testy API - gorest.co.in

	Test Case	propozycja opisu	osoba	priority	progress	type	level	resources	method	expected result	link
TC-01	Tworzenie użytkownika o takim samym adresie email (o zajętym już adresie email)	As an authenticated api consumer I **cannot** enter an already used email (POST)	Paweł	0	Pass	functional	unit	user	post	User shouldn't be created, error output	
TC-02	Tworzenie użytkownika o niepoprawnym adresie email	As an authenticated api consumer I **cannot** enter an email that doesn't pass the email regex (POST)	Paweł	0	Pass	functional	unit	user	post	User shouldn't be created, error output	
TC-03	Tworzenie użytkownika o poprawnym adresie email	As an authenticated api consumer I can create an user with a valid email that hasn't been used before (POST)	Paweł	0	Pass	functional	unit	user	post	User should be created	
TC-04	Dodaj użytkownika, który już istnieje	As an authenticated api consumer I **cannot** create an user that has the same id with an already existing user (POST)	Szymon	0	Pass	functional	unit	user	post	User shouldn't be created, error output	
TC-05	Utworzenie użytkownika z określonym ID	As an authenticated api consumer I **cannot** create an user with a specified ID (POST)	Ola	0	Pass	functional	unit	user	post	User should be created, but with independent ID	https://www.codepile.net/pile/py3Ypn5l
TC-06	Podmień całego użytkownika	As an authenticated api consumer I can replace the entire user (PUT)	Szymon	0	In progress	functional	unit	user	put	User should be replaced	
TC-07	Podmień niektóre dane użytkownika	As an authenticated api consumer I can replace the entire user (PATCH)	Szymon	0	In progress	functional	unit	user	patch	Some user data should be replaced	
TC-08	Pobierz wszystkich użytkowników	As an **anonymous** and authenticated api consumer I can get first page of users (GET)	Szymon	0	Pass	functional	unit	user	get	A list of all users should be displayed	
TC-09	Pobierz wszystkich z konkretnym pierw	As an **anonymous** api consumer I can all users with a certain name (GET)	Szymon	0	Pass	functional	unit	user	get	User with a specific first name should be displayed	
TC-10	Pobierz pierwszą stronę użytkowników jako użytkownik bez tokena	As an **anonymous** api consumer I can get first page of users (GET)	Magda	1	Pass	functional	unit	user	get	I see first page of users	
TC-11	Pobierz wybraną stronę użytkowników jako użytkownik bez tokena	As an **anonymous** api consumer I can get a given page of users (GET)	Magda	1	Pass	functional	unit	user	get	I see page x of users	
TC-12	Usuwanie użytkownika	As an authenticated api consumer I can delete user (DELETE)	Szymon	0	Pass	functional	unit	user	delete	User should be deleted	
TC-13	Usuwanie użytkownika, który nie istnieje	As an authenticated api consumer I **cannot** delete not existing user (DELETE)	Ola	0	Pass	functional	unit	user	delete	Error message	
TC-14	Nie można usunąć użytkownika bez tokena	As an **anonymous** api consumer I **cannot** delete a given user (DELETE)	Magda	1	Pass	functional	unit	user	delete	You should not be able to delete an user	
TC-15	Utworzenie elementu listy to-do	As an authenticated api consumer I can create a user todo (POST)	Ola	0	Pass	functional	unit	todo	post	Todo should be created	
TC-16	Tworzenie elementu listy to-do przez użytkownika, który nie istnieje	As an authenticated api consumer I **cannot** create a todo of non-existing user (POST)	Ola	1	Pass	functional	unit	todo	post	Todo shouldn't be created	
TC-17	Pobranie listy to-do	As an authenticated api consumer I can retrieve a user todo (GET)	Ola	0	Pass	functional	unit	todo	get	Todo should be retrieved	
TC-18	Podmień składnik elementu listy to-do	As an authenticated api consumer I can update an user todo (PATCH)	Ola	1	Pass	functional	unit	todo	patch	Some user todo elements should be replaced	
TC-19	Podmień cały element listy to-do	As an authenticated api consumer I can update an user todo (PUT)	Ola	0	Pass	functional	unit	todo	put	Todo should be replaced	
TC-20	Przeglądanie wybranej strony list to-do jako użytkownik bez tokena	As an **anonymous** api consumer I can get first page of todos (GET)	Paweł	1	Pass	functional	unit	todo	get	I see first page of to-dos	
TC-21	Usunięcie nieistniejącego elementu listy to-do	As an authenticated api consumer I **cannot** delete an unexisting user todo (DELETE)	Ola	1	Pass	functional	unit	todo	delete	Error message	
TC-22	Usunięcie elementu listy to-do	As an authenticated api consumer I can delete a user todo (DELETE)	Ola	0	Pass	functional	unit	todo	delete	Todo should be deleted	
TC-23	Tworzenie nowego posta	As an authenticated api consumer I can create new user post (POST)	Magda	0	Pass	functional	unit	post	post	Post should be created	
TC-24	Tworzenie bardzo długiego posta (np cały pan tadeusz)	As an authenticated api consumer I can create very large post (POST)	Paweł	0	Pass	functional	unit	post	post	Post shouldn't be created, error output	
TC-25	Aktualizacja istniejącego posta	As an authenticated api consumer I can update already existing post (PATCH)	Magda	0	Pass	functional	unit	post	patch	Post should be updated	
TC-26	Aktualizacja istniejącego posta przez innego użytkownika	As an authenticated api consumer I **cannot** post update not by the author (PATCH)	Paweł	1	Fail	functional	unit	post	patch	Post shouldn't be updated, error output	
TC-27	Wyświetlanie wszystkich postów	As an authenticated api consumer I can display all users posts (GET)	Magda	0	Pass	functional	unit	post	get	All posts should be displayed	
TC-28	Przeglądanie pierwszej strony postów jako użytkownik bez tokena	As an **anonymous** api consumer I can get first page of posts (GET)	Paweł	1	Pass	functional	unit	post	get	I see first page of posts	
TC-29	Usunięcie posta	As an authenticated api consumer I can delete user post (DELETE)	Magda	0	Pass	functional	unit	post	delete	Post should be deleted	
TC-30	Tworzenie nowego komentarza	As an authenticated api consumer I can create new user comment (POST)	Monika	0	Pass	functional	unit	comment	post	Comment should be created	
TC-31	Aktualizacja istniejącego komentarza	As an authenticated api consumer I can update already existing comment (PATCH)	Monika	0	Pass	functional	unit	comment	patch	Comment should be updated	
TC-32	Aktualizacja istniejącego komentarza p	As an authenticated api consumer I **cannot** update comment update not by the author (PATCH)	Monika	2	Fail	non-functional	unit	comment	patch	Comment shouldn't be updated, error output	
TC-33	Wyświetlanie wszystkich komentarzy	As an authenticated api consumer I can display all users comments (GET)	Monika	0	Pass	functional	unit	comment	get	All comments should be displayed	
TC-34	Przeglądanie pierwszej strony komentarzy jako użytkownik bez tokena	As an **anonymous** api consumer I can get first page of comments (GET)	Monika	0	Pass	functional	unit	comment	get	I see first page of comments	
TC-35	Usunięcie komentarza	As an authenticated api consumer I can delete user comment (DELETE)	Monika	0	Pass	functional	unit	comment	delete	Comment should be deleted	
TC-36	Test efektywności	Api response takes less than 200 ms	Ola	1	Fail	non-functional	system	efficiency		Response should take under 200 ms	
TC-37	Test przeciążeniowy - za dużo requestów na raz	Api doesn't allow to send 100 requests one after another	Monika	1	Fail	non-functional	system	security		429: Too many requests. The request was rejected due to rate limiting.	
TC-38	Pen test	As an **anonymous** api consumer I cannot create an user inserting a script instead of a name		1	Ready to test	non-functional	system	security		User shouldn't be created	
TC-39	Test integracyjny	As an authenticated api consumer I can 1. Create new user (POST) 2. Create new user post (POST) 3. Delete just created user (DELETE)	Magda	1	Pass	functional	integration	multiple	multiple	All steps should be executed	
TC-40	Test integracyjny	As an authenticated api consumer I can 1. Create new user (POST) 2. Update name and email (PATCH) 3. Create a post (POST) 4. Create a comment (POST) 5. Update a post (PUT) 6. Update a comment (PUT)	Monika	1	Pass	functional	integration	multiple	multiple	All steps should be executed	
TC-41	Test integracyjny	As an authenticated api consumer I can 1. Retrieve an user created by another person (in a different test) based on an ID (GET) 3. Delete the existing user by ID(DELETE)		1	Ready to test	functional	integration	multiple	multiple	All steps should be executed	
TC-42	Test integracyjny	As an authenticated api consumer I can 1. Create new user (POST) 2. Create new user todo (POST) 3. Update user todo (PUT) 4. Delete a user todo (DELETE)	Ola	1	Pass	functional	integration	multiple	multiple	All steps should be executed	
TC-43	Test obciążeniowy	As an authenticated api consumer I can 1. Create 50 new comments (POST)	Monika	1	Pass	non-functional	system	efficiency		All the comments should be created	
TC-44	Pobierz użytkownika stworzonego poprzedniego dnia	As an authenticated api I can retrieve an user with specific ID created at least 12 hours earlier (GET)		1	Ready to test	functional	unit	get	user	I should be able to retrieve an user	# baza jest resetowana co jakiś czas, więc chyba ciężko to będzie przetestować
TC-45	Komentarz powyżej 1000 znaków	As an authenticated api consumer I cannot create new user comment longer than 1000 signs (POST)	Monika		Pass		unit				
TC-46	Aktualizacja istniejącego posta	As an authenticated api consumer I can update already existing post (PUT)	Magda		Pass	functional	unit	put	comment		
TC-47	Aktualizacja istniejącego komentarza	As an authenticated api consumer I can update already existing comment (PUT)			Ready to test	functional	unit	put	comment		
			Monika	2	Assigned		security				
				2			parsing				
				2			efficiency				
				2			multiplot				
				2			reliability				
				2			incomplete data				