PRZYPADKI TESTOWE SIMPLE BOOKS API W PROGRAMIE POSTMAN

Adres strony: https://simple-books-api.glitch.me

Dokumentacja: https://github.com/vdespa/introduction-to-postman-course/blob/main/simple-books-api.md

Autor: Marta Michalak-Pałosz

Data wykonania: 7 grudnia 2021 r.

Wersja 2.0

Tabela 1. Przypadki testowe „Status”

Lp.	Test	Warunki wstępne	Kroki wykonania	Oczekiwany rezultat
1.	Zwrócenie statusu API.	-	1. Wybranie typu requestu – GET. 2. Wysłanie requestu na adres: https://simple-books-api.glitch.me/status	Zwrócony zostaje response zawierający status API, w formacie JSON ze statusem 200.

Tabela 2. Przypadki testowe „List of books”

Lp.	Test	Warunki wstępne	Kroki wykonania	Oczekiwany rezultat
1.	Zwrócenie listy wszystkich książek.	-	1. Wybranie typu requestu – GET. 2. Wysłanie requestu na adres: https://simple-books-api.glitch.me/books	Zwrócony zostaje response zawierający listę wszystkich książek, w formacie JSON ze statusem 200.
2.	Zwrócenie listy wszystkich książek o wartości type=non-fiction	-	1. Wybranie typu requestu – GET. 2. Dodanie adresu requestu: https://simple-books-api.glitch.me/books 3. Wprowadzenie parametru type o wartości non-fiction. 4. Wysłanie requestu.	Zwrócony zostaje response zawierający listę wszystkich książek o wartości type=non-fiction, w formacie JSON ze statusem 200.
3.	Zwrócenie listy wszystkich książek z podaniem niepoprawnego typu.	-	1. Wybranie typu requestu – GET. 2. Dodanie adresu requestu: https://simple-books-api.glitch.me/books 3. Wprowadzenie parametru type o wartości lorem ipsum. 4. Wysłanie requestu.	Zwrócony zostaje response zawierający informację o błędnym parametrze, w formacie JSON ze statusem błędu.

4.	Zwrócenie książek o długości listy = 1 (parametr limit ustawiony na 1, możliwy przedział pomiędzy 1 a 20).	-	1. Wybranie typu requestu – GET. 2. Dodanie adresu requestu: https://simple-books-api.glitch.me/books 3. Wprowadzenie parametru limit o minimalnej wartości = 1. 4. Wysłanie requestu.	Zwrócony zostaje response zawierający listę z pojedynczą książką, w formacie JSON ze statusem 200.
5.	Zwrócenie książek o maksymalnej długości listy = 20 (parametr limit ustawiony na 20, możliwy przedział pomiędzy 1 a 20).	-	1. Wybranie typu requestu – GET. 2. Dodanie adresu requestu: https://simple-books-api.glitch.me/books 3. Wprowadzenie parametru limit o maksymalnej wartości = 20. 4. Wysłanie requestu.	Zwrócony zostaje response zawierający listę 20 książek, w formacie JSON ze statusem 200.
6.	Zwrócenie listy książek o długości spoza zakresu 1 – 20 (limit = 21).	-	1. Wybranie typu requestu – GET. 2. Dodanie adresu requestu: https://simple-books-api.glitch.me/books 3. Wprowadzenie parametru limit o wartości = 21, niemieszczącej się w zakresie 1 - 20. 4. Wysłanie requestu.	Zwrócony zostaje response zawierający informację o błędnym parametrze, w formacie JSON ze statusem błędu.
7.	Zwrócenie listy książek o długości spoza zakresu 1 – 20 (limit = 0).	-	1. Wybranie typu requestu – GET. 2. Dodanie adresu requestu: https://simple-books-api.glitch.me/books 3. Wprowadzenie parametru limit o wartości = 0, niemieszczącej się w zakresie 1 - 20. 4. Wysłanie requestu.	Zwrócony zostaje response zawierający informację o błędnym parametrze, w formacie JSON ze statusem błędu.

8.	Zwrócenie listy książek o błędnym parametrze limit = chomik (wartość parametru nie jest wartością liczbową).	-	1. Wybranie typu requestu – GET. 2. Dodanie adresu requestu: https://simple-books-api.glitch.me/books 3. Wprowadzenie parametru limit o błędnej wartości = chomik. 4. Wysłanie requestu.	Zwrócony zostaje response zawierający informację o błędnym parametrze, w formacie JSON ze statusem błędu.
9.	Zwrócenie listy książek bez podawania wartości parametru limit.	-	1. Wybranie typu requestu – GET. 2. Dodanie adresu requestu: https://simple-books-api.glitch.me/books 3. Pozostawienie pustego parametru limit. 4. Wysłanie requestu.	Zwrócony zostaje response zawierający informację o błędnym parametrze, w formacie JSON ze statusem błędu.
10.	Zwrócenie listy wszystkich książek o długości=2 i type=fiction.	-	1. Wybranie typu requestu – GET. 2. Dodanie adresu requestu: https://simple-books-api.glitch.me/books 3. Wprowadzenie parametru limit=2 i type=fiction. 4. Wysłanie requestu	Zwrócony zostaje response, zawierający listę maksymalne dwóch książek o wartości type=fiction, w formacie JSON ze statusem 200.

Tabela 3. Przypadki testowe „Get a single book”

Lp.	Test	Warunki wstępne	Kroki wykonania	Oczekiwany rezultat
1.	Zwrócenie informacji o książce o ID=2.	Baza danych API zawiera książkę o ID=2.	1. Wybranie typu requestu – GET. 2. Wysłanie requestu na adres: https://simple-books-api.glitch.me/books/2	Zwrócony zostaje response zawierający książkę o ID=2, w formacie JSON ze statusem 200.

2.	Zwrócenie informacji o książce o nieistniejącym ID=-5.	Baza danych API nie zawiera książki o ID=-5.	1. Wybranie typu requestu – GET. 2. Wysłanie requestu na adres: https://simple-books-api.glitch.me/books/-5	Zwrócony zostaje response zawierający informację o nieistniejącej książce o ID=-5, w formacie JSON ze statusem błędu.
3.	Zwrócenie informacji o książce o nieistniejącym ID=50.	Baza danych API nie zawiera książki o ID=50.	1. Wybranie typu requestu – GET. 2. Wysłanie requestu na adres: https://simple-books-api.glitch.me/books/50	Zwrócony zostaje response zawierający informację o nieistniejącej książce o ID=50, w formacie JSON ze statusem błędu.
4.	Zwrócenie informacji o książce o nieistniejącym ID=chomik.	Baza danych API nie zawiera książki o ID=chomik.	1. Wybranie typu requestu – GET. 2. Wysłanie requestu na adres: https://simple-books-api.glitch.me/books/chomik	Zwrócony zostaje response zawierający informację o nieistniejącej książce o ID=chomik, w formacie JSON ze statusem błędu.

Tabela 4. Przypadki testowe „Authorization and register API client”

Lp.	Test	Warunki wstępne	Kroki wykonania	Oczekiwany rezultat
1.	Zarejestrowanie API client i uzyskanie tokenu dostępu.	Brak zarejestrowanego API clienta na wybrany email.	1. Wybranie typu requestu – POST. 2. Dodanie adresu requestu: https://simple-books-api.glitch.me/books/api-clients 3. Wprowadzenie Body w formacie JSON podając poprawne, unikatowe wartości dla klucza: clientName i clientEmail. 4. Wysłanie requestu.	API client został zarejestrowany. Zwrócony zostaje response informujący o pomyślnym zarejestrowaniu API client oraz uzyskanym tokenie dostępu, w formacie JSON ze statusem 201.
2.	Ponowne zarejestrowanie API client na tego samego użytkownika.	Istnieje zarejestrowany API client na wybrany email.	1. Wybranie typu requestu – POST. 2. Dodanie adresu requestu: https://simple-books-api.glitch.me/books/api-clients	API client nie zostaje zarejestrowany. Zwrócony zostaje response zawierający informację o istniejącym użytkowniku, w formacie JSON ze statusem błędu.

			3. Wprowadzenie body w formacie JSON podając dane uprzednio zarejestrowanego API clienta: clientName i clientEmail. 4. Wysłanie requestu.	
3.	Zarejestrowanie API client bez podania nazwy.	-	1. Wybranie typu requestu – POST. 2. Dodanie adresu requestu: https://simple-books-api.glitch.me/books/api-clients 3. Wprowadzenie body w formacie JSON zawierającym nieuzupełnioną wartość dla clientName oraz clientEmail w poprawnym formacie. 4. Wysłanie requestu.	API client nie zostaje zarejestrowany. Zwrócony zostaje response zawierający informację o braku nazwy użytkownika, w formacie JSON ze statusem błędu.
4.	Zarejestrowanie API client z błędnym adresem email.	-	1. Wybranie typu requestu – POST. 2. Dodanie adresu requestu: https://simple-books-api.glitch.me/books/api-clients 3. Wprowadzenie body w formacie JSON podając poprawną wartość dla clientName oraz niepoprawną (np. bez @) dla clientEmail. 4. Wysłanie requestu.	API client nie zostaje zarejestrowany. Zwrócony zostaje response zawierający informację o braku adresu email użytkownika, w formacie JSON ze statusem błędu.
5.	Zarejestrowanie API client bez podania wartości.	-	1. Wybranie typu requestu – POST. 2. Dodanie adresu requestu: https://simple-books-api.glitch.me/books/api-clients 3. Wprowadzenie body w formacie JSON bez podania danych. 4. Wysłanie requestu.	API client nie zostaje zarejestrowany. Zwrócony zostaje response zawierający informację o braku danych, w formacie JSON ze statusem błędu.

Tabela 5. Przypadki testowe „Submit an order”

Lp.	Test	Warunki wstępne	Kroki wykonania	Oczekiwany rezultat
1.	Złożenie zamówienia bez uwierzytelnienia.	Baza danych API zawiera dostępną książkę o ID=1 oraz klienta.	1. Wybranie typu requestu – POST. 2. Dodanie adresu requestu: https://simple-books-api.glitch.me/books/orders 3. Wprowadzenie body w formacie JSON podając: bookID: 1, customerName (dowolne istniejące). 4. Wysłanie requestu.	Zamówienie nie zostało złożone. Zwrócony zostaje response informujący o braku autoryzacji, w formacie JSON ze statusem błędu.
2.	Złożenie zamówienia z uwierzytelnieniem.	Baza danych API zawiera dostępną książkę o ID=1. Użytkownik posiada poprawny Bearer Token do autoryzacji w API.	1. Wybranie typu requestu – POST. 2. Dodanie adresu requestu: https://simple-books-api.glitch.me/books/orders 3. Wprowadzenie body w formacie JSON podając: bookID: 1, customerName (dowolne istniejące). 4. Wprowadzenie tokenu w zakładce Authorization i wybranym type=Bearer Token. 5. Wysłanie requestu.	Zamówienie zostało złożone. Zwrócony zostaje response zawierający informację o ID złożonego zamówienia, w formacie JSON ze statusem 201.
3.	Złożenie zamówienia z podaniem niepoprawnego tokenu uwierzytelniającego.	Baza danych API zawiera dostępną książkę o ID=1.	1. Wybranie typu requestu – POST. 2. Dodanie adresu requestu: https://simple-books-api.glitch.me/books/orders 3. Wprowadzenie body w formacie JSON podając: bookID: 1, customerName (dowolne istniejące). 4. Wprowadzenie niepoprawnego tokenu w zakładce Authorization i wybranym type=Bearer Token 5. Wysłanie requestu.	Zamówienie nie zostało złożone. Zwrócony zostaje response informujący o braku autoryzacji z powodu niepoprawnego tokenu, w formacie JSON ze statusem błędu.

4.	Złożenie kolejnego zamówienia na tą samą książkę (z uwierzytelnieniem).	Baza danych API zawiera dostępną książkę o ID=1. Użytkownik posiada poprawny Bearer Token do autoryzacji w API.	1. Wybranie typu requestu – POST. 2. Dodanie adresu requestu: https://simple-books-api.glitch.me/books/orders 3. Wprowadzenie body w formacie JSON podając: bookID: 1 customerName (dowolne istniejące). 4. Wprowadzenie tokenu w zakładce Authorization i wybranym type=Bearer Token. 5. Wysłanie requestu.	Zamówienie zostało złożone. Zwrócony zostaje response zawierający informację o złożonym zamówieniu, w formacie JSON ze statusem 201.
5.	Złożenie zamówienia na niedostępną książkę (z uwierzytelnieniem).	Książka o ID=2 nie jest dostępna. Użytkownik posiada poprawny Bearer Token do autoryzacji w API.	1. Wybranie typu requestu – POST. 2. Dodanie adresu requestu: https://simple-books-api.glitch.me/books/orders 3. Wprowadzenie body w formacie JSON podając: bookID: 2, customerName (dowolne istniejące). 4. Wprowadzenie tokenu w zakładce Authorization i wybranym type=Bearer Token. 5. Wysłanie requestu.	Zamówienie nie zostało złożone. Zwrócony zostaje response zawierający informację o niedostępności książki, w formacie JSON z kodem błędu.
6.	Złożenie zamówienia na książkę o niewłaściwym ID (z uwierzytelnieniem).	Baza danych API nie zawiera książki o ID=chomik. Użytkownik posiada poprawny Bearer Token do autoryzacji w API.	1. Wybranie typu requestu – POST. 2. Dodanie adresu requestu: https://simple-books-api.glitch.me/books/orders 3. Wprowadzenie body w formacie JSON podając: bookID: chomik, customerName (dowolne istniejące). 4. Wprowadzenie tokenu w zakładce Authorization i wybranym type=Bearer Token. 5. Wysłanie requestu.	Zamówienie nie zostało złożone. Zwrócony zostaje response zawierający informację o niewłaściwym ID książki, w formacie JSON z kodem błędu.

Tabela 6. Przypadki testowe „Get all orders”

Lp.	Test	Warunki wstępne	Kroki wykonania	Oczekiwany rezultat
1.	Wyświetlenie wszystkich zamówień (z użyciem uwierzytelnienia).	Użytkownik posiada poprawny Bearer Token do autoryzacji w API.	1. Wybranie typu requestu – GET. 2. Dodanie adresu requestu: https://simple-books-api.glitch.me/orders 3. Wprowadzenie tokenu w zakładce Authorization i wybranym type=Bearer Token. 4. Wysłanie requestu.	Zwrócony zostaje response zawierający wszystkie zamówienia, w formacie JSON ze statusem 200.
2.	Wyświetlenie wszystkich zamówień (bez uwierzytelnienia).	-	1. Wybranie typu requestu – GET. 2. Dodanie adresu requestu: https://simple-books-api.glitch.me/orders 3. Wysłanie requestu.	Zwrócony zostaje response zawierający informację o braku uwierzytelnienia, w formacie JSON ze statusem błędu.
3.	Wyświetlenie wszystkich zamówień (z podaniem niepoprawnego tokenu uwierzytelnieniającego).	-	1. Wybranie typu requestu – GET. 2. Dodanie adresu requestu: https://simple-books-api.glitch.me/orders 3. Wprowadzenie niepoprawnego tokenu w zakładce Authorization i wybranym type=Bearer Token. 4. Wysłanie requestu.	Zwrócony zostaje response informujący o braku autoryzacji z powodu niepoprawnego tokenu, w formacie JSON ze statusem błędu.

Tabela 7. Przypadki testowe „Get an order”

Lp.	Test	Warunki wstępne	Kroki wykonania	Oczekiwany rezultat
1.	Wyświetlenie istniejącego zamówienia (z użyciem uwierzytelnienia).	Baza danych API zawiera zamówienie o przykładowym ID= PqvG8gFMJGlulrOFPahCl. Użytkownik posiada poprawny Bearer Token do autoryzacji w API.	1. Wybranie typu requestu – GET. 2. Dodanie adresu requestu: https://simple-books-api.glitch.me/orders/ PqvG8gFMJGlulrOFPahCl 3. Wprowadzenie tokenu w zakładce Authorization i wybranym type=Bearer Token. 4. Wysłanie requestu.	Zamówienie zostało wyświetlone. Zwrócony zostaje response zawierający informację o zamówieniu, w formacie JSON ze statusem 200.
2.	Wyświetlenie istniejącego zamówienia (bez uwierzytelnienia).	Baza danych API zawiera zamówienie o przykładowym ID= PqvG8gFMJGlulrOFPahCl.	1. Wybranie typu requestu – GET. 2. Dodanie adresu requestu: https://simple-books-api.glitch.me/orders/ PqvG8gFMJGlulrOFPahCl 3. Wysłanie requestu.	Zamówienie nie zostało wyświetlone. Zwrócony zostaje response zawierający informację o braku uwierzytelnienia, w formacie JSON ze statusem błędu.
3.	Wyświetlenie istniejącego zamówienia (z podaniem niepoprawnego tokenu uwierzytelnieniającego).	Baza danych API zawiera zamówienie o przykładowym ID= PqvG8gFMJGlulrOFPahCl.	1. Wybranie typu requestu – GET. 2. Dodanie adresu requestu: https://simple-books-api.glitch.me/orders/ PqvG8gFMJGlulrOFPahCl 3. Wprowadzenie niepoprawnego tokenu w zakładce Authorization i wybranym type=Bearer Token. 4. Wysłanie requestu.	Zamówienie nie zostało wyświetlone. Zwrócony zostaje response informujący o braku autoryzacji z powodu niepoprawnego tokenu, w formacie JSON ze statusem błędu.

4.	Wyświetlenie nieistniejącego zamówienia (z użyciem uwierzytelnienia).	Baza danych API nie zawiera zamówienia o ID= chomik. Użytkownik posiada poprawny Bearer Token do autoryzacji w API.	1. Wybranie typu requestu – GET. 2. Dodanie adresu requestu: https://simple-books-api.glitch.me/orders/chomik 3. Wprowadzenie tokenu w zakładce Authorization i wybranym type=Bearer. 4. Wysłanie requestu.	Zwrócony zostaje response zawierający informację o nieistniejącym zamówieniu, w formacie JSON ze statusem błędu.

Tabela 8. Przypadki testowe „Update an order”

Lp.	Test	Warunki wstępne	Kroki wykonania	Oczekiwany rezultat
1.	Zaktualizowanie istniejącego zamówienia (z użyciem uwierzytelnienia).	Baza danych API zawiera zamówienie o przykładowym ID= PqvG8gFMJGlulrOFPahCl. Użytkownik posiada poprawny Bearer Token do autoryzacji w API.	1. Wybranie typu requestu – PATCH. 2. Dodanie adresu requestu: https://simple-books-api.glitch.me/orders/ PqvG8gFMJGlulrOFPahCl 3. Wprowadzenie tokenu w zakładce Authorization i wybranym type=Bearer Token. 4. Wprowadzenie nowej nazwy klienta w body w formacie JSON (customerName). 5. Wysłanie requestu.	Zamówienie zostało zaktualizowane. Zwrócony zostaje response ze statusem 204.
2.	Zaktualizowanie istniejącego zamówienia (bez uwierzytelnienia).	Baza danych API zawiera zamówienie o przykładowym ID= PqvG8gFMJGlulrOFPahCl.	1. Wybranie typu requestu – PATCH. 2. Dodanie adresu requestu: https://simple-books-api.glitch.me/orders/ PqvG8gFMJGlulrOFPahCl 3. Wprowadzenie nowej nazwy klienta w body w formacie JSON: (customerName). 4. Wysłanie requestu.	Zamówienie nie zostało zaktualizowane. Zwrócony zostaje response zawierający informację o braku uwierzytelnienia, w formacie JSON ze statusem błędu.

3.	Zaktualizowanie istniejącego zamówienia (z podaniem niepoprawnego tokenu uwierzytelniającego).	Baza danych API zawiera zamówienie o ID= PqvG8gFMJGlulrOFPahCl.	1. Wybranie typu requestu – PATCH. 2. Dodanie adresu requestu: https://simple-books-api.glitch.me/orders/ PqvG8gFMJGlulrOFPahCl 3. Wprowadzenie niepoprawnego tokenu w zakładce Authorization i wybranym type=Bearer Token. 4. Wprowadzenie nowej nazwy klienta w body w formacie JSON dla customerName. 5. Wysłanie requestu.	Zamówienie nie zostało zaktualizowane. Zwrócony zostaje response informujący o braku autoryzacji z powodu niepoprawnego tokenu, w formacie JSON ze statusem błędu.
4.	Zaktualizowanie nieistniejącego zamówienia (z użyciem uwierzytelnienia).	Baza danych API nie zawiera zamówienia o ID=5. Użytkownik posiada poprawny Bearer Token do autoryzacji w API.	1. Wybranie typu requestu – PATCH. 2. Dodanie adresu requestu: https://simple-books-api.glitch.me/orders/5 3. Wprowadzenie tokenu w zakładce Authorization i wybranym type=Bearer Token. 4. Wprowadzenie nowej nazwy klienta w body w formacie JSON dla customerName. 5. Wysłanie requestu.	Zamówienie nie zostało zaktualizowane. Zwrócony zostaje response zawierający informację o nieistniejącym ID, w formacie JSON ze statusem błędu.

Tabela 9. Przypadki testowe „Delete an order”

Lp.	Test	Warunki wstępne	Kroki wykonania	Oczekiwany rezultat
1.	Usunięcie istniejącego zamówienia bez uwierzytelnienia.	Baza danych API zawiera zamówienie o przykładowym ID= VD88zm6jBkKijnM3pZ54G	1. Wybranie typu requestu – DELETE. 2. Dodanie adresu requestu: https://simple-books-api.glitch.me/orders/ VD88zm6jBkKijnM3pZ54G 3. Wysłanie requestu.	Zamówienie nie zostaje usunięte. Zwrócony zostaje response zawierający informację o braku uwierzytelnienia, w formacie JSON ze statusem błędu.

2.	Usunięcie istniejącego zamówienia z podaniem niepoprawnego tokenu uwierzytelniającego.	Baza danych API zawiera zamówienie o danym ID.	1. Wybranie typu requestu – DELETE. 2. Dodanie adresu requestu: https://simple-books-api.glitch.me/orders/VD88zm6jBkKijnM3pZ54G 3. Wprowadzenie niepoprawnego tokenu w zakładce Authorization i wybranym type=Bearer Token. 4. Wysłanie requestu.	Zamówienie nie zostaje usunięte. Zwrócony zostaje response informujący o braku autoryzacji z powodu niepoprawnego tokenu, w formacie JSON ze statusem błędu.
3.	Usunięcie istniejącego zamówienia z użyciem uwierzytelnienia.	Baza danych API zawiera zamówienie o przykładowym ID= VD88zm6jBkKijnM3pZ54G Użytkownik posiada poprawny Bearer Token do autoryzacji w API.	1. Wybranie typu requestu – DELETE. 2. Dodanie adresu requestu: https://simple-books-api.glitch.me/orders/VD88zm6jBkKijnM3pZ54G 3. Wprowadzenie tokenu w zakładce Authorization i wybranym type=Bearer Token. 4. Wysłanie requestu.	Zamówienie zostaje usunięte. Zwrócony zostaje response zawierający informację o usuniętym zamówieniu, w formacie JSON ze statusem 200.
4.	Ponowne usunięcie tego samego zamówienia (z użyciem uwierzytelnienia).	Baza danych API nie zawiera zamówienia o przykładowym ID= VD88zm6jBkKijnM3pZ54G Użytkownik posiada poprawny Bearer Token do autoryzacji w API.	1. Wybranie typu requestu – DELETE. 2. Dodanie adresu requestu: https://simple-books-api.glitch.me/orders/VD88zm6jBkKijnM3pZ54G 3. Wprowadzenie tokenu w zakładce Authorization i wybranym type=Bearer Token. 4. Wysłanie requestu.	Zwrócony zostaje response zawierający informację o braku zamówienia o danym ID, w formacie JSON ze statusem błędu.