Superleap Leads Management API – Complete Test Cases

Module: Leads Management
Base URL: https://aakash.superleap.com/leads_object
APIs Covered:

- POST /leads_object

- GET /leads_object/{id}

- PUT /leads_object/{id}

- DELETE /leads_object/{id}

- GET /leads_object

---

1. Create Lead (POST /leads_object)

TC ID	Description	Input Data (JSON)	Expected Result
TC_LEADS_001	Create lead with all valid fields	`{ "name": "John Doe", "email": "john.doe@example.com", "phone": "+1234567890", "company": "Example Corp", "status": "New" }`	✅ HTTP 201 Created Response contains non-empty `id`, all input fields, valid timestamp
TC_LEADS_002	Create lead with minimum mandatory fields (name + email)	`{ "name": "Jane Doe", "email": "jane.doe@example.com" }`	✅ HTTP 201 Created Optional fields are null/default
TC_LEADS_003	Missing mandatory field (email)	`{ "name": "John Doe" }`	❌ HTTP 400 Bad Request Error: missing email

TC_LEADS_004	Invalid email format	{ "name": "John Doe", "email": "invalidemail" }	❌ HTTP 400 Bad Request Error: invalid email
TC_LEADS_005	Invalid phone number format	{ "name": "John Doe", "email": "john.doe@example.com", "phone": "abcd" }	❌ HTTP 400 Bad Request Error: invalid phone format
TC_LEADS_006	Duplicate email (if uniqueness enforced)	{ "name": "John Smith", "email": "john.doe@example.com" }	❌ HTTP 409 Conflict Error: duplicate email
TC_LEADS_007	Empty payload	{}	❌ HTTP 400 Bad Request Error: missing mandatory fields
TC_LEADS_008	Unauthorized access	Valid payload, but no token	❌ HTTP 401 Unauthorized

2. Get Lead by ID (GET /leads_object/{id})

TC ID	Description	Input Data / ID	Expected Result
TC_LEADS_009	Fetch existing lead by valid ID	/leads_object/12345	✅ HTTP 200 OK Response contains all lead fields matching ID
TC_LEADS_010	Fetch lead with invalid ID	/leads_object/99999	❌ HTTP 404 Not Found
TC_LEADS_011	Fetch lead without authentication	/leads_object/12345	❌ HTTP 401 Unauthorized

3. Update Lead (PUT /leads_object/{id})

TC ID	Description	Input Data (JSON)	Expected Result

TC_LEADS_012	Update lead status and company	{ "status": "Contacted", "company": "Updated Corp" }	✅ HTTP 200 OK Lead updated, response reflects changes
TC_LEADS_013	Update with invalid field	{ "invalid_field": "test" }	❌ HTTP 400 Bad Request
TC_LEADS_014	Update non-existent lead	{ "status": "Lost" }	❌ HTTP 404 Not Found
TC_LEADS_015	Update without authentication	{ "status": "Lost" }	❌ HTTP 401 Unauthorized

4. Delete Lead (DELETE /leads_object/{id})

TC ID	Description	Input / ID	Expected Result
TC_LEADS_016	Delete existing lead	/leads_object/1 2345	✅ HTTP 200 OK or 204 No Content Lead removed successfully
TC_LEADS_017	Delete non-existent lead	/leads_object/9 9999	❌ HTTP 404 Not Found
TC_LEADS_018	Delete without authentication	/leads_object/1 2345	❌ HTTP 401 Unauthorized

5. List Leads (GET /leads_object)

TC ID	Description	Parameters	Expected Result
TC_LEADS_019	Fetch all leads	None	✅ HTTP 200 OK Returns list of leads
TC_LEADS_020	Filter leads by status	?status= New	✅ HTTP 200 OK Only leads with "New" status returned

| TC_LEADS_021 | Fetch leads without authentication | None | ❌ HTTP 401 Unauthorized |

6. Optional / Advanced Scenarios

TC ID	Description	Parameters	Expected Result
TC_LEADS_022	Pagination - limit and offset	`?limit=10&offset=0`	✅ Returns first 10 leads
TC_LEADS_023	Sorting by name	`?sort=name`	✅ Leads sorted alphabetically by name
TC_LEADS_024	Sorting by created date	`?sort=created_at`	✅ Leads sorted by date
TC_LEADS_025	Bulk create (if supported)	Array of multiple lead objects	✅ HTTP 201, multiple leads created
TC_LEADS_026	Bulk delete (if supported)	Array of lead IDs	✅ HTTP 200, all specified leads deleted

✅ Summary

- **Total Test Cases:** 26

- **Positive:** 10

- **Negative:** 16

- **Coverage:** Create, Read, Update, Delete, List, Filter, Pagination, Sorting, Auth