

Swagger Editor, Laravel REST API – autoryzacja tokenami JWT

Początek laboratorium:

- przejść pod adres <https://editor.swagger.io/>,
- usunąć całą zawartość, wkleić zawartość pliku *Countries API & JWT Lab006 start.yaml*,
- włączyć panel zarządzania XAMPP oraz uruchomić moduły Apache oraz MySQL, następnie kliknąć przycisk *Admin* przy MySQL, w celu przejścia do panelu *phpMyAdmin*,
- pobrać na pulpit archiwum *Lab006_PAB_start.zip*, w którym umieszczony jest projekt startowy do wykonania zadań oraz rozpakować to archiwum,
- przejść do rozpakowanego folderu oraz w przypadku posiadania innych ustawień połączenia z bazą niż domyślne, wykonać ich zmianę w pliku *.env.example*,
- uruchomić skrypt *start.bat* 2x kliknięciem.

Zadania (Swagger Editor, Laravel):

- <https://swagger.io/blog/api-design/design-first-or-code-first-api-development/>

Zadanie 6.1:

Uruchomić aplikację. Sprawdzić ustawienie ścieżek związanych z uwierzytelnianiami i autoryzacją użytkowników w pliku *routes\api.php*.

POST	<code>api/auth/login</code>	<code>AuthController@login</code>
POST	<code>api/auth/logout</code>	<code>AuthController@logout</code>
POST	<code>api/auth/me</code>	<code>AuthController@me</code>
POST	<code>api/auth/refresh</code>	<code>AuthController@refresh</code>
POST	<code>api/auth/forgot-password</code>	<code>AuthController@forgotPassword</code>
POST	<code>api/auth/reset-password</code>	<code>AuthController@resetPassword</code>

Zadanie 6.2:

Przejdź do *Swagger Editor*. Za pomocą **Try it out** sprawdź działanie *endpoint'u* „login”. Uzyskać *token'y* dla *admin'a Jana* i *user'a Marty* (oraz skopiować je do pliku tekstowego).

- email: jan@email.com, hasło: 1234,
- email: marta@email.com, hasło: 1234.

Dla *endpoint'ów* operacji *CRUD* krajów w aplikacji obowiązuje obecnie:

- tylko „zalogowani” (w znaczeniu „przedstawiający się ważnym *token'em*”) użytkownicy mogą odczytywać dane krajów/dane danego kraju,
- tylko użytkownicy z rolą „*admin*” mogą wykonywać operacje dodawania/modyfikacji/usuwania danego kraju.

–

Uzupełnić dokumentację *endpoint*ów wszystkich operacji *CRUD* dotyczących krajów o stosowanie tego schematu.

Method	URL	Description
DELETE	/countries/{countryId}	Delete a country

Uzupełnić dokumentację *endpoint*’ów dla wszystkich operacji *CRUD* dotyczących krajów o możliwy przypadek odpowiedzi wykorzystującej tą definicję.

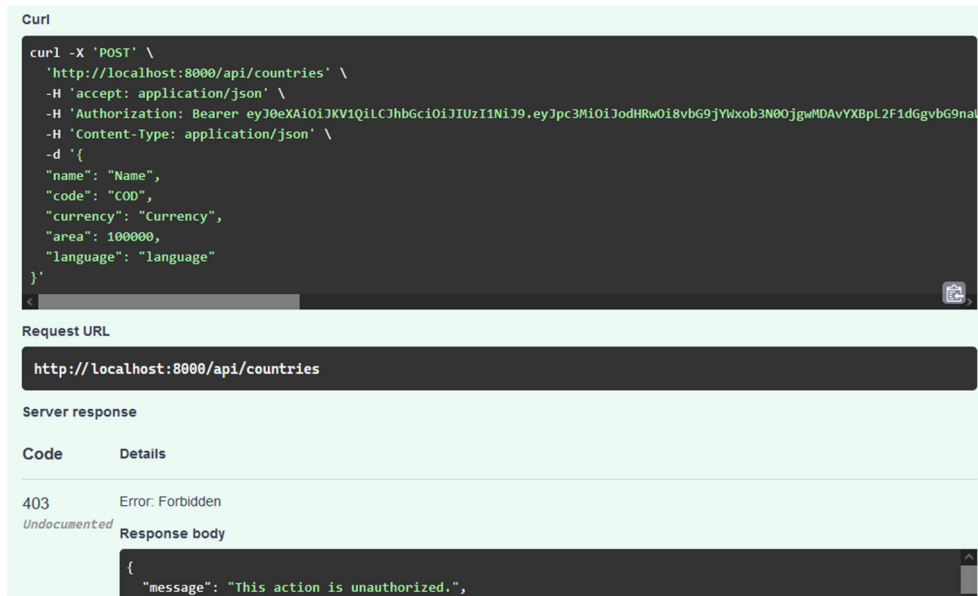
No links

„Zalogować” Jana (zapamiętać jego token). Następnie pobrać wszystkie kraje.

```
http://localhost:8000/api/countries
```

Zadanie 6.6:

„Wylogować” Jana (zapomnieć jego *token*). „Zalogować” Martę (wcześniej utworzonym *token’em*). Następnie spróbować dodać nowy kraj.



Zadanie 6.7:

Dodać (w *components* → *responses*) analogicznie do zadania 6.4 definicję dla odpowiedzi o błędnej *autoryzacji* żądania.

Uzupełnić dokumentację *endpoint’ów* dla wszystkich operacji *CRUD* dotyczących modyfikacji krajów o możliwy przypadek odpowiedzi wykorzystującej tą definicję.

403	This action is unauthorized	No links
-----	-----------------------------	----------

Zadanie 6.8: *

Napisać *endpoint* dla operacji pobrania danych „zalogowanego” użytkownika, tak, aby pokrywał się z działaniem funkcji kontrolera *me()*.

Dodać (w *components* → *responses*) definicję dla zwracanego obiektu użytkownika.



Zadanie 6.9: *

Napisać *endpoint* dla operacji „wylogowania” użytkownika (unieważnienie token’a), tak, aby pokrywał się z działaniem funkcji kontrolera *logout()*.



Zadanie 6.10: *

Napisać *endpoint* dla operacji „odświeżania” tokena (przypomnieć co robi ta operacja), tak, aby pokrywał się z działaniem funkcji kontrolera `refresh()`.

POST /auth/refresh Refresh a token



Zadania (Swagger i Laravel, resetowanie hasła):

Zadanie 6.11:

Napisać *endpoint'y* dla procesu resetowania (przypominania) hasła.

Z powodów technicznych akcja przygotowywania zawartości maila z linkiem oraz jego wysyłania zastąpiona jest wypisaniem samego *token'a* resetowania hasła w postaci logu na konsoli.

<https://laravel.com/docs/10.x/passwords>

Zadanie 6.12:

Napisać dokumentację *endpoint'u* obsługi żądania mającego na celu uzyskanie *token'a*, tak, aby pokrywał się z działaniem funkcji kontrolera `forgotPassword()`:

- metoda *POST*,
- ścieżka: `.../api/auth/forgot-password`,
- należy do grupy *auth* (w *tags*),
- żądanie nie potrzebuje uwierzytelniania,
- w wymaganym ciele żądania ma być *JSON* z *email'em użytkownika*, który chce dokonać zresetowania hasła,
- w przypadku:
 - pozytywnym oczekiwany jest status 202 (*Accepted*), z pustym ciałem odpowiedzi,
 - niepozytywnym wynikającym z nieodnalezienia użytkownika oczekiwany jest status 404,
 - niepozytywnym wynikającym z braku/podania pustego lub w nieprawidłowej postaci adresu email, oczekiwany status to 422.

<https://developer.mozilla.org/en-US/docs/Web/HTTP/Status/202>

POST /auth/forgot-password Get the token for a password reset request.



POST /auth/reset-password Reset a user's password.



Zadanie 6.13:

Przetestować za pomocą (**Try it out**) oczekiwane sytuacje ustalone w poprzednim zadaniu. Po pomyślnej, sprawdzić zawartość tabeli *password_resets*. Odczytać *token* z logu w konsoli.

http://localhost/phpmyadmin/index.php?route=/sql&db=pab_lab6&table=password_reset_tokens

Zadanie 6.14:

Napisać dokumentację *endpoint'u* obsługi żądania mającego na celu zrealizowanie zmiany hasła, tak, aby pokrywał się z działaniem funkcji kontrolera `resetPassword()`:

- metoda *POST*,
- ścieżka: `.../api/auth/reset-password`,
- należy do grupy *auth* (w *tags*),
- żądanie nie potrzebuje uwierzytelniania,
- w wymaganym ciele żądania ma być *JSON* z:
 - otrzymanym wcześniej *token'em*,
 - *email'em* użytkownika, który chce dokonać zresetowania hasła,
 - nowym hasłem użytkownika (min. 3 znaki),
- w przypadku:
 - pozytywnym oczekiwany jest status 200 (z wiadomością dla użytkownika),
 - niepozytywnym wynikającym z m.in. podania błędnego *token'a*, oczekiwany jest status 404,
 - niepozytywnym wynikającym z braku/podania pustego lub w nieprawidłowej postaci wymaganych danych, oczekiwany status to 422.

–

Zadanie 6.15:

Przetestować za pomocą (**Try it out**) oczekiwane sytuacje ustalone w poprzednim zadaniu. Po pomyślnej, sprawdzić zawartość tabeli *password_resets*. Następnie uzyskać nowy *token JWT* posługując się nowym hasłem.

–

Zadanie 6.16: *

Zapoznać się ze statusem 409. Zaproponować sytuacje, gdzie można ustalić zwracanie tego statusu zamiast 422.

<https://developer.mozilla.org/en-US/docs/Web/HTTP/Status/409>

* – zadania/podpunkty do samodzielnego dokończenia/wykonania,

* – zadania/podpunkty dla zainteresowanych.

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

Wersja pliku: v1.0

Inne: *

<https://auth0.com/blog/forbidden-unauthorized-http-status-codes/>

Don't let the client know...

An origin server that wishes to "hide" the current existence of a forbidden target resource MAY instead respond with a status code of 404 (Not Found).