ID	Test Case	Precondition	Test Steps	Test Data (Request Body)	Expected Result	Actual Result	Status	JIRA
				Courier - Creating a courier - /api/v1/courier				
	POSITIVE TEST Ensure that a new courier can be		Send a POST request to /api/v1/courier Enter valid login details	{ "login": "dad", "password": "1234", "firstName": "ryley" }	201 Created { "ok": true } The courier is	As expected	Passed	
1	successfully created with valid data, and the record appears in the Couriers	Database is up and accessible. API server is running	Run a PostgreSQL query in the database to verify the courier has reached the database with corresponding details	SELECT * FROM "Couriers" WHERE login = 'dad';	successfully created, and the corresponding record is present in the Couriers table.	, , , , , , ,		
		ı	Courie	- Creating a courier - /api/v1/courier. LOGIN FIELD				
	POSITIVE TEST/ BOUNDRY Ensure that a new courier can		Send a POST request to /api/v1/courier Enter 2 characters in login details	{ "login": "aj", "password": "1234", "firstName": "ryley" }	201 Created { "ok": true } } The courier is successfully created, and the	As expected		
2	be successfully created with minimum login field character count (2), and the record appears in the Couriers table in the database.	Database is up and accessible. API server is running	Run a PostgreSQL query in the database to verify the courier has reached the database with corresponding details	SELECT * FROM "Couriers" WHERE login = 'aj';	corresponding record is present in the Couriers table.			
	POSITIVE TEST/ BOUNDRY Ensure that a new courier can		Send a POST request to /api/v1/courier Enter 10 characters in login details	{ "login": "anastasiaa", "password": "5555", "firstName": "ana" }	201 Created { "ok": true } The courier is successfully	As expected		
3	be successfully created with maximum login field	Database is up and accessible. API server is running	Run a PostgreSQL query in the database to verify the courier has reached the database with corresponding details	SELECT * FROM "Couriers" WHERE login = 'anastasiaa';	created, and the corresponding record is present in the Couriers table.			
	POSITIVE TEST/ BOUNDRY Ensure that a new courier can be successfully created with		Send a POST request to /api/v1/courier Enter 3 characters in login details	{ "login": "ana", "password": "5555", "firstName": "anna" } }	201 Created { "ok": true } The courier is	As expected		
4	slightly above the minimum login field character count	Database is up and accessible. API server is running	Run a PostgreSQL query in the database to verify the courier has reached the database with corresponding details	SELECT * FROM "Couriers" WHERE login = 'ana';	successfully created, and the corresponding record is present in the Couriers table.			
					I	I I		
	NEGATIVE TEST/ BOUNDRY Ensure that a new courier can be NOT successfully created		Send a POST request to /api/v1/courier Enter 1 character in login details	{ "login": "a", "password": "5555", "firstName": "gary" }	HTTP/1.1 400 Bad Request { "message": "Not enough data to create an account" } The courier is successfully not	HTTP 201 CREATED { "ok": true }		https://rj178997.atlassian.net/browse/USWA-33
5		Database is up and accessible. API server is running	Database is up and 3. Run a PostgreSQL query in the database to verify the courier has NOT reached the database with SELECT * FROM "Couriers" WHERE login = 'a';	The courier is successfully created, and the corresponding record is present in the Couriers table				
	POSITIVE TEST/ BOUNDRY Ensure that a new courier can be successfully created with		Send a POST request to /api/v1/courier Enter 9 characters in login details	{ "login": "anastasia", "password": "5555", "firstName": "dale" }	HTTP 201 CREATED { "ok": true } The courier is successfully control and the	As expected		
6	slightly below the maximum login field character count (9), and the record appears in the Couriers table in the database.	Database is up and accessible. API server is running	Run a PostgreSQL query in the database to verify the courier has reached the database with corresponding details	SELECT * FROM "Couriers" WHERE login = 'anastasia';	created, and the corresponding record is present in the Couriers table HTTP/1.1 400 Bad Request	LITTO 004 005 ATED		
	NEGATIVE TEST/ BOUNDRY Ensure that a new courier can NOT be successfully created with slightly above the		Send a POST request to /api/v1/courier Enter 11 characters in login details	{ "login": "gusvansands", "password": "5555", "firstName": "gus" }	The courier is successfully not created, and the corresponding record is not present in the Couriers table.	HTTP 201 CREATED { "ok": true } The courier is successfully		https://rj178997.atlassian.net/browse/USWA-34
7	maximum login field character count (11), and the record DOES NOT appear in the Couriers table in the database.	Database is up and accessible. API server is running	Run a PostgreSQL query in the database to verify the courier has NOT reached the database with corresponding details	SELECT * FROM "Couriers" WHERE login = 'gusvansands';	LITTOM A 400 C - 1 C	created, and the corresponding record is present in the Couriers table		
	NEGATIVE TEST/ BOUNDRY Ensure that a new courier can NOT be successfully created		Send a POST request to /api/v1/courier Enter empty string in login details	{ "login": "", "password": "5555", "firstName": "gus" } }	HTTP/1.1 400 Bad Request { "code": 400, "message": "Not enough data to create an account." }	As expected		
8	with empty string as login field (character count 0), and	Database is up and accessible. API server is running	Run a PostgreSQL query in the database to verify the courier has NOT reached the database with corresponding details	SELECT * FROM "Couriers" WHERE firstName = 'gus';	The courier is successfully not created, and the corresponding record is not present in the Couriers table.			

ID	Test Case	Precondition	Test Steps	Test Data (Request Body)	Expected Result	Actual Result	Status	JIRA	
	1031 0430	1 1000Haltion		/ rest bata (request body)	HTTP/1.1 400 Bad Request	HTTP 201 CREATED	Otatas	Oliva	
	NEGATIVE TEST Ensure that a new courier can		Send a POST request to /api/v1/courier Enter numbers in login details	"login": "1234", "password": "5655", "firstName": "gus" }	{ "code": 400, "message": "Not enough data to create an account." }	{ "ok": true }		https://rj178997.atlassian.net/brows	se/USWA-3 <u>5</u>
9	NOT be successfully created with numbers as string as	Database is up and accessible. API server is running	3. Run a PostgreSQL query in the database to verify the courier has NOT reached the database with corresponding details	SELECT * FROM "Couriers" WHERE firstName = 'gus';	The courier is successfully not created, and the corresponding record is not present in the Couriers table.	The courier is successfully created, and the corresponding record is present in the Couriers table			
	NEGATIVE TEST		Send a POST request to /api/v1/courier Enter symbols in login details	{ "login": "\$\$\$\$", "password": "5555", "firstName": "gus" }	HTTP/1.1 400 Bad Request { "code": 400, "message": "Not enough data to create an account." }	HTTP 201 CREATED { "ok": true }		https://rj178997.atlassian.net/brows	se/LISWA-36
10	Ensure that a new courier can NOT be successfully created with symbols as login field, and the record DOES NOT appear in the Couriers table in the database.	Database is up and accessible. API server is running	3. Run a PostgreSQL query in the database to verify the courier has NOT reached the database with corresponding details	SELECT * FROM "Couriers" WHERE firstName = 'gus';	The courier is successfully not created, and the corresponding record is not present in the Couriers table.	The courier is successfully created, and the corresponding record is present in the Couriers table	i alieu	Ingest 1 Cost takes to the last of the l	
	NEGATIVE TEST		Send a POST request to /api/v1/courier Enter non latin letters in login details	{ "login": "фарбад", "password": "5555", "firstName": "gus" } }	HTTP/1.1 400 Bad Request { "code": 400, "message": "Not enough data to create an account." }	HTTP 201 CREATED { "ok": true }	Failed	https://rj178997.atlassian.net/brows	se/USWA-37
11	Ensure that a new courier can NOT be successfully created with non-latin letters in login field, and the record DOES NOT appear in the Couriers table in the database.	Database is up and accessible. API server is running	3. Run a PostgreSQL query in the database to verify the courier has NOT reached the database with corresponding details	SELECT * FROM "Couriers" WHERE firstName = 'gus';	The courier is successfully not created, and the corresponding record is not present in the Couriers table.	The courier is successfully created, and the corresponding record is present in the Couriers table	i alieu	THE STATE OF THE S	
	NEGATIVE TEST Ensure that a new courier can NOT be successfully created	Database is up and accessible. API server is	Send a POST request to /api/v1/courier Enter pre-existing login details	{ "login": "dad", "password": "1234", "firstName": "ryley" }	HTTP/1.1 409 Conflict { "message": "This login is not available" } The courier is successfully not	As expected			
12	with duplicate data in login field, and the record DOES NOT appear in the Couriers table in the database.	running 3. Have Pre-Exisiting Courier in database	Run a PostgreSQL query in the database to verify the courier is not duplicated in the database	SELECT * FROM "Couriers" WHERE login = 'dad';	created, and the corresponding record is duplicated in the Couriers table.				
	NEGATIVE TEST		Send a POST request to /api/v1/courier Enter NULL login details	{ "login": NULL, "password": "5555", "firstName": "gus"	HTTP/1.1 400 Bad Request { "code": 400, "message": "Not enough data to create an account." }				
13	Ensure that a new courier can NOT be successfully created with NULL in login field, and the record DOES NOT appear in the Couriers table in the database.	Database is up and accessible. API server is running	details and the second of the NOT	The courier is successfully not created, and the corresponding record is not present in the Couriers table.	As expected	Passed			
	database.			courier - /api/v1/courier. PASSWORD TESTS		1			
		Database is up and accessible.	Send a POST request to /api/v1/courier Omit the PASSWORD field and	{ "login": "ryley" "firstName": "Jim" }	"code": 400, "message": "Not enough data to create an account." } The courier is NOT created,	As expected	Passed		
14	submitting omitting the PASSWORD field, and the record does not appear in the Couriers table in the database.	2. API server is running	3. Run a PostgreSQL query in the database to verify there is not a courier added in the database	SELECT * FROM "Couriers" WHERE firstName = "Jim";	and the corresponding record is NOT in the Couriers table.				
	NEGATIVE TEST Ensure that a courier cannot be successfully created when submitting empty string in the	accessible.		{ "login": "jacky" "password": "", "firstName": "Jim" }	{ "code": 400, "message": "Not enough data to create an account." } The courier is NOT created, and the corresponding record is NOT in the Couriers table.	As expected			
15	PASSWORD field, and the record does not appear in the Couriers table in the database.	2. API server is running	Run a PostgreSQL query in the database to verify there is not a courier added in the database	SELECT * FROM "Couriers" WHERE firstName = "Jim";					
	POSITIVE TEST/ BOUNDRY TEST Ensure that a courier can be successfully created when	Database is up and accessible.	Send a POST request to /api/v1/courier Submit 0000 in the PASSWORD field and enter valid login and first name	{ "login": "james" "password": "0000", "firstName": "Jim" }	201 Created { "ok": true } The courier is successfully created, and the corresponding record is present in the Couriers table.	As expected			
16		2. API server is running	Run a PostgreSQL query in the database to verify the courier is added in the database	SELECT * FROM "Couriers" WHERE firstName = "Jim";					
	POSITIVE TEST/ BOUNDRY TEST Ensure that a courier can be	1 Database is up and	Send a POST request to /api/v1/courier Submit 9999 in the PASSWORD field and enter valid login and first name	{ "login": "james" "password": "9999", "firstName": "Jim" } }	201 Created { "ok": true } The courier is successfully created, and the	As expected			
	successfully created when submitting maximum	Database is up and accessible.			corresponding record is present in the Couriers table.	, to expedied			

ID	Test Case	Precondition	Test Steps	Test Data (Request Body)	Expected Result	Actual Result	Status	JIRA					
	character limit in the PASSWORD field, and the	2. API server is	3. Run a PostgreSQL query in the										
17	record appears in the Couriers table in the database.	running	database to verify the courier is added in the database	SELECT * FROM "Couriers" WHERE firstName = "Jim";									
	NEGATIVE TEST/ BOUNDRY		1. Send a POST request to	{ "!agin", "inmon"	{ "code": 400,	201 Created {							
	TEST		/api/v1/courier 2. Submit 10000 in the PASSWORD	"login": "james" "password": "10000",	"message": "Not enough data to create an account."	"ok": true							
	Ensure that a courier can NOT be successfully created when		field and enter valid login and first	"firstName": "Jim"	The courier is NOT created.	The courier is successfully							
	submitting ABOVE maximum character limit in the	Database is up and accessible.	name	1	and the corresponding record is NOT in the Couriers table.	created, and the corresponding record is		https://rj178997.atlassian.net/brov	wse/USWA-38				
	PASSWORD field, and the record DOES NOT appear in	2. API server is	3. Run a PostgreSQL query in the		S NOT III die councie date.	present in the Couriers							
18	the Couriers table in the database.	unning	running	unning	running	database to verify there is not a courier added in the database	SELECT * FROM "Couriers" WHERE firstName = "Jim";		table.				
	NEGATIVE TEST/ BOUNDRY		1. Send a POST request to	{	{ "code": 400	201 Created							
	TEST		/api/v1/courier 2. Submit 100 in the PASSWORD	"login": "james" "password": "100",	"message": "Not enough data to create an account."	"ok": true							
	Ensure that a courier can NOT be successfully created when		field and enter valid login and first	"firstName": "Jim"	}]							
	submitting BELOW minimum	1. Database is up and	name	}	The courier is NOT created, and the corresponding record	The courier is successfully created, and the		https://rj178997.atlassian.net/brov	wse/USWA-39				
	PASSWORD field, and the record DOES NOT appear in	accessible. 2. API server is	3. Run a PostgreSQL query in the		is NOT in the Couriers table.	corresponding record is present in the Couriers							
19	the Couriers table in the database.	running	database to verify there is not a courier added in the database	SELECT * FROM "Couriers" WHERE firstName = "Jim";		table.							
	NEGATIVE TEST/ BOUNDRY TEST		Send a POST request to /api/v1/courier	{ "login": "james"	{ "code": 400,	201 Created {							
			2. Submit 1 in the PASSWORD field	"password": "1",	"message": "Not enough data to create an account."	"ok": true							
	Ensure that a courier can NOT be successfully created when submitting single numberic	1. Database is up and	and enter valid login and first name	"firstName": "Jim"	The courier is NOT created,	The courier is successfully	Failed	https://rj178997.atlassian.net/brov	wse/LISWA_40				
	character in the PASSWORD	accessible. 2. API server is	3. Run a PostgreSQL query in the	1	and the corresponding record is NOT in the Couriers table.	created, and the corresponding record is present in the Couriers table.		https://ji/1000/.atlassian.nevblov	WSE/USWA-40				
20	field, and the record DOES NOT appear in the Couriers	running	database to verify there is not a courier added in the database	SELECT * FROM "Couriers" WHERE firstName = "Jim";									
20	table in the database.		Send a POST request to	{	{	201 Created							
	NEGATIVE TEST		/api/v1/courier 2. Submit latin letters in the	"login": "james" "password": "password",	"code": 400, "message": "Not enough data to create an account."	{ "ok": true							
			PASSWORD field and enter valid	"firstName": "Jim"	}	The courier is successfully created, and the							
	Ensure that a courier can NOT be successfully created when submitting latin letter	1. Database is up and	login and first name	}	The courier is NOT created, and the corresponding record is NOT in the Couriers table.			https://rj178997.atlassian.net/brov	wse/USWA-41				
	characters in the PASSWORD field, and the record DOES	accessible. 2. API server is	3. Run a PostgreSQL query in the		is NOT in the Couriers table.	corresponding record is present in the Couriers		naponj 17 ocor .aaacolannoebrot					
21	NOT appear in the Couriers table in the database.	running	database to verify there is not a courier added in the database	SELECT * FROM "Couriers" WHERE firstName = "Jim":		table.							
	table in the database.		1. Send a POST request to	{	{ "code": 400,	201 Created							
	NEGATIVE TEST		/api/v1/courier 2. Submit symbols in the	"login": "james" "password": "%%%",	"message": "Not enough data to create an account."	{ "ok": true							
	Ensure that a courier can NOT		PASSWORD field and enter valid	"firstName": "Jim"	}	The courier is successfully							
	be successfully created when submitting symbols in the	Database is up and accessible.	login and first name	}	The courier is NOT created, and the corresponding record is NOT in the Couriers table.	created, and the	cord is	Failed https://rj178997.	https://rj178997.atlassian.net/brov	wse/USWA-42			
	PASSWORD field, and the record DOES NOT appear in	2. API server is	3. Run a PostgreSQL query in the		is NOT in the Country table.	corresponding record is present in the Couriers table							
22	the Couriers table in the database.	running	database to verify there is not a courier added in the database	SELECT * FROM "Couriers" WHERE firstName = "Jim";		table.							
			1. Send a POST request to	{	HTTP/1.1 400 Bad Request {								
			/api/v1/courier 2. Enter NULL login details	"login": "james" "password": NULL,	"code": 400, "message": "Not enough data to create an account."								
	NEGATIVE TEST		Eor Hole login details	"firstName": "gus"	}	As expected							
	NOT be successfully created	1. Database is up and	3. Run a PostgreSQL query in the	,	The courier is successfully not	/10 CAPECIEU							
	with NULL in Password field, and the record DOES NOT	accessible. 2. API server is	database to verify the courier has NOT reached the database with	SELECT * FROM "Couriers" WHERE firstName = 'gus';	created, and the corresponding record is not present in the Couriers table.								
23	appear in the Couriers table in the database.	running	corresponding details		present in the Couners table.								
				Creating a courier - /api/v1/courier. FirstName FIELI									
			1. Send a POST request to /api/v1/courier	{ "login": "james",	201 Created { "ok": true								
	POSITIVE TEST/ BOUNDRY		Enter 2 characters in firstName details	"password": "1234", "firstName": "aj"	}								
	Ensure that a new courier can			}	The courier is successfully created, and the	As expected							
	be successfully created with minimum FirstName field	Database is up and accessible.	Run a PostgreSQL query in the database to verify the courier has		corresponding record is present in the Couriers								
	character count (2), and the record appears in the Couriers	2. API server is	reached the database with	SELECT * FROM "Couriers" WHERE login = 'james';	table.								
24	table in the database.	running	corresponding details	,	201 Created								
			1. Send a POST request to /api/v1/courier	{ "login": "james",	{ "ok": true								
	POSITIVE TEST/ BOUNDRY		2. Enter 10 characters in firstName details	"password": "5555", "firstName": "aaaaaaaaaa"	}								
	Ensure that a new courier can		- Columb	}	The courier is successfully created, and the	As expected							
	be successfully created with maximum firstName field	1. Database is up and	database to verify the sourier has		corresponding record is present in the Couriers								
	character count (10), and the record appears in the Couriers	accessible. 2. API server is	reached the database with	SELECT * FROM "Couriers" WHERE login = 'james';	table.								
25	table in the database.	running	corresponding details										

ID	Test Case	Precondition	Test Steps	Test Data (Request Body)	Expected Result	Actual Result	Status	JIRA		
	POSITIVE TEST/ BOUNDRY Ensure that a new courier can be successfully created with		Send a POST request to /apii/1/courier Enter 3 characters in firstName details	{ "login": "james", "password": "1234", "firstName": "abc" }	201 Created { "ok": true } The courier is	As expected	Passed			
26	slightly above the minimum firstName field character count (3), and the record appears in the Couriers table in the database.	Database is up and accessible. API server is running	3. Run a PostgreSQL query in the database to verify the courier has reached the database with corresponding details	SELECT * FROM "Couriers" WHERE login = 'james';	successfully created, and the corresponding record is present in the Couriers table.					
	NEGATIVE TEST/ BOUNDRY Ensure that a new courier can be NOT successfully created with slightly below the minimum firstName field character count (1), and the record DOES NOT	accessible.	Send a POST request to /api/v1/courier Enter 1 character in firstName details Run a PostgreSQL query in the database to verify the courier has NOT	{ "login": "james", "password": "5555", "firstName": "a" } SELECT * FROM "Couriers" WHERE login = "james";	HTTP/1.1 400 Bad Request { "message": "Not enough data to create an account" } The courier is successfully not created, and the corresponding record is not present in the Couriers table.	HTTP 201 CREATED { "ok": true } The courier is successfully created, and the corresponding record is present in the Couriers		https://ri178997.atlassian.net/brow	ie/USWA-43	
27	appear in the Couriers table in the database. POSITIVE TEST/ BOUNDRY Ensure that a new courier can be successfully created with slightly accepted with slightly for the courier count of the courier count of the courier count of the courier table record appears in the Couriers table	accessible. 2. API server is	reached the database with corresponding details 1. Send a POST request to /api/v1/courier 2. Enter 9 characters in firstName details 3. Run a PostgreSQL query in the database to verify the courier has reached the database with	{ "login": "james", "password": "1234", "firstName": "bbbbbbbbbb" } SELECT * FROM "Couriers" WHERE login = 'james';	HTTP 201 CREATED { "ok": true } The courier is successfully created, and the corresponding record is present in the Couriers table	table As expected	Passed			
28	NEGATIVE TEST/ BOUNDRY Ensure that a new courier can NOT	running 1 Database is up and	corresponding details 1. Send a POST request to /apit/v1/courier 2. Enter 11 characters in firstName details 3. Run a PostgreSQL query in the	{ "login": "james", "password": "5555", "firstName": "ccccccccc" }	HTTP/1.1 400 Bad Request The courier is successfully not created, and the corresponding record is not present in the Couriers table.	HTTP 201 CREATED { "ok": true } The courier is successfully created, and the	Failed	https://rj178997.atlassian.net/brows	se/USWA-44	
29	be successfully created with slightly above the maximum firstName field character count (11), and the record DOES NOT appear in the Couriers table in the database.	accessible. 2. API server is running	database to verify the courier has NOT reached the database with corresponding details	SELECT * FROM "Couriers" WHERE login = 'james';		corresponding record is present in the Couriers table				
	POSITIVE TEST Ensure that a new courier can be successfully created with empty string as firstName field (character count 0) and the	Database is up and accessible. API server is	Send a POST request to /api/v1/courier Enter empty string in firstName details Run a PostgreSQL query in the database to verify the courier has reached the database with	{ "login": "james", "password": "1234", "firstName": "" } SELECT * FROM "Couriers" WHERE login = "james";	HTTP 201 CREATED { "ok": true } The courier is successfully created, and the corresponding record is present in the Couriers table	As expected				
30	record appears in the Couriers table in the database. NEGATIVE TEST	running	corresponding details 1. Send a POST request to /api/v1/courier 2. Enter numbers in firstName details	{ "login": "james", "password": "5555", "firstName": "1234"	HTTP/1.1 400 Bad Request { "code": 400, "message": "Not enough data to create an account." }	HTTP 201 CREATED { "ok": true }		https://ri178997.atlassian.net/brows	all ISWA 45	
31	Ensure that a new courier can NOT be successfully created with numbers as string as firstName field, and the record DOES NOT appear in the Couriers table in the database.	Database is up and accessible. API server is running	3. Run a PostgreSQL query in the database to verify the courier has NOT reached the database with corresponding details	J SELECT * FROM *Couriers* WHERE login = 'james';	The courier is successfully not created, and the corresponding record is not present in the Couriers table.	The courier is successfully created, and the corresponding record is present in the Couriers table	Failed	III. III. III. III. III. III. III. III	eroswa-45	
			Send a POST request to /api/v1/courier Enter symbols in firstName details	{ "login": "james", "password": "1234", "firstName": "\$\$\$\$"	HTTP/1.1 400 Bad Request { "code": 400, "message": "Not enough data to create an account." }	HTTP 201 CREATED { "ok": true }	Foiled	https://rj178997.atlassian.net/brows	e/LISWA_46	
32	NEGATIVE TEST Ensure that a new courier can NOT be successfully created with symbols as FirstName field, and the record DOES NOT appear in the Couriers table in the database.	Database is up and accessible. API server is running	3. Run a PostgreSQL query in the database to verify the courier has NOT reached the database with corresponding details	SELECT * FROM "Couriers" WHERE login = 'james';	The courier is successfully not created, and the corresponding record is not present in the Couriers table.	The courier is successfully created, and the corresponding record is present in the Couriers table	Falled		S S S S S S S S S S S S S S S S S S S	
	NEGATIVE TEST		Send a POST request to /api/v1/courier Enter non latin letters in firstName details	{ "login": "james", "password": "5555", "firstName": "фуу" } }	HTTP/1.1 400 Bad Request { "code": 400, "message": "Not enough data to create an account." }	HTTP 201 CREATED { "ok": true }	Failed	https://ri178997.atlassian.net/brows	re/USWA-47	
33	Ensure that a new courier can NOT be successfully created with non-	Database is up and accessible. API server is running	3. Run a PostgreSQL query in the database to verify the courier has NOT reached the database with corresponding details	SELECT * FROM "Couriers" WHERE login = 'james';	The courier is successfully not created, and the corresponding record is not present in the Couriers table.	The courier is successfully created, and the corresponding record is present in the Couriers table				

ID	Test Case	Precondition	Test Steps	Test Data (Request Body)	Expected Result	Actual Result	Status	JIRA	
	POSITIVE TEST		Send a POST request to /api/v1/courier OMIT firstName field	{ "login": "james", "password": "5555" }	HTTP 201 CREATED { "ok": true } The courier is successfully	As expected	Passed		
34	Ensure that a new courier can be successfully created with OMITTING the firstName field, and the record appears in the Couriers table in the database.	Database is up and accessible. API server is running	Run a PostgreSQL query in the database to verify the courier has reached the database with corresponding details	SELECT * FROM "Couriers" WHERE login = 'james';	created, and the corresponding record is present in the Couriers table				
				Courier - Removing a courier					
	POSITIVE TEST Deletion of a Courier with no linked orders	Database is up and accessible. API server is running Courier has been created and store in the database	Send a POST request to {baseUrl} /api/v1/courier/ with valid courier details. Query the SQL database for the courier's details, and retrieve the ID. Send a DELETE request to {baseUrl} /api/v1/courier/id, where id is the courier's ID retrieved from the SQL database. Query the SQL database again using the courier's ID to ensure they no longer exist in the database.	SELECT * FROM "Couriers" WHERE firstName = 'tom';	1. Upon sending the POST request the API should return a 201 (Created) status code 2. After sending a query in the SQL database it should return the courier's data, including their ID. 3. Upon sending a DELETE request the API should return HTTP/1.1 200 OK { ok: true } 4. After sending query once more to the SQL database using the deleted courier ID, The query should return no rows, confirming that the courier has been	As expected	PASSED		
36	POSITIVE TEST Login not Possible after Deletion of a Courier with no linked orders	Perform all test steps in test case #35	Send a POST request to {baseURL} /api/v1/courier/login In the body of the request use the same login information used in test case #35	{ "login": "fred", "password": "1234" }	deleted. { "code": 404, "message": "Account not found." }	As expected	PASSED		
37	NEGATIVE TEST Deletion of non-existent Courier	Database is up and accessible. API server is running	Send a DELETE request to {baseUrl}/api/v1/courier/99, (non existent id)		{ "code": 404, "message": "There's no courier with this ID." }	As expected	PASSED		

ID	Test Case	Precondition	Test Steps	Test Data (Request Body)	Expected Result	Actual Result	Status	JIRA	
38	POSITIVE TEST Deletion of a Courier with linked orders	1. Database is up and accessible. 2. API server is running 3. Courier has been created and store in the database	Send a POST request to {baseUrl} /apit/1/courier/ with valid courier details. Query the SQL database for the courier's details, and retrieve the ID. Create an order by sending a POST request to {baseUrl}/apit/1/orders with	POST REQUEST (baseUrl)/api/v1/courier/ { "login": "fred", "password": "1234", "firstName": "tom" } SQL Query SELECT * FROM "Couriers" WHERE firstName = 'tom'; POST REQUEST (baseUrl)/api/v1/orders { "firstName": "Naruto", "lastName": "Uchiha", "address": "Konoha, 142 apt.", "metroStation": 4, "phone": "47 800 355 35 35", "rentTime": 5, "deliveryDate": "2020-06-06", "comment": "Saske, come back to Konoha", "color": "BLACK" "	1. Upon sending a DELETE request the API should return HTTP/1.1 200 OK { ok: true } 2. After sending query once more to the SQL database using the deleted courier ID, The query should return no rows, confirming that the courier has been deleted.	As expected	PASSED	JIKA	
39	POSITIVE TEST On Deletion of a Courier linked orders are deleted from orders table within the database	Database is up and accessible. API server is running Courier has been created and store in the database	Repeat all steps of test 38 Open Orders table within SQL database	SELECT * FROM "Orders" WHERE firstName = "Naruto";	Upon SQL query in orders table, it should contain zero results	SQL query results with past orders, indicating the orders are not deleted	FAILED	https://rj178997.atlassian. net/browse/USWA-49	
40	POSITIVE TEST Login not Possible after Deletion of a Courier with linked orders	1. Perform all test steps in test case #38	Send a POST request to {baseURL} /apii/v1/courier/login In the body of the request use the same login information used in test case #38	{ "login": "fred", "password": "1234" }	{ "code": 404, "message": "Account not found." }	As expected	PASSED		
	PASSED 24		FAILED 15						

ID Test Case Precondition Test Steps Test Data (Request Body) Expected Result Actual Result Status	JIRA	

ID	Test Case	Precondition	Test Steps	Test Data (Request Body)	Expected Result	Actual Result	Status	JIRA	