Задание 3 (Повторная аттестация)

Составленные API-тест-кейсы по методам для работы с видео YouTube Для начала работы с методами нам потребовалось привязать к Google-почте разработческий профиль, создать API key и access token

Nº	Название	Тестовые данные	Шаги	Ожидаемый результат	Приоритет	Статус			
Позит	Позитивные функциональные АРІ тест-кейсы								
1	Получить список видео по id jifUJrYPZQQ		1. Отправить Get запрос https://www.googleapis.com/yo utube/v3/videos?id={{id}}}	Запрос успешно отправлен на сервер					
			 Проверить код ответа Проверить название Проверить скорость ответа Проверить Response Size 	HTTP запрос - 200 OK Headers request: Content type: application / json Accept: application / json Ответ пришел в разумные ожидаемые сроки (695 ms Ответ пришел в разумных размерах (683 B)	Passed	Высокий			
			6. Проверить тело ответа от сервера на соответствие документации	Тело ответа возвращается и имеет вид, соответствующий документации					
2	Поставить like видео c id jifUJrYPZQQ	Тип запроса : https://www.googleapis.com/y outube/v3/videos/rate?id=jifU JryPZQQ&rating=like	1. Отправить POST запрос https://www.googleapis.com/yo utube/v3/videos/rate?id=jifUJrY PZQQ&rating=like	Запрос успешно отправлен на сервер					
			2. Проверить код ответа	HTTP запрос - 204 ОК					
			3. Проверить название	Headers request:					
				Content type: application / json					
		Раздел Tests var key = "issue_number" var value = pm.response.json().number		Accept: application / json	Passed	Высокий			
			4. Проверить скорость ответа	Ответ пришел в разумные ожидаемые сроки					
			5. Проверить отображение UI	В репозитории отображается созданная задача с заданными в запросе параметрами (название, автор и т.д.)					
			6. Проверить тело ответа от сервера на соответствие документации	Тело ответа возвращается и имеет вид, соответствующий документации					

3	Получить список самых популярных видео	https://youtube.googleapis.com/ youtube/v3/videos?part=snippet	ode=US 2. Проверить код ответа 3. Проверить название	Запрос успешно отправлен на сервер HTTP запрос - 200 OK Headers request: Content type: application / json Accept: application / json Ответ пришел в разумные ожидаемые сроки Тело ответа возвращается и имеет вид, соответствующий документации	Passed	Высокі
4	Проверить, что API вернул корректный список видео, отсортированный по популярности.	Тип запроса : GET Запросом https://	1. Отправить GET-запрос к API с параметром user_id идентификатора пользователя 2. Проверить код ответа 3. Проверить название	Получен успешный ответ от API с списком рекомендованных видео для указанного пользователя HTTP запрос - 200 OK Headers request: Content type: application / json Accept: application / json	Passed	Высок
			5. Проверить тело ответа от сервера на соответствие документации	Ответ пришел в разумные ожидаемые сроки Тело ответа возвращается и имеет вид, соответствующий документации		
5	Получить информацию о пользователе, загрузившем видео с id jifUJrYPZQQ	Тип запроса: GET Запрос: <u>https://</u>	1. Отправить GET запрос к API с параметром id=jifUJrYPZQQ и action=user 2. Проверить код ответа 3. Проверить название	Запрос успешно отправлен на сервер. Получен успешный ответ от API с информацией о пользователе, загрузившем видео HTTP запрос - 200 OK Headers request:		

		Тело запроса в формате Json { "id": "" }		Content type: application / json	Passed	Высокий
			4	Accept: application / json		
				Ответ пришел в разумные ожидаемые сроки		
			5. Проверить отооражение Ог	В репозитории отображается созданная задача с заданными в запросе параметрами (название, автор и т.д.)		
			6. Проверить тело ответа от	Тело ответа возвращается и имеет вид,		
			сервера на соответствие документации	соответствующий документации		
6	Добавить	Тип запроса: POST	1. Отправить POST-запрос к	Получен успешный ответ от АРІ, подтверждающий		
	комментарий к видео	Запрос: <u>https://</u>	АРІ с параметрами	успешное добавление комментария		
	c id jifUJrYPZQQ		id=jifUJrYPZQQ и текстом			
			комментария "Супер"			
		Тело запроса в формате Json { "id": "jifUJrYPZQQ", "body": "Супер", "assignee": "MiraS75",	2. Проверить код ответа	НТТР запрос - 204 ОК	Passed	Высокий
		ĺ				
			3. Проверить название	Headers request:		
				Content type: application / json		
			A Flancasian and a second	Accept: application / json		
1			4. проверить скорость ответа	Ответ пришел в разумные ожидаемые сроки		
			Б. Проворить тово ответе от	Тело ответа возвращается и имеет вид,		
			5. Проверить тело ответа от	соответствующий документации		
			сервера на соответствие			
	l		документации			

7	Получить список рекомендуемых видео для пользователя с определенным id	Тип запроса : GET Запрос: https://	АРІ с параметром user_id идентификатора пользователя 2. Проверить код ответа 3. Проверить название 4. Проверить скорость ответа 5. Проверить тело ответа от сервера на соответствие документации	Получен успешный ответ от API с списком рекомендованных видео для указанного пользователя HTTP запрос - 200 OK Headers request: Content type: application / json Ассерt: application / json Ответ пришел в разумные ожидаемые сроки Тело ответа возвращается и имеет вид, соответствующий документации	Passed	Высокий
8	Изменить описание видео с id jifUJrYPZQQ	Тип запроса : PUT Запрос: <u>https://</u>		Получен успешный ответ от API, подтверждающий успешное изменение описания видео HTTP запрос - 200 ОК Headers request: Content type: application / json Ассерt: application / json Ответ пришел в разумные ожидаемые сроки Тело ответа возвращается и имеет вид, соответствующий документации	Passed	Высокий
9	Удалить комментарий к видео с id jifUJrYPZQQ	Тип запроса : DELETE Запрос: <u>https://</u>	Отправить DELETE-запрос к API с параметрами id=jifUJrYPZQQ и id комментария Проверить код ответа Проверить название	Получен успешный ответ от API, подтверждающий успешное удаление комментария HTTP запрос - 200 ОК Headers request: Content type: application / json Accept: application / json Ответ пришел в разумные ожидаемые сроки В репозитории отображается созданная задача с заданными в запросе параметрами (название, автор и т.д.)	Passed	Высокий

			6. Проверить тело ответа от сервера на соответствие документации	Тело ответа возвращается и имеет вид, соответствующий документации		
Негат	ивные функциональн	ые API тест-кейсы				
10	Передать некорректный		1. Отправить Get запрос https://	Запрос успешно отправлен на сервер		
	идентификатор видео		2. Проверить код ответа	HTTP запрос - 404 NOT FOUND	1	
	(например, пустую		3. Проверить название	Headers request:		
	строку или случайную			Content type: application / json	1	
	строку) и проверить,			Accept: application / json	Passed	Низкий
	что API вернул		4. Проверить скорость ответа	Ответ пришел в разумные ожидаемые сроки		
	ошибку или пустой			Тело ответа возвращается и имеет вид,		
	список		5. Проверить тело ответа от	соответствующий документации		
			сервера на соответствие			
			документации			
11	Передать	Тип запроса: POST	1. Отправить POST запрос	Запрос успешно отправлен на сервер		
	некорректный тип	Запрос: https://	https://			
	идентификатора		2. Проверить код ответа	HTTP запрос - 404 NOT FOUND	1	
	(например, число		3. Проверить название	Headers request:	1	
	вместо строки) и			Content type: application / json	1	
	проверить, что АРІ			Accept: application / json	1	
	вернул ошибку или		4. Проверить скорость ответа	Ответ пришел в разумные ожидаемые сроки	B	
	пустой список			В репозитории отображается созданная задача с	Passed	Низкий
				заданными в запросе параметрами (название,		
				автор и т.д.)	1	
			6. Проверить тело ответа от	Тело ответа возвращается и имеет вид,		
			сервера на соответствие	соответствующий документации		
			документации			
12	Передать	Тип запроса : POST	1. Отправить POST запрос	Запрос успешно отправлен на сервер		
	несуществующий	Запрос: https://	https://			
	идентификатор видео		2. Проверить код ответа	HTTP sanpoc - 404 NOT FOUND	1	
	и проверить, что АРІ		3. Проверить название	Headers request:	1	
	вернул ошибку или		,	Content type: application / json	1	
	пустой список			Accept: application / json	1	
	'		4. Проверить скорость ответа	Ответ пришел в разумные ожидаемые сроки	Passod	Низкий

			 Проверить отображение UI Проверить тело ответа от сервера на соответствие документации 	В репозитории отображается созданная задача с заданными в запросе параметрами (название, автор и т.д.) Тело ответа возвращается и имеет вид, соответствующий документации	r asseu	ועועאכועו ו
13	Попытаться поставить like без авторизации и проверить, что API вернул ошибку или отказал в доступе.	Тип запроса : Get Запрос: https://	 Отправить Get запрос https:// Проверить код ответа Проверить название Проверить скорость ответа Проверить тело ответа от сервера на соответствие документации 	Запрос успешно отправлен на сервер HTTP запрос - 404 NOT FOUND Headers request: Content type: application / json Accept: application / json Ответ пришел в разумные ожидаемые сроки Тело ответа возвращается и имеет вид, соответствующий документации	Passed	Низкий
14	Передать некорректный идентификатор видео (например, пустую строку или случайную строку) и проверить, что API вернул ошибку или отказал в доступе	Тип запроса: POST Запрос: https://	 Отправить POST запрос <a <="" href="https://</td><td>Запрос успешно отправлен на сервер HTTP запрос - 404 NOT FOUND Headers request: Content type: application / json Accept: application / json Ответ пришел в разумные ожидаемые сроки В репозитории отображается созданная задача с заданными в запросе параметрами (название, автор и т.д.) Тело ответа возвращается и имеет вид, соответствующий документации</td><td>Passed</td><td>Низкий</td></tr><tr><td>15</td><td>Передать несуществующий идентификатор видео и проверить, что API вернул ошибку или отказал в доступе.</td><td>Тип запроса: POST
Запрос: <u>https://.</u></td><td> Отправить POST запрос <td>Запрос успешно отправлен на сервер HTTP запрос - 404 NOT FOUND Headers request: Content type: application / json Ассерt: application / json Ответ пришел в разумные ожидаемые сроки</td><td>Passad</td><td>Низучий</td>	Запрос успешно отправлен на сервер HTTP запрос - 404 NOT FOUND Headers request: Content type: application / json Ассерt: application / json Ответ пришел в разумные ожидаемые сроки	Passad	Низучий

		В репозитории отображается созданная задача с заданными в запросе параметрами (название, автор и т.д.)	r asseu	I INIAVEINI
	' '	Тело ответа возвращается и имеет вид, соответствующий документации		