

Test Case ID	Description	HTTP Request	Pre-conditions	Steps to reproduce	Expected Result	Actual Result	Status Code	Status Passed / Failed
TC-000	Check API Status	GET	User is connected with live server.	1. Run localhost 2. Enter <a href="http://127.0.0.1:3000/api/status">http://127.0.0.1:3000/api/status</a> 3. Click "Send" button.	Response: "API is running" Expected status: 200 OK	Response: "API is running"	200 OK	Passed
TC-001	Add user to contacts list.	POST	1. User is connected with live server. 2. Contact details: "first_name": "{{\$randomFirstName}}", "last_name": "{{\$randomLastName}}", "city": "{{\$randomCity}}", "phone_number": "{{\$randomPhoneNumber}}"	1. Choose POST method. 2. Enter <a href="http://127.0.0.1:3000/api/contacts">http://127.0.0.1:3000/api/contacts</a> in URL bar. 3. Go to body. 4. Choose raw JSON format. 5. Enter first_name, last_name, city, phone_number and its values. 6. Save. 7. Click send button.	Contact is added properly to contacts list.	User receives "Contact added successfully" message following by its contact id. Contact is added properly.	201 Created	Passed
TC-002	Add user with no values	POST	1. User is connected with live server. 2. Contact details: "first_name": "", "last_name": "", "city": "", "phone_number": ""	1. Choose POST method. 2. Enter <a href="http://127.0.0.1:3000/api/contacts">http://127.0.0.1:3000/api/contacts</a> in URL bar. 3. Go to body. 4. Choose raw JSON format. 5. Enter first_name, last_name, city, phone_number and its values. 6. Save. 7. Click send button.	Contact cannot be added to contacts list	Contact is not added to contacts list	400 Bad request	Passed
TC-003	Get entire contacts list information.	GET	1. User is connected with live server. 2. Contacts list is created and added to database.	1. Choose GET method. 2. Enter <a href="http://127.0.0.1:3000/api/contacts">http://127.0.0.1:3000/api/contacts</a> in URL bar. 3. Click 'Send' button.	Contacts list and its content should be visible in 'Body' field. User should be able to see all informations about each contact in database.	User is able to see all contacts informations.	200 OK	Passed
TC-004	Get single contact.	GET	1. User is connected with live server. 2. Contact id = 3 3. Contact with predefined id exists.	1. Choose GET method. 2. Enter <a href="http://127.0.0.1:3000/api/contacts/">http://127.0.0.1:3000/api/contacts/</a> : id in URL bar. 3. Enter existing Contact_id as a value in Path Variables. 4. Click 'Send' button.	Status code should be 200. User should be able to see all informations about choosen contact.	Chosen contact content is visible to a user.	200 OK	Passed
TC-005	Get single contact using non-existing contact id	GET	1. User is connected with live server. 2. Contact cannot exist in database. 3. Contact id = 5	1. Choose GET method. 2. Enter <a href="http://127.0.0.1:3000/api/contacts/">http://127.0.0.1:3000/api/contacts/</a> : id in URL bar. 3. Enter non-existing Contact_id as a value in Path Variables. 4. Click 'Send' button.	User should not be able to see chosen contact content. Instead, an error "Contact not found" should appear.	Error "Contact not found" appears. Chosen contact is not possible to be displayed.	404 Not Found	Passed
TC-006	Add existing user to contacts list.	POST	1. User is connected with live server. 2. Contact details: "first_name": "Felix", "last_name": "Rogahn", "city": "Susanberg", "phone_number": "471-537-0352"	1. Choose POST method. 2. Enter <a href="http://127.0.0.1:3000/api/contacts">http://127.0.0.1:3000/api/contacts</a> in URL bar. 3. Go to body. 4. Choose raw JSON format. 5. Enter existing first_name, last_name, city, phone_number and its values. 6. Save. 7. Click send button.	Contact should not be added to contacts list. Error message should be visible. Expected status code: 500	User can use existing contact data to create the same user, which receives another contact id.	201 Created	Failed

TC-007	Add new contact without filling in required data.	POST	1. User is connected with live server. 2. Contact data: "first_name": "Testname"	1. Choose POST method. 2. Enter http://127.0.0.1:3000/api/contacts in URL bar. 3. Go to body. 4. Choose raw JSON format. 5. Enter first_name, only. 6. Save. 7. Click send button.	Contact should not be created. Error "Internal Server Error" should appear.	Contact is not created. Internal server error appears.	500 Internal Server Error	Passed
TC-008	Update contact	PUT	1. User is connected with live server. 2. Contact exists in database. 3. Contact data: "first_name": "TestName", "last_name": "TestLastName", "city": "Warsaw", "phone_number": "122212213"	1. Choose PUT method. 2. Enter http://127.0.0.1:3000/api/contacts/:id in URL bar. 3. Enter existing contact id as a value in "Path Variables". 4. Go to body. 5. Choose raw JSON format. 6. Rename first_name, last_name, city and phone_number. 7. Save. 8. Click "Send" button.	As a response, message "Contact updated successfully" should appear.	Response "Contact updated successfully". All contact data updated.	204 No Content	Passed
TC-009	Update contact without filling in required data.	PUT	1. User is connected with live server. 2. Contact exists in database. 3. Contact_id = 4 4. Contact input: "last_name": "Test2"	1. Choose PUT method. 2. Enter http://127.0.0.1:3000/api/contacts/:id in URL bar. 3. Enter existing contact id as a value in "Path Variables". 4. Go to body. 5. Choose raw JSON format. 6. Fill in only last_name key. 7. Save. 8. Press "Send" button.	Contact should not be updated.	Contact update failed. "Contact data missed." should be visible.	400 Bad Request	Passed
TC-010	Update contact giving all the keys, but leaving blank values.	PUT	1. User is connected with live server. 2. Contact exists in database. 3. Contact_id = 4 4. Contact input: "first_name": "", "last_name": "", "city": "", "phone_number": ""	1. Choose PUT method. 2. Enter http://127.0.0.1:3000/api/contacts/:id in URL bar. 3. Enter existing contact id as a value in "Path Variables". 4. Go to body. 5. Choose raw JSON format. 6. Give only keys and leave all values empty. 7. Save. 8. Press "Send" button.	Contact should not be updated.	Contact update failed. "Contact data missed." should be visible.	400 Bad Request	Passed
TC-011	Update contact's single component.	PATCH	1. User is connected with live server. 2. Contact exists in database. 3. Contact input: "last_name": "Test2LastName"	1. Choose PATCH method. 2. Enter http://127.0.0.1:3000/api/contacts/:id in URL bar. 3. Enter contact id as a value in "Path Variables". 4. Go to body. 5. Choose raw JSON format. 6. Give last_name value. 7. Save. 8. Press "Send" button.	Contact should be updated successfully	Contact is updated.	200 OK	Passed

TC-012	Update contact's single component leaving value input empty.	PATCH	1. User is connected with live server. 2. Contact exists in database. 3. Contact_id = 4 4. Contact input: "first_name": ""	1. Choose PATCH method. 2. Enter http://127.0.0.1:3000/api/contacts/:id in URL bar. 3. Enter existing contact id as a value in "Path Variables". 4. Go to body. 5. Choose raw JSON format. 6. Leave first_name value empty. 7. Save. 8. Press "Send" button.	Contact should not be updated. An error should occur. Expected status code: 500	Contact is updated.	200 OK	Failed
TC-013	Update contact's single component leaving body empty.	PATCH	1. User is connected with live server. 2. Contact exists in database. 3. Contact_id = 4	1. Choose PATCH method. 2. Enter http://127.0.0.1:3000/api/contacts/:id in URL bar. 3. Enter existing contact id as a value in "Path Variables". 4. Go to body. 5. Choose raw JSON format. 6. Leave empty brackets in body field . 7. Save. 8. Press "Send" button.	Contact should not be updated. An error should occur	Contact is not updated. Internal server error appears.	500 Internal Server Error	Passed
TC-014	Update contact's single component of non-existing contact.	PATCH	1. User is connected with live server. 2. Contact is not created. 3. Contact_id = 20 4. Contact input: "phone_number": "123456789"	1. Choose PATCH method. 2. Enter http://127.0.0.1:3000/api/contacts/:id in URL bar. 3. Enter contact id as a value in "Path Variables". 4. Go to body. 5. Choose raw JSON format. 6. Enter phone_number as a key and 123456789 as a value. 7. Save. 8. Press "Send" button.	An error of non-existing contact should appear. Contact update fails. Expected status code: 500	Message "Contact is updated" received.	200 OK	Failed
TC-015	Delete existing contact.	Delete	1. User is connected with live server. 2. Contact_id = 5 3. Contact is created.	1. Choose DELETE method. 2. Enter http://127.0.0.1:3000/api/contacts/:id in URL bar. 3. Enter Contact_id as a value in "Path Variables". 4. Press "Send" button.	Contact is deleted from database.	Message "Contact deleted successfully" appears.	200 OK	Passed
TC-016	Delete non-existing contact.	Delete	1. User is connected with live server. 2. Contact is not created. 3. Contact_id = 21	1. Choose DELETE method. 2. Enter http://127.0.0.1:3000/api/contacts/:id in URL bar. 3. Enter non-existing Contact_id as a value in "Path Variables". 4. Press "Send" button.	Non-existing contact cannot be deleted.	Message "Contact not found" received.	404 Not Found	Passed