

Laravel – widoki z CRUD

Początek laboratorium:

- pobrać na pulpit archiwum `Lab007_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):

Zadanie 7.1: *

Wyjaśnić następujące zagadnienia:

- operacje *CRUD*,
- walidacja danych,
- walidacja danych po stronie *backend'u*,
- walidacja danych po stronie *frontend'u*,
- przekierowanie (*redirect*),
- przekierowanie do trasy nazwanej w *Laravel'u*.

<https://laravel.com/docs/12.x/redirects#redirecting-named-routes>

Zadanie 7.2:

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

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

```
php artisan serve
```

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

Zadanie 7.3:

Wykonać poniższe polecenie w celu wygenerowania nowego *kontrolera* dla modelu *Country* będącego „zasobem” wraz z klasami do ustalenia reguł walidacji dla wybranych rodzajów żądań (*app/Http/Requests/...*).

<https://laravel.com/docs/12.x/controllers#resource-controllers>
<https://laravel.com/docs/12.x/controllers#generating-form-requests>

```
php artisan make:controller CountryController --model=Country --resource --requests
```

Zadanie 7.4:

Ustawić trasy dla kontrolera `app/Http/Controllers/CountryController.php` dotyczące operacji *CRUD* na zasobie *Country*.
Wyświetlić obecnie skonfigurowane trasowanie.

<https://laravel.com/docs/12.x/controllers#generating-form-requests>

```
Route::resource('countries', CountryController::class);
```

```
php artisan route:list
```

GET HEAD	countries	countries.index	> CountryController@index
POST	countries	countries.store	> CountryController@store
GET HEAD	countries/create	countries.create	> CountryController@create
GET HEAD	countries/{country}	countries.show	> CountryController@show
PUT PATCH	countries/{country}	countries.update	> CountryController@update
DELETE	countries/{country}	countries.destroy	> CountryController@destroy
GET HEAD	countries/{country}/edit	countries.edit	> CountryController@edit

Zadanie 7.5:

Uzupełnić funkcję `index` w *CountryController* o zwracanie widoku z pliku `resources\views\countries\index.blade.php` wraz z przekazaniem do niego zmiennej `countries` zawierającej kolekcję wszystkich krajów pobranych z bazy danych.
Następnie w tym widoku uzupełnić generowanie adresu linku „Dodaj nowy kraj” (używając *helper’a route*) prowadzącego do trasy nazwanej `countries.create`.

<https://laravel.com/docs/12.x/eloquent#retrieving-models>

<https://laravel.com/docs/12.x/helpers#method-route>

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

Zadanie 7.6:

Uzupełnić funkcję `create` w *CountryController* o zwracanie widoku z pliku `resources\views\countries\create.blade.php`.
Następnie w tym widoku uzupełnić formularz umożliwiający wpisanie wszystkich danych nowego kraju o ustalenie przesyłania ich żądaniem ... (*metoda HTTP*) pod adres obsługiwany przez funkcję `store`.
Zadbać, aby po niepoprawnym przesłaniu formularza, w jego polach pozostały ostatnio wprowadzone wartości.

<https://laravel.com/docs/12.x/helpers#method-old>

W przeglądarce internetowej przejść pod adres: <http://localhost:8000/countries/create>

Zadanie 7.7:

Uzupełnić funkcję `store` w *CountryController* o dodanie do bazy danych nowego kraju, z danymi zebranymi z formularza z poprzedniego zadania.
Po udanym wstawieniu kraju do bazy → ma nastąpić przekierowanie pod `country.index`, po nieudanym → akcja domyślna.
W `app\Http\Requests\StoreCountryRequest.php`:

- w `authorize` zmienić `false` na `true`,
- w `rules` (na podstawie migracji) uzupełnić *reguły walidacji* nowych krajów m.in. opcjonalności, typu danych, unikalne nazwy, maks. długość nazwy, zakres powierzchni.

<https://laravel.com/docs/12.x/requests#retrieving-input>
<https://laravel.com/docs/12.x/eloquent#inserts>
<https://laravel.com/docs/12.x/redirects#redirecting-named-routes>

<https://laravel.com/docs/12.x/validation#quick-writing-the-validation-logic>
<https://laravel.com/docs/12.x/validation#creating-form-requests>

```
return [  
    'name' => '...',  
    'code' => '...',  
    'currency' => '...',  
    'area' => '...',  
    'language' => '...',  
];
```

Sprawdzić działanie formularza dodawania nowego kraju:

- podając kraj z prawidłowymi danymi,
- podając kraj z nieprawidłowymi danymi np. brakującymi wartościami, ujemną powierzchnią.

Zadanie 7.8:

W `resources\views\countries\index.blade.php` zmodyfikować tabelkę tak, aby była w niej dodatkowa kolumna zawierająca link „Edycja” w każdym wierszu.

```
<td><a href="{{route('countries.edit', $country->id)}}">Edycja</a></td>  
lub  
<td><a href="{{route('countries.edit', $country)}}">Edycja</a></td>
```

Edycja

Edycja

Edycja

Zadanie 7.9:

Uzupełnić funkcję `edit` w `CountryController` o zwracanie widoku z pliku `resources\views\countries\edit.blade.php` wraz z przekazaniem do niego danych kraju automatycznie zmapowanego i pobranego z bazy danych, wybranego do edycji (w celu wstępnego uzupełniania pól formularza danymi tego kraju).

Następnie w tym widoku uzupełnić formularz umożliwiający wpisanie wszystkich danych edycji danego kraju o ustalenie przesyłania ich żądaniem POST (z ukrytym polem `_method` z wartością „PUT” – specyficznie w *Laravel'u*) pod adres obsługiwany przez funkcję `update` z przekazaniem całego obiektu kraju (lub `id` kraju).

Zadbać, aby po wyświetleniu formularza, w jego polach zostały automatycznie wstawione bieżące dane edytowanego kraju.

<https://laravel.com/docs/12.x/routing#implicit-binding>
<https://laravel.com/docs/12.x/blade#method-field>

W przeglądarce internetowej przejść pod adres: <http://localhost:8000/countries/1/edit>

Zadanie 7.10:

Uzupełnić funkcję `update` w `CountryController` o aktualizację w bazie danych konkretnego kraju danymi przyjmowanymi w żądaniu.

Po udanej edycji kraju → ma nastąpić przekierowanie pod `country.index`, po nieudanej → akcja domyślna.

W `app\Http\Requests\UpdateCountryRequest.php`:

- w `authorize` zmienić `false` na `true`,

- w *rules* (na podstawie migracji) uzupełnić *reguły walidacji* nowych krajów m.in. opcjonalności, typu danych, unikalne nazwy, maks. długość nazwy, zakres powierzchni. Uwzględnić wykluczenie sprawdzania unikatowości wobec tego samego obiektu.

<https://laravel.com/docs/12.x/eloquent#updates>

```
public function update(UpdateCountryRequest $request, Country $country)
{
    $input = $request->all();
    $country->update($input);
    return redirect()->route('countries.index');
}
```

<https://www.csrhymes.com/2019/06/22/using-the-unique-validation-rule-in-laravel-form-request.html>

```
return [
    'name' => 'required|string|unique:countries,name, '.$this->country->id.' |max:50',
    'code' => 'required|string|unique:countries,code, '.$this->country->id.' |max:3',
    //...,
];
```

Sprawdzić działanie formularza edycji danego kraju:

- podając prawidłowe dane,
- podając nieprawidłowe dane np. brakujące wartości, ujemna powierzchnia.

Zadanie 7.11:

Uzupełnić funkcję *destroy* w *CountryController* o usunięcie z bazy danych konkretnego wybranego kraju.

Po udanym usunięciu kraju → ma nastąpić przekierowanie pod *country.index*, po nieudanym → akcja domyślna.

<https://laravel.com/docs/12.x/eloquent#deleting-models>

Zadanie 7.12:

W *resources\views\countries\index.blade.php* zmodyfikować tabelkę tak, aby była w niej dodatkowa kolumna zawierająca (mały) formularz z przyciskiem *Usuń* dotyczący usuwania danego kraju w każdym wierszu.

Przesłanie formularza ma być żądaniem *POST* (z ukrytym polem *_method* z wartością „DELETE” – specyficznie w *Laravel'u*) pod adres obsługiwany przez funkcję *destroy* z przekazaniem *id* kraju.

```
<td>
    <form method="POST" action="{{ route('countries.destroy', $country->id) }}">
        @csrf
        @method('DELETE')
        <input type="submit" class="btn btn-danger" value="Usuń"
            style="--bs-btn-padding-y: .25rem; --bs-btn-padding-x: .5rem; --bs-btn-font-size: .75rem;" />
    </form>
</td>
```

Sprawdzić działanie usuwania danego kraju:

- usuwając kraj z tabeli,
- usuwając kraj z tabeli ze zmodyfikowanym w narzędziach dewelopera (F12) nieistniejącym identyfikatorem np. 60.

Edycja	Usuń
Edycja	Usuń
Edycja	Usuń
Edycja	Usuń

```
<td>Australia</td>
<td>Auk</td>
<td>dolar australijski</td>
<td>768858 km²</td>
<td>angielski</td>
</td>
<td>
    <form method="POST" action="http://localhost:8000/countries/60"
        <input type="hidden" name="_token"
            value="oKnoQF9HREYSNzKs8i3DvVUuZK6vKvjvLvi"
            autocomplete="off">
        <input type="hidden" name="_method" value="DELETE">
        <input class="btn btn-danger" type="submit" value="Usuń"
            style="--bs-btn-padding-y: .25rem; --bs-btn-padding-x: .5rem;
            --bs-btn-font-size: .75rem;" />
    </form>
</td>
```

Kraje

[Dodaj nowy kraj](#)

#	Nazwa	Kod	Waluta	Powierzchnia	Język		
1	USA	US	dolar amerykański	9833520 km ²	angielski	Edycja	Usuń
2	Chiny	CN	yuan	9596960 km ²	mandaryński	Edycja	Usuń
3	Austria	AT	euro	83879 km ²	niemiecki	Edycja	Usuń
4	Tanzania	TZ	szyling tanzański	947300 km ²	suahili	Edycja	Usuń
5	Polska	PL	złoty	38179800 km ²	polski	Edycja	Usuń
6	Australia	AU	dolar australijski	7686850 km ²	angielski	Edycja	Usuń

Zadania (Laravel c.d.):

Zadanie 7.13: *

Spróbować usunąć pierwszy kraj. Wyjaśnić zaistniałą sytuację. Zaproponować rozwiązanie problemu.

Wykorzystać sposób wyświetlania błędów w sesji umieszczony w szablonie `resources\views\shared\session-error.blade.php`.

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

Nie można usunąć rekordu, dla którego istnieją rekordy podrzędne.

#	Nazwa	Kod	Waluta	Powierzchnia	Język		
1	USA	US	dolar amerykański	9833520 km ²	angielski	Edycja	Usuń

Zadanie 7.14: *

W `resources\views\countries\index.blade.php` zmodyfikować tabelkę tak, aby w kolumnie zawierającej `id` krajów były one linkami, a kliknięcie przenosiłoby do podstrony z informacjami o danym kraju (wykorzystanie funkcji `show` oraz widoku `show.blade.php`). Zaprojektować podstronę w tym widoku, np. karta/tabelka itp.

Kraj

Nazwa	USA
Kod	US
Waluta	dolar amerykański
Powierzchnia	9833520 km ²
Język	angielski
	Edycja
	Usuń

Nazwa

[1](#) USA

[2](#) Chiny

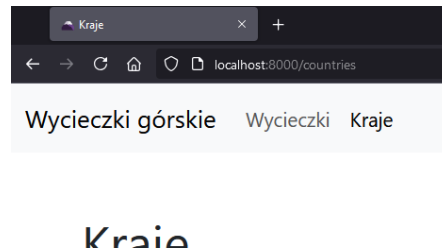
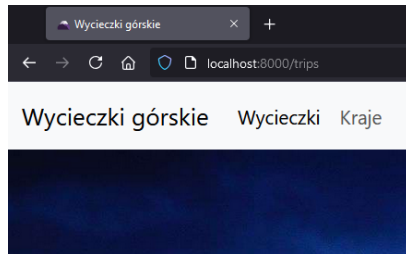


Zadanie 7.15: *

Zmodyfikować *navbar*, tak aby były w nim dwa linki:

- do strony głównej wycieczek,
- do tabelki z krajami.

Używając obiektu *request* w zależności od przeglądanej strony wyróżnić odpowiedni link.



Zadanie 7.16: *

Dodać do *TripController* dwie nowe funkcje: *edit* i *update*, w których analogicznie zaprogramować edycję danej wycieczki. Ustawić dla nich *routing*.

W formularzu ma być pole *select* do wygodnego dla użytkownika wyboru kraju (na podstawie krajów obecnych dostępnie w bazie danych), w którym odbywa się wycieczka. Umieścić załączenia szablonów *shared.session-error* i *shared.validation-error*.

```
Route::get('/trips/{id}/edit', 'edit')->name('trips.edit');
```

```
Route::put('/trips/{id}', 'update')->name('trips.update');
```

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

Edytuj dane wycieczki

Nazwa

Kolorado

Kontynent

Ameryka Północna

Okres

7

Opis

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.

Cena

19000.00

▼

PLN

Nazwa obrazka

colorado.jpg

Kraj

USA

▼

Wyślij

<https://developer.mozilla.org/en-US/docs/Web/HTML/Element/select>

Zadanie 7.17: *

Do poprzedniego zadania dodać: reguły walidacji i ich obsługę, ale tym razem innym sposobem: bezpośrednio określenie ich w funkcji *update* (zamiast klas *...TripRequest*) wraz z przekazaniem ich do *validate*.

Uwzględnić sprawdzanie istnienia wybranego kraju.

<http://localhost:8000/trips>

```
$request->validate([
    'name' => 'required|string|unique:trips,name, '.$id',
    //...,
]);
```

<https://laravel.com/docs/12.x/validation#basic-usage-of-exists-rule>

Zadanie 7.18: *

W *resources\views\trips\index.blade.php* zmodyfikować tabelkę tak, aby była w niej dodatkowa kolumna zawierająca w każdym wierszu link „Edycja”, prowadzący do podstrony edycji danej wycieczki.

Cena

19000.00 PLN	Edycja
24000.00 PLN	Edycja
22000.00 PLN	Edycja

* – 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.