

Домашняя работа  
по предмету:

Продвинутое программирование на PHP — Laravel

Выполнил: Байборodin Александр

Урок 2. Контроллеры, экшены и роутинг

Задание

1. Установите Laravel с помощью composer, выполнив команду `composer create-project laravel/laravel <имя проекта>`. В поле <имя проекта> впишите имя вашего проекта. Этому имени будет соответствовать имя папки, в которую вы поместите проект.
2. Создайте контроллер для вывода формы на страницу и её обработки. В командную строку введите команду `php artisan make:controller FormProcessor`.
3. После выполнения команды убедитесь, что контроллер создан, — соответствующий файл должен появиться в папке `app/Http/Controllers`.
4. Внутри контроллера опишите метод `index`: он должен выводить в браузер форму для заполнения.
  - Опишите форму в виде шаблона blade.
  - Внутри формы должны быть поля для ввода имени, фамилии и email пользователя.
  - Форма отправляется методом POST.
  - Параметр `action` пока оставьте пустым.
  - Не забудьте про CSRF.
5. Внутри файла `/routes/web.php` опишите новый роут (метод GET), который будет вызывать метод `index` контроллера `FormProcessor` по url `/userform`.
6. Запустите встроенный сервер Laravel командой `php artisan serve --port=8080` и убедитесь, что форма выводится по адресу `http://localhost:8080/userform`.
7. В контроллере `FormProcessor` создайте метод `store` для обработки формы. Этот метод должен принимать поля формы и отправлять ответ в виде JSON-объекта, содержащего значения полей формы (имя, фамилия, email).
8. Внутри файла `/routes/web.php` опишите новый роут (метод POST), который будет вызывать метод `store` контроллера `FormProcessor` по url `/store_form`.
9. Отредактируйте поле `action` формы в шаблоне и укажите адрес `/store_form`.
10. Откройте форму в браузере по адресу `http://localhost:8080/userform`, заполните её и попробуйте отправить на сервер, нажав кнопку Submit. Если всё сделано правильно, вы увидите в браузере объект JSON.
11. Создайте новый шаблон blade для приветствия пользователя (например: «Привет, <имя>!»).

12. Измените метод store контроллера FormProcessor таким образом, чтобы вместо JSON он возвращал шаблон, заполненный данными пользователя.

13. Сделайте коммит своих изменений с помощью git и отправьте push в репозиторий.

## Результат

Создадим laravel-проект веб-приложения my-laravel-app-lesson-two:

```
PS C:\laravelapp> composer create-project laravel/laravel my-laravel-app-lesson-two
Creating a "laravel/laravel" project at "./my-laravel-app-lesson-two"
Cannot use laravel/laravel's latest version v11.6.1 as it requires php ^8.2 which is not satisfied by your platform.
Installing laravel/laravel (v10.3.3)
- Installing laravel/laravel (v10.3.3): Extracting archive
Created project in C:\laravelapp\my-laravel-app-lesson-two
> @php -r "file_exists('.env') || copy('.env.example', '.env');"
Loading composer repositories with package information
Updating dependencies
Lock file operations: 111 installs, 0 updates, 0 removals
- Locking brick/math (0.12.1)
- Locking carbonphp/carbon-doctrine-types (2.1.0)
- Locking dflydev/dot-access-data (v3.0.3)
- Locking doctrine/inflector (2.0.10)
- Locking doctrine/lexer (3.0.1)
- Locking dragonmantank/cron-expression (v3.4.0)
- Locking egulias/email-validator (4.0.3)
- Locking fakerphp/faker (v1.24.1)
- Locking filp/whoops (2.17.0)
- Locking fruitcake/php-cors (v1.3.0)
- Locking graham-campbell/result-type (v1.1.3)
- Locking guzzlehttp/guzzle (7.9.2)
- Locking guzzlehttp/promises (2.0.4)
- Locking guzzlehttp/psr7 (2.7.0)
- Locking guzzlehttp/uri-template (v1.0.4)
- Locking hamcrest/hamcrest-php (v2.0.1)
- Locking laravel/framework (v10.48.28)
- Locking laravel/pint (v1.20.0)
- Locking laravel/prompts (v0.1.25)
- Locking laravel/sail (v1.41.0)
- Locking laravel/sanctum (v3.3.3)
- Locking laravel/serializable-closure (v1.3.7)
- Locking laravel/tinker (v2.10.1)
- Locking league/commonmark (2.6.1)
- Locking league/config (v1.2.0)
- Locking league/flysystem (3.29.1)
- Locking league/flysystem-local (3.29.0)
- Locking league/mime-type-detection (1.16.0)
- Locking mockery/mockery (1.6.12)
```

```
Installing webmozart/assert (1.11.0)
Writing lock file
Installing dependencies from lock file (including require-dev)
Package operations: 111 installs, 0 updates, 0 removals
- Installing doctrine/inflector (2.0.10): Extracting archive
- Installing doctrine/lexer (3.0.1): Extracting archive
- Installing symfony/polyfill-ctype (v1.31.0): Extracting archive
- Installing webmozart/assert (1.11.0): Extracting archive
- Installing dragonmantank/cron-expression (v3.4.0): Extracting archive
- Installing symfony/deprecation-contracts (v3.5.1): Extracting archive
- Installing psr/container (2.0.2): Extracting archive
- Installing fakerphp/faker (v1.24.1): Extracting archive
- Installing symfony/polyfill-php83 (v1.31.0): Extracting archive
- Installing symfony/polyfill-mbstring (v1.31.0): Extracting archive
- Installing symfony/http-foundation (v6.4.18): Extracting archive
- Installing fruitcake/php-cors (v1.3.0): Extracting archive
- Installing psr/http-message (2.0): Extracting archive
- Installing psr/http-client (1.0.3): Extracting archive
- Installing ralouphie/getallheaders (3.0.3): Extracting archive
- Installing psr/http-factory (1.1.0): Extracting archive
- Installing guzzlehttp/psr7 (2.7.0): Extracting archive
- Installing guzzlehttp/promises (2.0.4): Extracting archive
- Installing guzzlehttp/guzzle (7.9.2): Extracting archive
- Installing symfony/polyfill-php80 (v1.31.0): Extracting archive
- Installing guzzlehttp/uri-template (v1.0.4): Extracting archive
- Installing laravel/pint (v1.20.0): Extracting archive
- Installing symfony/polyfill-intl-normalizer (v1.31.0): Extracting archive
- Installing symfony/polyfill-intl-grapheme (v1.31.0): Extracting archive
- Installing symfony/string (v6.4.15): Extracting archive
- Installing symfony/service-contracts (v3.5.1): Extracting archive
- Installing symfony/console (v6.4.17): Extracting archive
- Installing voku/portable-ascii (2.0.3): Extracting archive
- Installing phpoption/phpooption (1.9.3): Extracting archive
- Installing graham-campbell/result-type (v1.1.3): Extracting archive
- Installing vlucas/phpdotenv (v5.6.1): Extracting archive
- Installing symfony/css-selector (v6.4.13): Extracting archive
```

```
Windows PowerShell
- Installing phpunit/php-text-template (3.0.1): Extracting archive
- Installing phpunit/php-invoker (4.0.0): Extracting archive
- Installing phpunit/php-file-iterator (4.1.0): Extracting archive
- Installing theseer/tokenizer (1.2.3): Extracting archive
- Installing sebastian/lines-of-code (2.0.2): Extracting archive
- Installing sebastian/complexity (3.2.0): Extracting archive
- Installing sebastian/code-unit-reverse-lookup (3.0.0): Extracting archive
- Installing phpunit/php-code-coverage (10.1.16): Extracting archive
- Installing phar-io/version (3.2.1): Extracting archive
- Installing phar-io/manifest (2.0.4): Extracting archive
- Installing myclabs/deep-copy (1.12.1): Extracting archive
- Installing phpunit/phpunit (10.5.45): Extracting archive
- Installing spatie/error-solutions (1.1.2): Extracting archive
- Installing spatie/backtrace (1.7.1): Extracting archive
- Installing spatie/flare-client-php (1.10.0): Extracting archive
- Installing spatie/ignition (1.15.0): Extracting archive
- Installing spatie/laravel-ignition (2.9.0): Extracting archive
No 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.

laravel/sail ..... DONE
laravel/sanctum ..... DONE
laravel/tinker ..... DONE
nesbot/carbon ..... DONE
nunomaduro/collision ..... DONE
nunomaduro/termwind ..... DONE
spatie/laravel-ignition ..... DONE

0 packages you are using are looking for funding.
see 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.
> @php artisan key:generate --ansi

INFO Application key set successfully.

PS C:\laravelapp>
```

Создадим контроллер для вывода формы на страницу и её обработки. Перейдем в каталог приложения и в командную строку введем команду **php artisan make:controller FormProcessor**:

```
PS C:\laravelapp> cd my-laravel-app-lesson-two
PS C:\laravelapp\my-laravel-app-lesson-two> php artisan make:controller FormProcessor

INFO Controller [C:\laravelapp\my-laravel-app-lesson-two\app\Http\Controllers\FormProcessor.php] created successfully.

PS C:\laravelapp\my-laravel-app-lesson-two> cd C:\laravelapp\my-laravel-app-lesson-two\app\Http\Controllers\
PS C:\laravelapp\my-laravel-app-lesson-two\app\Http\Controllers> dir

Каталог: C:\laravelapp\my-laravel-app-lesson-two\app\Http\Controllers

Mode                LastWriteTime         Length Name
----                -
-----          13.02.2024   5:23           299 Controller.php
-a-----          10.02.2025  22:37          120 FormProcessor.php

PS C:\laravelapp\my-laravel-app-lesson-two\app\Http\Controllers>
```

Изменим код контроллера на код:

```
<?php
```

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

```
use Illuminate\Http\Request;
```

```
class FormProcessor extends Controller
```

```
{
```

```
    public function index()
```

```
    {
```

```
        return view('userform'); // Будем использовать шаблон userform.blade.php
```

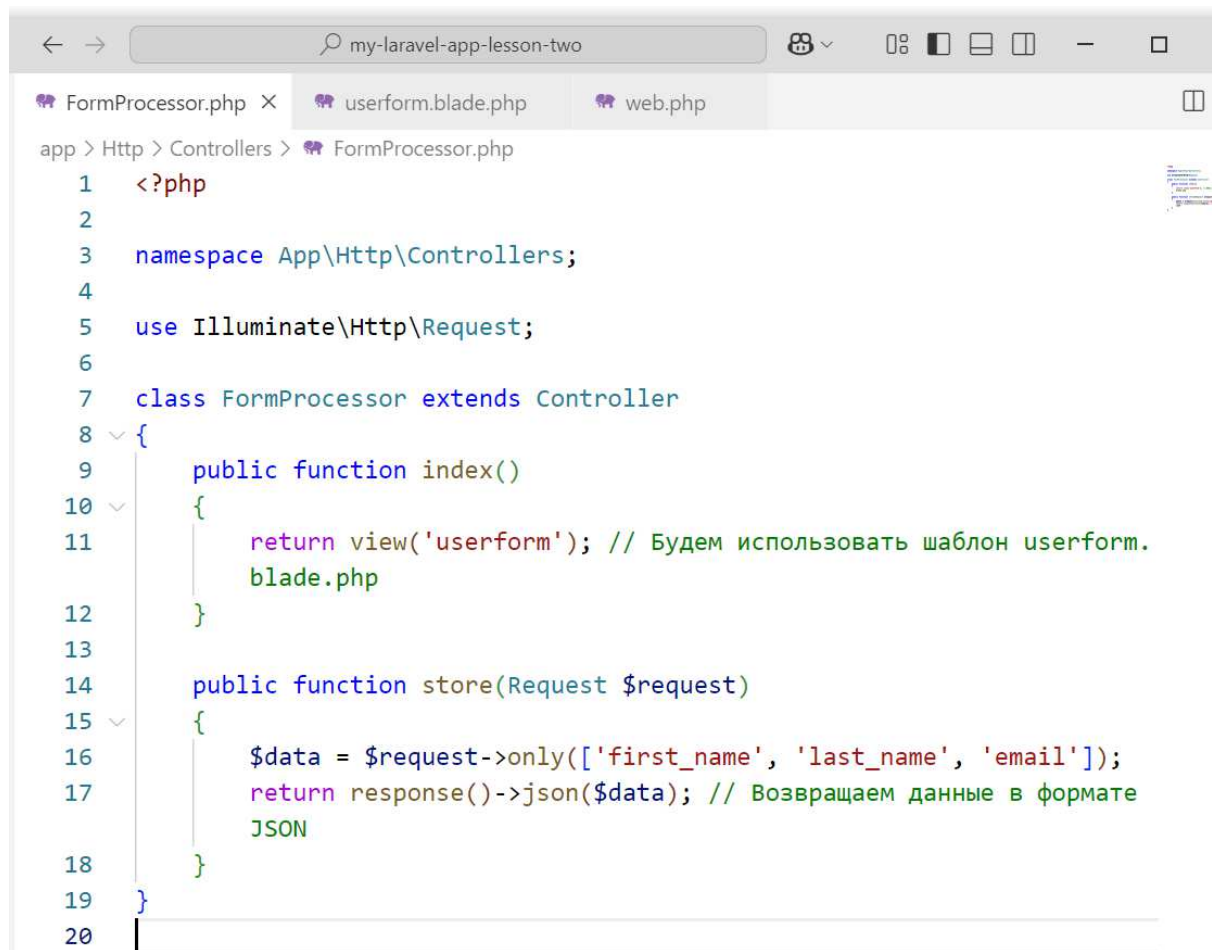
```
    }
```

```
    public function store(Request $request)
```

```

    {
        $data = $request->only(['first_name', 'last_name', 'email']);
        return response()->json($data); // Возвращаем данные в формате JSON
    }
}

```



The screenshot shows a web browser window at the top with the address bar displaying 'my-laravel-app-lesson-two'. Below the browser, a code editor displays the contents of 'FormProcessor.php'. The code defines a 'FormProcessor' class that extends 'Controller'. It has two methods: 'index()' which returns the 'userform' blade template, and 'store(Request \$request)' which extracts 'first\_name', 'last\_name', and 'email' from the request and returns them as a JSON response.

```

1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6
7  class FormProcessor extends Controller
8  {
9      public function index()
10     {
11         return view('userform'); // Будем использовать шаблон userform.
12         blade.php
13     }
14
15     public function store(Request $request)
16     {
17         $data = $request->only(['first_name', 'last_name', 'email']);
18         return response()->json($data); // Возвращаем данные в формате
19         JSON
20     }
21 }

```

Создадим представление:

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>User Form</title>
</head>
<body>
    <h1>User Form</h1>
    <form method="POST" action="/store_form">
        @csrf <!-- CSRF-токен для защиты -->
        <label for="first_name">Имя:</label>
        <input type="text" name="first_name" id="first_name" required>
        <br>
        <label for="last_name">Фамилия:</label>
        <input type="text" name="last_name" id="last_name" required>
        <br>
        <label for="email">Email:</label>
        <input type="email" name="email" id="email" required>
        <br>
        <button type="submit">Ок</button>
    </form>

```

```
</form>
</body>
</html>
```

```
FormProcessor.php  userform.blade.php X  web.php
resources > views > userform.blade.php
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1
6      <title>User Form</title>
7  </head>
8  <body>
9      <h1>User Form</h1>
10     <form method="POST" action="/store_form">
11         @csrf <!-- CSRF-токен для защиты -->
12         <label for="first_name">Имя:</label>
13         <input type="text" name="first_name" id="first_name" required>
14         <br>
15         <label for="last_name">Фамилия:</label>
16         <input type="text" name="last_name" id="last_name" required>
17         <br>
18         <label for="email">Email:</label>
19         <input type="email" name="email" id="email" required>
20         <br>
21         <button type="submit">Ок</button>
22     </form>
23 </body>
24 </html>
```

Создадим роуты:

```
<?php
```

```
use App\Http\Controllers\FormProcessor;
```

```
Route::get('/userform', [FormProcessor::class, 'index']);
```

```
Route::post('/store_form', [FormProcessor::class, 'store']);
```

```
FormProcessor.php  userform.blade.php  web.php  X
routes > web.php
1  <?php
2
3  use App\Http\Controllers\FormProcessor;
4
5  Route::get('/userform', [FormProcessor::class, 'index']);
6  Route::post('/store_form', [FormProcessor::class, 'store']);
```

Запустим приложение:

```
PS C:\laravelapp\my-laravel-app-lesson-two> dir

Каталог: C:\laravelapp\my-laravel-app-lesson-two

Mode                LastWriteTime         Length Name
----                -
d-----         13.02.2024          5:23         app
d-----         13.02.2024          5:23        bootstrap
d-----         13.02.2024          5:23        config
d-----         13.02.2024          5:23        database
d-----         13.02.2024          5:23        public
d-----         13.02.2024          5:23        resources
d-----         13.02.2024          5:23        routes
d-----         13.02.2024          5:23        storage
d-----         13.02.2024          5:23        tests
d-----         10.02.2025         22:13        vendor
-----         13.02.2024          5:23          258 .editorconfig
-a-----         10.02.2025         22:26         1148 .env
-----         13.02.2024          5:23         1097 .env.example
-----         13.02.2024          5:23          186 .gitattributes
-----         13.02.2024          5:23          243 .gitignore
-----         13.02.2024          5:23         1686 artisan
-----         13.02.2024          5:23         1882 composer.json
-a-----         10.02.2025         22:07        299580 composer.lock
-----         13.02.2024          5:23          248 package.json
-----         13.02.2024          5:23         1134 phpunit.xml
-----         13.02.2024          5:23         4109 README.md
-----         13.02.2024          5:23          263 vite.config.js

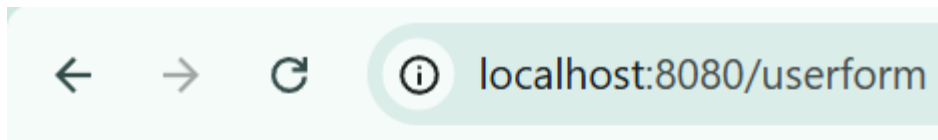
PS C:\laravelapp\my-laravel-app-lesson-two> php artisan serve --port=8080

INFO  Server running on [http://127.0.0.1:8080].

Press Ctrl+C to stop the server

2025-02-10 22:51:04 .....
2025-02-10 22:51:04 .....
2025-02-10 22:51:10 .....
2025-02-10 22:51:11 /favicon.ico .....
2025-02-10 22:51:11 .....
```

Откроем браузер по адресу:



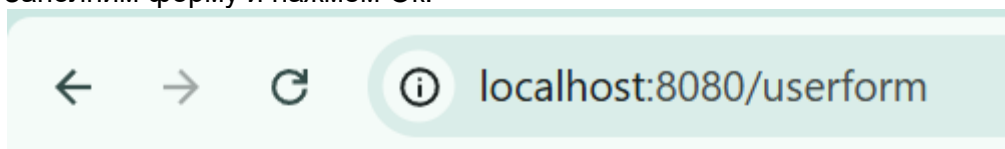
# User Form

Имя:

Фамилия:

Email:

Заполним форму и нажмем Ок:



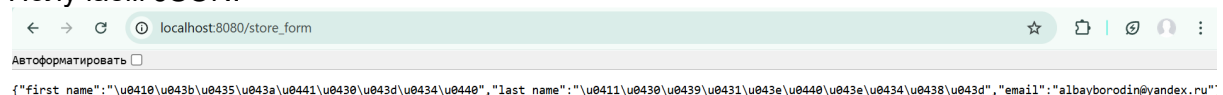
# User Form

Имя:

Фамилия:

Email:

Получаем JSON:



Создадим шаблон-представление для приветствия:



```
FormProcessor.php  userform.blade.php  web.php  welcome_user.blade.php ×
resources > views > welcome_user.blade.php
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Welcome</title>
7  </head>
8  <body>
9      <h1>Привет, {{ $first_name }}!</h1>
10     <p>Ваши данные:</p>
11     <ul>
12         <li><strong>Имя:</strong> {{ $first_name }}</li>
13         <li><strong>Фамилия:</strong> {{ $last_name }}</li>
14         <li><strong>Email:</strong> {{ $email }}</li>
15     </ul>
16 </body>
17 </html>
```

Изменим контроллер:

```
<?php
```

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

```
use Illuminate\Http\Request;
```

```
class FormProcessor extends Controller
```

```
{
```

```
    public function index()
```

```
    {
```

```
        return view('userform'); // Будем использовать шаблон userform.blade.php
```

```
    }
```

```
    // public function store(Request $request)
```

```
    // {
```

```
    //     $data = $request->only(['first_name', 'last_name', 'email']);
```

```
    //     return response()->json($data); // Возвращаем данные в формате JSON
```

```
    // }
```

```
    public function store(Request $request)
```

```
    {
```

```
    // Получаем данные из формы
```

```
    $data = $request->only(['first_name', 'last_name', 'email']);
```

```
    // Возвращаем шаблон welcome_user с данными
```

```
    return view('welcome_user', [
```

```
        'first_name' => $data['first_name'],
```

```
        'last_name' => $data['last_name'],
```

```
        'email' => $data['email'],
```

```
    ]);
```

```
    }
```

```
}
```

Перезагрузим страницу:



localhost:8080/store\_form

# Привет, Александр!

Ваши данные:

- **Имя:** Александр
- **Фамилия:** Байборodin
- **Email:** albayborodin@yandex.ru