Team 7-2. Test cases

@Kadi-Kristel Kööp

- Deals API
  - POST
  - DELETE
- Notes API
  - GET
  - PATCH

CRM API baseUrl: https://cerebrumhubcrmlite.bubbleapps.io/api/1.1

Authorization token: b30ca93d0a7ebe389e9326a22497bf38

Deals API

POST

Test case ID	**ND001. Create a new deal with maximum dataset and check the created object**
Description	It is possible to create a new deal and observe if it was created successfully with given data
Precondition	User is authorized

Test data	{
	
	"assignee__user": "1717560677241x142582647964403460",
	
	"company_name_text": "Company for team7",
	
	"deal_value_estimation_number": 50505,
	
	"description_text": "New deal for team7",
	
	"file_list_custom_data_file": [
	
	],
	
	"logo_image": "https://icons.getbootstrap.com/icons/7-circle/",
	
	"funnel_custom_funnel1": "1717249949887x707204560107654700",
	
	"name_text": "Deal2",
	
	"order_number": 6,
	
	"stage_custom_stage": "1717561532498x465020404734400200",
	
	"status_option_status": "Won",
	
	"visible_to_list_user": [
	
	"1717560677241x142582647964403460",
	
	"1625051658962x581289845978797800",
	
	"1717488737309x417287978861723650",
	
	"1717475368368x181690397246576420",
	
	"1717472985666x307445402784460200"
	
	]
	
	}

Test steps	1. Send **POST**: {baseUrl}/obj/deal request
	2. Copy ID of the created deal from the response
	3. Send **GET**: {baseUrl}/obj/deal/:UniqueID, set correct UniqueID (taken from Step 2)
Expected result	Step 1: Successful code: 201 Created, new deal is created
	Step 3: Successful code: 200 OK, correct deal is found, data corresponds to input.

Test case ID	**ND002. Create a new deal with minimum dataset and check the created object**
Description	It is possible to create a new deal with only inserting mandatory data and observe if it was created successfully with given data
Precondition	User is authorized

Test data	{
	
	  "assignee__user": "1717560677241x142582647964403460",
	
	  "company_name_text": "K's Company",
	
	  "deal_value_estimation_number": 13535,
	
	  "name_text": "Lottery",
	
	  "order_number": 5,
	
	  "status_option_status": "In progress",
	
	  "visible_to_list_user": [
	
	    "1717560677241x142582647964403460"
	
	  ]
	
	}

Test steps	1. Send **POST**: {baseUrl}/obj/deal request
	2. Copy ID of the created deal from the response
	3. Send **GET**: {baseUrl}/obj/deal/:UniqueID, set correct UniqueID (taken from Step 2)
Expected result	Step 1: Successful code: 201 Created, new deal is created
	Step 3: Successful code: 200 OK, correct deal is found, data corresponds to input.

Test case ID	**ND003. Create a new deal - no authorization**
Description	It is **NOT** possible to create a new deal without being authorized
Precondition	User is not authorized

Test data	{
	
	  "assignee__user": "1717560677241x142582647964403460",
	
	  "company_name_text": "Company for team7",
	
	  "deal_value_estimation_number": 50506,
	
	  "description_text": "Deal for team7 - no auth",
	
	  "file_list_custom_data_file": [
	
	  ],
	
	    "logo_image": "https://icons.getbootstrap.com/icons/7-circle/",
	
	  "funnel_custom_funnel1": "1717249949887x707204560107654700",
	
	  "name_text": "Deal2 no auth",
	
	  "order_number": 6,
	
	  "stage_custom_stage": "1717561532498x465020404734400200",
	
	  "status_option_status": "In progress",
	
	  "visible_to_list_user": [
	
	    "1717560677241x142582647964403460",
	
	    "1625051658962x581289845978797800",
	
	    "1717488737309x417287978861723650"
	
	  ]
	
	}

Test steps	1. Send **POST**: {baseUrl}/obj/deal request
Expected result	• Error code: 401 Unauthorized • Error message displayed: "Permission denied: cannot create this object"

Test case ID	**ND004. Create a new deal with empty body**
Description	It is **NOT** possible to create a new deal with empty body
Precondition	User is authorized

Test data	{
	
	}

Test steps	1. Send **POST**: {baseUrl}/obj/deal request with empty body
Expected result	• Error code: 400 Bad Request. • Error message displayed: "Request body is missing. Please provide required fields."

Test case ID	ND005. Create a new deal with empty fields
Description	It is **NOT** possible to create a new deal with empty fields
Precondition	User is authorized

Test data	{
	  "assignee__user": "",
	  "company_name_text": "",
	
	  "description_text": "",
	  "file_list_custom_data_file": [
	  ],
	  "logo_image": "",
	  "funnel_custom_funnel1": "",
	  "name_text": "",
	
	  "stage_custom_stage": "",
	  "status_option_status": "",
	  "visible_to_list_user": [
	
	  ]
	}

Test steps	1. Send **POST**: {baseUrl}/obj/deal request with empty fields
Expected result	• Error code: 400 Bad Request. • Error message displayed: "The following mandatory fields cannot be empty: assignee__user, company_name_text, deal_value_estimation_number, name_text, order_number, status_option_status, visible_to_list_user. Please provide values for these fields to create a new deal."

Test case ID	**ND006. Create a new deal without mandatory fields**
Description	It is **NOT** possible to create a new deal without adding data in mandatory fields
Precondition	User is authorized

Test data	{
	
	  "assignee__user": "",
	
	  "company_name_text": "Company for team7",
	
	  "description_text": "New deal for team7",
	
	  "file_list_custom_data_file": [
	
	  ],
	
	    "logo_image": "https://icons.getbootstrap.com/icons/7-circle/",
	
	  "funnel_custom_funnel1": "1717249949887x707204560107654700",
	
	  "name_text": "",
	
	  "stage_custom_stage": "1717561532498x465020404734400200",
	
	  "visible_to_list_user": [
	
	  ]
	}

Test steps	1. Send **POST**: {baseUrl}/obj/deal request without adding data in mandatory fields
Expected result	• Error code: 400 Bad Request • Error message displayed: "The following mandatory fields cannot be empty: assignee__user, company_name_text, deal_value_estimation_number, name_text, order_number, status_option_status, visible_to_list_user. Please provide values for these fields to create a new deal."

Test case ID	ND007. Create a new deal with non-existent visible_to_list_user
Description	It is **NOT** possible to create a new deal with non-existent visible_to_list user
Precondition	User is authorized

Test data	{
	
	  "assignee__user": "1717560677241x142582647964403460",
	
	  "company_name_text": "Company for team7",
	
	  "deal_value_estimation_number": 11,
	
	  "description_text": "New deal for team7",
	
	  "file_list_custom_data_file": [
	
	  ],
	
	    "logo_image": "https://icons.getbootstrap.com/icons/7-circle/",
	
	  "funnel_custom_funnel1": "1717249949887x707204560107654700",
	
	  "name_text": "Lorem ipsum 3",
	
	  "order_number": 14,
	
	  "stage_custom_stage": "1717561532498x465020404734400200",
	
	  "status_option_status": "Won",
	
	  "visible_to_list_user": [
	
	    "1717560677241"
	
	  ]
	
	}

Test steps	1. Send **POST**: {baseUrl}/obj/deal request with non-existent visible_to_list_user
Expected result	• Error code: 400 Bad Request. • Error message displayed: "Invalid data for field visible_to_list_user: object with this id does not exist: 1717560677241"

Test case ID	ND008. Create a new deal with stage_option_status type boolean
Description	It is **NOT** possible to create a new deal with stage_option_status type boolean
Precondition	User is authorized

Test data	{
	
	  "assignee__user": "1717560677241x142582647964403460",
	
	  "company_name_text": "7 Company",
	
	  "deal_value_estimation_number": 123,
	
	  "description_text": "lorem ipsum 7",
	
	  "file_list_custom_data_file": [
	
	  ],
	
	    "logo_image": "https://icons.getbootstrap.com/icons/7-circle/",
	
	 "funnel_custom_funnel1": "1717249949887x707204560107654700",
	
	  "name_text": "lorem ipsum7",
	
	  "order_number": 15,
	
	 "stage_custom_stage": "1717561532498x465020404734400200",
	
	  "status_option_status": false,
	
	  "visible_to_list_user": [
	
	    "1717560677241x142582647964403460",
	
	    "1717488737309x417287978861723650",
	
	    "1717475368368x181690397246576420",
	
	    "1717472985666x307445402784460200"
	
	  ]
	
	}

Test steps	1. Send **POST**: {baseUrl}/obj/deal request with stage_option_status type boolean
Expected result	• Error code: 400 Bad Request. • Error message displayed: "Invalid data for field stage_option_status: Expected a string, but got a boolean (original data: \"false\")"

Test case ID	ND009. Create a new deal with order_number type string
Description	It is **NOT** possible to create a new deal with order_number type string
Precondition	User is authorized

Test data	{
	
	"assignee__user": "1717560677241x142582647964403460",
	
	"company_name_text": "Company for team7",
	
	"deal_value_estimation_number": 120,
	
	"description_text": "lorem ipsum 7",
	
	"file_list_custom_data_file": [
	
	],
	
	"logo_image": "https://icons.getbootstrap.com/icons/7-circle/",
	
	"funnel_custom_funnel1": "1717249949887x707204560107654700",
	
	"name_text": "lorem ipsum 7",
	
	"order_number": "One",
	
	"stage_custom_stage": "1717561532498x465020404734400200",
	
	"status_option_status": "In progress",
	
	"visible_to_list_user": [
	
	"1717560677241x142582647964403460",
	
	"1717488737309x417287978861723650",
	
	"1717475368368x181690397246576420",
	
	"1717472985666x307445402784460200"
	
	]
	
	}

Test steps	1. Send **POST**: {baseUrl}/obj/deal request with order_number type string
Expected result	• Error code: 400 Bad Request. • Error message displayed: "status": "INVALID_DATA", "message": "Invalid data for field order_number: Expected a number, but got a string (original data: \"One\")"

Test case ID	ND010. Create a new deal with company_name_text type numeric
Description	It is **NOT** possible to create a new deal with company_name_text type numeric
Precondition	User is authorized

Test data	{
	
	  "assignee__user": "1717560677241x142582647964403460",
	  "company_name_text": 7,
	  "deal_value_estimation_number": 123,
	  "description_text": "lorem ipsum 7",
	  "file_list_custom_data_file": [
	  ],
	    "logo_image": "https://icons.getbootstrap.com/icons/7-circle/",
	  "funnel_custom_funnel1": "1717249949887x707204560107654700",
	  "name_text": "lorem ipsum7",
	  "order_number": 15,
	  "stage_custom_stage": "1717561532498x465020404734400200",
	  "status_option_status": "In progress",
	  "visible_to_list_user": [
	    "1717560677241x142582647964403460",
	    "1717488737309x417287978861723650",
	    "1717475368368x181690397246576420",
	    "1717472985666x307445402784460200"
	  ]
	}

Test steps	1. Send **POST**: {baseUrl}/obj/deal request with company_name_text type numeric
Expected result	• Error code: 400 Bad Request. • Error message displayed: "status": "INVALID_DATA", "message": "Invalid data for field company_name_text: Expected a string, but got a number (original data: 7)"

DELETE

Test case ID	DD001. Delete deal and check data
Description	It is possible to delete an existing deal
Precondition	- Some deals existing in the system - User is authorized
Test data	Existing deal ID
Test steps	1. Send **DELETE**: {baseUrl}obj/deal/:UniqueID 2. Send **GET**: {baseUrl}obj/deal/:UniqueID, set correct UniqueID (taken from Step 1)
Expected result	Step 1: Error code: 204 No Content, delete was successful Step 2: Error code: 404 Not Found, "status": "MISSING_DATA", "message": "Missing object of type deal: object with id {{UniqueId}} does not exist"

Test case ID	DD002. Delete deal - no authorization
Description	It is **NOT** possible to delete an existing deal without being authorized
Precondition	- Some deals existing in the system - User is not authorized
Test data	Existing deal ID
Test steps	1. Send **DELETE**: {baseUrl}obj/deal/:UniqueID
Expected result	- Error code: 401 Unauthorized. - Error message displayed: "You do not have permission to delete this object"

Test case ID	**DD003. Delete non-existent deal**
Description	It is **NOT** possible to delete a non-existent deal
Precondition	• Some deals existing in the system • User is authorized
Test data	Use previously deleted deal id (e.g 1717651411429x242358675937661700)
Test steps	1. Send **DELETE**: {baseUrl}obj/deal/:UniqueID
Expected result	• Error code: 404 Not Found • Error message displayed: "status": "MISSING_DATA", "Missing object of type deal: object with id 1717651411429x242358675937661700 does not exist"

Notes API

GET

Test case ID	NG001. Get the list of notes
Description	It is possible to retrieve all existing notes
Precondition	• Some notes existing in the application • User is authorized
Test data	-
Test steps	1. Send **GET**: {baseUrl}obj/note request
Expected result	• Successful code: 200 OK • List of all the notes are returned in response

Test case ID	NG002. Get the list of notes - no authorization
Description	It is **NOT** possible to retrieve all existing notes without being authorized
Precondition	• Some notes existing in the application • User is not authorized
Test data	-
Test steps	1. Send **GET**: {baseUrl}obj/note request
Expected result	• Error code: 401 Unauthorized • Error message displayed: "You do not have permission to get this object"

Test case ID	NG003. Get the list of notes with positive limit (25)
Description	It is possible to fetch first 25 existing notes
Precondition	• 25 notes existing in the application • User is authorized
Test data	-
Test steps	1. Send **GET**: {baseUrl}obj/note?limit=25 request
Expected result	• Successful code: 200 OK • List of the 25 notes are returned in response

Test case ID	**NG004. Get the list of notes with negative limit (-1)**
Description	It is **NOT** possible to retrieve existing notes with limit being negative
Precondition	• Some notes existing in the application • User is authorized
Test data	-
Test data	1. Send **GET**: {baseUrl}obj/note?limit=-1 request
Expected result	• Error code: 400 Bad Request. • Error message displayed: "Invalid data for field limit"

Test case ID	**NG005. Get the list of notes with negative cursor**
Description	It is **NOT** possible to retrieve existing notes with cursor being negative
Precondition	• Some notes existing in the application • User is authorized
Test data	-
Test steps	1. Send **GET**: {baseUrl}obj/note?cursor=-1 request
Expected result	• Error code: 400 Bad Request. • Error message displayed: "Invalid data for field cursor"

Test case ID	**NG006. Get the list of notes with cursor 0**
Description	It is possible to retrieve the list of notes from the first note
Precondition	• At least one note existing in the application • User is authorized
Test data	-
Test steps	1. Send **GET**: {baseUrl}obj/note?cursor=0 request
Expected result	• Successful code: 200 OK. • List of the notes retrieved in response, starting from the first note

Test case ID	NG007. Get the list of notes with cursor 5
Description	It is possible to retrieve existing notes starting from the 5th note
Precondition	• At least 5 notes existing in the application • User is authorized
Test data	-
Test steps	1. Send **GET**: {baseUrl}obj/note?cursor=5 request
Expected result	• Successful code: 200 OK. • List of the notes retrieved in response, starting from the 5th note

Test case ID	NG008. Get the list of notes - sorting by created date descending
Description	It is possible to retrieve all existing notes sorted by created dated descending
Precondition	• Some notes existing in the application • User is authorized
Test data	-
Test steps	1. Send **GET**: {baseUrl}obj/note?sort_field=Created Date&descending=true request
Expected result	• Successful code: 200 OK. • The most recent notes are shown first in the list of all the notes that are returned in response.

Test case ID	NG009. Get the list of notes - constraints
Description	It is possible to filter notes based on their content
Precondition	• Some notes existing in the application • User is authorized
Test data	-
Test steps	1. Send **GET**: {baseUrl}obj/note?constraints=[{ "key": "content_text", "constraint_type": "text contains", "value": "customer" }] request
Expected result	• Successful code: 200 OK. • Matching note is displayed in response

PATCH

Test case ID	**MND001. Modify note data and check the modified note**
Description	It is possible to modify existing note data and observe if it was modified successfully with given data
Precondition	• Some notes exisiting in the application • User is authorized

Test data	{
	    "content_text": "Contacted customer via email. Waiting for their response."
	}

Test steps	1. Send **PATCH**: {baseUrl}/obj/note/:UniqueID request 2. Copy ID of the modified note from the Params tab under Path Variables 3. Send **GET**: {baseUrl}/obj/note/:UniqueID request, set the correct ID (taken from Step 2)
Expected result	Step 1: Successful code: 204 No Content Step 3: Successful code: 200 OK. Note is modified successfully

Test case ID	**MND002. Modify note data - no authorization**
Description	It is **NOT** possible to modify existing note data without being authorized
Precondition	• Some notes existing in the application • User is not authorized

Test data	{
	    "content_text": "Contacted customer via email."
	}

Test steps	1. Send **PATCH**: {baseUrl}/obj/note/:UniqueID request
Expected result	• Error code: 401 Unauthorized. • Error message displayed: "You do not have permission to modify this object"

Test case ID	**MND003. Modify note data - content_text type boolean**
Description	It is **NOT** possible to modify note data when the content_text type is boolean
Precondition	• Some notes existing in the application • User is authorized

Test data	{
	   "content_text": true
	}

Test steps	1. Send **PATCH**: {baseUrl}/obj/note/:UniqueID request
Expected result	• Error code: 400 Bad Request. • Error message displayed: "status": "INVALID_DATA", "message": "Invalid data for field content_text: Expected a string, but got a boolean (original data: true)"

Test case ID	**MND004. Modify note data - content_text type numeric**
Description	It is **NOT** possible to modify note data when the content_text type is numeric
Precondition	• Some notes existing in the application • User is authorized

Test data	{
	   "content_text": 3
	}

Test steps	1. Send **PATCH**: {baseUrl}/obj/note/:UniqueID request
Expected result	• Error code: 400 Bad Request. • Error message displayed: "status": "INVALID_DATA", "message": "Invalid data for field content_text: Expected a string, but got a number (original data: 3)"

Test case ID	MND005. Modify note data - empty body
Description	It is **NOT** possible to modify note data when the body is empty
Precondition	• Some notes existing in the application • User is authorized

Test data	{
	
	}

Test steps	1. Send **PATCH**: {baseUrl}/obj/note/:UniqueID request
Expected result	• Error code: 400 Bad Request. • Error message displayed: "Request body is missing. Please provide required field."

Test case ID	MND006. Modify note data - empty content_text
Description	It is **NOT** possible to modify note data when the content_text field is empty
Precondition	• Some notes existing in the application • User is authorized

Test data	{
	    "content_text": ""
	}

Test steps	1. Send **PATCH**: {baseUrl}/obj/note/:UniqueID request
Expected result	• Error code: 400 Bad Request. • Error message displayed: "content_text cannot be empty. Please provide value for required field."