

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

ID	Test Case	Precondition	Test Steps	Test Data (Request Body)	Expected Result	Actual Result	Status	JIRA
9	NEGATIVE TEST Ensure that a new courier can NOT be successfully created with numbers as string as login field, and the record DOES NOT appear in the Couriers table in the database.	1. Database is up and accessible. 2. API server is running	1. Send a POST request to /api/v1/courier 2. Enter numbers in login details 3. Run a PostgreSQL query in the database to verify the courier has NOT reached the database with corresponding details	{ "login": "1234", "password": "5555", "firstName": "gus" }	HTTP/1.1 400 Bad Request { "code": 400, "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 table	Failed	https://rj178997.atlassian.net/browse/USWA-35
10	NEGATIVE TEST 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.	1. Database is up and accessible. 2. API server is running	1. Send a POST request to /api/v1/courier 2. Enter symbols in login details 3. Run a PostgreSQL query in the database to verify the courier has NOT reached the database with corresponding details	{ "login": "\$\$\$\$", "password": "5555", "firstName": "gus" }	HTTP/1.1 400 Bad Request { "code": 400, "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 table	Failed	https://rj178997.atlassian.net/browse/USWA-36
11	NEGATIVE TEST Ensure that a new courier can NOT be successfully created with non-latin letters as login field, and the record DOES NOT appear in the Couriers table in the database.	1. Database is up and accessible. 2. API server is running	1. Send a POST request to /api/v1/courier 2. Enter non latin letters in login details 3. Run a PostgreSQL query in the database to verify the courier has NOT reached the database with corresponding details	{ "login": "фapбaд", "password": "5555", "firstName": "gus" }	HTTP/1.1 400 Bad Request { "code": 400, "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 table	Failed	https://rj178997.atlassian.net/browse/USWA-37
12	NEGATIVE TEST Ensure that a new courier can NOT be successfully created with duplicate data in login field, and the record DOES NOT appear in the Couriers table in the database.	1. Database is up and accessible. 2. API server is running 3. Have Pre-Existing Courier in database	1. Send a POST request to /api/v1/courier 2. Enter pre-existing login details 3. Run a PostgreSQL query in the database to verify the courier is not duplicated in the database	{ "login": "dad", "password": "1234", "firstName": "riley" }	HTTP/1.1 409 Conflict { "message": "This login is not available" } The courier is successfully not created, and the corresponding record is not duplicated in the Couriers table.	As expected	Passed	
13	NEGATIVE TEST 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.	1. Database is up and accessible. 2. API server is running	1. Send a POST request to /api/v1/courier 2. Enter NULL login details 3. Run a PostgreSQL query in the database to verify the courier has NOT reached the database with corresponding details	{ "login": NULL, "password": "5555", "firstName": "gus" }	HTTP/1.1 400 Bad Request { "code": 400, "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.	As expected	Passed	
Courier - Creating a courier - /api/v1/courier. PASSWORD TESTS								
14	NEGATIVE TEST Ensure that a courier cannot be successfully created when submitting omitting the PASSWORD field, and the record does not appear in the Couriers table in the database.	1. Database is up and accessible. 2. API server is running	1. Send a POST request to /api/v1/courier 2. Omit the PASSWORD field and enter valid login and first name 3. Run a PostgreSQL query in the database to verify there is not a courier added in the database	{ "login": "riley" "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	Passed	
15	NEGATIVE TEST Ensure that a courier cannot be successfully created when submitting empty string in the PASSWORD field, and the record does not appear in the Couriers table in the database.	1. Database is up and accessible. 2. API server is running	1. Send a POST request to /api/v1/courier 2. Submit an empty string in the PASSWORD field and enter valid login and first name 3. Run a PostgreSQL query in the database to verify there is not a courier added in the database	{ "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	Passed	
16	POSITIVE TEST/ BOUNDRY TEST Ensure that a courier can be successfully created when submitting minimum character limit in the PASSWORD field, and the record appears in the Couriers table in the database.	1. Database is up and accessible. 2. API server is running	1. Send a POST request to /api/v1/courier 2. Submit 0000 in the PASSWORD field and enter valid login and first name 3. Run a PostgreSQL query in the database to verify the courier is added in the database	{ "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	Passed	
	POSITIVE TEST/ BOUNDRY TEST Ensure that a courier can be successfully created when submitting maximum	1. Database is up and accessible.	1. Send a POST request to /api/v1/courier 2. 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 corresponding record is present in the Couriers table.	As expected	Passed	

ID	Test Case	Precondition	Test Steps	Test Data (Request Body)	Expected Result	Actual Result	Status	JIRA
17	character limit in the PASSWORD field, and the record appears in the Couriers table in the database.	2. API server is running	3. Run a PostgreSQL query in the database to verify the courier is added in the database	SELECT * FROM "Couriers" WHERE firstName = "Jim";				
18	NEGATIVE TEST/ BOUNDRY TEST Ensure that a courier can NOT be successfully created when submitting ABOVE maximum character limit in the PASSWORD field, and the record DOES NOT appear in the Couriers table in the database.	1. Database is up and accessible. 2. API server is running	1. Send a POST request to /api/v1/courier 2. Submit 10000 in the PASSWORD field and enter valid login and first name 3. Run a PostgreSQL query in the database to verify there is not a courier added in the database	{ "login": "james" "password": "10000", "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.	201 Created { "ok": true } The courier is successfully created, and the corresponding record is present in the Couriers table.	Failed	https://rj178997.atlassian.net/browse/USWA-38
19	NEGATIVE TEST/ BOUNDRY TEST Ensure that a courier can NOT be successfully created when submitting BELOW minimum character limit in the PASSWORD field, and the record DOES NOT appear in the Couriers table in the database.	1. Database is up and accessible. 2. API server is running	1. Send a POST request to /api/v1/courier 2. Submit 100 in the PASSWORD field and enter valid login and first name 3. Run a PostgreSQL query in the database to verify there is not a courier added in the database	{ "login": "james" "password": "100", "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.	201 Created { "ok": true } The courier is successfully created, and the corresponding record is present in the Couriers table.	Failed	https://rj178997.atlassian.net/browse/USWA-39
20	NEGATIVE TEST/ BOUNDRY TEST Ensure that a courier can NOT be successfully created when submitting single numeric character in the PASSWORD field, and the record DOES NOT appear in the Couriers table in the database.	1. Database is up and accessible. 2. API server is running	1. Send a POST request to /api/v1/courier 2. Submit 1 in the PASSWORD field and enter valid login and first name 3. Run a PostgreSQL query in the database to verify there is not a courier added in the database	{ "login": "james" "password": "1", "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.	201 Created { "ok": true } The courier is successfully created, and the corresponding record is present in the Couriers table.	Failed	https://rj178997.atlassian.net/browse/USWA-40
21	NEGATIVE TEST Ensure that a courier can NOT be successfully created when submitting latin letter characters in the PASSWORD field, and the record DOES NOT appear in the Couriers table in the database.	1. Database is up and accessible. 2. API server is running	1. Send a POST request to /api/v1/courier 2. Submit latin letters in the PASSWORD field and enter valid login and first name 3. Run a PostgreSQL query in the database to verify there is not a courier added in the database	{ "login": "james" "password": "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.	201 Created { "ok": true } The courier is successfully created, and the corresponding record is present in the Couriers table.	Failed	https://rj178997.atlassian.net/browse/USWA-41
22	NEGATIVE TEST Ensure that a courier can NOT be successfully created when submitting symbols in the PASSWORD field, and the record DOES NOT appear in the Couriers table in the database.	1. Database is up and accessible. 2. API server is running	1. Send a POST request to /api/v1/courier 2. Submit symbols in the PASSWORD field and enter valid login and first name 3. Run a PostgreSQL query in the database to verify there is not a courier added in the database	{ "login": "james" "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.	201 Created { "ok": true } The courier is successfully created, and the corresponding record is present in the Couriers table.	Failed	https://rj178997.atlassian.net/browse/USWA-42
23	NEGATIVE TEST Ensure that a new courier can NOT be successfully created with NULL in Password field, and the record DOES NOT appear in the Couriers table in the database.	1. Database is up and accessible. 2. API server is running	1. Send a POST request to /api/v1/courier 2. Enter NULL login details 3. Run a PostgreSQL query in the database to verify the courier has NOT reached the database with corresponding details	{ "login": "james" "password": NULL, "firstName": "gus" }	HTTP/1.1 400 Bad Request { "code": 400, "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.	As expected	Passed	
Courier - Creating a courier - /api/v1/courier. FirstName FIELD TESTS								
24	POSITIVE TEST/ BOUNDRY Ensure that a new courier can be successfully created with minimum FirstName field character count (2), and the record appears in the Couriers table in the database.	1. Database is up and accessible. 2. API server is running	1. Send a POST request to /api/v1/courier 2. Enter 2 characters in firstName details 3. Run a PostgreSQL query in the database to verify the courier has reached the database with corresponding details	{ "login": "james", "password": "1234", "firstName": "aj" }	201 Created { "ok": true } The courier is successfully created, and the corresponding record is present in the Couriers table.	As expected	Passed	
25	POSITIVE TEST/ BOUNDRY Ensure that a new courier can be successfully created with maximum firstName field character count (10), and the record appears in the Couriers table in the database.	1. Database is up and accessible. 2. API server is running	1. Send a POST request to /api/v1/courier 2. Enter 10 characters in firstName details 3. Run a PostgreSQL query in the database to verify the courier has reached the database with corresponding details	{ "login": "james", "password": "5555", "firstName": "aaaaaaaaa" }	201 Created { "ok": true } The courier is successfully created, and the corresponding record is present in the Couriers table.	As expected	Passed	

ID	Test Case	Precondition	Test Steps	Test Data (Request Body)	Expected Result	Actual Result	Status	JIRA
26	POSITIVE TEST/ BOUNDRY Ensure that a new courier can be successfully created with slightly above the minimum firstName field character count (3) , and the record appears in the Couriers table in the database.	1. Database is up and accessible. 2. API server is running	1. Send a POST request to /api/v1/courier 2. Enter 3 characters in firstName details 3. Run a PostgreSQL query in the database to verify the courier has reached the database with corresponding details	{ "login": "james", "password": "1234", "firstName": "abc" }	201 Created { "ok": true }	As expected	Passed	
27	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 appear in the Couriers table in the database.	1. Database is up and accessible. 2. API server is running	1. Send a POST request to /api/v1/courier 2. Enter 1 character in firstName details 3. Run a PostgreSQL query in the database to verify the courier has NOT reached the database with corresponding details	{ "login": "james", "password": "5555", "firstName": "a" }	HTTP/1.1 400 Bad Request { "message": "Not enough data to create an account" }	HTTP 201 CREATED { "ok": true }	Failed	https://rj178997.atlassian.net/browse/USWA-43
28	POSITIVE TEST/ BOUNDRY Ensure that a new courier can be successfully created with slightly below the maximum firstName field character count (9) , and the record appears in the Couriers table in the database.	1. Database is up and accessible. 2. API server is running	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 corresponding details	{ "login": "james", "password": "1234", "firstName": "bbbbbbbbb" }	HTTP 201 CREATED { "ok": true }	As expected	Passed	
29	NEGATIVE TEST/ BOUNDRY Ensure that a new courier can NOT 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.	1. Database is up and accessible. 2. API server is running	1. Send a POST request to /api/v1/courier 2. Enter 11 characters in firstName details 3. Run a PostgreSQL query in the database to verify the courier has NOT reached the database with corresponding details	{ "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 }	Failed	https://rj178997.atlassian.net/browse/USWA-44
30	POSITIVE TEST Ensure that a new courier can be successfully created with empty string as firstName field (character count 0) , and the record appears in the Couriers table in the database.	1. Database is up and accessible. 2. API server is running	1. Send a POST request to /api/v1/courier 2. Enter empty string in firstName details 3. Run a PostgreSQL query in the database to verify the courier has reached the database with corresponding details	{ "login": "james", "password": "1234", "firstName": "" }	HTTP 201 CREATED { "ok": true }	As expected	Passed	
31	NEGATIVE TEST 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.	1. Database is up and accessible. 2. API server is running	1. Send a POST request to /api/v1/courier 2. Enter numbers in firstName details 3. Run a PostgreSQL query in the database to verify the courier has NOT reached the database with corresponding 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 }	Failed	https://rj178997.atlassian.net/browse/USWA-45
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.	1. Database is up and accessible. 2. API server is running	1. Send a POST request to /api/v1/courier 2. Enter symbols in firstName details 3. Run a PostgreSQL query in the database to verify the courier has NOT reached the database with corresponding 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 }	Failed	https://rj178997.atlassian.net/browse/USWA-46
33	NEGATIVE TEST Ensure that a new courier can NOT be successfully created with non-latin letters in firstName field , and the record DOES NOT appear in the Couriers table in the database.	1. Database is up and accessible. 2. API server is running	1. Send a POST request to /api/v1/courier 2. Enter non latin letters in firstName details 3. Run a PostgreSQL query in the database to verify the courier has NOT reached the database with corresponding details	{ "login": "james", "password": "5555", "firstName": "øyy" }	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/browse/USWA-47

ID	Test Case	Precondition	Test Steps	Test Data (Request Body)	Expected Result	Actual Result	Status	JIRA
34	POSITIVE TEST 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.	1. Database is up and accessible. 2. API server is running	1. Send a POST request to /api/v1/courier 2. OMIT firstName field 3. Run a PostgreSQL query in the database to verify the courier has reached the database with corresponding details	<pre>{ "login": "james", "password": "5555" }</pre> SELECT * FROM "Couriers" WHERE login = 'james';	HTTP 201 CREATED <pre>{ "ok": true }</pre> The courier is successfully created, and the corresponding record is present in the Couriers table	As expected	Passed	
Courier - Removing a courier								
35	POSITIVE TEST Deletion of a Courier with no linked orders	1. Database is up and accessible. 2. API server is running 3. Courier has been created and store in the database	1. Send a POST request to {baseUrl}/api/v1/courier/ with valid courier details. 2. Query the SQL database for the courier's details, and retrieve the ID. 3. Send a DELETE request to {baseUrl}/api/v1/courier/:id, where :id is the courier's ID retrieved from the SQL database. 4. Query the SQL database again using the courier's ID to ensure they no longer exist in the database	POST REQUEST <pre>{ "login": "fred", "password": "1234", "firstName": "tom" }</pre> SQL Query SELECT * FROM "Couriers" WHERE firstName = 'tom'; DELETE REQUEST send empty body to {baseUrl}/api/v1/courier/1 SQL Query SELECT * FROM "Couriers" WHERE id = 1;	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 <pre>{ ok: true }</pre> 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 deleted.	As expected	PASSED	
36	POSITIVE TEST Login not Possible after Deletion of a Courier with no linked orders	1. Perform all test steps in test case #35	1. Send a POST request to {baseUrl}/api/v1/courier/login 2. In the body of the request use the same login information used in test case #35	<pre>{ "login": "fred", "password": "1234" }</pre>	<pre>{ "code": 404, "message": "Account not found." }</pre>	As expected	PASSED	
37	NEGATIVE TEST Deletion of non-existent Courier	1. Database is up and accessible. 2. API server is running	1. Send a DELETE request to {baseUrl}/api/v1/courier/99, (non existent id)		<pre>{ "code": 404, "message": "There's no courier with this ID." }</pre>	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	1. Send a POST request to {baseUrl}/api/v1/courier/ with valid courier details. 2. Query the SQL database for the courier's details, and retrieve the ID. 3. Create an order by sending a POST request to {baseUrl}/api/v1/orders with valid order details and retrieve the track id in the response body 4. Now to retrieve the order id send a GET request {baseUrl}/api/v1/orders/track?t=:id, replacing the :id with your actual track ID Retrieve the order id in the response body 5. Using the order id as the first param and the courier id as the second param send a PUT request to /api/v1/orders/accept/1?courierId=213 6. remove a courier by sending a delete request to {baseUrl}/api/v1/courier/id 7. search the sql database again and ensure the courier is no longer there	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": "+7 800 355 35 35", "rentTime": 5, "deliveryDate": "2020-06-06", "comment": "Saske, come back to Konoha", "color": ["BLACK"] } PUT REQUEST {baseUrl}/api/v1/orders/accept/1?courierId=1 DELETE REQUEST send empty body to {baseUrl}/api/v1/courier/1 SQL Query SELECT * FROM "Couriers" WHERE id = 1;	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	
39	POSITIVE TEST On Deletion of a Courier linked orders are deleted from orders table within the database	1. Database is up and accessible. 2. API server is running 3. Courier has been created and store in the database	1. Repeat all steps of test 38 2. 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	1. Send a POST request to {baseURL}/api/v1/courier/login 2. 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	



