

## Team 1 2. Test cases

### Notes API - Agnes Nauka

Test case ID	N1 - Get the list of all notes
Description	Check if it is possible to retrieve the full list of all existing notes
Precondition	Some notes exists in the application
Test data	-
Test steps	1. Send <b>GET</b> <code>{{baseUrl}}/obj/note</code>
Expected result	<ul style="list-style-type: none"><li>• Successful code: 200 OK</li><li>• List of all notes is returned in response</li></ul>

Test case ID	N2 - Get the list of all notes and and sort them by creation date
Description	Check if it is possible to retrieve the full list of all existing notes sorted by the creation date
Precondition	Some notes exists in the application
Test data	-
Test steps	1. Send <b>GET</b> <code>{{baseUrl}}/obj/note?sort_field=Created Date</code>
Expected result	<ul style="list-style-type: none"><li>• Successful code: 200 OK</li><li>• List of all notes is returned in response and the list is sorted by the creation date</li></ul>

Test case ID	N3 - Get the list of all notes starting from cursor 1, not 0
Description	Check if it is possible to retrieve the full list of all existing notes starting from the cursor 1, leaving out the cursor 0
Precondition	Some notes exists in the application
Test data	-
Test steps	1. Send <b>GET</b> <code>{{baseUrl}}/obj/note?cursor=1</code>
Expected result	<ul style="list-style-type: none"><li>• Successful code: 200 OK</li><li>• List of all notes is returned in response and the list starts from cursor 1</li></ul>

Test case ID	N4 - Get the list of all notes and check param limit
Description	Check if it is possible to retrieve the list of notes when the limit is 10
Precondition	Some notes exists in the application
Test data	-

Test steps	1. Send <b>GET</b> <code>{{baseUrl}}/obj/note?limit=10</code>
Expected result	<ul style="list-style-type: none"> <li>Successful code: 200 OK</li> <li>List of notes is returned in response and the list shows 10 notes</li> </ul>

Test case ID	<b>N5 - Get the list of all notes and check boolean</b>
Description	Check if it is possible to retrieve the list of all notes sorted by the content_text ascending
Precondition	Some notes exists in the application
Test data	-
Test steps	1. Send <b>GET</b> <code>{{baseUrl}}/obj/note?sort_field=content_text&amp;descending=false</code>
Expected result	<ul style="list-style-type: none"> <li>Successful code: 200 OK</li> <li>List of notes is returned in response and the list shows all notes ascending by the content_text</li> </ul>

Test case ID	<b>N6 - Get the note when the constraints equals 'hey there'</b>
Description	Check if it is possible to retrieve only one note using constraints equals 'hey there'
Precondition	The note exists in the application
Test data	-
Test steps	1. Send <b>GET</b> <code>{{baseUrl}}/obj/note?constraints=[{"key": "content_text", "constraint_type": "equals", "value": "hey there" }]</code>
Expected result	<ul style="list-style-type: none"> <li>Successful code: 200 OK</li> <li>One note with the content_text 'hey there' is returned in response</li> </ul>

Test case ID	<b>N7 - Get the note when the constraint is text_contains 'client'</b>
Description	Check if it is possible to retrieve only one note using constraint text_contains 'client'
Precondition	The note exists in the application
Test data	-
Test steps	1. Send <b>GET</b> <code>{{baseUrl}}/obj/note?constraints=[{"key": "content_text", "constraint_type": "text contains", "value": "client" }]</code>
Expected result	<ul style="list-style-type: none"> <li>Successful code: 200 OK</li> <li>One note with the content_text that contains 'client' is returned in response</li> </ul>

Test case ID	<b>N8 - Get the note when the constraint is empty</b>
--------------	---

Description	Check if it is possible to retrieve only one note using constraint is empty
Precondition	The note exists in the application
Test data	-
Test steps	1. Send <b>GET</b> <code>{{baseUrl}}/obj/note?constraints=</code>
Expected result	<ul style="list-style-type: none"> <li>400 Bad request is expected</li> <li>Error message is returned in response <code>"status": "INVALID_DATA",</code>  <code>"message": "Issue with data for endpoint note: cannot be parsed as a JSON"</code></li> </ul>

<b>Test case ID</b>	<b>N9 - Get the note with unique ID</b>
Description	Check if it is possible to retrieve the note by unique ID, for example 1627393624856x199214596987551740
Precondition	The note exists in the application
Test data	-
Test steps	1. Send <b>GET</b> <code>{{baseUrl}}/obj/note/:UniqueID</code>
Expected result	<ul style="list-style-type: none"> <li>Successful code: 200 OK</li> <li>One note with the unique ID is returned in response</li> </ul>

<b>Test case ID</b>	<b>N10 - Create a new note and check the created object</b>
Description	Check if it is possible to create a new note 'Test for team 1' and observe if it was created successfully with given data
Precondition	-
Test data	<pre> 1 { 2   "content_text": "Test for team 1" 3 }</pre>
Test steps	1. Send <b>POST</b> <code>{{baseUrl}}/obj/note</code> request using body from test data 2. Copy id of the created note from the response 3. Send <b>GET</b> <code>{{baseUrl}}/obj/note/:UniqueID</code> , set correct UniqueID (taken from step 2)
Expected result	Step 1: Successful code: 201 Created, note is created  Step 3: Successful code: 200 OK, correct note is found, data corresponds to input

<b>Test case ID</b>	<b>N11 - Create a new note without any info</b>
---------------------	---

Description	Check if it is possible to create a new note without any info (content_text is empty)
Precondition	-
Test data	<pre> 1 { 2   "content_text": 3 }</pre>
Test steps	1. Send <b>POST</b> <code>{{baseUrl}}/obj/note</code>
Expected result	<ul style="list-style-type: none"> <li>• 400 Bad request is expected</li> <li>• Error message is returned in response</li> </ul>

Test case ID	<b>N12 -Create, Delete note and check data</b>
Description	Check if it is possible to delete a note after creating one, then check if it was deleted
Precondition	-
Test data	<pre> 1 { 2   "content_text": "123" 3 }</pre>
Test steps	<ol style="list-style-type: none"> <li>1. Send <b>POST</b> <code>{{baseUrl}}/obj/note</code> request using body from test data</li> <li>2. Copy id of the created note from the response</li> <li>3. Send <b>DELETE</b> <code>{{baseUrl}}/obj/lead/:UniqueID</code> , set correct UniqueID (taken from step 2)</li> <li>4. Send <b>GET</b> <code>{{baseUrl}}/obj/lead/:UniqueID</code> , set correct UniqueID (taken from step 2)</li> </ol>
Expected result	<p>Step 1: Successful code: 201 Created, note is created</p> <p>Step 3: Successful code: 204 No Content, delete was successful</p> <p>Step 4: Error code: 404 Not Found, "status": "MISSING_DATA", "message": "Missing object of type note: object with id {{UniqueID}} does not exist"</p>

Test case ID	<b>N13 - Create a new note when content_text is boolean</b>
Description	Check if it is possible to create a new note when content_text is boolean
Precondition	-
Test data	<pre> 1 { 2   "content_text": true 3 }</pre>
Test steps	1. Send <b>POST</b> <code>{{baseUrl}}/obj/note</code> request using body from test data

Expected result	<ul style="list-style-type: none"> <li>• 400 Bad request is expected</li> <li>• Error message is returned in response <code>"status": "INVALID_DATA", "message": "Invalid data for field content_text: Expected a string, but got a boolean (original data: true)"</code></li> </ul>
-----------------	--

Test case ID	<b>N14 - Create a note and check the created object</b>
Description	Check if it is possible to create a new note and observe if it was created successfully with given data
Precondition	-
Test data	<pre> 1 { 2   "content_text": "There are 3 members in this team" 3 }</pre>
Test steps	<ol style="list-style-type: none"> <li>1. Send <b>POST</b> <code>{baseUrl}/obj/note</code> request using body from test data</li> <li>2. Copy id of the created note from the response</li> <li>3. Send <b>GET</b> <code>{baseUrl}/obj/note/:UniqueID</code>, set correct UniqueID (taken from step 2)</li> </ol>
Expected result	<p>Step 1: Successful code: 201 Created, note is created</p> <p>Step 3: Successful code: 200 OK, correct note is found, data corresponds to input.</p>

Test case ID	<b>N15 - Update a note and check the data</b>
Description	Check if it is possible to update created note and observe if it was updated successfully
Precondition	N14 is successfully completed
Test data	<pre> 1 { 2   "content_text": "There are 11 members in this team" 3 }</pre>
Test steps	<ol style="list-style-type: none"> <li>1. Send <b>PATCH</b> <code>{baseUrl}/obj/note/:UniqueID</code>, set correct UniqueID (taken from step 2, N14)</li> <li>2. Send <b>GET</b> <code>{baseUrl}/obj/note/:UniqueID</code>, set correct UniqueID (taken from previous step)</li> </ol>
Expected result	<p>Step 1: Successful code: 204 No Content, update was successful</p> <p>Step 2: Successful code: 200 OK, correct note is found, note was successfully updated</p>

## Deals API - Tuuli Pesur

Test case ID	<b>D1 - Read a list of all deals</b>
--------------	--------------------------------------

Description	Check if it's possible to read a list of all deals.
Precondition	List of some deals exist.
Test data	-
Test steps	1. Send <b>GET</b> {baseUrl}/obj/deal .
Expected result	<ul style="list-style-type: none"> <li>• Success code: 200 OK.</li> <li>• List of all deals is shown.</li> </ul>

<b>Test case ID</b>	<b>D2 - Read a list of all deals with search constraints</b>
Description	Check if it's possible to read a list of all deals with adding search constraints.
Precondition	List of some deals exist.
Test data	Constraints: [{"key": "status_option_status", "constraint_type": "equals", "value": "Won" }]
Test steps	1. Send <b>GET</b> {{baseUrl}}/obj/deal?constraints=[{"key": "status_option_status", "constraint_type": "equals", "value": "Won" }]
Expected result	<ul style="list-style-type: none"> <li>• Success code: 200 OK.</li> <li>• List of all deals with searched constraint appears.</li> </ul>

<b>Test case ID</b>	<b>D3 - Search a deal with specific title (filtering by name)</b>
Description	Check if it's possible to search a deal with specific title (filtering by name).
Precondition	Deal with this specific title must exist.
Test data	Title "Test1"
Test steps	1. Send <b>GET</b> {{baseUrl}}/obj/deal?constraints=[{"key": "company_name_text", "constraint_type": "equals", "value": "Test1"}]
Expected result	<ul style="list-style-type: none"> <li>• Success code: 200 OK.</li> <li>• Deal with searched title is shown.</li> </ul>

<b>Test case ID</b>	<b>D4 - Search a deal with wrong not existing conditions</b>
Description	Check if it's possible to search a deal with wrong not existing conditions (proper error handling in expected).
Precondition	A deal with conditions in search does not exist.
Test data	Company name that doesn't exist in the list. In this case "Buildco".
Test steps	1. Send <b>GET</b> {{baseUrl}}/obj/deal?constraints=[{"key": "company_text", "constraint_type": "equals", "value": "Buildco"}]
Expected result	<ul style="list-style-type: none"> <li>• Error code: Error 404.</li> </ul>

	<ul style="list-style-type: none"> <li>• A deal with searched name doesn't appear.</li> </ul>
--	---

Test case ID	D5 - Read a deal data by its unique Id
Description	Check if it's possible to read a deal data by it's unique ID.
Precondition	Data by this ID is created.
Test data	UniqueID
Test steps	1. Send <b>GET</b> <code>{{baseUrl}}/obj/deal/:UniqueID</code> and enter previously existing deal's ID.
Expected result	<ul style="list-style-type: none"> <li>• Success code: 200 OK is presented with deal's information.</li> </ul>

Test case ID	D6 - Read a deal with random not existing ID
Description	Check if it's possible to read a deal with random not existing ID.
Precondition	Deal with searched ID doesn't exist.
Test data	ID that doesn't exist. In this case "1695974436660x97513119675864330".
Test steps	1. Send <b>GET</b> <code>{{base URL}}/obj/deal?constraints=[{"key": "_id", "constraint_type": "equals", "value": "1695974436660x97513119675864330"}]</code>
Expected result	<ul style="list-style-type: none"> <li>• Error code: Error 404.</li> <li>• Lead with this specific not existing ID doesn't appear.</li> </ul>

Test case ID	D7 - Create a new deal
Description	Check if it's possible to create new deal.
Precondition	Authorization is needed.
Test data	<pre> 1  { 2    "assignee__user": "1625047362532x398219546419355650", 3    "company_name_text": "Test1", 4    "deal_value_estimation_number": 0, 5    "description_text": "Test1", 6    "file_list_custom_data_file": [], 7    "name_text": "Test1", 8    "note_list_custom_note": [], 9    "order_number": 0, 10   "stage_custom_stage": "1625047822374x314234657054522700", 11   "status_option_status": "Won", 12   "visible_to_list_user": [ 13     "1625047362532x398219546419355650" 14   ] 15 }</pre>
Test steps	1. Send <b>POST</b> <code>{baseUrl}/obj/deal</code> and enter correct JSON text. 2. Send <b>GET</b> <code>{baseUrl}/obj/deal/:UniqueID</code> with UniqueID received in the last step.

Expected result	<ol style="list-style-type: none"> <li>1. Success code: Status 201 Created and ID for the new deal shown.</li> <li>2. Success code: Status 200 OK and new deal is shown.</li> </ol>
-----------------	---

Test case ID	<b>D8 - Create a new deal without authorization</b>
Description	Check if it's possible to create new deal without entering authorization key.
Precondition	No authorization.
Test data	<pre> 1  { 2    "assignee__user": "1625047362532x398219546419355650", 3    "company_name_text": "Test1", 4    "deal_value_estimation_number": 0, 5    "description_text": "Test1", 6    "file_list_custom_data_file": [], 7    "name_text": "Test1", 8    "note_list_custom_note": [], 9    "order_number": 0, 10   "stage_custom_stage": "1625047822374x314234657054522700", 11   "status_option_status": "Won", 12   "visible_to_list_user": [ 13     "1625047362532x398219546419355650" 14   ] 15  } </pre>
Test steps	1. Send <b>POST</b> {baseUrl}/obj/deal and enter correct JSON text.
Expected result	<ul style="list-style-type: none"> <li>• Error code: Status 401 Unauthorized.</li> <li>• New deal is not created.</li> </ul>

Test case ID	<b>D9 - Create a new deal without assignee user text</b>
Description	Check if it's possible to create a new deal without adding assignee user text.
Precondition	All other requirements are filled except "assignee user" text.
Test data	<pre> 1  { 2    "company_name_text": "Test1", 3    "deal_value_estimation_number": 0, 4    "description_text": "Test1", 5    "file_list_custom_data_file": [], 6    "name_text": "Test1", 7    "note_list_custom_note": [], 8    "order_number": 0, 9    "stage_custom_stage": "1625047822374x314234657054522700", 10   "status_option_status": "Won", 11   "visible_to_list_user": [ 12     "1625047362532x398219546419355650" 13   ] 14  } </pre>
Test steps	1. Send <b>POST</b> {baseUrl}/obj/deal and enter test data JSON text.
Expected result	<ul style="list-style-type: none"> <li>• Error code: Status 400 Bad request.</li> <li>• New deal is not created.</li> </ul>



Test case ID	D10 - Create a new deal without company name text
Description	Check if it's possible to create a new deal without adding company name text.
Precondition	All other requirements are filled except "company name text".
Test data	<pre> 1  { 2    "assignee__user": "1625047362532x398219546419355650", 3    "deal_value_estimation_number": 0, 4    "description_text": "Test1", 5    "file_list_custom_data_file": [], 6    "name_text": "Test1", 7    "note_list_custom_note": [], 8    "order_number": 0, 9    "stage_custom_stage": "1625047822374x314234657054522700", 10   "status_option_status": "Won", 11   "visible_to_list_user": [ 12     "1625047362532x398219546419355650" 13   ] 14 }</pre>
Test steps	1. Send <b>POST</b> {baseUrl}/obj/deal and enter test data JSON text.
Expected result	<ul style="list-style-type: none"> <li>• Error code: Status 400 Bad request.</li> <li>• New deal is not created.</li> </ul>

Test case ID	D11 - Create a new deal without deal value estimation number
Description	Check if it's possible to create a new deal without adding deal value estimation number.
Precondition	All other requirements are filled except "deal value estimation".
Test data	<pre> 1  { 2    "assignee__user": "1625047362532x398219546419355650", 3    "company_name_text": "Test1", 4    "description_text": "Test1", 5    "file_list_custom_data_file": [], 6    "name_text": "Test1", 7    "note_list_custom_note": [], 8    "order_number": 0, 9    "stage_custom_stage": "1625047822374x314234657054522700", 10   "status_option_status": "Won", 11   "visible_to_list_user": [ 12     "1625047362532x398219546419355650" 13   ] 14 }</pre>
Test steps	1. Send <b>POST</b> {baseUrl}/obj/deal and enter test data JSON text.
Expected result	<ul style="list-style-type: none"> <li>• Error code: Status 400 Bad request.</li> <li>• New deal is not created.</li> </ul>

Test case ID	D12 - Create a new deal without name text
Description	Check if it's possible to create a new deal without adding name text.

Precondition	All other requirements are filled except " <i>name text</i> ".
Test data	<pre> 1  { 2    "assignee__user": "1625047362532x398219546419355650", 3    "company_name_text": "Test1", 4    "deal_value_estimation_number": 0, 5    "description_text": "Test1", 6    "file_list_custom_data_file": [], 7    "note_list_custom_note": [], 8    "order_number": 0, 9    "stage_custom_stage": "1625047822374x314234657054522700", 10   "status_option_status": "Won", 11   "visible_to_list_user": [ 12     "1625047362532x398219546419355650" 13   ] 14 }</pre>
Test steps	1. Send <b>POST</b> {baseUrl}/obj/deal and enter test data JSON text.
Expected result	<ul style="list-style-type: none"> <li>• Error code: Status 400 Bad request.</li> <li>• New deal is not created.</li> </ul>

Test case ID	<b>D13 - Create a new deal without order number</b>
Description	Check if it's possible to create a new deal without adding order number.
Precondition	All other requirements are filled except " <i>order number</i> ".
Test data	<pre> 1  { 2    "assignee__user": "1625047362532x398219546419355650", 3    "company_name_text": "Test1", 4    "deal_value_estimation_number": 0, 5    "description_text": "Test1", 6    "file_list_custom_data_file": [], 7    "name_text": "Test1", 8    "note_list_custom_note": [], 9    "stage_custom_stage": "1625047822374x314234657054522700", 10   "status_option_status": "Won", 11   "visible_to_list_user": [ 12     "1625047362532x398219546419355650" 13   ] 14 }</pre>
Test steps	1. Send <b>POST</b> {baseUrl}/obj/deal and enter test data JSON text.
Expected result	<ul style="list-style-type: none"> <li>• Error code: Status 400 Bad request.</li> <li>• New deal is not created.</li> </ul>

Test case ID	<b>D14 - Create a new deal without status option status</b>
Description	Check if it's possible to create a new deal without adding status option status text.
Precondition	All other requirements are filled except " <i>status option status</i> ".
Test data	<pre> 1  { 2    "assignee__user": "1625047362532x398219546419355650", 3    "company_name_text": "Test1",</pre>

	<pre> 4    "deal_value_estimation_number": 0, 5    "description_text": "Test1", 6    "file_list_custom_data_file": [], 7    "name_text": "Test1", 8    "note_list_custom_note": [], 9    "order_number": 0, 10   "stage_custom_stage": "1625047822374x314234657054522700", 11   "visible_to_list_user": [ 12     "1625047362532x398219546419355650" 13   ] 14 }</pre>
Test steps	1. Send <b>POST</b> {baseUrl}/obj/deal and enter test data JSON text.
Expected result	<ul style="list-style-type: none"> <li>• Error code: Status 400 Bad request.</li> <li>• New deal is not created.</li> </ul>

<b>Test case ID</b>	<b>D15 - Create a new deal without visible to list user text</b>
Description	Check if it's possible to create a new deal without adding visible to list user text.
Precondition	All other requirements are filled except "visible to list user".
Test data	<pre> 1  { 2    "assignee__user": "1625047362532x398219546419355650", 3    "company_name_text": "Test1", 4    "deal_value_estimation_number": 0, 5    "description_text": "Test1", 6    "file_list_custom_data_file": [], 7    "name_text": "Test1", 8    "note_list_custom_note": [], 9    "order_number": 0, 10   "stage_custom_stage": "1625047822374x314234657054522700", 11   "status_option_status": "Won", 12 }</pre>
Test steps	1. Send <b>POST</b> {baseUrl}/obj/deal and enter test data JSON text.
Expected result	<ul style="list-style-type: none"> <li>• Error code: Status 400 Bad request.</li> <li>• New deal is not created.</li> </ul>

<b>Test case ID</b>	<b>D16 - When creating a new deal entering string text instead of integer</b>
Description	Check if it's possible to create a new deal with having string text instead of integer in <i>deal_value_estimation_number</i> and <i>order_number</i> .
Precondition	All other requirements are filled correctly.
Test data	<pre> 1  { 2    "assignee__user": "1625047362532x398219546419355650", 3    "company_name_text": "Test1", 4    "deal_value_estimation_number": TEST, 5    "description_text": "Test1", 6    "file_list_custom_data_file": [], 7    "name_text": "Test1", 8    "note_list_custom_note": [], 9    "order_number": TEST,</pre>

	<pre> 10  "stage_custom_stage": "1625047822374x314234657054522700", 11  "status_option_status": "won", 12  "visible_to_list_user": [ 13      "1625047362532x398219546419355650" 14  ] 15  } </pre>
Test steps	1. Send <b>POST</b> {baseUrl}/obj/deal and enter test data JSON text.
Expected result	<ul style="list-style-type: none"> <li>• Error code: Status 400 Bad request.</li> <li>• New deal is not created.</li> </ul>

Test case ID	<b>D17 - Fill “status_option_status” with something other than won, in progress, lost.</b>
Description	Check if it's possible to create a new deal with having some other string text entered into the “status_option_status” than expected results (won, in progress, lost).
Precondition	All other requirements are filled correctly.
Test data	<pre> 1  { 2      "assignee__user": "1625047362532x398219546419355650", 3      "company_name_text": "Test1", 4      "deal_value_estimation_number": 0, 5      "description_text": "Test1", 6      "file_list_custom_data_file": [], 7      "name_text": "Test1", 8      "note_list_custom_note": [], 9      "order_number": 0, 10     "stage_custom_stage": "1625047822374x314234657054522700", 11     "status_option_status": "TEST", 12     "visible_to_list_user": [ 13         "1625047362532x398219546419355650" 14     ] 15 } </pre>
Test steps	1. Send <b>POST</b> {baseUrl}/obj/deal and enter test data JSON text.
Expected result	<ul style="list-style-type: none"> <li>• Error code: Status 400 Bad request.</li> <li>• New deal is not created.</li> </ul>

Test case ID	<b>D18 - Modify lead by UniqueID</b>
Description	Check if it's possible to modify the data of existing deal by it's UniqueID.
Precondition	Data with this ID exists and valid authorization entered.
Test data	<p>Unique ID 1696326541475x671956756528809300</p> <p>Company_name_text is modified - Test1 ➡ Test 12</p> <pre> 1  { 2      "assignee__user": "1625047362532x398219546419355650", 3      "company_name_text": "Test12", 4      "deal_value_estimation_number": 0, 5      "description_text": "Test1", 6      "file_list_custom_data_file": [], </pre>

	<pre> 7  "note_list_custom_note": [], 8  "order_number": 0, 9  "stage_custom_stage": "1625047822374x314234657054522700", 10 "status_option_status": "Won", 11 "visible_to_list_user": [ 12     "1625047362532x398219546419355650" 13 ] 14 }</pre>
Test steps	<ol style="list-style-type: none"> <li>1. Send <b>PATCH</b> <code>{{baseUrl}}/obj/deal/:UniqueID</code> and enter test data JSON text.</li> <li>2. Send <b>GET</b> <code>{{baseUrl}}/obj/deal/:UniqueID</code> and enter test data UniqueID.</li> </ol>
Expected result	<ol style="list-style-type: none"> <li>1. Success code: Status 204 No content is shown.</li> <li>2. Success code: Status 200 OK and modified lead is shown.</li> </ol>

Test case ID	<b>D19 - Try to modify not existing deal with random ID</b>
Description	Check if it's possible to try to modify not existing deal with random ID (proper error handling is expected).
Precondition	Lead with this Unique ID doesn't exist.
Test data	<p>Unique ID 1696326541475x6719567565288093</p> <p>Company_name_text is modified - Test1 ➡ Test 12</p> <pre> 1  { 2    "assignee__user": "1625047362532x398219546419355650", 3    "company_name_text": "Test12", 4    "deal_value_estimation_number": 0, 5    "description_text": "Test1", 6    "file_list_custom_data_file": [], 7    "note_list_custom_note": [], 8    "order_number": 0, 9    "stage_custom_stage": "1625047822374x314234657054522700", 10   "status_option_status": "Won", 11   "visible_to_list_user": [ 12       "1625047362532x398219546419355650" 13   ] 14 }</pre>
Test steps	<ul style="list-style-type: none"> <li>• Send <b>PATCH</b> <code>{{baseUrl}}/obj/deal/:UniqueID</code> and enter test data JSON text.</li> </ul>
Expected result	<ul style="list-style-type: none"> <li>• Error code: Status 404 Not Found.</li> <li>• No lead is modified.</li> </ul>

Test case ID	<b>D20 - Replace the deal data with a new set of data</b>
Description	Check if it's possible to replace the deal data with a new set of data.
Precondition	
Test data	<p>UniqueID 1696343382927x352045287752024100</p> <p>Old JSON text:</p>

	<pre> 1  { 2    "assignee__user": "1625047362532x398219546419355650", 3    "company_name_text": "Test1", 4    "description_text": "Test1", 5    "file_list_custom_data_file": [], 6    "name_text": "Test1", 7    "note_list_custom_note": [], 8    "order_number": 0, 9    "stage_custom_stage": "1625047822374x314234657054522700", 10   "status_option_status": "Won", 11   "visible_to_list_user": [ 12     "1625047362532x398219546419355650" 13   ] 14 }</pre> <p>New JSON text:</p> <pre> 1  { 2    "assignee__user": "1625047362532x398219546419355650", 3    "company_name_text": "Test12", 4    "deal_value_estimation_number": 0, 5    "description_text": "Test12", 6    "file_list_custom_data_file": [], 7    "name_text": "Test12", 8    "note_list_custom_note": [], 9    "order_number": 0, 10   "stage_custom_stage": "1625047822374x314234657054522700", 11   "visible_to_list_user": [ 12     "1625047362532x398219546419355650" 13   ] 14 }</pre>
Test steps	<ul style="list-style-type: none"> <li>Send <b>PUT</b> <code>{{baseUrl}}/obj/deal/:UniqueID</code> and fill new test data JSON text</li> </ul>
Expected result	<ul style="list-style-type: none"> <li>Successful code: 204 No content</li> <li>Data is replaced without any errors</li> </ul>

<b>Test case ID</b>	<b>D21 - Delete existing deal without authorization</b>
Description	Check if it's possible to delete existing deal without entering authorization.
Precondition	No authorization filled and deal exists.
Test data	Deal ID 1625048593556x124292100920442880
Test steps	<ol style="list-style-type: none"> <li>Remove authorization from the request.</li> <li>Send <b>DELETE</b> <code>{{baseUrl}}/obj/deal/:UniqueID</code></li> </ol>
Expected result	<ol style="list-style-type: none"> <li>Authorization is removed.</li> <li>Error code: 401 Permission denied and lead is not deleted.</li> </ol>

<b>Test case ID</b>	<b>D22 - Delete existing deal with authorization</b>
Description	Check if it's possible to delete existing deal.
Precondition	Authorization filled and deal exists.

Test data	Deal ID 1625048593556x124292100920442880
Test steps	<ol style="list-style-type: none"><li>1. Send <b>DELETE</b> {{baseUrl}}/obj/lead/:UniqueID</li><li>2. Send <b>GET</b> {{baseUrl}}/obj/lead/:UniqueID</li></ol>
Expected result	<ol style="list-style-type: none"><li>1. Success code: 204 Success code and deal is deleted.</li><li>2. Error code: 404 Not found code and deal is not found.</li></ol>