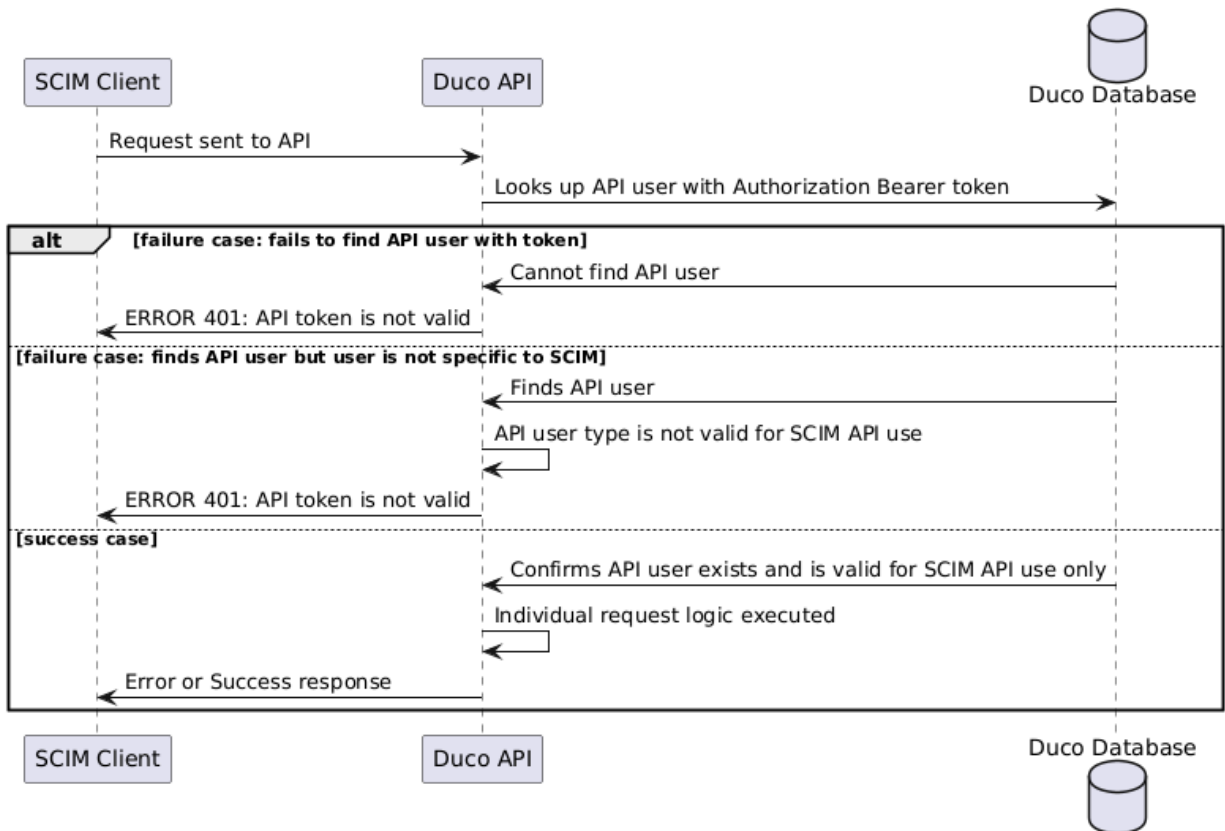


SCIM API Sequence Diagrams

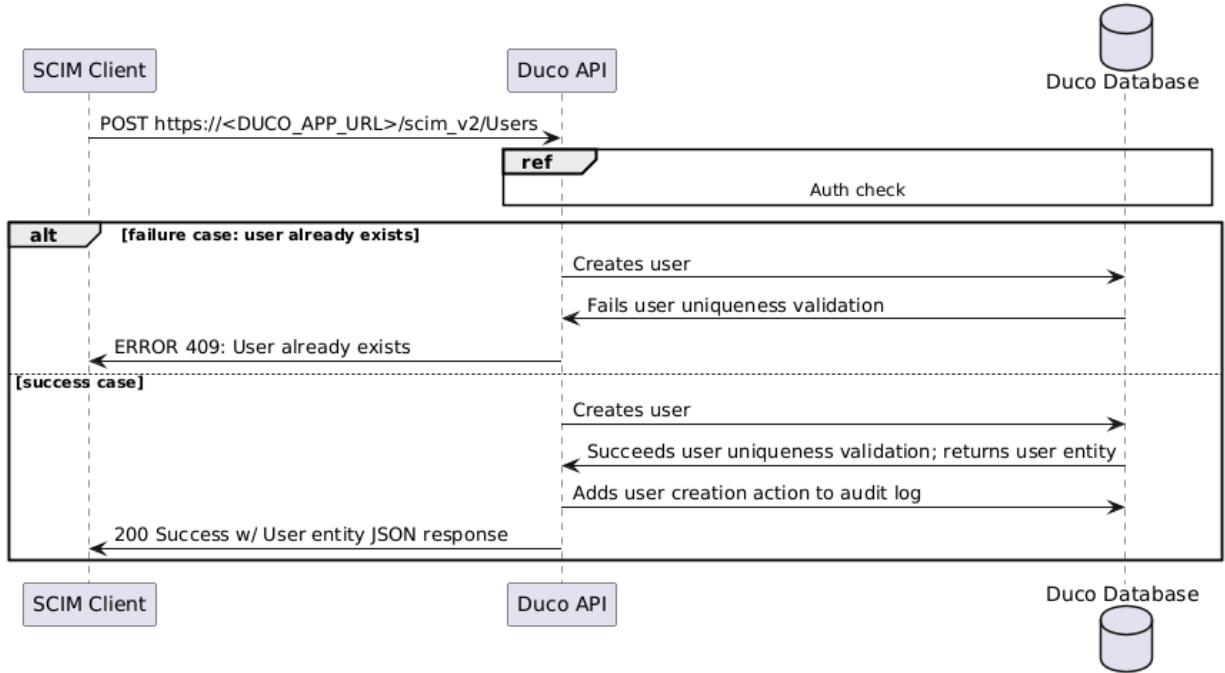
Authorization Bearer Token verification

This authorization logic is executed for every request sent to Duco's SCIM API. The Authorization Bearer token will be verified before executing any logic for a request. The diagram below shows the failure and success cases. All other diagrams in this document will shorten this check and assume the success case.



Users Endpoint

Creating a User



Request Payload:

```
POST https://<DUCO_APP_URL>/scim_v2/Users
Accept: application/scim+json
Content-Type: application/scim+json
Authorization: Bearer <BEARER_TOKEN>

{
  "schemas": [
    "urn:ietf:params:scim:schemas:core:2.0:User"
  ],
  "userName": "jane.doe@example.com",
  "name": {
    "familyName": "Doe",
    "givenName": "Jane"
  }
}
```

Failure case JSON response:

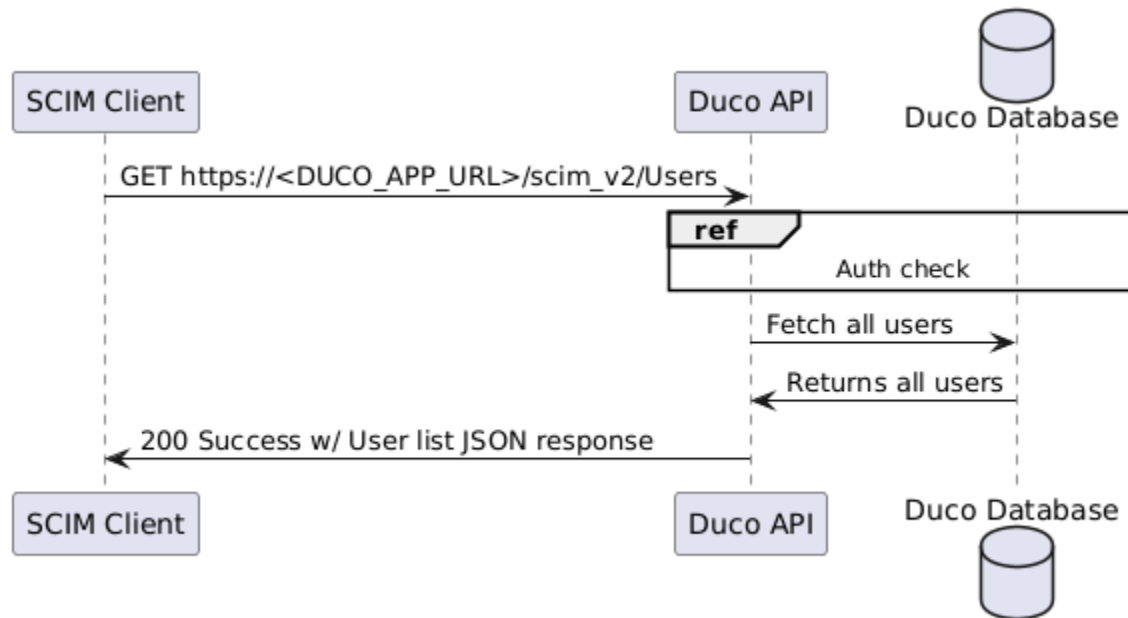
```
{
  "schemas": [
    "urn:ietf:params:scim:api:messages:2.0:Error"
  ],
  "detail": "Email has already been taken",
  "status": "409",
  "scimType": "uniqueness"
}
```

Success case JSON response:

```
{
  "userName": "jane.doe@example.com",
  "name": {
    "familyName": "Doe",
    "givenName": "Jane"
  },
  "meta": {
    "resourceType": "User",
    "created": "2024-06-27T15:57:28Z",
    "lastModified": "2024-06-27T15:57:28Z",
    "location":
    "https://<DUCO_APP_URL>/scim_v2/Users/49052add-0c37-45f2-a7c0-026acc06ce3d"
  },
  "id": "49052add-0c37-45f2-a7c0-026acc06ce3d",
  "emails": [],
  "phoneNumbers": [],
  "groups": [],
  "active": true,
  "schemas": [
    "urn:ietf:params:scim:schemas:core:2.0:User"
  ]
}
```

Fetching (Reading) Users

Fetching All Users



If there are no users are found, the response should still return an empty JSON list (e.g. '[]')

Request example:

```
GET https://<DUCO_APP_URL>/scim_v2/Users
Accept: application/scim+json
Content-Type: application/scim+json
Authorization: Bearer <BEARER_TOKEN>
```

User list JSON response:

```
{
  "schemas": [
    "urn:ietf:params:scim:api:messages:2.0:ListResponse"
  ],
  "totalResults": 20,
  "startIndex": 1,
  "itemsPerPage": 100,
  "Resources": [
    {
      "id": "1",
      "userName": "admin@example.com",
```

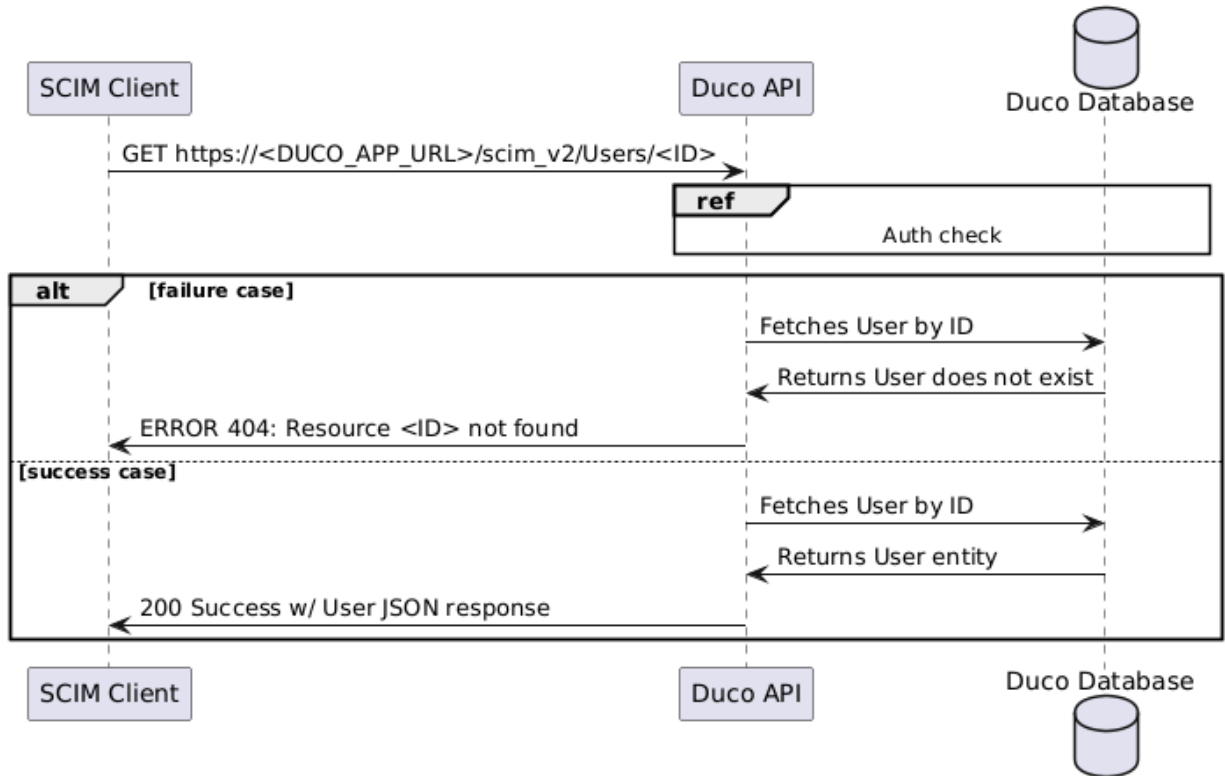
```
    "name": {
      "givenName": "Admin",
      "familyName": "Admin"
    },
    "emails": [],
    "phoneNumbers": [],
    "groups": [
      {
        "value": "3d1e291a-ae8a-4376-a6e0-65c55978b1c6",
        "display": "Administrators"
      }
    ],
    "active": true,
    "meta": {
      "resourceType": "User",
      "created": "2024-07-12T13:43:24Z",
      "lastModified": "2024-07-12T13:43:24Z",
      "location":
"http://<DUCO_APP_URL>/scim_v2/Users/3d1e291a-ae8a-4376-a6e0-65c55978b1c6"
    },
    "schemas": [
      "urn:ietf:params:scim:schemas:core:2.0:User"
    ]
  },
  {
    "id": "49052add-0c37-45f2-a7c0-026acc06ce3d",
    "userName": "jane.doe@example.com",
    "name": {
      "givenName": "Jane",
      "familyName": "Doe"
    },
    "emails": [],
    "phoneNumbers": [],
    "groups": [
      {
        "value": "961588b2-c4c9-4ec3-81dc-591f277f648e",
        "display": "Operations Managers"
      }
    ],
    "active": true,
    "meta": {
      "resourceType": "User",
      "created": "2024-07-12T13:43:24Z",
      "lastModified": "2024-07-12T13:43:24Z",
      "location":
```

```

"<DUCO_APP_URL>/scim_v2/Users/49052add-0c37-45f2-a7c0-026acc06ce3d"
  },
  ...
}

```

Fetching a Single User



Request example:

```

GET
https://<DUCO_APP_URL>/scim_v2/Users/49052add-0c37-45f2-a7c0-026acc06ce3d
Accept: application/scim+json
Content-Type: application/scim+json
Authorization: Bearer <BEARER_TOKEN>

```

Failure case JSON response:

```

{
  "schemas": [

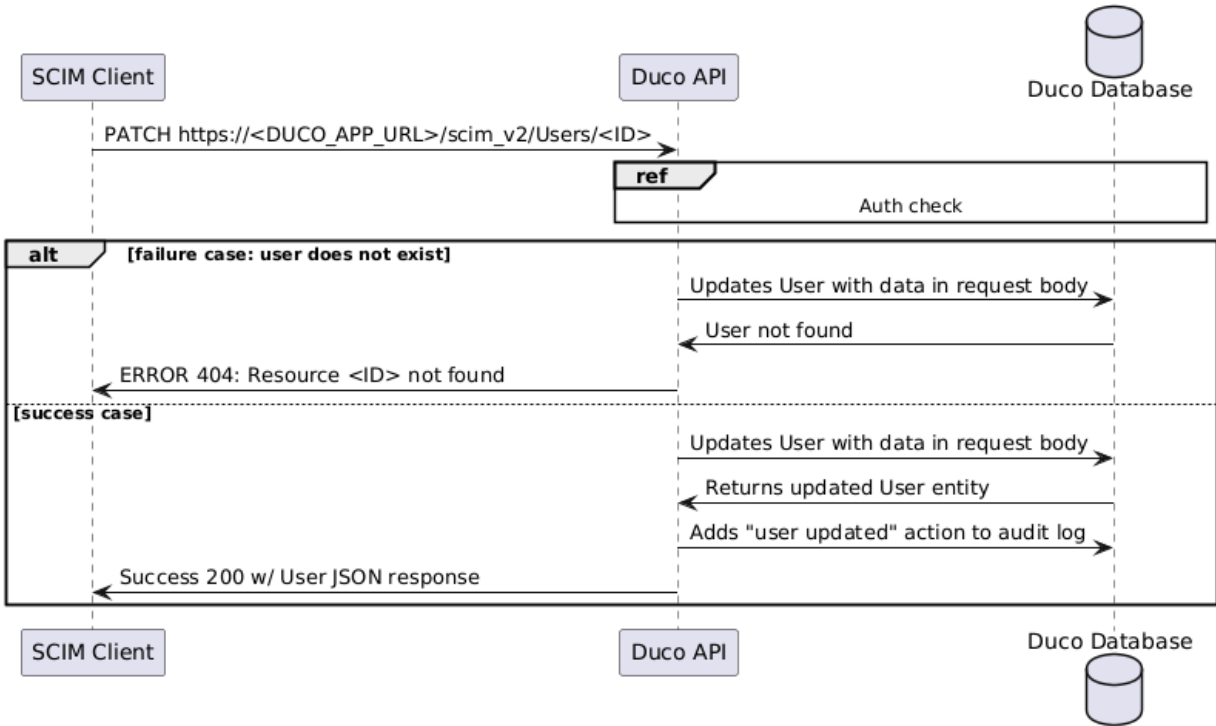
```

```
    "urn:ietf:params:scim:api:messages:2.0:Error"
  ],
  "detail": "Resource \"49052add-0c37-45f2-a7c0-026acc06ce3d\" not found",
  "status": "404"
}
```

Success case JSON response:

```
{
  "id": "49052add-0c37-45f2-a7c0-026acc06ce3d",
  "userName": "jane.doe@example.com",
  "name": {
    "givenName": "Jane",
    "familyName": "Doe"
  },
  "emails": [],
  "phoneNumbers": [],
  "groups": [],
  "active": true,
  "meta": {
    "resourceType": "User",
    "created": "2024-07-12T15:28:10Z",
    "lastModified": "2024-07-12T15:28:10Z",
    "location":
"http://<DUCO_APP_URL>/scim_v2/Users/49052add-0c37-45f2-a7c0-026acc06ce3d"
  },
  "schemas": [
    "urn:ietf:params:scim:schemas:core:2.0:User"
  ]
}
```

Updating a User's Attributes



Request Payload:

```

PATCH
https://<DUCO_APP_URL>/scim_v2/Users/49052add-0c37-45f2-a7c0-026acc06ce3d
Accept: application/scim+json
Content-Type: application/scim+json
Authorization: Bearer <BEARER_TOKEN>

{
  "schemas": ["urn:ietf:params:scim:api:messages:2.0:PatchOp"],
  "Operations": [{
    "op": "replace",
    "path": "name.familyName",
    "value": "Doe-Smith"
  }]
}
  
```

Failure case JSON response:

```

{
  "schemas": [
    "urn:ietf:params:scim:api:messages:2.0:Error"
  ]
}
  
```

```
],
"detail": "Resource \"49052add-0c37-45f2-a7c0-026acc06ce3d\" not found",
"status": "404"
}
```

Success case JSON response:

```
{
  "id": "49052add-0c37-45f2-a7c0-026acc06ce3d",
  "userName": "jane.doe@example.com",
  "name": {
    "givenName": "Jane",
    "familyName": "Doe-Smith"
  },
  "emails": [],
  "phoneNumbers": [],
  "groups": [
    {
      "value": "1",
      "display": "Duco"
    }
  ],
  "active": true,
  "meta": {
    "resourceType": "User",
    "created": "2024-07-12T14:24:39Z",
    "lastModified": "2024-07-12T19:49:22Z",
    "location":
"http://<DUCO_APP_URL>/scim_v2/Users/49052add-0c37-45f2-a7c0-026acc06ce3d"
  },
  "schemas": [
    "urn:ietf:params:scim:schemas:core:2.0:User"
  ]
}
```

Supported attributes

These are the supported user attributes that can be updated via patch operation:

Path	Example	Description
name.givenName	"Jane"	The first name of the user
name.familyName	"Doe"	The last name of the user

active	"true" or "false"	If the user account should be enabled/disabled or active/deactivated
--------	-------------------	--

In the example below, the following would need to be changed according to what is needed:

- <DUCO_APP_URL> - the base URL for Duco
- <USER_ID> - the user UUID
- <BEARER_TOKEN> - the bearer token
- <PATH> - a path to the attribute to update (see table above)
- <VALUE> - the new value of the attribute

```

PATCH https://<DUCO_APP_URL>/scim_v2/Users/<USER_ID>
Accept: application/scim+json
Content-Type: application/scim+json
Authorization: Bearer <BEARER_TOKEN>

{
  "schemas": ["urn:ietf:params:scim:api:messages:2.0:PatchOp"],
  "Operations": [{
    "op": "replace",
    "path": "<PATH>",
    "value": "<VALUE>"
  }]
}

```

This example changes the PATH and VALUE to "active" and "false" to disable/deactivate the given user.

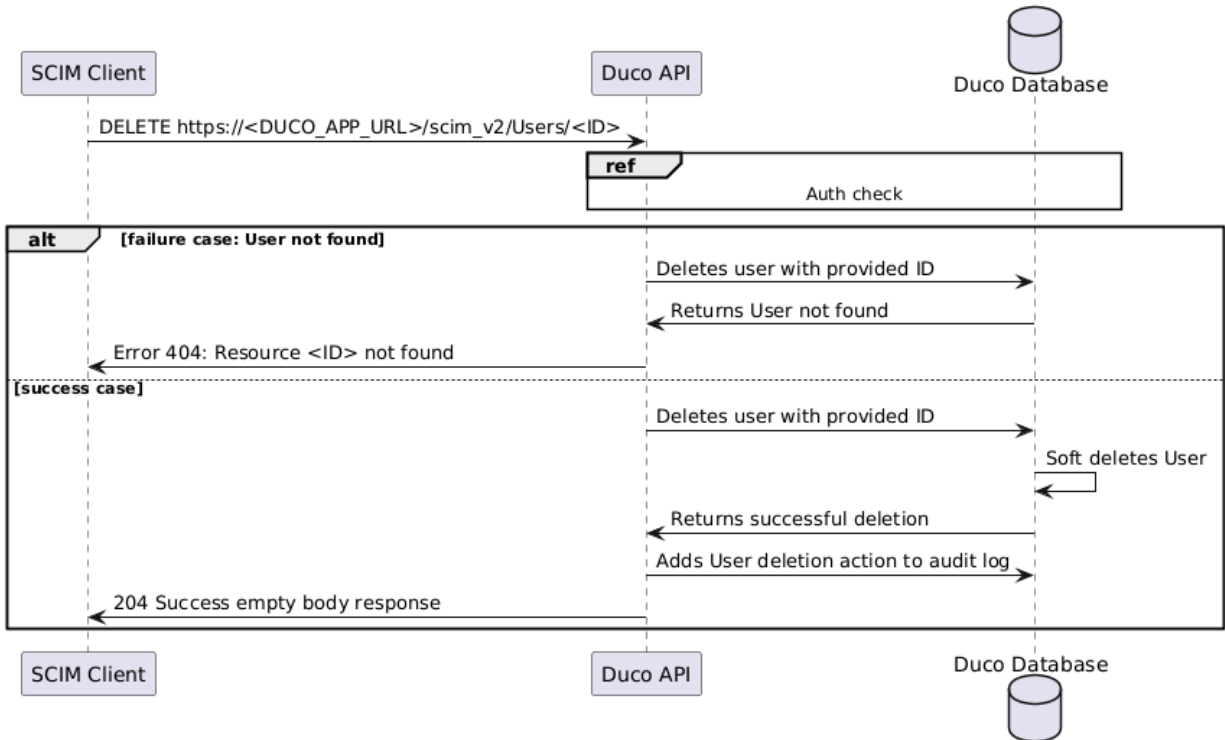
```

PATCH
https://<DUCO_APP_URL>/scim_v2/Users/49052add-0c37-45f2-a7c0-026acc06ce3d
Accept: application/scim+json
Content-Type: application/scim+json
Authorization: Bearer <BEARER_TOKEN>

{
  "schemas": ["urn:ietf:params:scim:api:messages:2.0:PatchOp"],
  "Operations": [{
    "op": "replace",
    "path": "active",
    "value": "false"
  }]
}

```

Deleting a User



“Soft delete” means setting a “deleted_at” timestamp for the user and treating it as deleted. This is done to preserve user ID referential integrity.

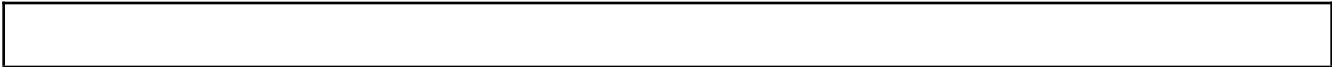
Request example:

```
DELETE
https://<DUCO_APP_URL>/scim_v2/Users/961588b2-c4c9-4ec3-81dc-591f277f648e
Accept: application/scim+json
Content-Type: application/scim+json
Authorization: Bearer <BEARER_TOKEN>
```

Failure case JSON response:

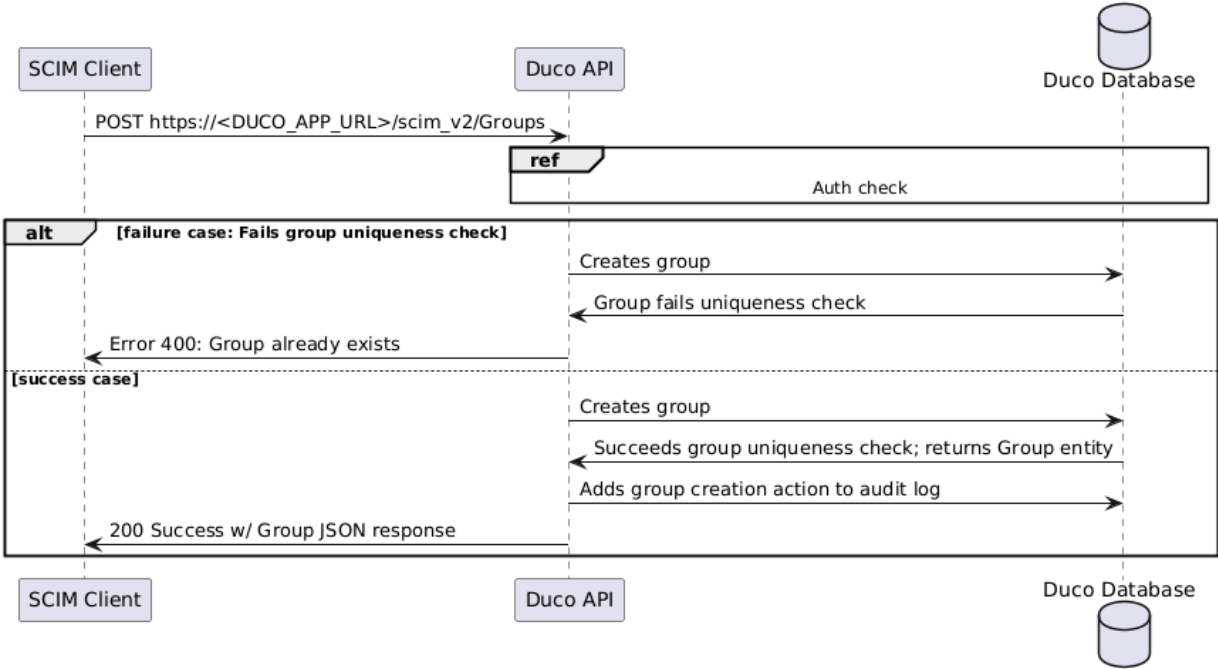
```
{
  "schemas": [
    "urn:iETF:params:scim:api:messages:2.0:Error"
  ],
  "detail": "Resource \"961588b2-c4c9-4ec3-81dc-591f277f648e\" not found",
  "status": "404"
}
```

Success case response (empty body):



Groups Endpoint

Creating a Group



Request Payload example:

```

POST https://<DUCO_APP_URL>/scim_v2/Groups
Accept: application/scim+json
Content-Type: application/scim+json
Authorization: Bearer <BEARER_TOKEN>

{
  "schemas": [
    "urn:ietf:params:scim:schemas:core:2.0:Group"
  ],
  "displayName": "Example Group"
}
  
```

```
}
```

Failure case JSON response:

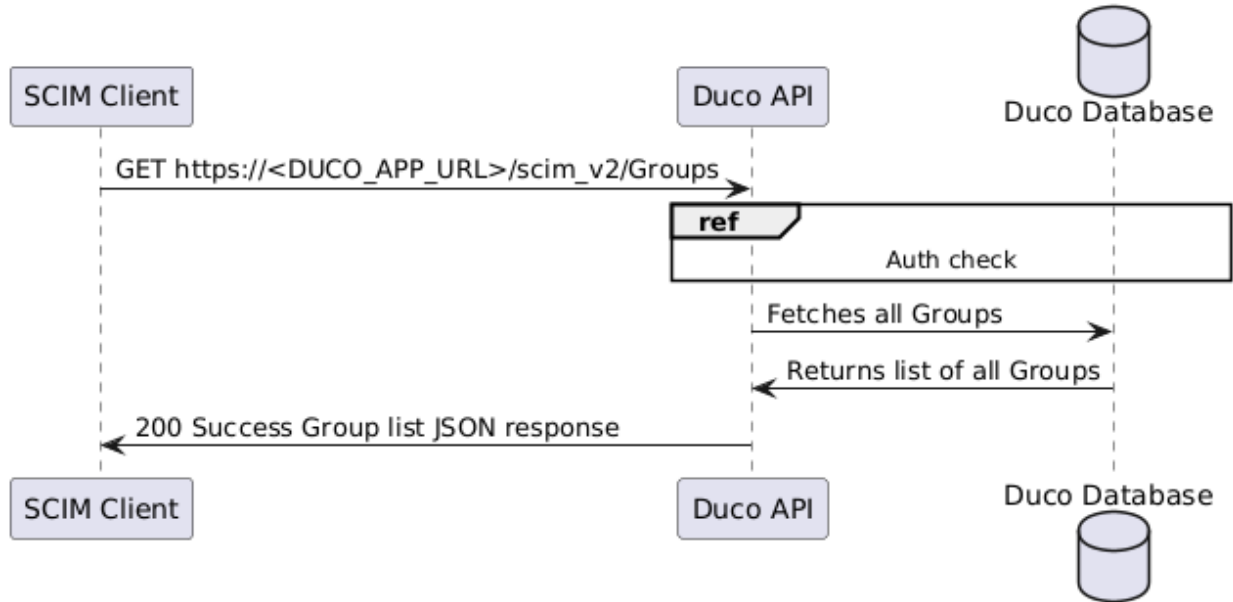
```
{
  "schemas": [
    "urn:ietf:params:scim:api:messages:2.0:Error"
  ],
  "detail": "Operation failed since record has become invalid: Name has
already been taken",
  "status": "400",
  "scimType": "invalidValue"
}
```

Success case JSON response:

```
{
  "displayName": "Example Group",
  "meta": {
    "resourceType": "Group",
    "location":
"http://<DUCO_APP_URL>/scim_v2/Groups/9b780f95-9c11-4b8f-8b6b-a834db900226"
  },
  "id": "9b780f95-9c11-4b8f-8b6b-a834db900226",
  "members": [],
  "schemas": [
    "urn:ietf:params:scim:schemas:core:2.0:Group"
  ]
}
```

Fetching a Group

All Groups



If there are no groups are found, the response should still return an empty JSON list (e.g. `[]`)

Request example:

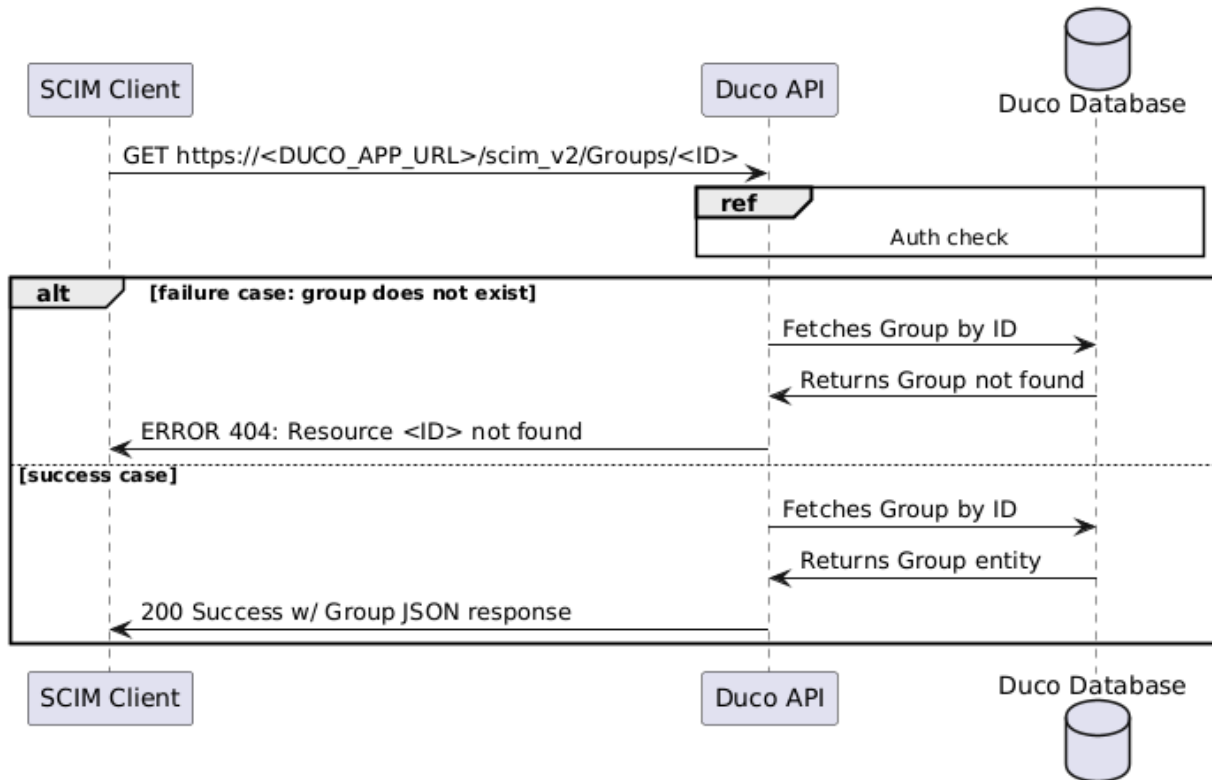
```
GET https://<DUCO_APP_URL>/scim_v2/Groups
Accept: application/scim+json
Content-Type: application/scim+json
Authorization: Bearer <BEARER_TOKEN>
```

Success case JSON response:

```
{
  "schemas": [
    "urn:ietf:params:scim:api:messages:2.0:ListResponse"
  ],
  "totalResults": 4,
  "startIndex": 1,
  "itemsPerPage": 100,
  "Resources": [
    {
      "displayName": "Administrators",
      "meta": {
```

```
    "resourceType": "Group",
    "location":
"http://<DUCO_APP_URL>/scim_v2/Groups/45c535f1-bd49-4ea9-9cff-9813d9278971
"
    },
    "id": "45c535f1-bd49-4ea9-9cff-9813d9278971",
    ...
  },
  {
    "displayName": "Operations Managers",
    "meta": {
      "resourceType": "Group",
      "location":
"http://<DUCO_APP_URL>/scim_v2/Groups/d654c6cf-f184-4de1-a2f7-af68345821c1
"
    },
    "id": "d654c6cf-f184-4de1-a2f7-af68345821c1",
    "members": [
      {
        "value": "28c6d1ae-034f-4fed-ba4b-3df3bc0514ea",
        "display": "User 1"
      },
      {
        "value": "cdb2955c-eb03-4b32-8c7e-d3946d70b672",
        "display": "User 2"
      },
      {
        "value": "0897b759-2bd6-4e63-8135-1d05313d2eb1",
        "display": "User 3"
      },
      ...
    ]
  }
}
```

An Individual Group



Request example:

```
GET
https://<DUCO_APP_URL>/scim_v2/Groups/98d97f4b-b3b3-4bb6-b533-dc48c744c2334
Accept: application/scim+json
Content-Type: application/scim+json
Authorization: Bearer <BEARER_TOKEN>
```

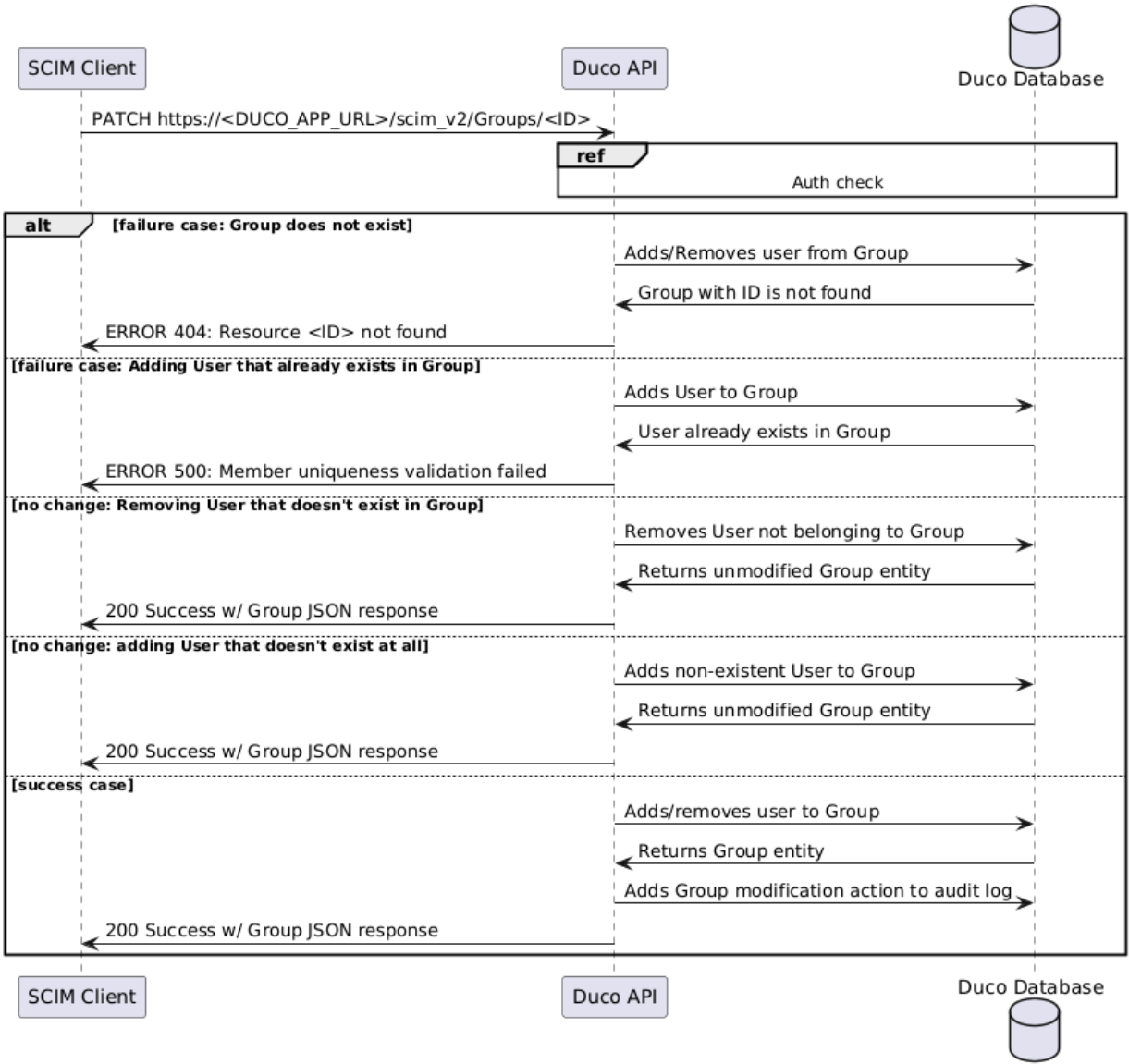
Failure case JSON response:

```
{
  "schemas": [
    "urn:ietf:params:scim:api:messages:2.0:Error"
  ],
  "detail": "Resource \"98d97f4b-b3b3-4bb6-b533-dc48c744c233\" not found",
  "status": "404"
}
```

Success case JSON response:

```
{
  "displayName": "Product Engineers",
  "meta": {
    "resourceType": "Group",
    "location":
"https://<DUCO_APP_URL>/scim_v2/Groups/98d97f4b-b3b3-4bb6-b533-dc48c744c233
"
  },
  "id": "98d97f4b-b3b3-4bb6-b533-dc48c744c233",
  "members": [],
  "schemas": [
    "urn:ietf:params:scim:schemas:core:2.0:Group"
  ]
}
```

Adding or Removing a User from a Group



Request Payload (Adding User) example:

```

PATCH
https://<DUCO_APP_URL>/scim_v2/Groups/98d97f4b-b3b3-4bb6-b533-dc48c744c2334
Accept: application/scim+json
Content-Type: application/scim+json
Authorization: Bearer <BEARER_TOKEN>

{
  "schemas": ["urn:ietf:params:scim:api:messages:2.0:PatchOp"],
  "operations": [{
  
```

```
    "op": "add",
    "value":
      {
        "members" : [{
          "value": "d0b5c202-2905-402d-bbc2-d8a6104e39af"
        }]
      }
  ]
}
```

Request Payload (Removing User) example:

```
PATCH
https://<DUCO_APP_URL>/scim_v2/Groups/98d97f4b-b3b3-4bb6-b533-dc48c744c2334
Accept: application/scim+json
Content-Type: application/scim+json
Authorization: Bearer <BEARER_TOKEN>

{
  "schemas": ["urn:ietf:params:scim:api:messages:2.0:PatchOp"],
  "Operations": [{
    {
      "op": "remove",
      "path": "members",
      "value": { "value": "28c6d1ae-034f-4fed-ba4b-3df3bc0514ea" }
    }
  ]
}
```

Failure case: "Group does not exist" JSON response:

```
{
  "schemas": [
    "urn:ietf:params:scim:api:messages:2.0:Error"
  ],
  "detail": "Resource \"98d97f4b-b3b3-4bb6-b533-dc48c744c2334\" not found",
  "status": "404"
}
```

Failure case: "Adding a User who already belongs to the Group) JSON response:

```
{
  "schemas": [
    "urn:ietf:params:scim:api:messages:2.0:Error"
  ],
  "detail": "Validation failed: User has already been assigned to this
group",
  "status": "500"
}
```

No change case: "Removal of a user that does not exist in the Group" JSON response:

```
{
  "id": "98d97f4b-b3b3-4bb6-b533-dc48c744c233",
  "displayName": "Product Engineers",
  "members": [
    {
      "value": "28c6d1ae-034f-4fed-ba4b-3df3bc0514ea",
      "display": "Jane Doe"
    }
  ],
  "meta": {
    "resourceType": "Group",
    "location":
"https://<DUCO_APP_URL>/scim_v2/Groups/98d97f4b-b3b3-4bb6-b533-dc48c744c23
3"
  },
  "schemas": [
    "urn:ietf:params:scim:schemas:core:2.0:Group"
  ]
}
```

No change case "Group is unmodified from adding non-existent User" JSON response:

```
{
  "id": "98d97f4b-b3b3-4bb6-b533-dc48c744c233",
  "displayName": "Product Engineers",
  "members": [
    {
      "value": "28c6d1ae-034f-4fed-ba4b-3df3bc0514ea",
      "display": "Jane Doe"
    }
  ],
}
```

```
"meta": {
  "resourceType": "Group",
  "location":
"https://<DUCO_APP_URL>/scim_v2/Groups/98d97f4b-b3b3-4bb6-b533-dc48c744c23
3"
},
"schemas": [
  "urn:ietf:params:scim:schemas:core:2.0:Group"
]
}
```

Success case (adding user) JSON response:

```
{
  "id": "98d97f4b-b3b3-4bb6-b533-dc48c744c233",
  "displayName": "Product Engineers",
  "members": [
    {
      "value": "28c6d1ae-034f-4fed-ba4b-3df3bc0514ea",
      "display": "Jane Doe"
    },
    {
      "value": "d0b5c202-2905-402d-bbc2-d8a6104e39af",
      "display": "John Smith"
    }
  ],
  "meta": {
    "resourceType": "Group",
    "location":
"https://<DUCO_APP_URL>/scim_v2/Groups/98d97f4b-b3b3-4bb6-b533-dc48c744c23
3"
  },
  "schemas": [
    "urn:ietf:params:scim:schemas:core:2.0:Group"
  ]
}
```

Success case (removing user) JSON response:

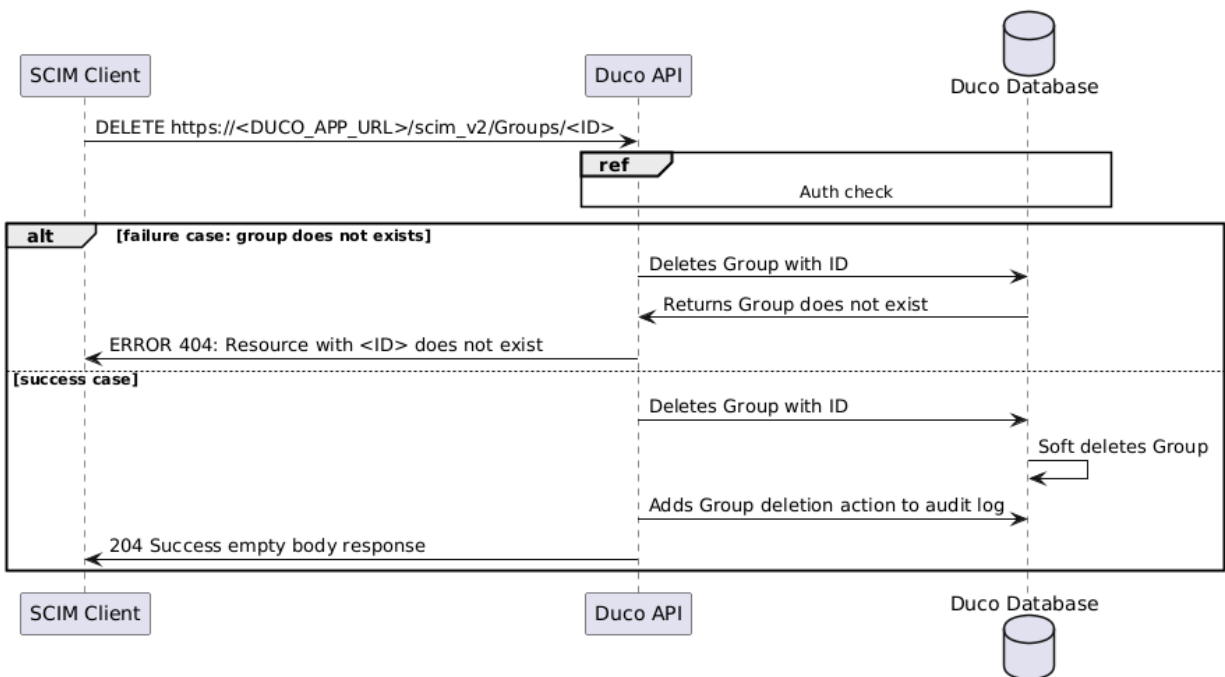
```
{
  "id": "98d97f4b-b3b3-4bb6-b533-dc48c744c233",
  "displayName": "Product Engineers",
  "members": [],
}
```

```

"meta": {
  "resourceType": "Group",
  "location":
"https://<DUCO_APP_URL>/scim_v2/Groups/98d97f4b-b3b3-4bb6-b533-dc48c744c23
3"
},
"schemas": [
  "urn:ietf:params:scim:schemas:core:2.0:Group"
]
}

```

Deleting a Group



“Soft delete” means setting a “deleted_at” timestamp for the group and treating it as deleted. This is done to preserve group ID referential integrity.

Request example:

```

DELETE
https://<DUCO_APP_URL>/scim_v2/Groups/98d97f4b-b3b3-4bb6-b533-dc48c744c233
4
Accept: application/scim+json
Content-Type: application/scim+json
Authorization: Bearer <BEARER_TOKEN>

```

Failure case JSON response:

```
{
  "schemas": [
    "urn:ietf:params:scim:api:messages:2.0:Error"
  ],
  "detail": "Resource \"98d97f4b-b3b3-4bb6-b533-dc48c744c2334\" not
found",
  "status": "404"
}
```

Success case (empty body) JSON response: