



Politechnika
Śląska

Przedmiot: Podstawy informatyki

Nr. Ćwiczenia: 4

Imię i Nazwisko: Michał Witczak

Nr. Albumu 305423

Link do repozytorium:

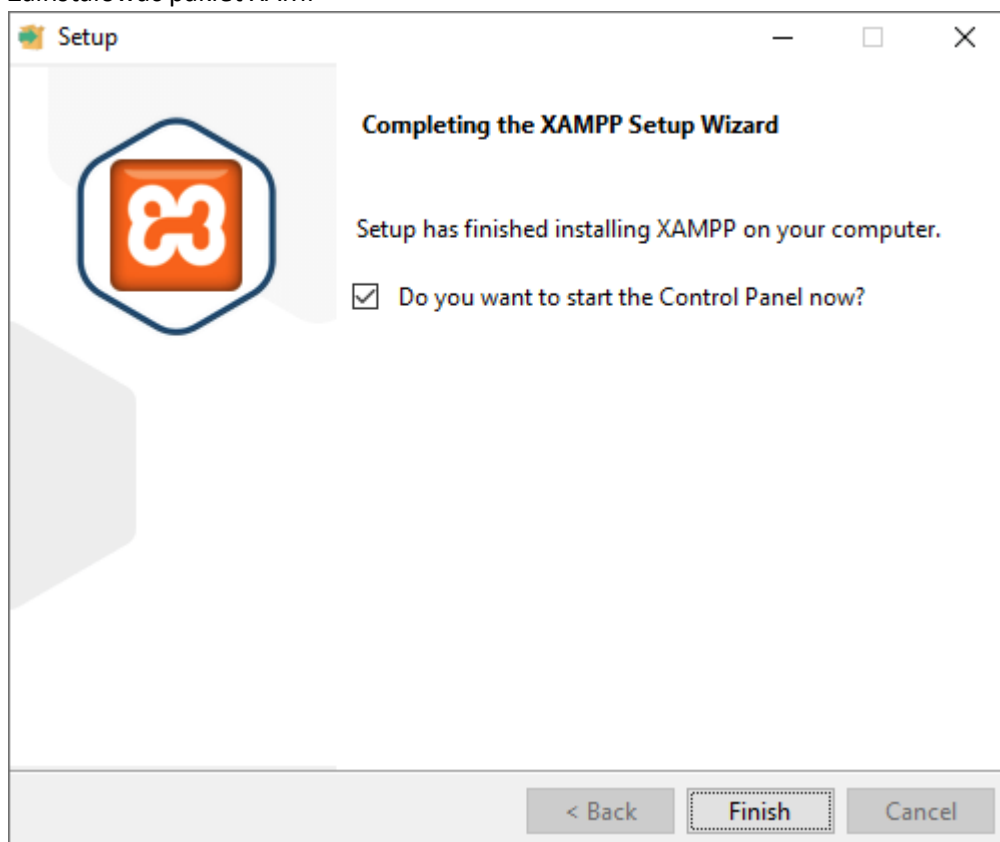
<https://github.com/MichalWitczakmw/restapipoprawaA.git>

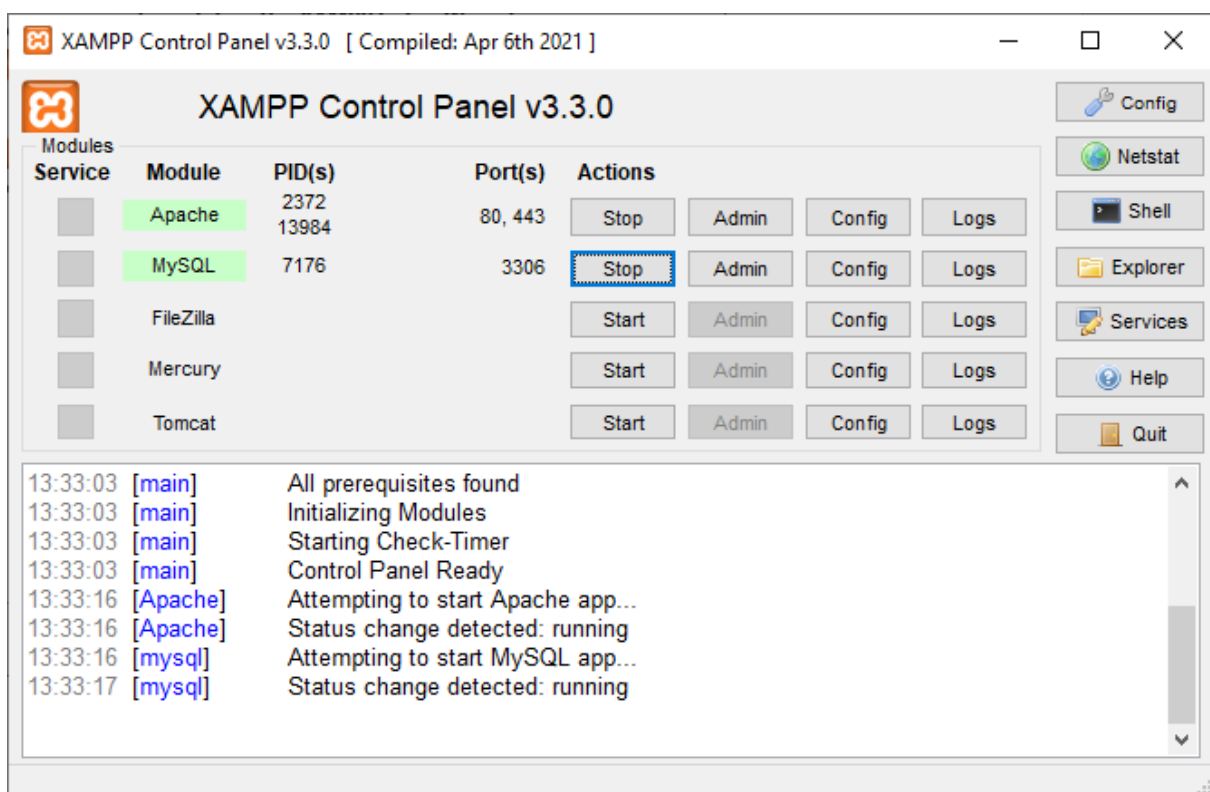
Poprawa.



Politechnika
Śląska

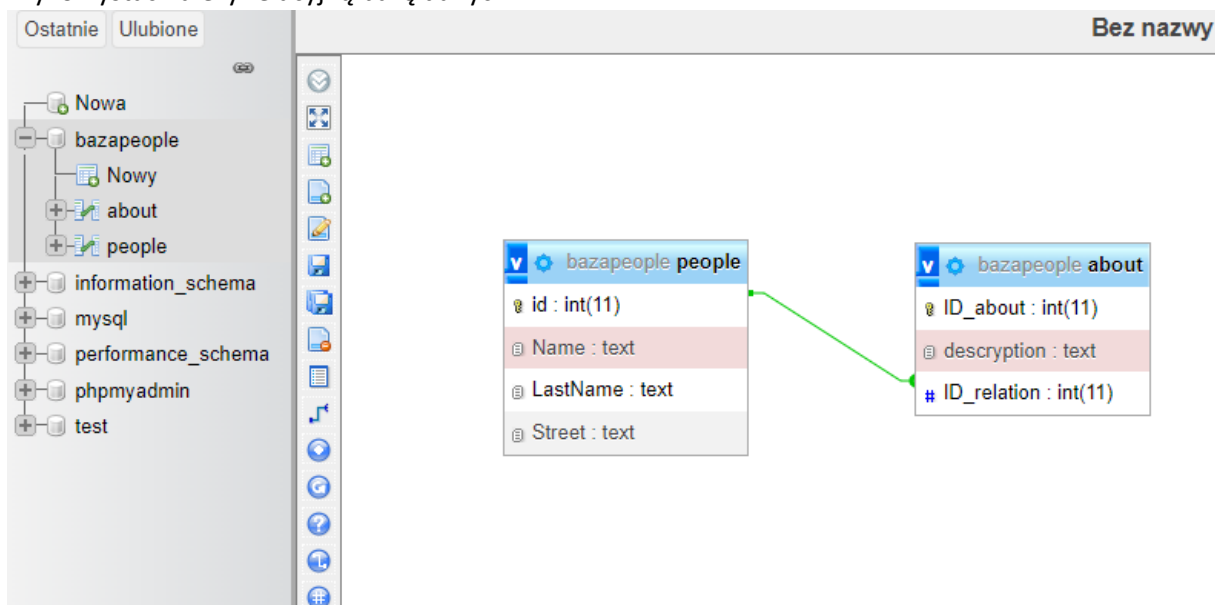
1. Zainstalować pakiet XAMP





Nie było problemów z instalacją ani z uruchomieniem.

- Wykorzystać należy relacyjną bazę danych.

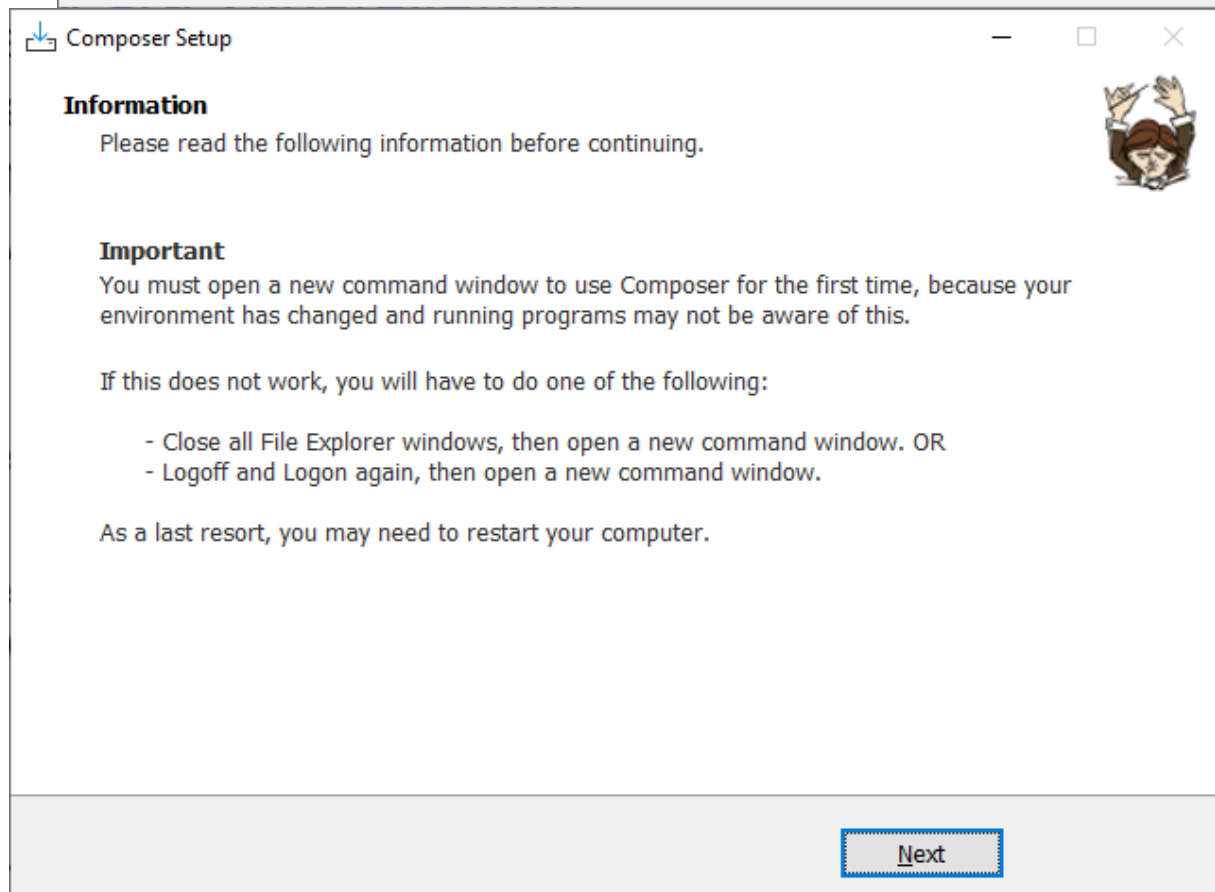
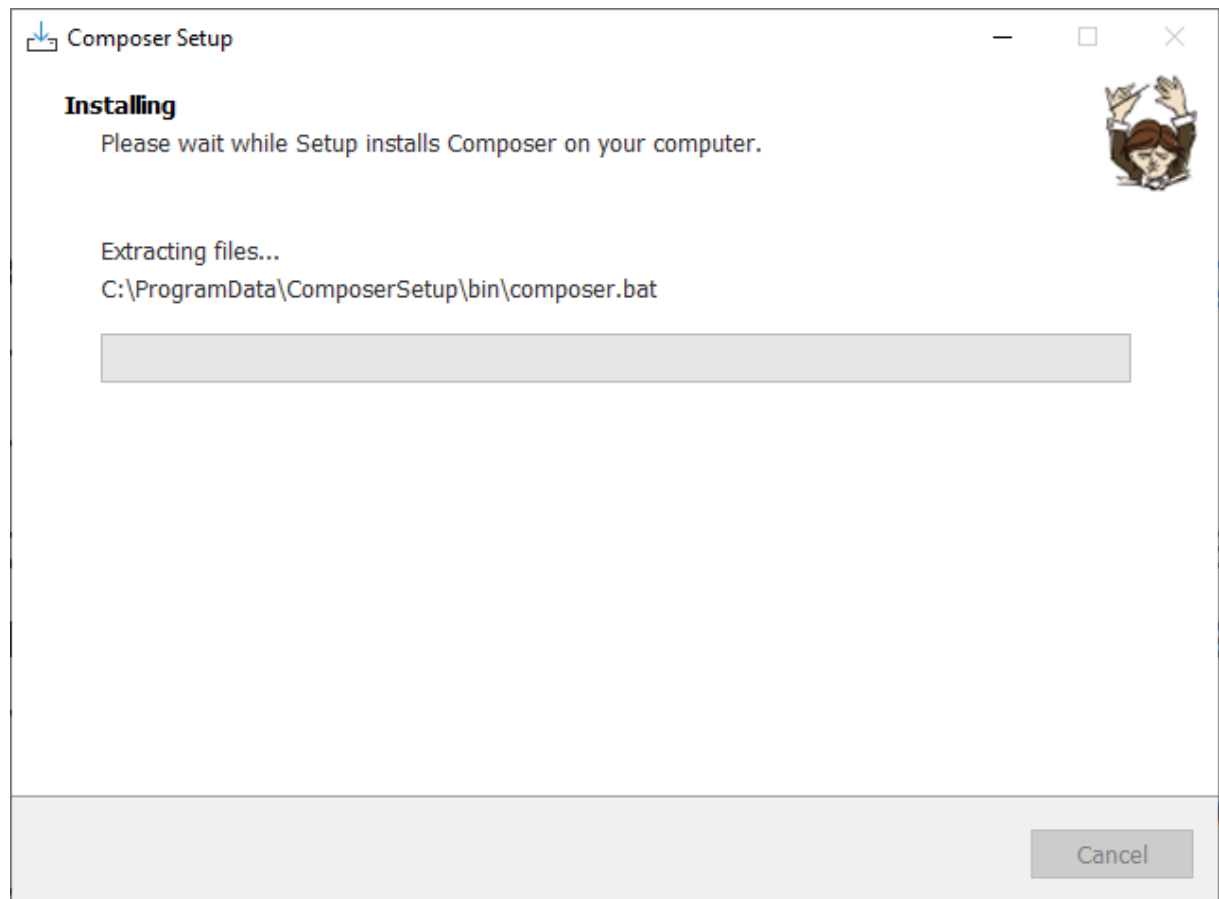




```
People.php × 2023_01_31_181751_create_people_table.php × .env ×
Plugins supporting .env files found.
1 APP_NAME=Laravel
2 APP_ENV=local
3 APP_KEY=base64:eFFXQNZkJjjvAc0mqkJB+mY3p0y+2V62xZV3/tCD3ew=
4 APP_DEBUG=true
5 APP_URL=http://localhost
6
7 LOG_CHANNEL=stack
8 LOG_DEPRECATIONS_CHANNEL=null
9 LOG_LEVEL=debug
10
11 DB_CONNECTION=mysql
12 DB_HOST=127.0.0.1
13 DB_PORT=3306
14 DB_DATABASE=bazapeople
15 DB_USERNAME=Michał
16 DB_PASSWORD=
17
18 BROADCAST_DRIVER=log
19 CACHE_DRIVER=file
20 FILESYSTEM_DISK=local
```

Udało się bez problemów utworzyć szkielet relacyjnej bazy danych plus dodać konto z hasłem i dostępem do niej.

3. Zainstalować program composer.





4. Instalacja laravel

Bez problemów

```
MINGW64:/c:/xampp/htdocs/rest/PEOPLE
Michał Witczak@DESKTOP-I1LFK3E MINGW64 /c:/xampp/htdocs/rest
$ composer create-project laravel/laravel PEOPLE
Creating a "laravel/laravel" project at "./PEOPLE"
Info from https://repo.packagist.org: #StandWithUkraine
Installing laravel/laravel (v9.5.2)
- Downloading laravel/laravel (v9.5.2)
- Installing laravel/laravel (v9.5.2): Extracting archive
Created project in C:\xampp\htdocs\rest\PEOPLE
> @php -r "file_exists('.env') || copy('.env.example', '.env');"
Loading composer repositories with package information
Updating dependencies
Lock file operations: 106 installs, 0 updates, 0 removals
- Locking brick/math (0.10.2)
- Locking dflydev/dot-access-data (v3.0.2)
- Locking doctrine/inflector (2.0.6)
```

5 utworzenie modelu wraz z migracją

```
Michał Witczak@DESKTOP-I1LFK3E MINGW64 /c:/xampp/htdocs/rest/PEOPLE
$ php artisan make:model People -m

[INFO] Model [C:\xampp\htdocs\rest\PEOPLE\app\Models\People.php] created successfully.

[INFO] Migration [C:\xampp\htdocs\rest\PEOPLE\database\migrations/2023_01_31_181751_create_people_table.php] created successfully.

Michał Witczak@DESKTOP-I1LFK3E MINGW64 /c:/xampp/htdocs/rest/PEOPLE
$ S
```

5. Konfiguracja Modelu i migracji.

```
People.php x 2023_01_31_181751_create_people_table.php x
1 <?php
2
3 namespace App\Models;
4
5 use ...
6
7 |
8 no usages
9
10 class People extends Model
11 {
12     use HasFactory;
13     no usages
14     protected $fillable=['Name','LastName','Street'];
15 }
16 }
```



```
People.php x 2023_01_31_181751_create_people_table.php x
1 <?php
2
3 use Illuminate\Database\Migrations\Migration;
4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Support\Facades\Schema;
6
7 return new class extends Migration
8 {
9     no usages
10    public function up()
11    {
12        Schema::create( table: 'people', function (Blueprint $table) {
13            $table->id();
14            $table->string( column: 'Name');
15            $table->string( column: 'LastName');
16            $table->string( column: 'Street');
17            $table->timestamps();
18        });
19    }
20    no usages
21    public function down()
22    {
23        Schema::dropIfExists( table: 'people');
24    }
25};
```

```
MINGW64:/c:/xampp/htdocs/rest/PEOPLE
Michał Witczak@DESKTOP-I1LFK3E MINGW64 /c:/xampp/htdocs/rest/PEOPLE
$ php artisan migrate:fresh

Dropping all tables ..... 77ms DONE

[INFO] Preparing database.

Creating migration table ..... 25ms DONE

[INFO] Running migrations.

2014_10_12_000000_create_users_table ..... 37ms DONE
2014_10_12_100000_create_password_resets_table ..... 99ms DONE
2019_08_19_000000_create_failed_jobs_table ..... 38ms DONE
2019_12_14_000001_create_personal_access_tokens_table ..... 47ms DONE
2023_01_31_181751_create_people_table ..... 24ms DONE

Michał Witczak@DESKTOP-I1LFK3E MINGW64 /c:/xampp/htdocs/rest/PEOPLE
$ php artisan migrate

[INFO] Nothing to migrate.

Michał Witczak@DESKTOP-I1LFK3E MINGW64 /c:/xampp/htdocs/rest/PEOPLE
$
```



6 Utworzenie Controller'a z konfiguracją:

```
Michał Witczak@DESKTOP-I1LFK3E MINGW64 /c/xampp/htdocs/rest/PEOPLE
$ php artisan make:controller PeopleController --model=People

INFO Controller [C:\xampp\htdocs\rest\PEOPLE\app\Http\Controllers\PeopleController.php] created successfully.

Michał Witczak@DESKTOP-I1LFK3E MINGW64 /c/xampp/htdocs/rest/PEOPLE
$
```

```
class PeopleController extends Controller
{
    no usages
    public function index()
    {
        $people = People::all();

        return response()->json([
            'status' => true,
            'people' => $people
        ]);
    }
    public function create()
    {
        //
    }
    no usages
    public function store(Request $request)
    {
        $people = People::create($request->all());

        return response()->json([
            'status' => true,
            'people' => $people
        ], status: 200);
    }
}
```




```
}  
no usages  
public function edit(People $people)  
{  
    //  
}  
no usages  
public function update(Request $request, People $people)  
{  
    $people->update($request->all());  
  
    return response()->json([  
        'status' => true,  
        'people' => $people  
    ], status: 200);  
}  
no usages  
public function destroy(People $people)  
{  
    $people->delete();  
  
    return response()->json([  
        'status' => true,  
    ], status: 200);  
}
```

Controllers

7 Tworzenie Request'a i konfiguracja

```
Michał Witczak@DESKTOP-I1LFK3E MINGW64 /c/xampp/htdocs/rest/PEOPLE  
$ php artisan make:request StorePeopleRequest  
  
INFO Request [C:\xampp\htdocs\rest\PEOPLE\app\Http\Requests/StorePeopleRequest.php] created successfully.  
  
Michał Witczak@DESKTOP-I1LFK3E MINGW64 /c/xampp/htdocs/rest/PEOPLE  
$
```



```
<?php

namespace App\Http\Requests;

use Illuminate\Foundation\Http\FormRequest;

1 usage
class StorePeopleRequest extends FormRequest
{
    no usages
    public function authorize()
    {
        return true;
    }
    no usages
    public function rules()
    {
        return [
            "Name" => "required|max:70",
            "LastName" => "required",
            "Street"=>"required"
        ];
    }
}
```

8 konfiguracja Route:

```
<?php
use App\Http\Controllers\PeopleController;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Route;

Route::middleware('auth:sanctum')->get('/user', function (Request $request) {
    return $request->user();
});
Route::apiResource('Witozak/305423/people', PeopleController::class);
```

9 Utworzenie instancji



```
MINGW64:/c:/xampp/htdocs/rest/PEOPLE
GET|HEAD api/user .....
GET|HEAD sanctum/csrf-cookie sanctum.csrf-cookie > Laravel\Sanctum ...

Showing [11] routes

Michał Witczak@DESKTOP-I1LFK3E MINGW64 /c:/xampp/htdocs/rest/PEOPLE
$ php artisan make:seeder PeopleSeeder

[INFO] Seeder [C:\xampp\htdocs\rest\PEOPLE\database\seeders\PeopleSeeder.php]
created successfully.

Michał Witczak@DESKTOP-I1LFK3E MINGW64 /c:/xampp/htdocs/rest/PEOPLE
$ php artisan make:factory PeopleFactory

[INFO] Factory [C:\xampp\htdocs\rest\PEOPLE\database\factories\PeopleFactory.p
hp] created successfully.

Michał Witczak@DESKTOP-I1LFK3E MINGW64 /c:/xampp/htdocs/rest/PEOPLE
$ php artisan migrate

[INFO] Nothing to migrate.
```

```
Michał Witczak@DESKTOP-I1LFK3E MINGW64 /c:/xampp/htdocs/rest/PEOPLE
$ php artisan migrate:fresh --seed

Dropping all tables ..... 72ms DONE

[INFO] Preparing database.

Creating migration table ..... 30ms DONE

[INFO] Running migrations.

2014_10_12_000000_create_users_table ..... 36ms DONE
2014_10_12_100000_create_password_resets_table ..... 66ms DONE
2019_08_19_000000_create_failed_jobs_table ..... 30ms DONE
2019_12_14_000001_create_personal_access_tokens_table ..... 50ms DONE
2023_01_31_181751_create_people_table ..... 29ms DONE

[INFO] Seeding database.

Database\Seeders\PeopleSeeder ..... RUNNING
Database\Seeders\PeopleSeeder ..... 173.40 ms DONE

Michał Witczak@DESKTOP-I1LFK3E MINGW64 /c:/xampp/htdocs/rest/PEOPLE
$
```



Politechnika
Śląska

phpMyAdmin

Server: 127.0.0.1 » Baza danych: bazapeople » tabela: people

Przeglądaj Struktura SQL Szukaj Wstaw Eksport Import Uprawnienia Operacje Śledzenie Więcej

Ostatnie Ulubione

1 > >> Pokaż wszystko Liczba wierszy: 25 Filtrowanie wierszy: Przeszukaj tę tabelę Sortuj wg klucza: Żaden

Extra options

				id	Name	LastName	Street	created_at	updated_at
<input type="checkbox"/>	Edytuj	Kopiuj	Usuń	1	Alyce Wolff	Murphy	Hansen Trace	2023-01-31 20:27:06	2023-01-31 20:27:06
<input type="checkbox"/>	Edytuj	Kopiuj	Usuń	2	Felicia Lemke	King	Melba Mills	2023-01-31 20:27:06	2023-01-31 20:27:06
<input type="checkbox"/>	Edytuj	Kopiuj	Usuń	3	Prof. Kendall Armstrong MD	Strosin	Bednar View	2023-01-31 20:27:06	2023-01-31 20:27:06
<input type="checkbox"/>	Edytuj	Kopiuj	Usuń	4	Sydney Fay Jr.	Satterfield	Heathcote Throughway	2023-01-31 20:27:06	2023-01-31 20:27:06
<input type="checkbox"/>	Edytuj	Kopiuj	Usuń	5	Micaela Muller	Shanahan	Feeney Forges	2023-01-31 20:27:06	2023-01-31 20:27:06
<input type="checkbox"/>	Edytuj	Kopiuj	Usuń	6	Gilda Sanford	Cummings	Lang Road	2023-01-31 20:27:06	2023-01-31 20:27:06
<input type="checkbox"/>	Edytuj	Kopiuj	Usuń	7	Prof. Darrion Heathcote PhD	Mosciski	Herzog Knoll	2023-01-31 20:27:06	2023-01-31 20:27:06
<input type="checkbox"/>	Edytuj	Kopiuj	Usuń	8	Shayna Feeney	Ullrich	Murphy Garden	2023-01-31 20:27:06	2023-01-31 20:27:06
<input type="checkbox"/>	Edytuj	Kopiuj	Usuń	9	Dr. Giles Dietrich	Sporer	Eino Drives	2023-01-31 20:27:06	2023-01-31 20:27:06
<input type="checkbox"/>	Edytuj	Kopiuj	Usuń	10	Jonatan Cremin	O'Kon	Horace Track	2023-01-31 20:27:06	2023-01-31 20:27:06
<input type="checkbox"/>	Edytuj	Kopiuj	Usuń	11	Alessandro Robel	Gutkowski	Sawayn Key	2023-01-31 20:27:06	2023-01-31 20:27:06
<input type="checkbox"/>	Edytuj	Kopiuj	Usuń	12	Courtney Sporer	Rath	Amina Rue	2023-01-31 20:27:06	2023-01-31 20:27:06

Konsola Zakładki Opcje Historia W

				id	Name	LastName	Street	created_at	updated_at
<input type="checkbox"/>	Edytuj	Kopiuj	Usuń	37	Winifred Bergstrom	Cormier	Carter Causeway	2023-01-31 20:27:06	2023-01-31 20:27:06
<input type="checkbox"/>	Edytuj	Kopiuj	Usuń	38	Sarina Raynor	Homenick	White Hills	2023-01-31 20:27:06	2023-01-31 20:27:06
<input type="checkbox"/>	Edytuj	Kopiuj	Usuń	39	Efrain Stoltenberg	Schaefer	Lebsack Oval	2023-01-31 20:27:06	2023-01-31 20:27:06
<input type="checkbox"/>	Edytuj	Kopiuj	Usuń	40	Fritz Haley	Aufderhar	Harry Cape	2023-01-31 20:27:06	2023-01-31 20:27:06
<input type="checkbox"/>	Edytuj	Kopiuj	Usuń	41	Lelah Parker	Konopelski	Rhett Prairie	2023-01-31 20:27:06	2023-01-31 20:27:06
<input type="checkbox"/>	Edytuj	Kopiuj	Usuń	42	Prof. Lucie Turner Sr.	Paucek	Nathanial Underpass	2023-01-31 20:27:06	2023-01-31 20:27:06
<input type="checkbox"/>	Edytuj	Kopiuj	Usuń	43	Gladysce Schmidt MD	Labadie	Mohammed Plain	2023-01-31 20:27:06	2023-01-31 20:27:06
<input type="checkbox"/>	Edytuj	Kopiuj	Usuń	44	Brandon Yundt III	Toy	King Freeway	2023-01-31 20:27:06	2023-01-31 20:27:06
<input type="checkbox"/>	Edytuj	Kopiuj	Usuń	45	Keshawn Shields DVM	Strosin	Reichel Junctions	2023-01-31 20:27:06	2023-01-31 20:27:06
<input type="checkbox"/>	Edytuj	Kopiuj	Usuń	46	Joanne Paucek IV	Parisian	Emerson Parks	2023-01-31 20:27:06	2023-01-31 20:27:06
<input type="checkbox"/>	Edytuj	Kopiuj	Usuń	47	Dr. Neva VonRueden	Schamberger	Ruth View	2023-01-31 20:27:06	2023-01-31 20:27:06
<input type="checkbox"/>	Edytuj	Kopiuj	Usuń	48	Mr. Melvina Baumbach	Doyle	Von Rapid	2023-01-31 20:27:06	2023-01-31 20:27:06
<input type="checkbox"/>	Edytuj	Kopiuj	Usuń	49	Vito Hickie	Grimes	Zetta Valley	2023-01-31 20:27:06	2023-01-31 20:27:06
<input type="checkbox"/>	Edytuj	Kopiuj	Usuń	50	Jedediah Powlowski	Kris	Braun Shoal	2023-01-31 20:27:06	2023-01-31 20:27:06



Politechnika
Śląska

Seeders:

```
PeopleSeeder / run
PeopleController.php x StorePeopleRequest.php x api.php x PeopleSeeder.php x PeopleFactory.php x DatabaseSeeder.php x
1 <?php
2
3 namespace Database\Seeders;
4
5 use App\Models\People;
6 use Illuminate\Database\Seeder;
7 use Illuminate\Support\Facades\DB;
8 use Illuminate\Support\Str;
9
10 class PeopleSeeder extends Seeder
11 {
12     public function run()
13     {
14         People::factory()
15             ->count( 50)
16             ->create();
17     }
18 }
19
```

```
PeopleController.php x StorePeopleRequest.php x api.php x PeopleSeeder.php x PeopleFactory.php x DatabaseSeeder.php x
namespace Database\Seeders;

// use Illuminate\Database\Console\Seeds\WithoutModelEvents;
use Illuminate\Database\Seeder;

no usages
class DatabaseSeeder extends Seeder
{
    /**
     * Seed the application's database.
     *
     * @return void
     */
    public function run()
    {
        $this->call([
            PeopleSeeder::class,
        ]);
    }
}
```

Factory:



```
PeopleController.php x StorePeopleRequest.php x api.php x PeopleSeeder.php x PeopleFactory.php x DatabaseSeeder.php x
namespace Database\Factories;

use Illuminate\Database\Eloquent\Factories\Factory;

/**
 * @extends \Illuminate\Database\Eloquent\Factories\Factory<\App\Models\People>
 */
no usages
class PeopleFactory extends Factory
{
    no usages
    public function definition()
    {
        return [
            'Name' => fake()->name(),
            'LastName'=>fake()->lastName(),
            'Street'=>fake()->streetName
        ];
    }
}
```

10Postman:

Pokazanie całej tabeli:

http://127.0.0.1:8000/api/Witczak/305423/people

GET http://127.0.0.1:8000/api/Witczak/305423/people

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION	...	Bulk Edit
Key	Value	Description		

body Cookies (2) Headers (10) Test Results 200 OK 293 ms 8.51 KB Save Response

Pretty Raw Preview Visualize JSON

```
26      "updated_at": "2023-01-31T20:27:06.000000Z"
27    },
28    {
29      "id": 4,
30      "Name": "Sydnee Fay Jr.",
31      "LastName": "Satterfield",
32      "Street": "Heathcote Throughway",
33      "created_at": "2023-01-31T20:27:06.000000Z",
34      "updated_at": "2023-01-31T20:27:06.000000Z"
35    },
36    {
37      "id": 5,
```



GET ⌵ http://127.0.0.1:8000/api/Witczak/305423/people Send ⌵

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

Query Params

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
	Key	Value	Description		

Body Cookies (2) Headers (10) Test Results 200 OK 259 ms 8.65 KB Save Response ⌵

Pretty Raw Preview Visualize JSON ⌵ ⌵

```
394     "updated_at": "2023-01-31T20:27:06.000000Z"
395   },
396   {
397     "id": 50,
398     "Name": "Jedediah Powlowski",
399     "LastName": "Kris",
400     "Street": "Braun Shoal",
401     "created_at": "2023-01-31T20:27:06.000000Z",
402     "updated_at": "2023-01-31T20:27:06.000000Z"
403   },
404 }
```

Pokazanie pojedynczej instancji:

GET ⌵ http://127.0.0.1:8000/api/Witczak/305423/people/7 Send ⌵

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

Query Params

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
	Key	Value	Description		

Body Cookies (2) Headers (10) Test Results 200 OK 397 ms 510 B Save Response ⌵

Pretty Raw Preview Visualize JSON ⌵ ⌵

```
1  {
2    "status": true,
3    "people": {
4      "id": 7,
5      "Name": "Prof. Darrion Heathcote PhD",
6      "LastName": "Mosciski",
7      "Street": "Herzog Knoll",
8      "created_at": "2023-01-31T20:27:06.000000Z",
9      "updated_at": "2023-01-31T20:27:06.000000Z"
10   }
11 }
```

Tworzenie instancji:



POST ▼ http://127.0.0.1:8000/api/Witczak/305423/people Send ▼

Params Authorization Headers (11) **Body** ● Pre-request Script Tests Settings Cookies

● none ● form-data ● x-www-form-urlencoded ● raw ● binary ● GraphQL **JSON** ▼ Beautify

```
1 {
2   ... "Name": "Tomek",
3   ... "LastName": "Fajno",
4   ... "Street": "Beka"
5 }
```

body Cookies (2) Headers (10) Test Results 200 OK 271 ms 478 B Save Response ▼

Pretty Raw Preview Visualize **JSON** ▼ 🔍

```
1 {
2   "status": true,
3   "people": {
4     "Name": "Tomek",
5     "LastName": "Fajno",
6     "Street": "Beka",
7     "updated_at": "2023-01-31T20:31:00.000000Z",
8     "created_at": "2023-01-31T20:31:00.000000Z",
9     "id": 51
10  }
11 }
```

Aktualizowanie instancji:

PUT ▼ http://127.0.0.1:8000/api/Witczak/305423/people/51 Send ▼

Params Authorization Headers (11) **Body** ● Pre-request Script Tests Settings Cookies

● none ● form-data ● x-www-form-urlencoded ● raw ● binary ● GraphQL **JSON** ▼ Beautify

```
1 {
2   ... "Name": "TomekGK",
3   ... "LastName": "FajnoXDD",
4   ... "Street": "Beka trzy"
5 }
```

body Cookies (2) Headers (10) Test Results 202 Accepted 217 ms 341 B Save Response ▼

Pretty Raw Preview Visualize **JSON** ▼ 🔍

```
1 {
2   "status": true,
3   "people": []
4 }
```




```
Raw Preview Visualize JSON ↕
{
  "updated_at": "2023-01-31T20:27:06.000000Z",
  "id": 51,
  "Name": "TomekGK",
  "LastName": "FajnoXDD",
  "Street": "Beka trzy",
  "created_at": "2023-01-31T20:31:00.000000Z",
  "updated_at": "2023-01-31T20:31:00.000000Z"
}
```

Usunięcie instancji:

http://127.0.0.1:8000/api/Witczak/305423/people/51 Save ✎ 💬

DELETE ⌵ http://127.0.0.1:8000/api/Witczak/305423/people/51 Send ⌵

Params Authorization Headers (9) Body Pre-request Script Tests Settings Cookies

Query Params

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
	Key	Value	Description		

ody Cookies (2) Headers (9) Test Results 🌐 204 No Content 179 ms 284 B Save Response ⌵

Pretty Raw Preview Visualize Text ⌵ ☰ 📄 🔍

1



Cookies (2) Headers (10) Test Results 200 OK 302 ms 8.51 KB

Raw Preview Visualize JSON

```
3 {
4   "id": 49,
5   "Name": "Vito Hickie",
6   "LastName": "Grimes",
7   "Street": "Zetta Valley",
8   "created_at": "2023-01-31T20:27:06.000000Z",
9   "updated_at": "2023-01-31T20:27:06.000000Z"
10 },
11 {
12   "id": 50,
13   "Name": "Jedediah Powlowski",
14   "LastName": "Kris",
15   "Street": "Braun Shoal",
16   "created_at": "2023-01-31T20:27:06.000000Z",
17   "updated_at": "2023-01-31T20:27:06.000000Z"
18 }
19 ]
```

Cookies Capture requests Run

Co to jest migracja?

Migration tool to narzędzie w Laravelu, które pozwoli Ci stworzyć nową tabelę w bazie bazując na definicji w kodzie PHP. Wewnątrz metody PHP wskazujesz, jakie kolumny ma zawierać tabela, jakiego typu i jakiej nazwy, po czym narzędzie do migracji stworzy je w bazie danych za Ciebie.

Co to są kody HTTP?

Kod HTTP to informacja zwrócona przez serwer do użytkownika. Każdy posiada odrębne oznaczenie numeryczne. Niektóre z nich są czysto informacyjne, inne potwierdzają powodzenie, lub przekierowanie. Wyróżniamy również błędy HTTP, które strony internetowej bądź serwera.

Co to jest programowanie obiektowe?

Programowanie obiektowe to programowania, w którym programy definiuje się za pomocą obiektów – elementów łączących stan i zachowanie.

Użyłem:

<https://techtolindia.com/how-to-make-rest-api-in-laravel-9>

<https://laravel.com/docs/9.x/seeding>

<https://laravel.com/docs/9.x/eloquent-factories>