

2.1 Deals API

- [Deals API](#)
 - Get the list of all deals
 - Get deal by unique ID
 - Create a new deal
 - Modify deal data
 - Replace deal data
 - Delete deal

CRM app baseUrl: <https://cerebrumhubcrmlite.bubbleapps.io/api/1.1>

Bearer token, aka authToken: b30ca93d0a7ebe389e9326a22497bf38

Deals API

Get the list of all deals

Test case ID	D1.1 Get a list of all deals
Description	It is possible to retrieve the list of all deals, by default list size is 50
Precondition	Some deals exists in the application
Test data	
Test steps	1. Send <code>GET {{baseUrl}}/obj/deal</code>
Expected result	<ul style="list-style-type: none">Successful code: 200 OKList of 50 deals is returned in response, if there is atleast 50 deals exists in the system. If there is less than 50 deals, then is returned the number of deals that exists.

Test case ID	D1.2 Get a list of deals without authorization
Description	Verify that unauthorized access attempts to retrieve the list of deals are denied.
Precondition	Some deals exists in the application
Test data	
Test steps	1. Send <code>GET {{baseUrl}}/obj/deal</code>
Expected result	<ul style="list-style-type: none">Error code: 401 Permission deniedList of all deals is not returned in response

Test case ID	D1.3 Get a list of all deals - maximum limit value 100
Description	Verify that it's possible to retrieve a list of all deals with a maximum limit value of 100.
Precondition	More than 100 deals exist in the application

Test data	
Test steps	1. Send <code>GET {{baseUrl}}/obj/deal</code> with parameter limit set to 100
Expected result	<ul style="list-style-type: none"> Successful code: 200 OK List of 100 deals is returned in response. If there is less than 100 deals in the application, then is returned the number of deals that exist.

Test case ID	D1.4 Get a list of all deals - limit value 101
Description	Verify that it is not possible to retrieve a list of all deals with a limit value of 101.
Precondition	More than 100 deals exist in the application
Test data	
Test steps	1. Send <code>GET {{baseUrl}}/obj/deal</code> with parameter limit set to 101
Expected result	<ul style="list-style-type: none"> Successful code: 200 OK List of 100 deals is returned in response. If there is less than 100 deals in the application, then is returned the number of deals that exist.

Test case ID	D1.5 Get a list of all deals - limit value -10
Description	Verify that the retrieval of a list of all deals with a negative limit value is not possible.
Precondition	Some deals exists in the application
Test data	
Test steps	1. Send <code>GET {{baseUrl}}/obj/deal</code> with parameter limit set to -10
Expected result	<ul style="list-style-type: none"> Error code and message are shown

Test case ID	D1.6 Get a list of all deals - limit value 0
Description	Verify that it is not possible to retrieve a list of all deals with a limit value of 0.
Precondition	Some deals exists in the application
Test data	
Test steps	1. Send <code>GET {{baseUrl}}/obj/deal</code> with parameter limit set to 0
Expected result	<ul style="list-style-type: none"> Error code and message are shown

Test case ID	D1.7 Get a list of all deals - limit non integer
Description	Verify that attempting to retrieve a list of all deals with a limit value that is not an integer is not allowed.
Precondition	Some deals exists in the application
Test data	

Test steps	1. Send <code>GET {{baseUrl}}/obj/deal</code> with parameter limit set to non integer "text".
Expected result	<ul style="list-style-type: none"> Error code and message are shown

Test case ID	D1.8 Get a list of all deals - cursor default value 0
Description	Verify that get a list of all deals cursor default value is 0.
Precondition	Some deals exists in the application
Test data	
Test steps	1. Send <code>GET {{baseUrl}}/obj/deal</code>
Expected result	<ul style="list-style-type: none"> Successful code: 200 OK The cursor value in the response is set to 0 by default.

Test case ID	D1.9 Get a list of all deals - cursor -10
Description	Verify that the retrieval of a list of all deals with a negative cursor value is not possible.
Precondition	Some deals exists in the application
Test data	
Test steps	1. Send <code>GET {{baseUrl}}/obj/deal</code> with cursor set to -10
Expected result	<ul style="list-style-type: none"> Error code and message are shown

Test case ID	D1.10 Get a list of all deals - cursor 50
Description	Verify that it's possible to retrieve a list of all deals with cursor value of 50.
Precondition	More than 50 deals exist in the application
Test data	
Test steps	1. Send <code>GET {{baseUrl}}/obj/deal</code> with cursor set to 50
Expected result	<ul style="list-style-type: none"> Successful code: 200 OK The cursor value in the response is set to 50.

Test case ID	D1.11 Get a list of all deals - cursor non integer
Description	Verify that attempting to retrieve a list of all deals with a non-integer cursor value is not possible.
Precondition	Some deals exists in the application
Test data	
Test steps	1. Send <code>GET {{baseUrl}}/obj/deal</code> with cursor set to non integer "text"
Expected result	<ul style="list-style-type: none"> Error code and message are shown

Test case ID	D1.12 Get a list of all deals - sorted by deal name
Description	Verify that you can retrieve the complete list of deals sorted by the deal names.
Precondition	Some deals exists in the application
Test data	
Test steps	1. Send <code>GET {{baseUrl}}/obj/deal</code> with sort_field parameter and the value name_text
Expected result	<ul style="list-style-type: none"> Successful code: 200 OK The list of deals is sorted by deal names

Test case ID	D1.13 Get a list of all deals - sort descending
Description	Verify that you can retrieve the complete list of deals sorted by the deal names in descending order.
Precondition	Some deals exists in the application
Test data	
Test steps	1. Send <code>GET {{baseUrl}}/obj/deal</code> with sort_field parameter value set to name_text and descending to true
Expected result	<ul style="list-style-type: none"> Successful code: 200 OK The list of deals is sorted by deal names in descending order.

Test case ID	D1.14 Get a list of all deals - sort ascending
Description	Verify that you can retrieve the complete list of deals sorted by the deal names in ascending order.
Precondition	Some deals exists in the application
Test data	
Test steps	1. Send <code>GET {{baseUrl}}/obj/deal</code> with sort_field parameter set to name_text and descending to false
Expected result	<ul style="list-style-type: none"> Successful code: 200 OK The list of deals is sorted by deal names in ascending order.

Test case ID	D1.15 Get a list of all deals in funnel "Lead acquisition" (filtering)
Description	The "Lead Acquisition" funnel is available and contains deals.
Precondition	
Test data	"Lead Acquisition" funnel ID: 1678717895619x961701523337510900
Test steps	1. Send <code>GET {{baseUrl}}/obj/deal</code> with constraints set to <code>[{"key": "funnel_custom_funnel1", "constraint_type": "equals", "value": "1678717895619x961701523337510900"}]</code>

Expected result	<ul style="list-style-type: none"> Successful code: 200 OK All the deals in the response are part of the "Lead Acquisition" funnel.
-----------------	---

Test case ID	D1.16 Search a deal with existing name
Description	Verify that the system successfully retrieves a deal using a valid and existing deal name in the search conditions.
Precondition	At least one deal with the specified name exists in the system.
Test data	In step one search for deal with name: "Pirate", but you can replace it with company of your choice
Test steps	1. Send <code>GET {{baseUrl}}/obj/deal</code> with constraints set to <code>[{"key": "name_text", "constraint_type": "equals", "value": "Pirate"}]</code>
Expected result	<ul style="list-style-type: none"> Successful code: 200 OK The response body contains a representation of the deal with the specified name, confirming its existence.

Test case ID	D1.17 Search a deal with incorrect or non-existing deal name (proper error handling is expected)
Description	Verify that it is not possible to for a deal using an incorrect or non-existing deal name.
Precondition	Deal with given name does not exist in the system
Test data	Use randomly generated company name
Test steps	1. Send <code>GET {{baseUrl}}/obj/deal</code> with constraints set to <code>[{"key": "name_text", "constraint_type": "equals", "value": "{\$randomCompanyName}"}]</code>
Expected result	<ul style="list-style-type: none"> Successful code: 200 OK Empty list is returned in response

Get deal by unique ID

Test case ID	D1.18 Get deal by existing Id
Description	Verify that it is possible to retrieve the full information of a deal by unique Id
Precondition	A deal with unique Id exists in the application
Test data	-
Test steps	1. Send <code>GET {{baseUrl}}/obj/deal/:UniqueId</code>
Expected result	<ul style="list-style-type: none"> Successful code: 200 OK A deal matching the Id is returned in response

Test case ID	D1.19 Get deal by Id unauthorized
--------------	--

Description	Verify that it is not possible to retrieve the full information of a deal by unique Id unauthorized
Precondition	A deal with unique Id exists in the application
Test data	-
Test steps	<ol style="list-style-type: none"> Send <code>GET {{baseUrl}}/obj/deal/:UniqueId</code> Change authorization type to No Auth
Expected result	<ul style="list-style-type: none"> Error code: 401 Unauthorized No deal is returned in response

Test case ID	D1.20 Get deal by empty Id value
Description	Verify that when leaving UniqueId value empty, the request response is error code: 404 Not Found
Precondition	Some deals with unique Id exists in the application
Test data	-
Test steps	<ol style="list-style-type: none"> Send <code>GET {{baseUrl}}/obj/deal/:UniqueId</code> Leave UniqueId value empty
Expected result	<ul style="list-style-type: none"> Error code: 404 Not Found No deal is shown, error message is displayed <code>"message": "Missing object of type deal: object with id :UniqueId does not exist"</code>

Test case ID	D1.21 Get deal by partial Id value
Description	Verify that when entering partial id to UniqueId value, the request response is error code: 404 Not Found
Precondition	Some deals with unique Id exists in the application
Test data	-
Test steps	<ol style="list-style-type: none"> Send <code>GET {{baseUrl}}/obj/deal/:UniqueId</code> Fill UniqueId value with partial Id
Expected result	<ul style="list-style-type: none"> Error code: 404 Not Found No deal is shown, error message is displayed <code>"message": "Missing object of type deal: object with id does not exist"</code>

Test case ID	D1.22 Get deal by not existent Id
Description	Verify that when entering non existent Id to UniqueId value, the request response is error code: 404 Not Found
Precondition	Some deals with unique Id exists in the application
Test data	<code>{{\$randomWords}}</code>
Test steps	<ol style="list-style-type: none"> Send <code>GET {{baseUrl}}/obj/deal/:UniqueId</code>

	2. Fill UniqueId value test data
Expected result	<ul style="list-style-type: none"> Error code: 404 Not Found No deal is shown, error message is displayed "message": "Missing object of type deal: object with id does not exist"

Test case ID	D1.23 Get deal by different Id (stage Id)
Description	Verify that when entering different kind of Id (e.g stage Id) to UniqueId value, the request response is error code: 404 Not Found
Precondition	A deal exists in the application with a stage linked to it.
Test data	Stage Id
Test steps	<ol style="list-style-type: none"> Send GET {{baseUrl}}/obj/deal/:UniqueId Fill UniqueId value stage Id
Expected result	<ul style="list-style-type: none"> Error code: 404 Not Found No deal is shown, error message is displayed "message": "Missing object of type deal: object with id does not exist"

Create a new deal

Test case ID	D2.1 Create new Deal with minimum dataset
Description	Check that it's possible to create new Deal with only mandatory fields.
Precondition	-
Test data	<pre> 1 { 2 "assignee__user": "1625047373081x778685405315052900", 3 "company_name_text": "Team1 tester", 4 "deal_value_estimation_number": 1, 5 "name_text": "Team1", 6 "order_number": 1, 7 "status_option_status": "In progress", 8 "visible_to_list_user": [9 "1625047373081x778685405315052900" 10] 11 }</pre>
Test steps	1. Send POST {{baseUrl}}/obj/deal with only required fields.
Expected result	<ul style="list-style-type: none"> Successful code: 201 Created. Object is created without any errors.

Test case ID	D2.2 Create a new Deal without authorization
Description	Check that it's not possible to create new Deal without being authorized.
Precondition	-
Test data	<pre> 1 { 2 "assignee__user": "1625047373081x778685405315052900", 3 "company_name_text": "Team1 tester",</pre>

```

4   "deal_value_estimation_number": 1,
5   "name_text": "Team1",
6   "order_number": 1,
7   "status_option_status": "In progress",
8   "visible_to_list_user": [
9     "1625047373081x778685405315052900"
10   ]
11 }

```

Test steps	1. Send POST {baseUrl}/obj/deal without being authorized.
Expected result	<ul style="list-style-type: none"> • Permission denied code: 401 Unauthorized. • Error code is displayed.

Test case ID	D2.3 Create a new Deal with empty body
Description	Check that it's not possible to create new Deal with empty body.
Precondition	-
Test data	<pre> 1 { 2 3 } </pre>
Test steps	1. Send POST {baseUrl}/obj/deal with empty body.
Expected result	<ul style="list-style-type: none"> • Database retrieval failure: 400 Bad Request. • Error code is displayed.

Test case ID	D2.4 Create a new Deal with empty assignee user
Description	Check that it's not possible to create a new Deal with an empty assignee user.
Precondition	-
Test data	<pre> 1 { 2 "assignee__user": "", 3 "company_name_text": "{\$randomCompanyName}", 4 "deal_value_estimation_number": 2, 5 "description_text": "{\$randomLoremSentence}", 6 "name_text": "Team1 test", 7 "order_number": 1, 8 "status_option_status": "In progress", 9 "visible_to_list_user": [10 "1625047373081x778685405315052900" 11] 12 } </pre>
Test steps	1. Send POST {baseUrl}/obj/deal with empty assignee user.
Expected result	<ul style="list-style-type: none"> • Database retrieval failure: 400 Bad Request. • Error code is displayed.

Test case ID	D2.5 Create a new Deal with assignee user in integer
--------------	---

Description	Check that it's not possible to create a new Deal with an assigned user in integer .
Precondition	-
Test data	<pre> 1 { 2 "assignee__user": 0, 3 "company_name_text": "{\$randomCompanyName}", 4 "deal_value_estimation_number": 2, 5 "description_text": "{\$randomLoremSentence}", 6 "name_text": "Team1 test", 7 "order_number": 1, 8 "status_option_status": "In progress", 9 "visible_to_list_user": [10 "1625047373081x778685405315052900" 11] 12 } </pre>
Test steps	1. Send POST {baseUrl}/obj/deal with assignee user in integer.
Expected result	<ul style="list-style-type: none"> • Database retrieval failure: 400 Bad Request. • Error code is displayed.

Test case ID	D2.6 Create a new Deal with empty company name
Description	Check that it's not possible to create a new Deal with an empty company name.
Precondition	-
Test data	<pre> 1 { 2 "assignee__user": "1625047373081x778685405315052900", 3 "company_name_text": "", 4 "deal_value_estimation_number": 2, 5 "description_text": "{\$randomLoremSentence}", 6 "name_text": "Team1 test", 7 "order_number": 1, 8 "status_option_status": "In progress", 9 "visible_to_list_user": [10 "1625047373081x778685405315052900" 11] 12 } </pre>
Test steps	1. Send POST {baseUrl}/obj/deal with an empty company name.
Expected result	<ul style="list-style-type: none"> • Database retrieval failure: 400 Bad Request. • Error code is displayed.

Test case ID	D2.7 Create a new Deal with company name in integer
Description	Check that it's not possible to create a new Deal with an company name in integer.
Precondition	-
Test data	<pre> 1 { 2 "assignee__user": "1625047373081x778685405315052900", 3 "company_name_text": 0, </pre>

	<pre> 4 "deal_value_estimation_number": 2, 5 "description_text": "{\$randomLoremSentence}", 6 "name_text": "Team1 test", 7 "order_number": 1, 8 "status_option_status": "In progress", 9 "visible_to_list_user": [10 "1625047373081x778685405315052900" 11] 12 }</pre>
--	--

Test steps	1. Send POST {baseUrl}/obj/deal with company namer in integer.
Expected result	<ul style="list-style-type: none"> • Database retrieval failure: 400 Bad Request. • Error code is displayed.

Test case ID	D2.8 Create a new Deal with empty deal value estimation number
Description	Check that it's not possible to create a new Deal with an empty estimation number
Precondition	-
Test data	<pre> 1 { 2 "assignee__user": "1625047373081x778685405315052900", 3 "company_name_text": "Team1 test", 4 "deal_value_estimation_number": , 5 "description_text": "{\$randomLoremSentence}", 6 "name_text": "Team1 test", 7 "order_number": 1, 8 "status_option_status": "In progress", 9 "visible_to_list_user": [10 "1625047373081x778685405315052900" 11] 12 }</pre>
Test steps	1. Send POST {baseUrl}/obj/deal with empty estimation number.
Expected result	<ul style="list-style-type: none"> • Database retrieval failure: 400 Bad Request. • Error code is displayed.

Test case ID	D2.9 Create a new Deal with Deal value estimation number in string
Description	Check that it's not possible to create a new Deal with Deal value estimation number in string.
Precondition	-
Test data	<pre> 1 { 2 "assignee__user": "1625047373081x778685405315052900", 3 "company_name_text": "Team1 test", 4 "deal_value_estimation_number": "test", 5 "description_text": "{\$randomLoremSentence}", 6 "name_text": "Team1 test", 7 "order_number": 1, 8 "status_option_status": "In progress", 9 "visible_to_list_user": [10 "1625047373081x778685405315052900" 11] 12 }</pre>

	<pre> 11] 12 }</pre>
Test steps	1. Send POST {baseUrl}/obj/deal with Deal value estimation number in string.
Expected result	<ul style="list-style-type: none"> • Database retrieval failure: 400 Bad Request. • Error code is displayed.

Test case ID	D2.10 Create a new Deal with empty name
Description	Check that it's not possible to create a new Deal with an empty name.
Precondition	-
Test data	<pre> 1 { 2 "assignee__user": "1625047373081x778685405315052900", 3 "company_name_text": "Team1 test", 4 "deal_value_estimation_number":1, 5 "description_text": "{\$randomLoremSentence}", 6 "name_text": "", 7 "order_number": 1, 8 "status_option_status": "In progress", 9 "visible_to_list_user": [10 "1625047373081x778685405315052900" 11] 12 }</pre>
Test steps	1. Send POST {baseUrl}/obj/deal with empty name.
Expected result	<ul style="list-style-type: none"> • Database retrieval failure: 400 Bad Request. • Error code is displayed.

Test case ID	D2.11 Create a new Deal with name in integer
Description	Check that it's not possible to create a new Deal with name in integer.
Precondition	-
Test data	<pre> 1 { 2 "assignee__user": "1625047373081x778685405315052900", 3 "company_name_text": "Team1 test", 4 "deal_value_estimation_number":1, 5 "description_text": "{\$randomLoremSentence}", 6 "name_text": 0, 7 "order_number": 1, 8 "status_option_status": "In progress", 9 "visible_to_list_user": [10 "1625047373081x778685405315052900" 11] 12 }</pre>
Test steps	1. Send POST {baseUrl}/obj/deal with name in integer.
Expected result	<ul style="list-style-type: none"> • Database retrieval failure: 400 Bad Request. • Error code is displayed.

Test case ID	D2.12 Create a new Deal with empty order number
Description	Check that it's not possible to create a new Deal with an empty order number.
Precondition	-
Test data	<pre> 1 { 2 "assignee__user": "1625047373081x778685405315052900", 3 "company_name_text": "Team1 test", 4 "deal_value_estimation_number":1, 5 "description_text": "{\$randomLoremSentence}", 6 "name_text": "Team1 test", 7 "order_number":, 8 "status_option_status": "In progress", 9 "visible_to_list_user": [10 "1625047373081x778685405315052900" 11] 12 }</pre>
Test steps	1. Send POST {baseUrl}/obj/deal with empty order number.
Expected result	<ul style="list-style-type: none"> Database retrieval failure: 400 Bad Request. Error code is displayed.

Test case ID	D2.13 Create a new Deal with order number in string
Description	Check that it's not possible to create a new Deal with order number in string.
Precondition	-
Test data	<pre> 1 { 2 "assignee__user": "1625047373081x778685405315052900", 3 "company_name_text": "Team1 test", 4 "deal_value_estimation_number":1, 5 "description_text": "{\$randomLoremSentence}", 6 "name_text": "Team1 test", 7 "order_number": "test", 8 "status_option_status": "In progress", 9 "visible_to_list_user": [10 "1625047373081x778685405315052900" 11] 12 }</pre>
Test steps	1. Send POST {baseUrl}/obj/deal with order number in string.
Expected result	<ul style="list-style-type: none"> Database retrieval failure: 400 Bad Request. Error code is displayed.

Test case ID	D2.14 Create a new Deal with empty status option
Description	Check that it's not possible to create a new Deal with an empty status option.
Precondition	-
Test data	<pre> 1 { 2 "assignee__user": "1625047373081x778685405315052900", 3 "company_name_text": "Team1 test",</pre>

	<pre> 4 "deal_value_estimation_number":1, 5 "description_text": "{\$randomLoremSentence}", 6 "name_text": "Team1 test", 7 "order_number":0, 8 "status_option_status": "", 9 "visible_to_list_user": [10 "1625047373081x778685405315052900" 11] 12 }</pre>
--	---

Test steps	1. Send POST {baseUrl}/obj/deal with empty status option.
Expected result	<ul style="list-style-type: none"> Database retrieval failure: 400 Bad Request. Error code is displayed.

Test case ID	D2.15 Create a new Deal with status option in integer
Description	Check that it's not possible to create a new Deal with status option in integer.
Precondition	-
Test data	<pre> 1 { 2 "assignee__user": "1625047373081x778685405315052900", 3 "company_name_text": "Team1 test", 4 "deal_value_estimation_number":1, 5 "description_text": "{\$randomLoremSentence}", 6 "name_text": "Team1 test", 7 "order_number":0, 8 "status_option_status": 1, 9 "visible_to_list_user": [10 "1625047373081x778685405315052900" 11] 12 }</pre>
Test steps	1. Send POST {baseUrl}/obj/deal with status option in integer.
Expected result	<ul style="list-style-type: none"> Database retrieval failure: 400 Bad Request. Error code is displayed.

Test case ID	D2.16 Create a new Deal with empty visible to list user
Description	Check that it's not possible to create a new Deal with an empty visible to list user
Precondition	-
Test data	<pre> 1 { 2 "assignee__user": "1625047373081x778685405315052900", 3 "company_name_text": "Team1 test", 4 "deal_value_estimation_number":1, 5 "description_text": "{\$randomLoremSentence}", 6 "name_text": "Team1 test", 7 "order_number":0, 8 "status_option_status": "In progress", 9 "visible_to_list_user": [10 "" 11] 12 }</pre>

	<pre> 11] 12 }</pre>
Test steps	1. Send POST {baseUrl}/obj/deal with empty visible to list user .
Expected result	<ul style="list-style-type: none"> • Database retrieval failure: 400 Bad Request. • Error code is displayed.

Test case ID	D2.17 Create a new Deal with visible list of user in integer
Description	Check that it's not possible to create a new Deal with visible list of user in integer.
Precondition	-
Test data	<pre> 1 { 2 "assignee__user": "1625047373081x778685405315052900", 3 "company_name_text": "Team1 test", 4 "deal_value_estimation_number":1, 5 "description_text": "{\$randomLoremSentence}", 6 "name_text": "Team1 test", 7 "order_number":0, 8 "status_option_status": "In progress", 9 "visible_to_list_user": [10 0 11] 12 }</pre>
Test steps	1. Send POST {baseUrl}/obj/deal with visible to list user in integer.
Expected result	<ul style="list-style-type: none"> • Database retrieval failure: 400 Bad Request. • Error code is displayed.

Modify deal data

Test case ID	D3.1 Modify the data of existing deal - all fields
Description	Verify that the system successfully allows the modification of an existing deal with valid data provided for all fields.
Precondition	An existing deal is available with data that needs to be updated
Test data	<pre> 1 { 2 "assignee__user": "1625047373081x778685405315052900", 3 "company_name_text": "Team1 Karin Patch1", 4 "deal_value_estimation_number": 1, 5 "description_text": "Test123", 6 "file_list_custom_data_file": [], 7 "funnel_custom_funnel1": "1697712429736x604475436022720900", 8 "lead_custom_lead": "1697609042937x450280473509041340", 9 "logo_image": "//s3.amazonaws.com/appforest_uf/f1625048400000 10 "name_text": "Team1 Karin Patch1", 11 "note_list_custom_note": [12 "1678718690483x188510294479732740" 13], 14 "order_number": 1, 15 "stage_custom_stage": "1625047818390x664413796313397600", 16 "status_option_status": "In progress",</pre>

```

17     "visible_to_list_user": [
18         "1625047373081x778685405315052900",
19         "1625047362532x398219546419355650"
20     ]
21 }

```

Test steps	1. Send PATCH {baseUrl}/obj/deal/:UniqueID
Expected result	<ul style="list-style-type: none"> Successful code: 204 Success. Deal data updated.

Test case ID	D3.2 Modify the data of existing deal - mandatory fields
Description	Verify that the system successfully allows the modification of an existing deal by updating only the mandatory fields.
Precondition	An existing deal is available with data that needs to be updated
Test data	<pre> 1 { 2 "assignee__user": "1625047373081x778685405315052900", 3 "company_name_text": "Team1 Karin Patch2", 4 "deal_value_estimation_number": 1, 5 "name_text": "Team1 Karin Patch2", 6 "order_number": 1, 7 "status_option_status": "In progress", 8 "visible_to_list_user": [9 "1625047373081x778685405315052900", 10 "1625047362532x398219546419355650" 11] 12 } </pre>
Test steps	1. Send PATCH {baseUrl}/obj/deal/:UniqueID modifying only mandatory fields.
Expected result	<ul style="list-style-type: none"> Successful code: 204 Success. Deal data updated.

Test case ID	D3.3 Modify the data of existing deal - no authorization
Description	Verify that modifying an existing deal without proper authorization is not allowed.
Precondition	An existing deal is available with data that needs to be updated
Test data	<pre> 1 { 2 "assignee__user": "1625047373081x778685405315052900", 3 "company_name_text": "Team1 Karin Patch2", 4 "deal_value_estimation_number": 1, 5 "name_text": "Team1 Karin Patch2", 6 "order_number": 1, 7 "status_option_status": "In progress", 8 "visible_to_list_user": [9 "1625047373081x778685405315052900", 10 "1625047362532x398219546419355650" 11] 12 } </pre>
Test steps	1. Change authorization type to No Auth

	2. Send PATCH {baseUrl}/obj/deal/:UniqueID
Expected result	<ul style="list-style-type: none"> Error code: 401 Unauthorized Object data is not replaced

Test case ID	D3.4 Modify existing deal - empty body
Description	Verify that modifying an existing deal with an empty request body is not possible
Precondition	An existing deal is available with data that needs to be updated
Test data	<pre> 1 { 2 3 }</pre>
Test steps	1. Send PATCH {baseUrl}/obj/deal/:UniqueID with empty body
Expected result	<ul style="list-style-type: none"> Error code and message is shown Object data is not replaced

Test case ID	D3.5 Modify not existing deal with empty Id (proper error handling is expected)
Description	Verify it's not possible to modify deal with empty ID.
Precondition	There are no deals with empty IDs in the system
Test data	<pre> 1 { 2 "name_text": "Team1 Karin Patch2" 3 }</pre>
Test steps	1. Send PATCH {baseUrl}/obj/deal/:UniqueID with empty ID value
Expected result	<ul style="list-style-type: none"> Error code: 404 Not found Error message is displayed: <pre>"message": "Missing object of type deal: object with id :UniqueID does not exist"</pre>

Test case ID	D3.6 Modify not existing deal with random Id (proper error handling is expected)
Description	Verify it's not possible to modify a non-existing deal with a random ID.
Precondition	There are no deals with the randomly generated ID in the system.
Test data	<pre> 1 { 2 "name_text": "Team1 Karin Patch2" 3 }</pre>
Test steps	1. Send PATCH {baseUrl}/obj/deal/:UniqueID with random ID, for example: 123456

Expected result	<ul style="list-style-type: none"> Error code: 404 Not found Error message is displayed: <pre>"message": "Missing object of type deal: object with id 123456 does not exist"</pre>
-----------------	--

Test case ID	D3.7 Modify the data of existing deal - assignee_user
Description	Verify that it is possible to modify "assignee_user" field
Precondition	An existing deal is available with data that needs to be updated
Test data	<pre>1 { 2 "assignee_user": "1625047362532x398219546419355650" 3 }</pre>
Test steps	1. Send PATCH {baseUrl}/obj/deal/:UniqueID with updated assignee_user value
Expected result	<ul style="list-style-type: none"> Successful code: 204 Success. Deal data updated.

Test case ID	D3.8 Modify the data of existing deal - assignee_user integer
Description	Verify that it is not possible to modify an existing deal while leaving the "assigned user" field with an integer value.
Precondition	An existing deal is available with data that needs to be updated
Test data	<pre>1 { 2 "assignee_user": 111 3 }</pre>
Test steps	1. Send PATCH {baseUrl}/obj/deal/:UniqueID with assignee_user value as integer
Expected result	<ul style="list-style-type: none"> Error code: 400 Database retrieval failure. Error message is displayed: <pre>"message": "Invalid data for field assignee_user: object with this id does not exist: 111"</pre>

Test case ID	D3.9 Modify the data of existing deal - assignee_user empty (mandatory field)
Description	Verify that it is not possible to modify an existing deal while leaving the mandatory field "assigned user" with an empty value.
Precondition	An existing deal is available with data that needs to be updated
Test data	<pre>1 { 2 "assignee_user": 3 }</pre>

Test steps	1. Send PATCH {baseUrl}/obj/deal/:UniqueID with field assignee__user value empty
Expected result	<ul style="list-style-type: none"> • Error code: 400 Database retrieval failure. • Error message is displayed: "Error parsing request body: Unexpected token } in JSON at position 22"

Test case ID	D3.10 Modify the data of existing deal - company_name_text
Description	Verify that it is possible to modify "company_name_text" field
Precondition	An existing deal is available with data that needs to be updated
Test data	<pre> 1 { 2 "company_name_text": "CHANGED Team 1 Karin Patch" 3 }</pre>
Test steps	1. Send PATCH {baseUrl}/obj/deal/:UniqueID with updated company_name_text
Expected result	<ul style="list-style-type: none"> • Successful code: 204 Success. Deal data updated.

Test case ID	D3.11 Modify the data of existing deal - company_name_text empty (mandatory field)
Description	Verify that it is not possible to modify an existing deal while leaving the mandatory field "company_name_text" with empty value.
Precondition	An existing deal is available with data that needs to be updated
Test data	<pre> 1 { 2 "company_name_text": "CHANGED Team 1 Karin Patch" 3 }</pre>
Test steps	1. Send PATCH {baseUrl}/obj/deal/:UniqueID with field "company_name_text" with empty value.
Expected result	<ul style="list-style-type: none"> • Successful code: 204 Success. Deal data updated.

Test case ID	D3.12 Modify the data of existing deal - company_name_text integer
Description	Verify that it is not possible to modify an existing deal while leaving the "company_name_text" field with an integer value.
Precondition	An existing deal is available with data that needs to be updated
Test data	<pre> 1 { 2 "company_name_text": 111 3 }</pre>

Test steps	<p>1. Send PATCH <code>{baseUrl}/obj/deal/:UniqueID</code> while leaving the "company_name_text" field with an integer value.</p>
Expected result	<ul style="list-style-type: none"> Error code: 400 Database retrieval failure Error message is displayed: <pre>"message": "Invalid data for field company_name_text: Expected a string, but got a number (original data: 111)"</pre>

Test case ID	D3.13 Modify the data of existing deal - deal_value_estimation_number
Description	Verify that it is possible to modify "deal_value_estimation_number" field
Precondition	An existing deal is available with data that needs to be updated
Test data	<pre>1 { 2 "deal_value_estimation_number": 1000 3 }</pre>
Test steps	<p>1. Send PATCH <code>{baseUrl}/obj/deal/:UniqueID</code> with updated "deal_value_estimation_number" value</p>
Expected result	<ul style="list-style-type: none"> Successful code: 204 Success. Deal data updated.

Test case ID	D3.14 Modify the data of existing deal - deal_value_estimation_number string
Description	Verify that it is not possible to modify an existing deal while leaving the "deal_value_estimation_number" field with an string value.
Precondition	An existing deal is available with data that needs to be updated
Test data	<pre>1 { 2 "deal_value_estimation_number": "Test1" 3 }</pre>
Test steps	<p>1. Send PATCH <code>{baseUrl}/obj/deal/:UniqueID</code> while leaving the "deal_value_estimation_number" field with an string value.</p>
Expected result	<ul style="list-style-type: none"> Error code: 400 Database retrieval failure Error message is displayed: <pre>"message": "Invalid data for field deal_value_estimation_number : Expected a number, but got a string (original data: \"Test1\")"</pre>

Test case ID	D3.15 Modify the data of existing deal - deal_value_estimation_number negative integer
Description	Verify that it is not possible to modify an existing deal while leaving the "deal_value_estimation_number" field with a negative integer value.

Precondition	An existing deal is available with data that needs to be updated
Test data	<pre> 1 { 2 "deal_value_estimation_number": -1, 3 }</pre>
Test steps	1. Send PATCH {baseUrl}/obj/deal/:UniqueID while leaving the "deal_value_estimation_number" field with a negative integer value.
Expected result	<ul style="list-style-type: none"> Error code: 400 Database retrieval failure Error message is shown, deal is not modified.

Test case ID	D3.16 Modify the data of existing deal - description_text
Description	Verify that it is possible to modify "description_text" field
Precondition	An existing deal is available with data that needs to be updated
Test data	<pre> 1 { 2 "description_text": "Muudetud description" 3 }</pre>
Test steps	1. Send PATCH {baseUrl}/obj/deal/:UniqueID with updated "description_text" value
Expected result	<ul style="list-style-type: none"> Successful code: 204 Success. Deal data updated.

Test case ID	D3.17 Modify the data of existing deal - description_text integer
Description	Verify that it is not possible to modify "description_text" field with integer
Precondition	An existing deal is available with data that needs to be updated
Test data	<pre> 1 { 2 "description_text": 111 3 }</pre>
Test steps	1. Send PATCH {baseUrl}/obj/deal/:UniqueID with "description_text" field as integer
Expected result	<ul style="list-style-type: none"> Error code: 400 Database retrieval failure Error message is displayed: <pre>"message": "Invalid data for field description_text: Expected a string, but got a number (original data: 111)"</pre>

Test case ID	D3.18 Modify the data of existing deal - file_list_custom_data_file integer
Description	Verify that it is not possible to modify an existing deal while leaving the "file_list_custom_data_file" field with an integer value.

Precondition	An existing deal is available with data that needs to be updated
Test data	<pre> 1 { 2 "file_list_custom_data_file": 111 3 }</pre>
Test steps	1. Send PATCH {baseUrl}/obj/deal/:UniqueID while leaving the "file_list_custom_data_file" field with an integer value.
Expected result	<ul style="list-style-type: none"> Error code: 400 Database retrieval failure Error message is displayed: <pre>"message": "Invalid data for field file_list_custom_data_file: expecting an array, but this is a number"</pre>

Test case ID	D3.19 Modify the data of existing deal - funnel_custom_funnel1 integer
Description	Verify that it is not possible to modify an existing deal while leaving the "funnel_custom_funnel1 integer" field with an integer value.
Precondition	An existing deal is available with data that needs to be updated
Test data	<pre> 1 { 2 "funnel_custom_funnel1": 111 3 }</pre>
Test steps	1. Send PATCH {baseUrl}/obj/deal/:UniqueID while leaving the "funnel_custom_funnel1 integer" field with an integer value.
Expected result	<ul style="list-style-type: none"> Error code: 400 Database retrieval failure Error message is displayed: <pre>"message": "Invalid data for field funnel_custom_funnel1: object with this id does not exist: 111"</pre>

Test case ID	D3.20 Modify the data of existing deal - lead_custom_lead integer
Description	Verify that it is not possible to modify an existing deal while leaving the "lead_custom_lead" field with an integer value.
Precondition	An existing deal is available with data that needs to be updated
Test data	<pre> 1 { 2 "lead_custom_lead": 111 3 }</pre>
Test steps	1. Send PATCH {baseUrl}/obj/deal/:UniqueID while leaving the "lead_custom_lead" field with an integer value.
Expected result	<ul style="list-style-type: none"> Error code: 400 Database retrieval failure Error message is displayed: <pre>"message": "Invalid data for field lead_custom_lead: object with</pre>

	this id does not exist: 111"
--	------------------------------

Test case ID	D3.21 Modify the data of existing deal - name_text
Description	Verify that it is possible to modify "name_text" field
Precondition	An existing deal is available with data that needs to be updated
Test data	<pre> 1 { 2 "name_text": "Muudetud TEAM1 PATCH Karin" 3 }</pre>
Test steps	1. Send PATCH {baseUrl}/obj/deal/:UniqueID with updated "name_text" value
Expected result	<ul style="list-style-type: none"> Successful code: 204 Success. Deal data updated.

Test case ID	D3.22 Modify the data of existing deal - name_text integer
Description	Verify that it is not possible to modify an existing deal while leaving the "name_text" field with an integer value.
Precondition	An existing deal is available with data that needs to be updated
Test data	<pre> 1 { 2 "name_text": 111 3 }</pre>
Test steps	1. Send PATCH {baseUrl}/obj/deal/:UniqueID while leaving the "name_text" field with an integer value.
Expected result	<ul style="list-style-type: none"> Error code: 400 Database retrieval failure Error message is displayed: <code>"message": "Invalid data for field name_text: Expected a string, but got a number (original data: 111)"</code>

Test case ID	D3.23 Modify the data of existing deal - note_list_custom_note integer
Description	Verify that it is not possible to modify an existing deal while leaving the "note_list_custom_note" field with an integer value.
Precondition	An existing deal is available with data that needs to be updated
Test data	<pre> 1 { 2 "note_list_custom_note": [3 111 4] 5 }</pre>

Test steps	1. Send PATCH {baseUrl}/obj/deal/:UniqueID while leaving the "note_list_custom_note" field with an integer value.
Expected result	<ul style="list-style-type: none"> Error code: 400 Database retrieval failure Error message is displayed: <pre>"message": "Invalid data for field note_list_custom_note: object with this id does not exist: 111"</pre>

Test case ID	D3.24 Modify the data of existing deal - order_number
Description	Verify that it is possible to modify "order_number" field
Precondition	An existing deal is available with data that needs to be updated
Test data	<pre>1 { 2 "order_number": 27 3 }</pre>
Test steps	1. Send PATCH {baseUrl}/obj/deal/:UniqueID with updated "order_number" value
Expected result	<ul style="list-style-type: none"> Successful code: 204 Success. Deal data updated.

Test case ID	D3.25 Modify the data of existing deal - order_number string
Description	Verify that it is not possible to modify an existing deal while leaving the "order_number" field with a string value.
Precondition	An existing deal is available with data that needs to be updated
Test data	<pre>1 { 2 "order_number": "team1" 3 }</pre>
Test steps	1. Send PATCH {baseUrl}/obj/deal/:UniqueID while leaving the "order_number" field with a string value.
Expected result	<ul style="list-style-type: none"> Error code: 400 Database retrieval failure Error message is displayed: <pre>"message": "Invalid data for field order_number: Expected a number, but got a string (original data: \"team1\")"</pre>

Test case ID	D3.26 Modify the data of existing deal - stage_custom_stage
Description	Verify that it is possible to modify "stage_custom_stage" field
Precondition	An existing deal is available with data that needs to be updated
Test data	<pre>1 { 2 "stage_custom_stage": "1625047833003x411752455944692100" 3 }</pre>

Test steps	1. Send PATCH {baseUrl}/obj/deal/:UniqueID with updated "stage_custom_stage" value
Expected result	<ul style="list-style-type: none"> Successful code: 204 Success. Deal data updated.

Test case ID	D3.27 Modify the data of existing deal - status_option_status
Description	Verify that it is possible to modify "status_option_status" field
Precondition	An existing deal is available with data that needs to be updated
Test data	<pre> 1 { 2 "status_option_status": "Lost" 3 }</pre>
Test steps	1. Send PATCH {baseUrl}/obj/deal/:UniqueID with updated "status_option_status" field
Expected result	<ul style="list-style-type: none"> Successful code: 204 Success. Deal data updated.

Test case ID	D3.28 Modify the data of existing deal - status_option_status invalid data
Description	Verify that it is not possible to modify an existing deal while leaving the "status_option_status" field with invalid data.
Precondition	An existing deal is available with data that needs to be updated
Test data	<pre> 1 { 2 "status_option_status": "Invalid data" 3 }</pre>
Test steps	1. Send PATCH {baseUrl}/obj/deal/:UniqueID while leaving the "status_option_status" field with invalid data.
Expected result	<ul style="list-style-type: none"> Error code: 400 Database retrieval failure Error message is displayed: <pre>"message": "Invalid data for field status_option_status: could not parse this as a Status"</pre>

Test case ID	D3.29 Modify the data of existing deal - visible_to_list_user
Description	Verify that it is possible to modify "visible_to_list_user" field
Precondition	An existing deal is available with data that needs to be updated
Test data	<pre> 1 { 2 "visible_to_list_user": [3 "1625047362532x398219546419355650" 4] 5 }</pre>

Test steps	1. Send PATCH {{baseUrl}}/obj/deal/:UniqueID with updated "visible_to_list_user" value
Expected result	<ul style="list-style-type: none"> Successful code: 204 Success. Deal data updated.

Test case ID	D3.30 Modify the data of existing deal - visible_to_list_user integer
Description	Verify that it is not possible to modify existing deal, while leaving while leaving visible_to_list_user with integer value
Precondition	An existing deal is available with data that needs to be updated
Test data	<pre> 1 { 2 "visible_to_list_user": [3 111 4] 5 }</pre>
Test steps	1. Send PATCH {{baseUrl}}/obj/deal/:UniqueID while leaving while leaving visible_to_list_user with integer value
Expected result	<ul style="list-style-type: none"> Error code: 400 Database retrieval failure Error message is displayed: <pre>"message": "Invalid data for field visible_to_list_user: object with this id does not exist: 111"</pre>

Replace deal data

Test case ID	D4.1 Replace deal data - minimum set of fields
Description	Check that it's possible to replace only mandatory fields on a deal.
Precondition	A deal exists in the database
Test data	<pre> 1 { 2 "assignee__user": "1625047373081x778685405315052900", 3 "company_name_text": "Team1 test A-M PUT", 4 "deal_value_estimation_number": 2, 5 "name_text": "Team1_A-M PUT", 6 "order_number": 2, 7 "status_option_status": "In progress", 8 "visible_to_list_user": [9 "1625047373081x778685405315052900", 10 "1625047362532x398219546419355650" 11] 12 }</pre>
Test steps	1. Send PUT request {{baseUrl}}/obj/deal/:UniqueID
Expected result	<ul style="list-style-type: none"> Successful code: 204 No Content Object data is replaced

Test case ID	D4.2 Replace deal data - all fields valid
Description	Check that it's possible to replace all fields with valid data on a deal.
Precondition	A deal exists in the database
Test data	<pre> 1 { 2 "assignee__user": "1625047373081x778685405315052900", 3 "company_name_text": "Team1 test A-M PUT2", 4 "deal_value_estimation_number": 1, 5 "description_text": "This is a test", 6 "file_list_custom_data_file": [], 7 "funnel_custom_funnel1": "1697712429736x604475436022720900", 8 "lead_custom_lead": "1697609042937x450280473509041340", 9 "logo_image": "//s3.amazonaws.com/appforest_uf/f1625048400000", 10 "name_text": "Team1_A-M PUT2", 11 "note_list_custom_note": [12 "1678718690483x188510294479732740" 13], 14 "order_number": 1, 15 "stage_custom_stage": "1625047818390x664413796313397600", 16 "status_option_status": "Won", 17 "visible_to_list_user": [18 "1625047373081x778685405315052900", 19 "1625047362532x398219546419355650" 20] 21 }</pre>
Test steps	1. Send PUT request {{baseUrl}}/obj/deal/:UniqueID
Expected result	<ul style="list-style-type: none"> Successful code: 204 No Content Object data is replaced

Test case ID	D4.3 Replace deal data - no authorization
Description	Verify that it is not possible to replace data on a deal unauthorized
Precondition	A deal exists in the database
Test data	<pre> 1 { 2 "assignee__user": "1625047373081x778685405315052900", 3 "company_name_text": "Team1 test A-M PUT", 4 "deal_value_estimation_number": 2, 5 "name_text": "Team1_A-M PUT", 6 "order_number": 2, 7 "status_option_status": "In progress", 8 "visible_to_list_user": [9 "1625047373081x778685405315052900", 10 "1625047362532x398219546419355650" 11] 12 }</pre>
Test steps	1. Send PUT request {{baseUrl}}/obj/deal/:UniqueID 2. Change authorization type to No Auth
Expected result	<ul style="list-style-type: none"> Error code: 401 Unauthorized Object data is not replaced

Test case ID	D4.4 Replace deal data - empty body
Description	Verify that it is not possible to replace deal data with empty body
Precondition	A deal exists in the database
Test data	<pre> 1 { 2 3 }</pre>
Test steps	1. Send PUT request <code> {{baseUrl}}/obj/deal/:UniqueId</code>
Expected result	<ul style="list-style-type: none"> Error code and message is shown Object data is not replaced

Test case ID	D4.5 Replace deal data - empty Id
Description	Check that error message is shown when UniqueId field is left empty
Precondition	-
Test data	<pre> 1 { 2 "assignee__user": "1625047373081x778685405315052900", 3 "company_name_text": "Team1 test A-M PUT", 4 "deal_value_estimation_number": 2, 5 "name_text": "Team1_A-M PUT", 6 "order_number": 2, 7 "status_option_status": "In progress", 8 "visible_to_list_user": [9 "1625047373081x778685405315052900", 10 "1625047362532x398219546419355650" 11] 12 }</pre>
Test steps	1. Send PUT request <code> {{baseUrl}}/obj/deal/:UniqueId</code> with empty UniqueId
Expected result	<ul style="list-style-type: none"> Error code: 404 Not Found Error message is displayed <code>"message": "Missing object of type deal: object with id :UniqueId does not exist"</code>

Test case ID	D4.6 Replace deal data: assignee__user - empty
Description	Check that it's not possible to leave assignee__user empty in PUT request.
Precondition	A deal exists in the database
Test data	<pre> 1 { 2 "assignee__user": "", 3 "company_name_text": "Team1 test A-M PUT2", 4 "deal_value_estimation_number": 1, 5 "description_text": "This is a test", 6 "file_list_custom_data_file": [], 7 "funnel_custom_funnel1": "1697712429736x604475436022720900", 8 "lead_custom_lead": "1697609042937x450280473509041340", 9 "logo_image": "//s3.amazonaws.com/appforest_uf/f1625048400000 10 "name_text": "Team1_A-M PUT2",</pre>

	<pre> 11 "note_list_custom_note": [12 "1678718690483x188510294479732740" 13], 14 "order_number": 1, 15 "stage_custom_stage": "1625047818390x664413796313397600", 16 "status_option_status": "Won", 17 "visible_to_list_user": [18 "1625047373081x778685405315052900", 19 "1625047362532x398219546419355650" 20] 21 }</pre>
Test steps	1. Send PUT request <code> {{baseUrl}}/obj/deal/:UniqueID</code> with empty assignee__user
Expected result	<ul style="list-style-type: none"> Error code: 400 Bad Request Proper error message is displayed

Test case ID	D4.7 Replace deal data: assignee__user - integer
Description	Check that assignee__user data cannot be replaced with integer.
Precondition	A deal exists in the database
Test data	<pre> 1 { 2 "assignee__user": 123, 3 "company_name_text": "Team1 test A-M PUT2", 4 "deal_value_estimation_number": 1, 5 "description_text": "This is a test", 6 "file_list_custom_data_file": [], 7 "funnel_custom_funnel1": "1697712429736x604475436022720900", 8 "lead_custom_lead": "1697609042937x450280473509041340", 9 "logo_image": "//s3.amazonaws.com/appforest_uf/f1625048400000 10 "name_text": "Team1_A-M PUT2", 11 "note_list_custom_note": [12 "1678718690483x188510294479732740" 13], 14 "order_number": 1, 15 "stage_custom_stage": "1625047818390x664413796313397600", 16 "status_option_status": "Won", 17 "visible_to_list_user": [18 "1625047373081x778685405315052900", 19 "1625047362532x398219546419355650" 20] 21 }</pre>
Test steps	1. Send PUT request <code> {{baseUrl}}/obj/deal/:UniqueID</code> with integer value in assignee__user
Expected result	<ul style="list-style-type: none"> Error code: 400 Bad Request Object data is not replaced. Error message is displayed "message": <code>"Invalid data for field assignee__user: object with this id does not exist: 123"</code>

Test case ID	D4.8 Replace deal data: company_name_text - empty
--------------	---

Description	Check that it's not possible to leave company_name_text empty in PUT request.
Precondition	A deal exists in the database
Test data	<pre> 1 { 2 "assignee__user": "1625047373081x778685405315052900", 3 "company_name_text": "", 4 "deal_value_estimation_number": 1, 5 "description_text": "This is a test", 6 "file_list_custom_data_file": [], 7 "funnel_custom_funnel1": "1697712429736x604475436022720900", 8 "lead_custom_lead": "1697609042937x450280473509041340", 9 "logo_image": "//s3.amazonaws.com/appforest_uf/f1625048400000 10 "name_text": "Team1_A-M PUT2", 11 "note_list_custom_note": [12 "1678718690483x188510294479732740" 13], 14 "order_number": 1, 15 "stage_custom_stage": "1625047818390x664413796313397600", 16 "status_option_status": "Won", 17 "visible_to_list_user": [18 "1625047373081x778685405315052900", 19 "1625047362532x398219546419355650" 20] 21 }</pre>
Test steps	<ol style="list-style-type: none"> Send PUT request {{baseUrl}}/obj/deal/:UniqueID with empty company_name_text
Expected result	<ul style="list-style-type: none"> Error code: 400 Bad Request Proper error message is displayed

Test case ID	D4.9 Replace deal data: company_name_text - integer
Description	Check that company_name_text data cannot be replaced with integer.
Precondition	A deal exists in the database
Test data	<pre> 1 { 2 "assignee__user": "1625047373081x778685405315052900", 3 "company_name_text": 123, 4 "deal_value_estimation_number": 1, 5 "description_text": "This is a test", 6 "file_list_custom_data_file": [], 7 "funnel_custom_funnel1": "1697712429736x604475436022720900", 8 "lead_custom_lead": "1697609042937x450280473509041340", 9 "logo_image": "//s3.amazonaws.com/appforest_uf/f1625048400000 10 "name_text": "Team1_A-M PUT2", 11 "note_list_custom_note": [12 "1678718690483x188510294479732740" 13], 14 "order_number": 1, 15 "stage_custom_stage": "1625047818390x664413796313397600", 16 "status_option_status": "Won", 17 "visible_to_list_user": [18 "1625047373081x778685405315052900", 19 "1625047362532x398219546419355650" 20] 21 }</pre>

	<pre> 20] 21 }</pre>
Test steps	<p>1. Send PUT request {{baseUrl}}/obj/deal/:UniqueID with integer value in company_name_text</p>
Expected result	<ul style="list-style-type: none"> Error code: 400 Bad Request Object data is not replaced. Error message is displayed "message": <code>"Invalid data for field company_name_text: Expected a string, but got a number (original data: 123)"</code>

Test case ID	D4.10 Replace deal data: deal_value_estimation_number - empty
Description	Check that it's not possible to leave deal_value_estimation_number empty in PUT request.
Precondition	A deal exists in the database
Test data	<pre> 1 { 2 "assignee__user": "1625047373081x778685405315052900", 3 "company_name_text": "Team1 test A-M PUT2", 4 "deal_value_estimation_number": , 5 "description_text": "This is a test", 6 "file_list_custom_data_file": [], 7 "funnel_custom_funnel1": "1697712429736x604475436022720900", 8 "lead_custom_lead": "1697609042937x450280473509041340", 9 "logo_image": "//s3.amazonaws.com/appforest_uf/f1625048400000", 10 "name_text": "Team1_A-M PUT2", 11 "note_list_custom_note": [12 "1678718690483x188510294479732740" 13], 14 "order_number": 1, 15 "stage_custom_stage": "1625047818390x664413796313397600", 16 "status_option_status": "Won", 17 "visible_to_list_user": [18 "1625047373081x778685405315052900", 19 "1625047362532x398219546419355650" 20] 21 }</pre>
Test steps	<p>1. Send PUT request {{baseUrl}}/obj/deal/:UniqueID with empty deal_value_estimation_number</p>
Expected result	<ul style="list-style-type: none"> Error code: 400 Bad Request Error message is displayed <code>Error: Error parsing request body: Unexpected token</code>

Test case ID	D4.11 Replace deal data: deal_value_estimation_number - negative
Description	Check that it's possible to have negative value in deal_value_estimation_number.
Precondition	A deal exists in the database
Test data	<pre> 1 { 2 "assignee__user": "1625047373081x778685405315052900",</pre>

```

3   "company_name_text": "Team1 test A-M PUT2",
4   "deal_value_estimation_number": -10000,
5   "description_text": "This is a test",
6   "file_list_custom_data_file": [],
7   "funnel_custom_funnel1": "1697712429736x604475436022720900",
8   "lead_custom_lead": "1697609042937x450280473509041340",
9   "logo_image": "//s3.amazonaws.com/appforest_uf/f1625048400000
10  "name_text": "Team1_A-M PUT2",
11  "note_list_custom_note": [
12      "1678718690483x188510294479732740"
13 ],
14  "order_number": 1,
15  "stage_custom_stage": "1625047818390x664413796313397600",
16  "status_option_status": "Won",
17  "visible_to_list_user": [
18      "1625047373081x778685405315052900",
19      "1625047362532x398219546419355650"
20 ]
21 }

```

Test steps	1. Send PUT request {{baseUrl}}/obj/deal/:UniqueID with negative deal_value_estimation_number
Expected result	<ul style="list-style-type: none"> Successful code: 204 No Content Object data is replaced

Test case ID	D4.12 Replace deal data: deal_value_estimation_number - string
Description	Check that deal_value_estimation_number data cannot be replaced with string.
Precondition	A deal exists in the database
Test data	<pre> 1 { 2 "assignee__user": "1625047373081x778685405315052900", 3 "company_name_text": "Team1 test A-M PUT2", 4 "deal_value_estimation_number": "1", 5 "description_text": "This is a test", 6 "file_list_custom_data_file": [], 7 "funnel_custom_funnel1": "1697712429736x604475436022720900", 8 "lead_custom_lead": "1697609042937x450280473509041340", 9 "logo_image": "//s3.amazonaws.com/appforest_uf/f1625048400000 10 "name_text": "Team1_A-M PUT2", 11 "note_list_custom_note": [12 "1678718690483x188510294479732740" 13], 14 "order_number": 1, 15 "stage_custom_stage": "1625047818390x664413796313397600", 16 "status_option_status": "Won", 17 "visible_to_list_user": [18 "1625047373081x778685405315052900", 19 "1625047362532x398219546419355650" 20] 21 } </pre>
Test steps	1. Send PUT request {{baseUrl}}/obj/deal/:UniqueID with string value in deal_value_estimation_number
Expected result	<ul style="list-style-type: none"> Error code: 400 Bad Request

- Object data is not replaced. Error message is displayed "message":
 "Invalid data for field company_name_text: Expected an integer,
 but got a string (original data: "1")"

Test case ID	D4.13 Replace deal data: description_text - integer
Description	Check that description_text data cannot be replaced with integer.
Precondition	A deal exists in the database
Test data	<pre> 1 { 2 "assignee__user": "1625047373081x778685405315052900", 3 "company_name_text": "Team1 test A-M PUT2", 4 "deal_value_estimation_number": 1, 5 "description_text": 123, 6 "file_list_custom_data_file": [], 7 "funnel_custom_funnel1": "1697712429736x604475436022720900", 8 "lead_custom_lead": "1697609042937x450280473509041340", 9 "logo_image": "//s3.amazonaws.com/appforest_uf/f1625048400000 10 "name_text": "Team1_A-M PUT2", 11 "note_list_custom_note": [12 "1678718690483x188510294479732740" 13], 14 "order_number": 1, 15 "stage_custom_stage": "1625047818390x664413796313397600", 16 "status_option_status": "Won", 17 "visible_to_list_user": [18 "1625047373081x778685405315052900", 19 "1625047362532x398219546419355650" 20] 21 }</pre>
Test steps	<ol style="list-style-type: none"> Send PUT request {{baseUrl}}/obj/deal/:UniqueID with integer value in description_text
Expected result	<ul style="list-style-type: none"> Error code: 400 Bad Request Object data is not replaced. Error message is displayed "message": "Invalid data for field company_name_text: Expected a string, but got a number (original data: 123)"

Test case ID	D4.14 Replace deal data: file_list_custom_data_file - integer
Description	Check that file_list_custom_data_file data cannot be replaced with integer.
Precondition	A deal exists in the database
Test data	<pre> 1 { 2 "assignee__user": "1625047373081x778685405315052900", 3 "company_name_text": "Team1 test A-M PUT2", 4 "deal_value_estimation_number": 1, 5 "description_text": "This is a test", 6 "file_list_custom_data_file": 123, 7 "funnel_custom_funnel1": "1697712429736x604475436022720900", 8 "lead_custom_lead": "1697609042937x450280473509041340", 9 "logo_image": "//s3.amazonaws.com/appforest_uf/f1625048400000 10 "name_text": "Team1_A-M PUT2", 11 "note_list_custom_note": [</pre>

	<pre> 12 "1678718690483x188510294479732740" 13], 14 "order_number": 1, 15 "stage_custom_stage": "1625047818390x664413796313397600", 16 "status_option_status": "Won", 17 "visible_to_list_user": [18 "1625047373081x778685405315052900", 19 "1625047362532x398219546419355650" 20] 21 } </pre>
Test steps	<p>1. Send PUT request {{baseUrl}}/obj/deal/:UniqueID with integer value in file_list_custom_data_file</p>
Expected result	<ul style="list-style-type: none"> Error code: 400 Bad Request Object data is not replaced. Error message is displayed "message": <code>"Invalid data for field file_list_custom_data_file: expecting an array, but this is a number"</code>

Test case ID	D4.15 Replace deal data: funnel_custom_funnel1 - integer
Description	Check that funnel_custom_funnel1 data cannot be replaced with integer.
Precondition	A deal exists in the database
Test data	<pre> 1 { 2 "assignee__user": "1625047373081x778685405315052900", 3 "company_name_text": "Team1 test A-M PUT2", 4 "deal_value_estimation_number": 1, 5 "description_text": "This is a test", 6 "file_list_custom_data_file": [], 7 "funnel_custom_funnel1": 123, 8 "lead_custom_lead": "1697609042937x450280473509041340", 9 "logo_image": "//s3.amazonaws.com/appforest_uf/f1625048400000 10 "name_text": "Team1_A-M PUT2", 11 "note_list_custom_note": [12 "1678718690483x188510294479732740" 13], 14 "order_number": 1, 15 "stage_custom_stage": "1625047818390x664413796313397600", 16 "status_option_status": "Won", 17 "visible_to_list_user": [18 "1625047373081x778685405315052900", 19 "1625047362532x398219546419355650" 20] 21 } </pre>
Test steps	<p>1. Send PUT request {{baseUrl}}/obj/deal/:UniqueID with integer value in funnel_custom_funnel1</p>
Expected result	<ul style="list-style-type: none"> Error code: 400 Bad Request Object data is not replaced. Error message is displayed "message": <code>"Invalid data for field funnel_custom_funnel1: object with this id does not exist: 123"</code>

Test case ID	D4.16 Replace deal data: funnel_custom_funnel1 - partial funnel Id
--------------	--

Description	Check that inserting partial funnel Id does not link a funnel to the deal.
Precondition	A deal exists in the database
Test data	<pre> 1 { 2 "assignee__user": "1625047373081x778685405315052900", 3 "company_name_text": "Team1 test A-M PUT2", 4 "deal_value_estimation_number": 1, 5 "description_text": "This is a test", 6 "file_list_custom_data_file": [], 7 "funnel_custom_funnel1": "1697712429736x604475", 8 "lead_custom_lead": "1697609042937x450280473509041340", 9 "logo_image": "//s3.amazonaws.com/appforest_uf/f1625048400000 10 "name_text": "Team1_A-M PUT2", 11 "note_list_custom_note": [12 "1678718690483x188510294479732740" 13], 14 "order_number": 1, 15 "stage_custom_stage": "1625047818390x664413796313397600", 16 "status_option_status": "Won", 17 "visible_to_list_user": [18 "1625047373081x778685405315052900", 19 "1625047362532x398219546419355650" 20] 21 }</pre>
Test steps	1. Send PUT request {{baseUrl}}/obj/deal/:UniqueID with partial funnel Id in funnel_custom_funnel1
Expected result	<ul style="list-style-type: none"> Error code: 400 Bad Request Object data is not replaced. Error message is displayed "message": <pre>"Invalid data for field funnel_custom_funnel1: object with this id does not exist: 1697712429736x604475"</pre>

Test case ID	D4.17 Replace deal data: lead_custom_lead - integer
Description	Check that lead_custom_lead data cannot be replaced with integer.
Precondition	A deal exists in the database
Test data	<pre> 1 { 2 "assignee__user": "1625047373081x778685405315052900", 3 "company_name_text": "Team1 test A-M PUT2", 4 "deal_value_estimation_number": 1, 5 "description_text": "This is a test", 6 "file_list_custom_data_file": [], 7 "funnel_custom_funnel1": "1697712429736x604475436022720900", 8 "lead_custom_lead": 123, 9 "logo_image": "//s3.amazonaws.com/appforest_uf/f1625048400000 10 "name_text": "Team1_A-M PUT2", 11 "note_list_custom_note": [12 "1678718690483x188510294479732740" 13], 14 "order_number": 1, 15 "stage_custom_stage": "1625047818390x664413796313397600", 16 "status_option_status": "Won", 17 "visible_to_list_user": [18 "1625047373081x778685405315052900", 19 "1625047362532x398219546419355650" 20] 21 }</pre>

	<pre> 19 "1625047362532x398219546419355650" 20] 21 } </pre>
Test steps	1. Send PUT request <code>{{baseUrl}}/obj/deal/:UniqueID</code> with integer value in <code>lead_custom_lead</code>
Expected result	<ul style="list-style-type: none"> Error code: 400 Bad Request Object data is not replaced. Error message is displayed "message": <code>"Invalid data for field lead_custom_lead: object with this id does not exist: 123"</code>

Test case ID	D4.18 Replace deal data: lead_custom_lead - partial lead Id
Description	Check that inserting partial lead Id does not link a lead to a deal.
Precondition	A deal and a lead exists in the database
Test data	<pre> 1 { 2 "assignee__user": "1625047373081x778685405315052900", 3 "company_name_text": "Team1 test A-M PUT2", 4 "deal_value_estimation_number": 1, 5 "description_text": "This is a test", 6 "file_list_custom_data_file": [], 7 "funnel_custom_funnel1": "1697712429736x604475436022720900", 8 "lead_custom_lead": "042937x450280473509041340", 9 "logo_image": "//s3.amazonaws.com/appforest_uf/f1625048400000", 10 "name_text": "Team1_A-M PUT2", 11 "note_list_custom_note": [12 "1678718690483x188510294479732740" 13], 14 "order_number": 1, 15 "stage_custom_stage": "1625047818390x664413796313397600", 16 "status_option_status": "Won", 17 "visible_to_list_user": [18 "1625047373081x778685405315052900", 19 "1625047362532x398219546419355650" 20] 21 } </pre>
Test steps	1. Send PUT request <code>{{baseUrl}}/obj/deal/:UniqueID</code> with partial lead Id in <code>lead_custom_lead</code>
Expected result	<ul style="list-style-type: none"> Error code: 400 Bad Request Object data is not replaced. Error message is displayed "message": <code>"Invalid data for field lead_custom_lead: object with this id does not exist: 042937x450280473509041340"</code>

Test case ID	D4.19 Replace deal data: name_text - empty
Description	Check that it's not possible to leave name_text empty in PUT request.
Precondition	A deal exists in the database
Test data	<pre> 1 { 2 "assignee__user": "1625047373081x778685405315052900", </pre>

```

3   "company_name_text": "Team1 test A-M PUT2",
4   "deal_value_estimation_number": 1,
5   "description_text": "This is a test",
6   "file_list_custom_data_file": [],
7   "funnel_custom_funnel1": "1697712429736x604475436022720900",
8   "lead_custom_lead": "1697609042937x450280473509041340",
9   "logo_image": "//s3.amazonaws.com/appforest_uf/f1625048400000
10  "name_text": "",
11  "note_list_custom_note": [
12      "1678718690483x188510294479732740"
13 ],
14  "order_number": 1,
15  "stage_custom_stage": "1625047818390x664413796313397600",
16  "status_option_status": "Won",
17  "visible_to_list_user": [
18      "1625047373081x778685405315052900",
19      "1625047362532x398219546419355650"
20 ]
21 }

```

Test steps	1. Send PUT request <code>{{baseUrl}}/obj/deal/:UniqueID</code> with empty name_text
Expected result	<ul style="list-style-type: none"> Error code: 400 Bad Request Proper error message is displayed

Test case ID	D4.20 Replace deal data: name_text - integer
Description	Check that name_text data cannot be replaced with integer.
Precondition	A deal exists in the database
Test data	<pre> 1 { 2 "assignee__user": "1625047373081x778685405315052900", 3 "company_name_text": "Team1 test A-M PUT2", 4 "deal_value_estimation_number": 1, 5 "description_text": "This is a test", 6 "file_list_custom_data_file": [], 7 "funnel_custom_funnel1": "1697712429736x604475436022720900", 8 "lead_custom_lead": "1697609042937x450280473509041340", 9 "logo_image": "//s3.amazonaws.com/appforest_uf/f1625048400000 10 "name_text": 123, 11 "note_list_custom_note": [12 "1678718690483x188510294479732740" 13], 14 "order_number": 1, 15 "stage_custom_stage": "1625047818390x664413796313397600", 16 "status_option_status": "Won", 17 "visible_to_list_user": [18 "1625047373081x778685405315052900", 19 "1625047362532x398219546419355650" 20] 21 } </pre>
Test steps	1. Send PUT request <code>{{baseUrl}}/obj/deal/:UniqueID</code> with integer value in name_text
Expected result	<ul style="list-style-type: none"> Error code: 400 Bad Request

- Object data is not replaced. Error message is displayed "message":
 "Invalid data for field note_list_custom_note: object with this id does not exist: 123"

Test case ID	D4.21 Replace deal data: note_list_custom_note - integer
Description	Check that note_list_custom_note data cannot be replaced with integer.
Precondition	A deal exists in the database
Test data	<pre> 1 { 2 "assignee__user": "1625047373081x778685405315052900", 3 "company_name_text": "Team1 test A-M PUT2", 4 "deal_value_estimation_number": 1, 5 "description_text": "This is a test", 6 "file_list_custom_data_file": [], 7 "funnel_custom_funnel1": "1697712429736x604475436022720900", 8 "lead_custom_lead": "1697609042937x450280473509041340", 9 "logo_image": "//s3.amazonaws.com/appforest_uf/f1625048400000 10 "name_text": "Team1_A-M PUT2", 11 "note_list_custom_note": [12 123 13], 14 "order_number": 1, 15 "stage_custom_stage": "1625047818390x664413796313397600", 16 "status_option_status": "Won", 17 "visible_to_list_user": [18 "1625047373081x778685405315052900", 19 "1625047362532x398219546419355650" 20] 21 }</pre>
Test steps	<ol style="list-style-type: none"> Send PUT request {{baseUrl}}/obj/deal/:UniqueID with integer value in note_list_custom_note
Expected result	<ul style="list-style-type: none"> Error code: 400 Bad Request Object data is not replaced. Error message is displayed "message": "Invalid data for field note_list_custom_note: object with this id does not exist: 123"

Test case ID	D4.22 Replace deal data: note_list_custom_note - partial note Id
Description	Check that inserting partial note Id does not link a note to a deal.
Precondition	A deal and a note exists in the database
Test data	<pre> 1 { 2 "assignee__user": "1625047373081x778685405315052900", 3 "company_name_text": "Team1 test A-M PUT2", 4 "deal_value_estimation_number": 1, 5 "description_text": "This is a test", 6 "file_list_custom_data_file": [], 7 "funnel_custom_funnel1": "1697712429736x604475436022720900", 8 "lead_custom_lead": "1697609042937x450280473509041340", 9 "logo_image": "//s3.amazonaws.com/appforest_uf/f1625048400000 10 "name_text": "Team1_A-M PUT2", 11 "note_list_custom_note": [</pre>

	<pre> 12 "78718690483x188510294479732740" 13], 14 "order_number": 1, 15 "stage_custom_stage": "1625047818390x664413796313397600", 16 "status_option_status": "Won", 17 "visible_to_list_user": [18 "1625047373081x778685405315052900", 19 "1625047362532x398219546419355650" 20] 21 } </pre>
Test steps	<p>1. Send PUT request {{baseUrl}}/obj/deal/:UniqueID with partial note Id in note_list_custom_note</p>
Expected result	<ul style="list-style-type: none"> Error code: 400 Bad Request Object data is not replaced. Error message is displayed "message": <code>"Invalid data for field note_list_custom_note: object with this id does not exist: 78718690483x188510294479732740"</code>

Test case ID	D4.23 Replace deal data: order_number - empty
Description	Check that it's not possible to leave order_number empty in PUT request.
Precondition	A deal exists in the database
Test data	<pre> 1 { 2 "assignee__user": "1625047373081x778685405315052900", 3 "company_name_text": "Team1 test A-M PUT2", 4 "deal_value_estimation_number": 1, 5 "description_text": "This is a test", 6 "file_list_custom_data_file": [], 7 "funnel_custom_funnel1": "1697712429736x604475436022720900", 8 "lead_custom_lead": "1697609042937x450280473509041340", 9 "logo_image": "//s3.amazonaws.com/appforest_uf/f1625048400000 10 "name_text": "Team1_A-M PUT2", 11 "note_list_custom_note": [12 "1678718690483x188510294479732740" 13], 14 "order_number": , 15 "stage_custom_stage": "1625047818390x664413796313397600", 16 "status_option_status": "Won", 17 "visible_to_list_user": [18 "1625047373081x778685405315052900", 19 "1625047362532x398219546419355650" 20] 21 } </pre>
Test steps	<p>1. Send PUT request {{baseUrl}}/obj/deal/:UniqueID with empty order_number</p>
Expected result	<ul style="list-style-type: none"> Error code: 400 Bad Request Error message is displayed Error: Error parsing request body: <code>Unexpected token</code>

Test case ID	D4.24 Replace deal data: order_number - string
--------------	--

Description	Check that order_number data cannot be replaced with string.
Precondition	A deal exists in the database
Test data	<pre> 1 { 2 "assignee__user": "1625047373081x778685405315052900", 3 "company_name_text": "Team1 test A-M PUT2", 4 "deal_value_estimation_number": 1, 5 "description_text": "This is a test", 6 "file_list_custom_data_file": [], 7 "funnel_custom_funnel1": "1697712429736x604475436022720900", 8 "lead_custom_lead": "1697609042937x450280473509041340", 9 "logo_image": "//s3.amazonaws.com/appforest_uf/f1625048400000 10 "name_text": "Team1_A-M PUT2", 11 "note_list_custom_note": [12 "1678718690483x188510294479732740" 13], 14 "order_number": "123", 15 "stage_custom_stage": "1625047818390x664413796313397600", 16 "status_option_status": "Won", 17 "visible_to_list_user": [18 "1625047373081x778685405315052900", 19 "1625047362532x398219546419355650" 20] 21 }</pre>
Test steps	1. Send PUT request {{baseUrl}}/obj/deal/:UniqueID with string value in order_number
Expected result	<ul style="list-style-type: none"> Error code: 400 Bad Request Object data is not replaced. Error message is displayed "message": "Invalid data for field company_name_text: Expected an integer, but got a string (original data: "1")"

Test case ID	D4.25 Replace deal data: stage_custom_stage - integer
Description	Check that stage_custom_stage data cannot be replaced with integer.
Precondition	A deal exists in the database
Test data	<pre> 1 { 2 "assignee__user": "1625047373081x778685405315052900", 3 "company_name_text": "Team1 test A-M PUT2", 4 "deal_value_estimation_number": 1, 5 "description_text": "This is a test", 6 "file_list_custom_data_file": [], 7 "funnel_custom_funnel1": "1697712429736x604475436022720900", 8 "lead_custom_lead": "1697609042937x450280473509041340", 9 "logo_image": "//s3.amazonaws.com/appforest_uf/f1625048400000 10 "name_text": "Team1_A-M PUT2", 11 "note_list_custom_note": [12 "1678718690483x188510294479732740" 13], 14 "order_number": 1, 15 "stage_custom_stage": 123, 16 "status_option_status": "Won", 17 "visible_to_list_user": [18 "1625047373081x778685405315052900", 19 "1625047362532x398219546419355650" 20] 21 }</pre>

	<pre> 19 "1625047362532x398219546419355650" 20] 21 } </pre>
Test steps	1. Send PUT request {{baseUrl}}/obj/deal/:UniqueID with integer value in stage_custom_stage
Expected result	<ul style="list-style-type: none"> Error code: 400 Bad Request Object data is not replaced. Error message is displayed "message": "Invalid data for field stage_custom_stage: object with this id does not exist: 123"

Test case ID	D4.26 Replace deal data: stage_custom_stage - partial stage Id
Description	Check that inserting partial stage Id does not link a stage to a deal.
Precondition	A deal and a stage exists in the database
Test data	<pre> 1 { 2 "assignee__user": "1625047373081x778685405315052900", 3 "company_name_text": "Team1 test A-M PUT2", 4 "deal_value_estimation_number": 1, 5 "description_text": "This is a test", 6 "file_list_custom_data_file": [], 7 "funnel_custom_funnel1": "1697712429736x604475436022720900", 8 "lead_custom_lead": "1697609042937x450280473509041340", 9 "logo_image": "//s3.amazonaws.com/appforest_uf/f1625048400000 10 "name_text": "Team1_A-M PUT2", 11 "note_list_custom_note": [12 "1678718690483x188510294479732740" 13], 14 "order_number": 1, 15 "stage_custom_stage": "25047818390x664413796313397600", 16 "status_option_status": "Won", 17 "visible_to_list_user": [18 "1625047373081x778685405315052900", 19 "1625047362532x398219546419355650" 20] 21 } </pre>
Test steps	1. Send PUT request {{baseUrl}}/obj/deal/:UniqueID with partial stage Id in stage_custom_stage
Expected result	<ul style="list-style-type: none"> Error code: 400 Bad Request Object data is not replaced. Error message is displayed "message": "Invalid data for field stage_custom_stage: object with this id does not exist: 25047818390x664413796313397600"

Test case ID	D4.27 Replace deal data: status_option_status - integer
Description	Check that status_option_status data cannot be replaced with integer.
Precondition	A deal exists in the database
Test data	<pre> 1 { 2 "assignee__user": "1625047373081x778685405315052900", </pre>

```

3   "company_name_text": "Team1 test A-M PUT2",
4   "deal_value_estimation_number": 1,
5   "description_text": "This is a test",
6   "file_list_custom_data_file": [],
7   "funnel_custom_funnel1": "1697712429736x604475436022720900",
8   "lead_custom_lead": "1697609042937x450280473509041340",
9   "logo_image": "//s3.amazonaws.com/appforest_uf/f1625048400000
10  "name_text": "Team1_A-M PUT2",
11  "note_list_custom_note": [
12      "1678718690483x188510294479732740"
13  ],
14  "order_number": 1,
15  "stage_custom_stage": "1625047818390x664413796313397600",
16  "status_option_status": 123,
17  "visible_to_list_user": [
18      "1625047373081x778685405315052900",
19      "1625047362532x398219546419355650"
20  ]
21 }

```

Test steps	1. Send PUT request {{baseUrl}}/obj/deal/:UniqueID with integer value in status_option_status
Expected result	<ul style="list-style-type: none"> Error code: 400 Bad Request Object data is not replaced. Error message is displayed "message": "Invalid data for field status_option_status: could not parse this as a Status"

Test case ID	D4.28 Replace deal data: status_option_status - invalid data
Description	Check that status_option_status data cannot be replaced with string other than "Won", "In progress", "Lost"
Precondition	A deal exists in the database
Test data	<pre> 1 { 2 "assignee__user": "1625047373081x778685405315052900", 3 "company_name_text": "Team1 test A-M PUT2", 4 "deal_value_estimation_number": 1, 5 "description_text": "This is a test", 6 "file_list_custom_data_file": [], 7 "funnel_custom_funnel1": "1697712429736x604475436022720900", 8 "lead_custom_lead": "1697609042937x450280473509041340", 9 "logo_image": "//s3.amazonaws.com/appforest_uf/f1625048400000 10 "name_text": "Team1_A-M PUT2", 11 "note_list_custom_note": [12 "1678718690483x188510294479732740" 13], 14 "order_number": 1, 15 "stage_custom_stage": "1625047818390x664413796313397600", 16 "status_option_status": "Incorrect", 17 "visible_to_list_user": [18 "1625047373081x778685405315052900", 19 "1625047362532x398219546419355650" 20] 21 } </pre>

Test steps	1. Send PUT request {{baseUrl}}/obj/deal/:UniqueID with "Incorrect" in status_option_status
Expected result	<ul style="list-style-type: none"> Error code: 400 Bad Request Object data is not replaced. Error message is displayed "message": "Invalid data for field status_option_status: could not parse this as a Status"

Test case ID	D4.29 Replace deal data: visible_to_list_user - empty
Description	Check that it's not possible to leave visible_to_list_user empty in PUT request.
Precondition	A deal exists in the database
Test data	<pre> 1 { 2 "assignee__user": "1625047373081x778685405315052900", 3 "company_name_text": "Team1 test A-M PUT2", 4 "deal_value_estimation_number": 1, 5 "description_text": "This is a test", 6 "file_list_custom_data_file": [], 7 "funnel_custom_funnel1": "1697712429736x604475436022720900", 8 "lead_custom_lead": "1697609042937x450280473509041340", 9 "logo_image": "//s3.amazonaws.com/appforest_uf/f1625048400000 10 "name_text": "Team1_A-M PUT2", 11 "note_list_custom_note": [12 "1678718690483x188510294479732740" 13], 14 "order_number": 1, 15 "stage_custom_stage": "1625047818390x664413796313397600", 16 "status_option_status": "Won", 17 "visible_to_list_user": [18 19] 20 }</pre>
Test steps	1. Send PUT request {{baseUrl}}/obj/deal/:UniqueID with empty visible_to_list_user
Expected result	<ul style="list-style-type: none"> Error code: 400 Bad Request Proper error message is displayed

Test case ID	D4.30 Replace deal data: visible_to_list_user - integer
Description	Check that visible_to_list_user data cannot be replaced with integer.
Precondition	A deal exists in the database
Test data	<pre> 1 { 2 "assignee__user": "1625047373081x778685405315052900", 3 "company_name_text": "Team1 test A-M PUT2", 4 "deal_value_estimation_number": 1, 5 "description_text": "This is a test", 6 "file_list_custom_data_file": [], 7 "funnel_custom_funnel1": "1697712429736x604475436022720900", 8 "lead_custom_lead": "1697609042937x450280473509041340", 9 "logo_image": "//s3.amazonaws.com/appforest_uf/f1625048400000</pre>

	<pre> 10 "name_text": "Team1_A-M PUT2", 11 "note_list_custom_note": [12 "1678718690483x188510294479732740" 13], 14 "order_number": 1, 15 "stage_custom_stage": "1625047818390x664413796313397600", 16 "status_option_status": "Won", 17 "visible_to_list_user": [18 123 19] 20 } </pre>
Test steps	<p>1. Send PUT request <code>{{baseUrl}}/obj/deal/:UniqueID</code> with integer value in <code>visible_to_list_user</code></p>
Expected result	<ul style="list-style-type: none"> Error code: 400 Bad Request Object data is not replaced. Error message is displayed "message": <code>"Invalid data for field visible_to_list_user: object with this id does not exist: 123"</code>

Test case ID	D4.31 Replace deal data: <code>visible_to_list_user</code> - partial user Id
Description	Check that inserting partial user Id does not link a user to a deal.
Precondition	A deal and a user exists in the database
Test data	<pre> 1 { 2 "assignee__user": "1625047373081x778685405315052900", 3 "company_name_text": "Team1 test A-M PUT2", 4 "deal_value_estimation_number": 1, 5 "description_text": "This is a test", 6 "file_list_custom_data_file": [], 7 "funnel_custom_funnel1": "1697712429736x604475436022720900", 8 "lead_custom_lead": "1697609042937x450280473509041340", 9 "logo_image": "//s3.amazonaws.com/appforest_uf/f1625048400000", 10 "name_text": "Team1_A-M PUT2", 11 "note_list_custom_note": [12 "1678718690483x188510294479732740" 13], 14 "order_number": 1, 15 "stage_custom_stage": "1625047818390x664413796313397600", 16 "status_option_status": "Won", 17 "visible_to_list_user": [18 "1625", 19 "047362532x398219546419355650" 20] 21 } </pre>
Test steps	<p>1. Send PUT request <code>{{baseUrl}}/obj/deal/:UniqueID</code> with partial user Id in <code>visible_to_list_user</code></p>
Expected result	<ul style="list-style-type: none"> Error code: 400 Bad Request Object data is not replaced. Error message is displayed "message": <code>"Invalid data for field visible_to_list_user: object with this id does not exist: 1625"</code>

Delete deal

Test case ID	D5.1 Delete Deals with existing ID
Description	Check that it's possible to delete Deals with existing ID
Precondition	-
Test data	-
Test steps	1. Send DELETE {baseUrl}/obj/deal/:UniqueID with existing ID
Expected result	<ul style="list-style-type: none"> Successful code: 204 .Funnel deleted Object is deleted without any errors.

Test case ID	D5.2 Delete Deals without being authorized.
Description	Check taht it's not possible to delete Deals without being authorized.
Precondition	-
Test data	-
Test steps	1. Send DELETE {baseUrl}/obj/deal/:UniqueID without being authorized
Expected result	<ul style="list-style-type: none"> Permission denied code: 401 Unauthorized Error code is displayed.

Test case ID	D5.3 Delete Deals with partial ID
Description	Check that it's not possible to delete Deals with partial ID.
Precondition	-
Test data	-
Test steps	1. Send DELETE {baseUrl}/obj/deal/:UniqueID with partial ID
Expected result	<ul style="list-style-type: none"> Error code: 404 Not found Error code is displayed.

Test case ID	D5.4 Delete same Deal twice
Description	Check that it's not possible to delete same Deal twice.
Precondition	-
Test data	-
Test steps	1. Send DELETE {baseUrl}/obj/deal/:UniqueID twice
Expected result	<ul style="list-style-type: none"> Error code: 404 Not found Error code is displayed.

Test case ID	D5.5 Delete Deals with empty Deal ID
---------------------	---

Description	Check that it's not possible to delete Deals without ID
Precondition	-
Test data	-
Test steps	1. Send DELETE {baseUrl}/obj/deal/:UniqueID without ID
Expected result	<ul style="list-style-type: none">• Error code: 404 Not found• Error code is displayed.