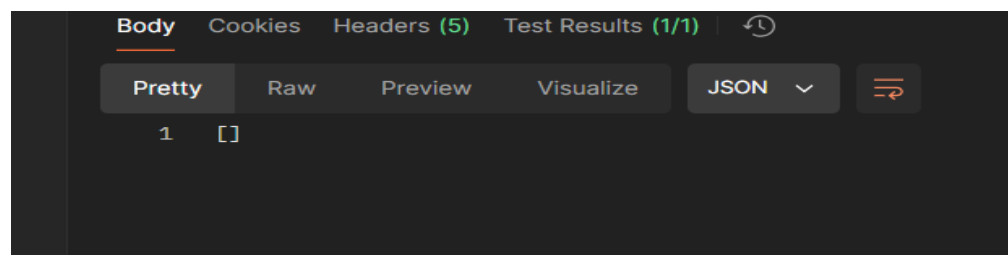


Scenariusze testowe API dla książek:

Zaktualizowane scenariusze testowe dla API książek

1. Pobranie pustej listy książek

- Nazwa testu: Get Books List Empty
- Opis: Weryfikacja, czy API zwraca pustą listę książek.
- Metoda HTTP: GET
- Endpoint: /books
- Kroki testowe:
 - a. Wyślij żądanie GET na /books.
 - b. Oczekuj odpowiedzi z kodem statusu 200 OK.
 - c. Sprawdź, czy odpowiedź zawiera pustą listę książek (pusta tablicę obiektów).
- Oczekiwany wynik:
 - a. Kod statusu: 200 OK
 - b. Body odpowiedzi



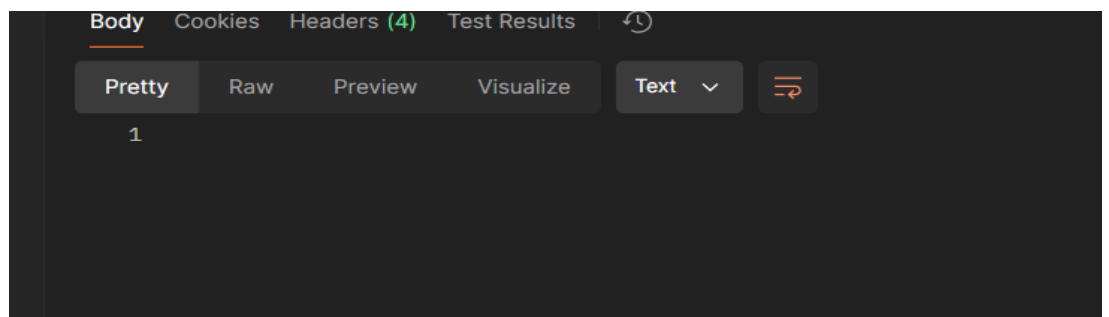
c.Body testu:

```
1 pm.test("Status code is 200", function () {  
2   pm.response.to.have.status(200);  
3 });
```

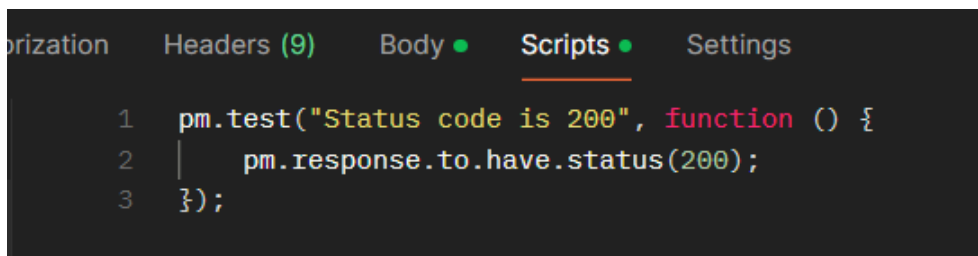
2. Dodanie książki do pustej listy.

- Nazwa testu: Post Added book to the list
- Opis: Weryfikacja czy API dodaje książki do listy.

- Metoda HTTP:POST
- Endpoint:/books
- Kroki testowe:
 - a. Wyślij żądanie POST na /books.
 - b. Oczekuj odpowiedzi z kodem statusu 200 OK.
 - c. Sprawdź czy dodano autora: J.R.R Tolkien i title:Lord of the Rings.
 - d. Sprawdź czy odpowiedź zawiera tablicę dodanej książki
- Oczekiwany wynik:
 - a. Kod statusu 200 OK
 - b. Body odpowiedzi.



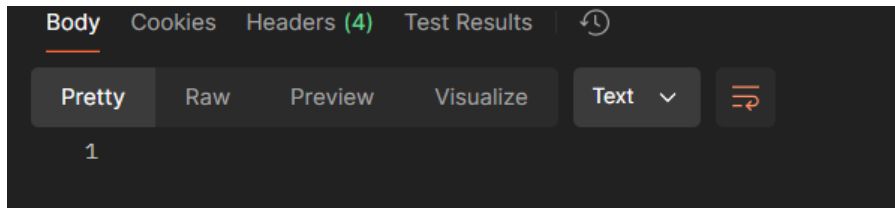
c.Body testu:



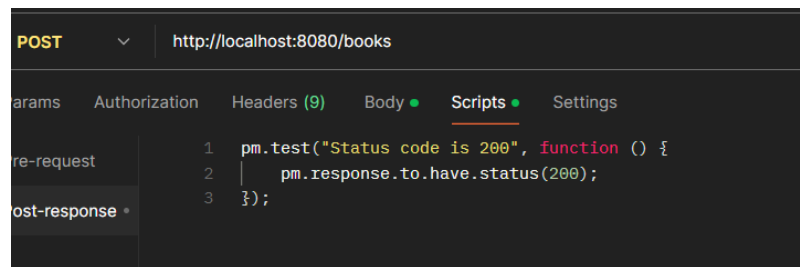
3.Dodanie kolejnej książki do listy.

- Nazwa testu:Post Added next book to the list
- Opis:Weryfikacja czy API dodaje kolejne książki do listy.
- Metoda HTTP:POST
- Endpoint:/books
- Kroki testowe:
 - a. Wyślij żądanie POST na /books.
 - b. Oczekuj odpowiedzi z kodem statusu 200 OK.

- c. Sprawdź czy dodano autora: Remigiorz Mroz i title: Listy z za grobu.
 - d. Sprawdź czy odpowiedź zawiera tablicę dodanej książki
- Oczekiwany wynik:
 - a. Kod statusu 200 OK
 - b. Body odpowiedzi.



c.Body testu:



4. Pobranie istniejącej listy książek

- Nazwa testu: Get Added Books to the list
- Opis: Weryfikacja, czy API zwraca pustą listę książek.
- Metoda HTTP: GET
- Endpoint: /books
- Kroki testowe:
 - a. Wyślij żądanie GET na /books.
 - b. Oczekuj odpowiedzi z kodem statusu 200 OK.
 - c. Sprawdź, czy odpowiedź zawiera listę książek (tablicę obiektów).
- Oczekiwany wynik:
 - a. Kod statusu: 200 OK
 - b. Body odpowiedzi

```

6      {
7        "title": "Lord of the Rings",
8        "author": "J.R.R. Tolkien"
9      },
10     {
11       "title": "Listy z za grobu",
12       "author": "Remigiorz Mroz"
13     }

```

c.Body testu:

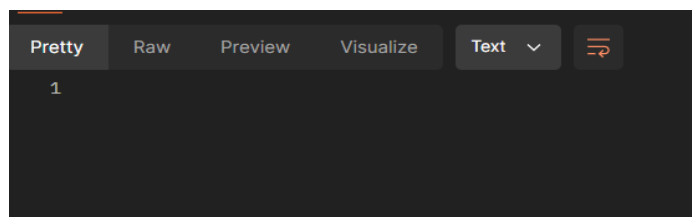
```

1  pm.test("Status code is 200", function () {
2    |    pm.response.to.have.status(200);
3  });

```

5. Usunięcie wszystkich książek z listy.

- Nazwa testu: Delete Delete all books from the list
- Opis: Weryfikacja, czy API zwraca pusta listę książek.
- Metoda HTTP: DELETE
- Endpoint: /books
- Kroki testowe:
 - a. Wyślij żądanie Delete na /books.
 - b. Oczekuj odpowiedzi z kodem statusu 200 OK.
 - c. Sprawdź, czy odpowiedź zawiera listę książek (tablicę obiektów).
- Oczekiwany wynik:
 - a. Kod statusu: 200 OK
 - b. Body odpowiedzi



The screenshot shows a REST client interface with tabs for 'Pretty', 'Raw', 'Preview', 'Visualize', and 'Text'. The 'Text' tab is selected, and it displays a single line of text: '1'.

c.Body testu:

POST

▼

http://localhost:8080/books

Params

Authorization

Headers (9)

Body ●

Scripts ●

Settings

Pre-request

Post-response ●

1

2

3

pm.test("Status code is 200", function () {

pm.response.to.have.status(200);

});