообщения в Telegram перейдя на маршрут и видим, что пришло сообщение в Telegram <http://localhost:8000/test-telegram>:



Для тестирования отправки почты запустил приложение Mailpit.exe. Открыл <http://localhost:8000/test-email>. Получил успешное сообщение:

Email sent successfully! Check Mailpit at <http://localhost:8025>

Перешел на <http://localhost:8025> - увидел письмо:

From Laravel <test@example.com>

To <test@example.com>

Subject Test Email

Date Sun, 30 Mar 2025, 7:56 pm (507 B)

Tags

```
C:\Windows\System32\cmd.exe - mailpit.exe

Microsoft Windows [Version 10.0.19045.5608]
(c) Корпорация Майкрософт (Microsoft Corporation). Все права за

C:\Users\A\Downloads\mailpit-windows-amd64>dir
Том в устройстве C имеет метку Windows-SSD
Серийный номер тома: AA56-C587

Содержимое папки C:\Users\A\Downloads\mailpit-windows-amd64

30.03.2025  19:48    <DIR>          .
30.03.2025  19:48    <DIR>          ..
29.03.2025  12:37             1 085 LICENSE
29.03.2025  12:37          23 092 224 mailpit.exe
29.03.2025  12:37             7 064 README.md
                3 файлов          23 100 373 байт
                2 папок          3 202 744 320 байт свободно
```

←

→

↺

ⓘ

localhost:8025/view/bsYPAgYqwqU7GRQhi8hyTp

Mailpit

🔍

Mark unread

↶ Return to inbox

Laravel

To: test@example.com

Test Email

6 minutes ago

From

Larave

To

<test@

Subject

Test E

Date

Sun, 30

Tags

Add tag

Теперь зарегистрируем нового пользователя и проверим телеграмм, видим, что пришло сообщение:

Бот для напоминаний

бот

Пора ужинать 22:16

Пора завтракать 22:17

Пора завтракать 22:17

Пора ужинать 22:17

Пора завтракать 22:18

Пора завтракать 22:18

Пора ужинать 22:18

Пора завтракать 11:33

Пора завтракать 11:33

Пора ужинать 11:33

Также видим, что письмо о регистрации пришло на почту:

- ↶ Return to inbox

Laravel3 minutes ago

To: user5@gmail.com

Добро пожаловать в Laravel

Laravel15 minutes ago

To: test@example.com

Test Email

From

Laravel <test@example.com>

To

<user5@gmail.com>

Subject

Добро пожаловать в Laravel

Date

Sun, 30 Mar 2025, 8:08 pm (11.4 kB)

Tags

Add tags...

HTML

HTML Source

Text

Headers

Raw