

## Testowanie backendu narzędziami Postma i SoapUI

### Postman – Zadania

#### Zadanie P\_1 - Pobierz listę ras

Utwórz folder o nazwie **“Exercise P\_1 – Get all races”**, a w nim przygotuj request, który będzie pobierał listę wszystkich ras dostępnych pod zasobami **“dictionaries”**

Asercje:

- A. Dodaj asercję sprawdzającą czy http code = 200
- B. dodaj asercję sprawdzającą czy ilość elementów na liście w JSONie zwrótnym **jest równa 10**
- C. dodaj asercję sprawdzającą czy na 5 pozycji nazwa kota to **“Kot egzotyczny”**

**Hint:** Stronka wspierająca w czytaniu JsonPath: <https://jsonpath.com/>

**(czas: 15 min)**

#### Zadanie P\_2 - Zarejestruj nowego użytkownika

**Krok 1.** Utwórz nowy folder o nazwie **“Exercise P\_2 – Register new user”**, a w nim utwórz request, który rejestruje nowego użytkownika.

**Hint:** Przykład zawartości body requestu tworzącego użytkownika odnajdziesz w dokumentacji Swagger na stronie <https://norad-beta.duckdns.org/norad/swagger-ui.html>

**Krok 2.** Zmień request w taki sposób, ażeby można było wywołać request utworzenia **wielokrotnie**.

Czyli e-mail nie może się powtarzać, za każdym wywołaniem powinien generować się nowy unikalny ze stałym prefixem - przyjmijmy Twoja pierwsza litera imienia i nazwisko. Na przykład: **mszpiler**

**Hint:** Przeszukaj projekt **postman-soapui-jmeter-qa-training** w katalogu **java-script** znajdując się przykłady, które mogą Tobie pomóc rozwiązać zadanie.

**Krok 3.** Dodaj asercję sprawdzającą, że w odpowiedzi będzie w strukturze JSON w polu **email** istniała wartość oczekiwana wygenerowana przed wywołaniem requesta tworzącego nowego użytkownika.

**Krok 4.** Przygotuj Pre-request Script, który pobiera Access Token.

**Hint:** Wklej zawartość **/postman-soapui-jmeter-qa-training/exercises-answers/postman/ex-p02-step-04-pre-req-fetch-access-token.js** do zakładki **“Pre-request Script”**. Wywołaj dowolny request do backendu wymagający autoryzacji.

**Krok 5.** Dodaj request sprawdzający istnienie rekordu użytkownika w bazie danych. Napisz test do tego requesta, który potwierdzi istnienie encji użytkownika w backendowej bazie danych.

(czas: 30min)

### Zadanie P\_3 - Dodaj nowe zwierzę do nowego użytkownika

Utwórz nowy folder **“Exercise P\_3 – Add new animal for new user”**, który zawierać będzie następujące kroki (requesty):

**Krok 1.** Rejestracja nowego użytkownika.

**Krok 2.** Pobierz Access Token dla utworzonego użytkownika w kroku 1 i zapisz pod zmienną **ACCESS\_TOKEN**.

**Krok 3.** Dodaj request pobierania wszystkich ras. Pobierz wybrany losowo element z listy ras i zastosuj go przy tworzeniu zwierzęcia. Zapisz go w zmiennych środowiskowych pod nazwą **RACE\_ID**

**Krok 4.** Dodaj nowe zwierzę dla użytkownika z kroku 1, dodane zwierzę powinno zawierać 3 zdjęcia oraz rasę ustaloną w kroku 3.

Wykorzystaj **ACCESS\_TOKEN** zapamiętany w poprzednich ćwiczeniach w zmiennych środowiskowych.

Dodatkowo, przed wywołaniem tworzenia nowego zwierzęcia przygotuj pre-request script, który wygeneruje losową nazwę zwierzęcia ze stałym prefixem na przykład pierwsza litera imienia i nazwisko np. **mszpiler** . Wygenerowaną nazwę zapisz pod zmienną środowiskową **ANIMAL\_NAME**.

Prefix pozwoli lepiej identyfikować Twoje dane w bazie danych, która jest wspólna dla wszystkich uczestników szkolenia.

Napisz skrypt w “Tests”, który zapisuje uuid zwierzęcia w zmiennych środowiskowych pod nazwą **ANIMAL\_UUID**.

#### Asercje w kroku 4

- A. Napisz test, który sprawdzi czy w odpowiedzi lista zdjęć jest oczekiwana i wynosi 3
- B. Napisz test, który sprawdzi czy każde ze zdjęć posiada niepuste pole uuid.

**Krok 5.** Utwórz request pobierający szczegóły zwierzęcia.

Wykorzystaj **ACCESS\_TOKEN** zapamiętany w poprzednich ćwiczeniach w zmiennych środowiskowych.

#### Asercje w kroku 5

- A. Sprawdź czy rasa w odpowiedzi jest taka sama jak rasa wylosowana w kroku 3.

**Krok 6.** Utwórz request pobierający szczegóły użytkownika.

Wykorzystaj ACCESS\_TOKEN zapamiętany w poprzednich ćwiczeniach w zmiennych środowiskowych.

*Asercje w kroku 6*

- A. Sprawdź czy UUID użytkownika powiązane jest z użytkownikiem utworzonym w kroku 1.

**(czas: 45min)**

### Zadanie P\_4 – Przetestuj upload zdjęcia - załącznik wskazany ręcznie

W kolekcji **“Training Collection”** utwórz folder **“Exercise P\_4 – Upload file”**, który będzie wykonywał test uploadu wybranego contentu zdjęcia na serwer. Wykorzystaj encję zdjęcia utworzonego w zadaniu P\_3.

**Krok 1.** Popraw **“Step 4 – Create new animal”** w zadaniu **“Exercise P\_3 – Add new animal for new user”** w taki sposób ażeby z responsa zapisać w zmiennych środowiskowych UUID pierwszego zdjęcia. Zapisz tą wartość pod zmienną **FIRST\_IMAGE\_UUID**

**Krok 2.** Dodaj request o nazwie **“Upload file”**, który wywołuje endpoint odpowiedzialny za upload plików na serwer.

**Krok 3.** Dodaj załącznik ręcznie i wykonaj request, który wykona upload obrazka na serwer.

Do autoryzacji wykorzystaj ACCESS\_TOKEN zapamiętany w poprzednich ćwiczeniach w zmiennych środowiskowych.

*Asercje*

- A. Sprawdź czy odpowiedź http = 200

**(czas: 20min)**

### Zadanie P\_5 - Dodaj zwierzęta pobierając dane z pliku CSV

W kolekcji **“Training Collection”** utwórz folder **“Exercise P\_5 – Add many animals from CSV file”**, który utworzy wiele zwierząt dla konkretnego użytkownika.

**Krok 1.** W folderze **“Exercise P\_5 – Add many animals from CSV file”**, utwórz 2 dodatkowe foldery:

- **Register new user** – tutaj będzie request do utworzenia nowego użytkownika

- **Create many animals** - tutaj będzie request do utworzenia zwierzęcia, folder zostanie użyty do wywołania Runnera z listą zwierząt znajdujących się w pliku CSV.

**Krok 2.** Utwórz step rejestrujący nowego użytkownika.

W kroku tym również podobnie jak w zadaniu P\_2, email użytkownika powinien być generowany losowo.

**Krok 3.** Pobierz aktualny Access Token użytkownika utworzonego w kroku 2.

**Krok 4.** W folderze **“Create many animals”** utwórz request tworzący jedno zwierzę - ciało requesta ma sparametryzowane wartości **name**, **racelId**, **userUUID**. Parametr **userUUID** pobieramy ze zmiennej **EX5\_USER\_UUID**, natomiast parametry **name** i **racelId** będą pobrane z pliku CSV w trakcie wywołania testu za pomocą Runnera.

**Krok 5.** Utwórz asercję sprawdzającą poprawne wykonanie kroku 4 – http code = 200.

Lista zwierząt znajduje się w pliku CSV. Dane na temat zwierzęcia w pliku, to nazwa zwierzęcia oraz rasa.

**Krok 6.** Wywołaj request z folderu **“Create many animals”** za pomocą **Runnera**. W Runnerze podaj środowisko oraz wskaż plik CSV z danymi do testów. Przeanalizuj wyniki testów.

**(czas: 45min)**