

# Laravel – pozostałe zagadnienia (do cz.1)

Początek laboratorium:

- pobrać na pulpit archiwum `Lab009_AI1_start.zip`, w którym umieszczony jest projekt startowy do wykonania zadań oraz rozpakować to archiwum,
- uruchomić skrypt `start.bat` (Windows, 2x kliknięciem) lub `start.sh` (inne systemy, przez polecenie `bash start.sh`),
- wyświetlić zawartość bazy danych `SQLite` z pliku `database.sqlite` za pomocą np. `DBeaver'a` lub rozszerzenia do `VSCode` „`DevDb`”.

Zadania (Laravel, problem N+1):

### Zadanie 9.1: \*

Wyjaśnić następujące zagadnienia:

- *problem N+1*,
- *Eloquent ORM*, dwa tryby ładowania danych powiązanych relacjami,
- który powoduje, a który zapobiega problemowi N+1,
- rozwiązanie problemu N+1.

–

### Zadanie 9.2:

Otworzyć terminal `cmd` (Command Prompt) w `VSCode`.

Uruchomić serwer deweloperski `php` dla przy użyciu komendy `serve artisan'a`.

```
php artisan serve
```

### Zadanie 9.3:

Otworzyć drugą kartę terminala `cmd` (Command Prompt) w `VSCode`.

Zainstalować pakiet `Laravel Debugbar` oraz sprawdzić jego działanie.

<https://github.com/barryvdh/laravel-debugbar>

```
composer require barryvdh/laravel-debugbar --dev
```

### Zadanie 9.4:

Przejsć na podstronę wycieczek.

Przejsć do zakładki `Queries` w `debugbar'u`.

Wyjaśnić, dlaczego wykonywane jest 8 zapytań (1 + 7).

<https://laravel.com/docs/12.x/eloquent-relationships#eager-loading>

W przeglądarce internetowej przejść pod adres: <http://localhost:8000/auth/login>

```
select * from "trips"
select * from "countries" where "countries"."id" = 1 limit 1
select * from "countries" where "countries"."id" = 1 limit 1
select * from "countries" where "countries"."id" = 2 limit 1
select * from "countries" where "countries"."id" = 3 limit 1
select * from "countries" where "countries"."id" = 4 limit 1
select * from "countries" where "countries"."id" = 5 limit 1
select * from "countries" where "countries"."id" = 5 limit 1
```

#### Zadanie 9.5:

Zrealizować naprawę *problemu N+1* poprzez zmianę ładowania „leniwego” (*lazy*) na „zachłanne” (*eager*) kolekcji wycieczek z krajami w funkcji *index* w *TripController*. Przejść ponownie na podstronę wycieczek. Zaobserwować różnicę budowy zapytań. Dodatkowo ograniczyć pobieranie tylko potrzebnych kolumn dotyczących krajów. Ponownie zaobserwować różnicę.

<https://laravel.com/docs/12.x/eloquent-relationships#eager-loading>

```
select * from "trips"
select * from "countries" where "countries"."id" in (1, 2, 3, 4, 5)
```

<https://laravel.com/docs/12.x/eloquent-relationships#eager-loading-specific-columns>

```
select * from "trips"
select "id", "name" from "countries" where "countries"."id" in (1, 2, 3, 4, 5)
```

#### Zadanie 9.6: \*

Zapoznać się z innymi przykładami występowania *problemu N+1*.

<https://laravel-news.com/laravel-n1-query-problems>

## Zadania (Laravel, warstwa serwisów):

#### Zadanie 9.7:

Wyjaśnić następujące zagadnienia:

- *logika biznesowa*,
- *warstwa serwisów* (zastosowanie, zalety),
- *wstrzykiwanie zależności*.

<https://www.imperva.com/learn/application-security/business-logic>

<https://medium.com/@miladev95/service-layer-in-laravel-a4fb8e45d9ee>

(w przypadku osiągnięcia limitu artykułów z tego portalu, otworzyć linki w oknie prywatnym)

<https://medium.com/@mohammad.roshandelpoor/service-layer-design-pattern-in-laravel-php-e132dcb4c2ab>

W artykule występuje także warstwa „*repozytoriów*”, natomiast zamiast nich, można dodać nowe funkcje w modelach (dotyczące danego modelu), które następnie używać je w serwisach.

<https://laravel.com/docs/12.x/container>

<https://laravel.com/docs/12.x/container#automatic-injection>

#### Zadanie 9.8:

W *TripController* dodać konstruktor, w którym umieścić „wstrzyknięcie” serwisu „*TripService*” wraz z przypisaniem do zmiennej klasy (właściwości) *tripService*.

```
protected $tripService;  
  
public function __construct(TripService $tripService)  
{  
    $this->tripService = $tripService;  
}
```

lub tylko

```
public function __construct(  
    protected TripService $tripService  
) {}
```

#### Zadanie 9.9:

W *TripService* uzupełnić funkcję *calculatePromoPrice()*, w której na podstawie wartości parametru *price* (obecna cena wycieczki) dokonać wyliczenia nowej ceny promocyjnej wycieczki:

- jeśli obecna cena jest mniejsza niż 5 000 zł, to promocja nie występuje,
- jeśli obecna cena jest mniejsza niż 10 000 zł, to pomniejszyć cenę o 2 500 zł,
- w pozostałym przypadku pomniejszyć cenę o 5 000 zł.

Zwrócić nową cenę wycieczki.

–

#### Zadanie 9.10:

W *TripController* w funkcji *show()* wywołać funkcję *calculatePromoPrice()* w celu uzyskania nowej ceny wycieczki, którą następnie przekazać do widoku *show* jako zmienną *promoPrice*.

Wyświetlanie nowej ceny w karcie zostało już przygotowane.

Sprawdzić wyliczenia cen dla różnych wycieczek.

### Wycieczka



#### Kolorado

jest wyżynno-górzystym stanem, którego średnia wysokość nad poziomem morza przekracza 2000 m. Najwyższy szczyt Kolorado, Mount Elbert, wznosi się na 4399 m n.p.m.

19000 14000 zł

### Wycieczka



#### Sudety

łańcuch górski na obszarze południowo-zachodniej Polski i północnych Czech, stosunkowo niewielki skrawek znajduje się w Niemczech; najwyższy szczyt Śnieżka - 1603 m n.p.m.

8000 5500 zł

## Zadania (Laravel, logi, middleware):

### Zadanie 9.11:

Zapoznać się z następującymi zagadnieniami:

- *logi* – logowanie (*logs* – *logging*),
- *kanały* (*channels*),
- *poziomy logów* (*log levels*),
- obecne ustawienia w pliku *config/logging.php*,
- *middleware*.

Wykonać poniższą własną komendę (zamieszczoną w *routes/console.php*).

<https://laravel.com/docs/12.x/logging>

<https://www.rfc-editor.org/rfc/rfc5424>

<https://laravel.com/docs/12.x/middleware>

```
php artisan log:clear
```

### Zadanie 9.12:

Dodać do funkcji *show* w *TripController* następujące linijki.

Przejsć na kilka podstron dotyczących danej wycieczki.

Następnie sprawdzić zawartość na końcu pliku *storage/logs/laravel.log*, oraz ostatnie linijki w terminalu (na którym uruchomiona jest aplikacja).

```
use Illuminate\Support\Facades\Log;
```

```
Log::info('Showing details for trip: '.$id);
```

```
Log::channel('stderr')->info('Showing details for trip: '.$id);
```

<http://localhost:8000/trips/1>

<http://localhost:8000/trips/4>

<http://localhost:8000/trips/6>

### Zadanie 9.13:

Dodać w *config/logging.php* w *channels* nowy kanał *lab9\_errors*, który będzie mógł zapisywać komunikaty o zaistniałych błędach do pliku *lab9\_errors.log*.

```
'lab9_errors' => [  
    'driver' => 'single',  
    'path' => storage_path('logs/lab9_errors.log'),  
    'level' => env('LOG_LEVEL', 'error'),  
],
```

### Zadanie 9.14:

Przejsć do *bootstrap/app.php* oraz zmienić „zgłaszanie błędów/wyjątków” z *std\_err* na kanał utworzony w poprzednim zadaniu.

Następnie:

- zalogować się na użytkownika *Jan* i spróbować usunąć kraj *USA*,
- zmienić nazwę pliku zawierającego bazę danych (*database/database.sqlite*) np. na *database2.sqlilte* oraz odświeżyć stronę,

Po tym sprawdzić zawartość *lab9\_errors.log*.

Zmienić nazwę z powrotem na *database.sqlite*.

```
storage > logs > lab9_errors.log  
1 [2025-04-04 15:18:27] local.ERROR: SQLSTATE[23000]: Integrity constraint violation: 19 FOREIGN KEY constraint failed (Connection: sc  
2 [2025-04-04 15:18:36] local.ERROR: Database file at path [C:\Lab009_AI1_start\database\database.sqlite] does not exist. Ensure this  
3 [2025-04-04 15:18:36] local.ERROR: Database file at path [C:\Lab009_AI1_start\database\database.sqlite] does not exist. Ensure this  
4 [2025-04-04 15:18:36] local.ERROR: Database file at path [C:\Lab009_AI1_start\database\database.sqlite] does not exist. Ensure this  
5 [2025-04-04 15:18:36] local.ERROR: Database file at path [C:\Lab009_AI1_start\database\database.sqlite] does not exist. Ensure this  
6 [2025-04-04 15:18:36] local.ERROR: Database file at path [C:\Lab009_AI1_start\database\database.sqlite] does not exist. Ensure this
```

### Zadanie 9.15:

Za pomocą poniższej komendy utworzyć nowy *middleware* o nazwie *LogActivity*. W *bootstrap/app.php* zarejestrować *middleware* *LogActivity* dla grupy *web*.

<https://martinjoo.dev/whats-new-in-laravel-11> (brak pliku `Http/Kernel.php` z Laravel'a 10)

```
php artisan make:middleware LogActivity

->withMiddleware(function (Middleware $middleware) {
    $middleware->web(LogActivity::class);
})
```

### Zadanie 9.16:

Przejsć do *app/http/Middleware/LogActivity.php* i uzupełnić funkcję *handle*, tak żeby była „monitorowana aktywność użytkowników”:

- na *logu w terminalu* ma pojawiać się ogólna informacja, że ktoś wykonał żądanie z daną metodą *HTTP* na daną ścieżkę (ogólna postać),
- do *logu w pliku laravel.log* ma trafiać informacja o tym który dokładnie użytkownik wykonał żądanie z daną metodą na daną ścieżkę (konkretna postać z wartościami parametrów),
- jeśli użytkownik nie jest zalogowany to do *logu w pliku laravel.log* zamiast emaila użytkownika ma być np. że jest on anonimowy.

```
public function handle(Request $request, Closure $next): Response
{
    Log::channel('stderr')->notice(
        "Someone did {$request->method()} on '{$request->route()->uri()}'"
    );

    $email = $request->user() ? $request->user()->email : 'anonymous';


    Log::channel('single')->notice(
        "'{$email}' did {$request->method()} on '{$request->getRequestUri()}'"
    );

    return $next($request);
}
```

### Zadanie 9.17:

Sprawdzić działanie *LogActivity.php* w praktyce poprzez wykonanie różnych akcji będąc zalogowanym na różnych użytkownikach.

```
.... ~ 511.82ms
local.NOTICE: Someone did GET on 'countries'
```

storage > logs >  laravel.log

```
1  ""
2  [2025-04-04 15:30:22] local.NOTICE: 'jan@email.com' did GET on '/countries'
3  [2025-04-04 15:30:47] local.NOTICE: 'jan@email.com' did GET on '/auth/logout'
4  [2025-04-04 15:30:47] local.NOTICE: 'anonymous' did GET on '/trips'
5  [2025-04-04 15:30:54] local.NOTICE: 'anonymous' did GET on '/auth/login'
6  [2025-04-04 15:31:02] local.NOTICE: 'anonymous' did POST on '/auth/login'
7  [2025-04-04 15:31:02] local.NOTICE: 'marta@email.com' did GET on '/trips'
8  [2025-04-04 15:31:06] local.NOTICE: 'marta@email.com' did GET on '/countries'
9  |
```

## Zadania (Laravel, locale):

### Zadanie 9.18:

Zainstalować pakiet z gotowymi tłumaczeniami: *laravel-lang*.  
Zmienić ustawienie *locale* w pliku *.env*.

<https://laravel-lang.com/>

```
composer require laravel-lang/common --dev
```

```
php artisan lang:add pl
```

```
php artisan lang:update
```

<https://laravel.com/docs/12.x/localization>

```
APP_LOCALE=pl
```

### Zadanie 9.19:

Zalogować się na *Jana*.

Przejsć do edycji dowolnej z wycieczek oraz przetestować działanie walidacji (w tym ujemny okres), zaobserwować obecne tłumaczenia komunikatów.

## Edytuj dane wycieczki

- Pole nazwa jest wymagane.
- Pole period jest wymagane.
- Pole cena jest wymagane.

### Zadanie 9.20:

Przejsć do pliku *lang/pl/validation.php* i uzupełnić go tak, aby komunikat dotyczący okresu był już całkowicie przetłumaczony.

## Edytuj dane wycieczki

- Pole okres musi być nie mniejsze od 0.
- Pole cena jest wymagane.

### Zadanie 9.21:

Przejsć do pliku *bootstrap/app.php* oraz komunikat z linii 30 przenieść jako wartość do pliku *lang/pl.json*. Zaproponować nazwę *klucza*.

Następnie w jego miejscu wstawić uzyskiwanie stringu poprzez *klucz*.

W pliku *lang/en.json* przygotować odpowiadające tłumaczenie komunikatu na *j.ang*.

Sprawdzić treść komunikatu dla ustawionych *locale: pl i en*.

<https://laravel.com/docs/12.x/localization#retrieving-translation-strings>

Nie można usunąć rekordu, dla którego istnieją rekordy podrzędne.

APP\_LOCALE=en

This record cannot be deleted because it has dependent records.

#### Zadanie 9.22:

Analogicznie do poprzedniego zadania przygotować tłumaczenie tytułu aplikacji: „Wycieczki górskie” na „Mountain Trips” i zastosować je w szablonie *footer.blade.php*. Analogicznie sprawdzić dla *locale: pl* i *en*.

```
<footer class="container-fluid bg-body-tertiary {{ $fixedBottom ?? 'fixed-bottom' }}">
  <div class="row text-center pt-2">
    <p>&copy;; {{__('Mountain Trips')}} &ndash; 2025</p>
  </div>
</footer>
```

© Wycieczki górskie – 2025

© Mountain Trips – 2025

### Zadania (pozostałe zagadnienia):

#### Zadanie 9.23: \*

*Jasny/ciemny motyw* – możliwy do przełączenia po kliknięciu w księżyc w *navbarze*, co za pomocą *JS* powoduje zmianę wartości atrybutu *data-bs-theme*. Ikona księżyca wykorzystana ze zbioru ikon *Bootstrap’a*.

<https://icons.getbootstrap.com/>

`data-bs-theme="..."`

<https://getbootstrap.com/docs/5.3/customize/color-modes/>

Wycieczki górskie Wycieczki Kraje

🌙 Jan, wyloguj się...

Wycieczki górskie Wycieczki Kraje

🌙 Jan, wyloguj się...

#### Zadanie 9.24: \*

Pamiętanie ulubionych wycieczek w ramach *sesji* – możliwość polubienia/odlubienia wycieczki poprzez kliknięcie serca w danym wierszu. Po wygaśnięciu/skasowaniu sesji ulubione wycieczki są utracone. Ikony serc wykorzystane ze zbioru ikon *Bootstrap’a*.

<https://laravel.com/docs/12.x/session>

<https://laravel.com/docs/12.x/session#interacting-with-the-session>



#	Nazwa wycieczki	Kontynent	Kraj	Okres	Cena		
1	Kolorado	Ameryka Północna	USA	7 dni	19000.00 PLN	♡	<a href="#">Edycja</a>
2	Alaska	Ameryka Północna	USA	10 dni	24000.00 PLN	♥	<a href="#">Edycja</a>
3	Everest	Azja	Chiny	7 dni	22000.00 PLN	♥	<a href="#">Edycja</a>
4	Alpy	Europa	Austria	6 dni	16000.00 PLN	♡	<a href="#">Edycja</a>
5	Kilimandżaro	Afryka	Tanzania	10 dni	25000.00 PLN	♡	<a href="#">Edycja</a>
6	Tatry	Europa	Polska	5 dni	7500.00 PLN	♥	<a href="#">Edycja</a>
7	Sudety	Europa	Polska	5 dni	8000.00 PLN	♡	<a href="#">Edycja</a>

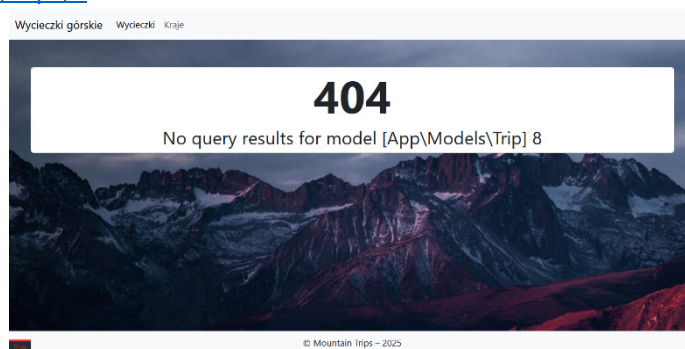
### Zadanie 9.25: \*

Strony błędów – dla wybranych kodów (statusów) *HTTP* można utworzyć widoki, zastępujące domyślny widok.

<https://www.ur.edu.pl/pl/kocur>

<https://laravel.com/docs/12.x/errors#custom-http-error-pages>

<http://localhost:8000/trips/8>



### Zadanie 9.26: \*

*Upload* obrazka wycieczki – formularz przyjmujący kilka obrazków w formacie *.jpg*, *.jpeg* oraz zapisujący je na dysku w *storage'u*.

Reguły walidacji obejmują walidację dla kilku plików na raz oraz zawierają własne komunikaty.

Po pomyślnym zapisaniu plików pokazuje się *toast* informacyjny.

## Cennik

[Dodaj nowy obrazek](#)

# Nazwa wycieczki Kontynent

<https://laravel.com/docs/12.x/filesystem>

<https://laravel.com/docs/12.x/filesystem#automatic-streaming>

<https://laravel.com/docs/12.x/validation#customizing-the-error-messages>

<https://laravel.com/docs/12.x/validation#validating-nested-array-input>

## Przesyłanie obrazków

Przesyłanie kilku (max. 5x) plików:

Przeglądaj... Nie wybrano plików.

Wyślij

<https://getbootstrap.com/docs/5.3/components/toasts/>

Wycieczki górskie

Wycieczki

Kraje

Przesyłanie plików zakończone pomyślnie.

Jan, wyloguj się...



**Zadanie 9.27: \***

*Transakcje – ...*

<https://laravel.com/docs/12.x/database#database-transactions>

\* – zadania/podpunkty do samodzielnego dokończenia/wykonania,

\* – zadania/podpunkty dla zainteresowanych.

Do spakowania projektu należy wykorzystać skrypt *archiwizacja.bat*. Po rozpakowaniu projektu należy użyć ponownie *start.bat*.

Po zakończonym laboratorium należy skasować wszystkie pobrane oraz utworzone przez siebie pliki z komputera w sali laboratoryjnej.

Wersja pliku: v1.0