Окружение: Postman v11.33.4								
ID	**Приоритет**	**Название**	**Предусловия**	**Тестовые данные**	**Шаги**	**Ожидаемый результат**	**Статус прохождения Passed or Failed**	**Комментарий**
Позитивное тестирование API								
PT1	Medium	Получить данные из пустой таблицы (Вызов метода GET)	Headers request: Content-Type: application/json Authorization: Type: OAuth 2.0 Body: grand_type: client_credentials client_id: psql-api client_secret: "your_client_secret" Отправить POST запрос http://localhost:4000/realms/master/protocol/openid-connect/token, скопировать access_token Использовать при запросах в Authorization: Type: bearer token, token: "your_access_token" Отправить DELETE запрос http://localhost:5001/data для очистки БД перед выполнением тестов		Отправить GET запрос http://localhost:5001/data	Запрос успешно инициирован	Passed	
					Проверить код состояния	Status: 200 OK	Passed	
					Проверить тело ответа от сервера	Response body: []	Passed	

PT2	**High**	Добавить данные в таблицу (Вызов метода POST)	Headers request:
			Content-Type: application/json
			
			Authorization:
			Type: OAuth 2.0
			Body:
			grand_type: client_credentials
			client_id: psql-api
			client_secret: "your_client_secret"
			Отправить POST запрос http://localhost:4000/realms/master/protocol/openid-connect/token, скопировать access_token
			Использовать при запросах в Authorization:
			Type: bearer token, token: "your_access_token"
				Body -> Raw -> JSON
				
				{
				"text": "Test string"
				}
					Отправить POST запрос http://localhost:5001/data	Запрос успешно инициирован	Passed	

					Проверить код состояния	Status: 200 OK	Passed	
					Проверить тело ответа от сервера	Response body: Data added successfully and sent to Kafka.	Passed	
PT3	**High**	Получить данные из таблицы после добавления (Вызов метода GET)	Headers request: Content-Type: application/json Authorization: Type: OAuth 2.0 Body: grand_type: client_credentials client_id: psql-api client_secret: "your_client_secret" Отправить POST запрос http://localhost:4000/realms/master/protocol/openid-connect/token, скопировать access_token Использовать при запросах в Authorization: Type: bearer token, token: "your_access_token"		Отправить GET запрос http://localhost:5001/data	Запрос успешно инициирован	Passed	
					Проверить код состояния	Status: 200 OK	Passed	

					Проверить тело ответа от сервера	Response body:
						
						[
						{
						"id": 1,
						"text": "Test string",
						"time": YYYY-MM-DDTHH:mm:ss.SSSSSS+HH:MM
						}
						]
							Passed	

PT4	**High**	Обновление данных по ID (Вызов метода PUT)	Headers request:
			Content-Type: application/json
			
			Authorization:
			Type: OAuth 2.0
			Body:
			grand_type: client_credentials
			client_id: psql-api
			client_secret: "your_client_secret"
			Отправить POST запрос http://localhost:4000/realms/master/protocol/openid-connect/token, скопировать access_token
			Использовать при запросах в Authorization:
			Type: bearer token, token: "your_access_token"
				Body -> Raw -> JSON
				
				{
				"text": "Updated string"
				}
					Отправить PUT запрос http://localhost:5001/data/1	Запрос успешно инициирован	Passed	

					Проверить код состояния	Status: 200 OK	Passed	
					Проверить тело ответа от сервера	Response body: Data updated successfully and sent to Kafka.	Passed	

Окружение: Postman v11.33.4								
ID	**Приоритет**	**Название**	**Предусловия**	**Тестовые данные**	**Шаги**	**Ожидаемый результат**	**Статус прохождения Passed or Failed**	**Комментарий**
PT5	**High**	Получение обновленных данных из таблицы / Получение данных по ID (Вызов метода GET)	Headers request: Content-Type: application/json Authorization: Type: OAuth 2.0 Body: grand_type: client_credentials client_id: psql-api client_secret: "your_client_secret" Отправить POST запрос http://localhost:4000/realms/master/protocol/openid-connect/token, скопировать access_token Использовать при запросах в Authorization: Type: bearer token, token: "your_access_token"		Отправить GET запрос http://localhost:5001/data/1	Запрос успешно инициирован	Passed	
					Проверить код состояния	Status: 200 OK	Passed	

					Проверить тело ответа от сервера	Response body:
						
						[
						{
						"id": 1,
						"text": "Updated string",
						"time": YYYY-MM-DDTHH:mm:ss.SSSSSS+HH:MM
						}
						]
							Passed	

PT6	**High**	Удаление данных по ID (Вызов метода DELETE)	Headers request: Content-Type: application/json Authorization: Type: OAuth 2.0 Body: grand_type: client_credentials client_id: psql-api client_secret: "your_client_secret" Отправить POST запрос http://localhost:4000/realms/master/protocol/openid-connect/token, скопировать access_token Использовать при запросах в Authorization: Type: bearer token, token: "your_access_token"		Отправить DELETE запрос http://localhost:5001/data/1	Запрос успешно инициирован	Passed	
					Проверить код состояния	Status: 200 OK	Passed	
					Проверить тело ответа от сервера	Response body: Data deleted successfully and delete event sent to Kafka.	Passed	
PT7	Medium	Получение данных из таблицы после удаления данных по ID (Вызов метода GET)	Headers request: Content-Type: application/json Authorization: Type: OAuth 2.0 Body: grand_type: client_credentials client_id: psql-api client_secret: "your_client_secret" Отправить POST запрос http://localhost:4000/realms/master/protocol/openid-connect/token, скопировать access_token Использовать при запросах в Authorization: Type: bearer token, token: "your_access_token"		Отправить GET запрос http://localhost:5001/data	Запрос успешно инициирован	Passed	
					Проверить код состояния	Status: 200 OK	Passed	
					Проверить тело ответа от сервера	Response body: []	Passed	
PT8	Medium	Удаление всех данных из таблицы (Вызов метода DELETE)	Headers request: Content-Type: application/json Authorization:		Отправить DELETE запрос http://localhost:5001/data	Запрос успешно инициирован	Passed	
					Проверить код состояния	Status: 200 OK	Passed	

Окружение: Postman v11.33.4							
ID	**Приоритет**	**Название**	**Предусловия**	**Тестовые данные**	**Шаги**	**Ожидаемый результат**	**Статус прохождения Passed or Failed**	**Комментарий**

			Authorization:
			Type: OAuth 2.0
			Body:
			grand_type: client_credentials
			client_id: psql-api
			client_secret: "your_client_secret"
			Отправить POST запрос http://localhost:4000/realms/master/protocol/openid-connect/token, скопировать access_token
			Использовать при запросах в Authorization:
			Type: bearer token, token: "your_access_token"
			
			Выполнить несколько POST запросов http://localhost:5001/data для наполнения таблицы с телом
			{
			"text": "Test string"
			}
					Проверить тело ответа от сервера	Response body:
						
						Table cleared and sequence reset.
							Passed	

PT9	Medium	Получение данных после удаления всех данных из таблицы (Вызов метода GET)	Headers request: Content-Type: application/json Authorization: Type: OAuth 2.0 Body: grand_type: client_credentials client_id: psql-api client_secret: "your_client_secret" Отправить POST запрос http://localhost:4000/realms/master/protocol/openid-connect/token, скопировать access_token Использовать при запросах в Authorization: Type: bearer token, token: "your_access_token"		Отправить GET запрос http://localhost:5001/data	Запрос успешно инициирован	Passed	
					Проверить код состояния	Status: 200 OK	Passed	
					Проверить тело ответа от сервера	Response body: []	Passed	

PT10	Low	Добавление строки длиной 100 символов (Вызов метода POST)	Headers request:
			Content-Type: application/json
			
			Authorization:
			Type: OAuth 2.0
			Body:
			grand_type: client_credentials
			client_id: psql-api
			client_secret: "your_client_secret"
			Отправить POST запрос http://localhost:4000/realms/master/protocol/openid-connect/token, скопировать access_token
			Использовать при запросах в Authorization:
			Type: bearer token, token: "your_access_token"
				Body -> Raw -> JSON
				
				{
				"text": "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
				}
					Отправить POST запрос http://localhost:5001/data	Запрос успешно инициирован	Passed	

					Проверить код состояния	Status: 200 OK	Passed	
					Проверить тело ответа от сервера	Response body: Data added successfully and sent to Kafka.	Passed	
PT11	Low	Добавление строк из всех возможных символов (Вызов метода POST)	Headers request: Content-Type: application/json	Body -> Raw -> JSON Последовательно выполнить два	Отправить POST запрос http://localhost:5001/data	Запрос успешно инициирован	Passed	
					Проверить код состояния	Status: 200 OK	Passed	

Окружение: Postman v11.33.4								
ID	**Приоритет**	**Название**	**Предусловия**	**Тестовые данные**	**Шаги**	**Ожидаемый результат**	**Статус прохождения Passed or Failed**	**Комментарий**

		[illegible]	Authorization:
			Type: OAuth 2.0
			Body:
			grand_type: client_credentials
			client_id: psql-api
			client_secret: "your_client_secret"
			Отправить POST запрос http://localhost:4000/realms/master/protocol/openid-connect/token, скопировать access_token
			Использовать при запросах в Authorization:
			Type: bearer token, token:
			"your_access_token"
				[illegible] запроса:
				
				{
				"text": "~!@#$%^&*()_+[]{}|\\:;'<>?,./-
				=0123456789QWERTYUIOPASDFGHJKLZXCVBNMqwertyuiopasdfghjklzxcvbnm"
				}
				
				{
				"text":
				"ЁЙЦУКЕНГШЩЗХЪФЫВАПРОЛДЖЭЯЧСМИТЬБ\"
				Юёйцукенгшщзхъфывапролджэячсмитьбю"
				}
					Проверить тело ответа от сервера	Response body:
						
						Data added successfully and sent to Kafka.
							Passed	

PT12	Low	Проверка сброса sequence после полной очистки таблицы	Headers request:
			Content-Type: application/json
			
			Authorization:
			Type: OAuth 2.0
			Body:
			grand_type: client_credentials
			client_id: psql-api
			client_secret: "your_client_secret"
			Отправить POST запрос http://localhost:4000/realms/master/protocol/openid-connect/token, скопировать access_token
			Использовать при запросах в Authorization:
			Type: bearer token, token:
			"your_access_token"
			
			Выполнить несколько POST запросов http://localhost:5001/data для наполнения таблицы с телом
			{
			"text": "Test string"
			}
			
			Данные для подключения к PSQL:
			
			-h localhost -p 5432 -U postgres -d postgres
			password: whatislove
				Body -> Raw -> JSON
				
				{
				"text": "Test string"
				}
				
				SQL-запрос:
				
				SELECT last_value FROM data_study_id_seq;
					Отправить DELETE запрос http://localhost:5001/data	Запрос успешно инициирован
						Status: 200 OK
						Response body: Table cleared and sequence reset.
							Passed	

					Отправить POST запрос http://localhost:5001/data	Запрос успешно инициирован Status: 200 OK Response body: Data added successfully and sent to Kafka.	Passed	
					Вручную подключиться к PostgreSQL и выполнить SQL-запрос для проверки last_value последовательности	Значение last_value в sequence равно 1	Passed	
					Отправить POST-запрос на http://localhost:5001/data	Запрос успешно инициирован Status: 200 OK Response body: Data added successfully and sent to Kafka.	Passed	

					Отправить GET запрос http://localhost:5001/data	Запрос успешно инициирован
						Status: 200 OK
						Response body:
						[
						{
						"id": 1,
						"text": "Test string",
						"time": YYYY-MM-DDTHH:mm:ss.SSSSSS+HH:MM
						}
						]
							Passed	