SCIM Operations

HTTP Method

GET	Retrieves one or more complete or partial resources.
POST	Depending on the endpoint, creates new resources, creates a search request, or MAY be used to bulk-modify resources.
PUT	Modifies a resource by replacing existing attributes with a specified set of replacement attributes (replace). PUT MUST NOT be used to create new resources.
PATCH	Modifies a resource with a set of client-specified changes (partial update).
DELETE	Deletes a resource.

Get

A HTTP Get request is used to fetch a resource or a set of resources.

Read

```
http://<your-domain>/soffid/webservice/scim2/v1/{Resource}/{id}
```

- id: is the identifier of a specific resource

Search

```
http://<your-domain>/soffid/webservice/scim2/v1/{Resource}/?filter={attribute}{op}{value}&sortBy={attributeName}&sortOrder={ascending|descending}&attributes={attributes}
```

- **filter**: allows you to add filter to query.
  - attribute
  - op: SCIM has support for the filter operations equals, contains, starts with, and more.

    - value
- **sortBy**: the attribute used to sort the response.
- **sortOrder**: order to sort, ascending or descending. Ascending is the default order.

In addition, you can request specific attributes of the resource, or exclude attributes (separated by commas if there is more than one).

- **attributes**={attributes}
- **excludedAttributes**={attributes}

Example:

```
http://<your-domain>/soffid/webservice/scim2/v1/User?filter=lastName co ada and active eq true
&sortOrder=descending&sortBy=userName&attributes=userName,lastName&filter=userName co admin
```

Sorting

Parameter	Description
sortBy	Specifies the attribute whose value will be used to order ther returned responses.
sortOrder	Allowed values are "ascending" and "descending". If sortBy is provided, and sortOrder is nos provided, sortOrder will be "ascending" by default.

Sorting example:

```
http://<your-server>/soffid/webservice/scim2/v1/User?sortBy=lastName&sortOrder=descending
```

Pagination

Parameter	Description
startIndex	Index of the first query result. Default 1
count	Maximun numer of query results per page

Pagination example:

```
http://<your-server>/soffid/webservice/scim2/v1/User?startIndex=1&count=10
```

```
{
    "totalResults":100,
    "itemsPerPage":10,
    "startIndex":1,
    "schemas":["urn:ietf:params:scim:api:messages:2.0:ListResponse"],
    "Resources":[{
      ...
    }]
}
```

Post

A HTTP Post request is used to create a new resource

```
http://<your-domain>/soffid/webservice/scim2/v1/{Resource}
```

```
Content-Type: application/json
```

You must send the JSON with the attributes of the resource you want to create.

```
{
  "schemas":[{schema}],
  "attribute1":"value1",
  "attribute2":"value2",
  "attribute3":{
    "subattribute1":"valueX",
    "subattribute1":"valueX",
  },
  .......
}
```

- schema: is the schema url of the resource you are creating.
- attributes: name of the resource attributes.
- values: values for each attribute.

Put

A HTTP Put request is used to update resources. This operation replace all values of the resource

```
http://<your-domain>/soffid/webservice/scim2/v1/{Resource}/{id}
```

```
Content-Type: application/json
```

You must send the JSON with the attributes of the resource you want to update, which includes the ID.

```
{
  "schemas":[{schema}],
  "id": "idValue",
  "attribute1":"value1",
  "attribute2":"value2",
  "attribute3":{
    "subattribute1":"valueX",
    "subattribute1":"valueX",
  },
  .......
}
```

- schema: is the schema url of the resource you are creating.
- id: identifier of the resource
- attributes: name of the resource attributes.
- values: values for each attribute.

Patch

A HTTP Patch request is used to update partial resources

```
http://<your-domain>/soffid/webservice/scim2/v1/{Resource}/{id}
```

```
{
   "Operations": [
      {
         "op": "operation",
         "path": "attribute",
         "value": "value"
      },
      ............
   ]
}
```

- op: available operations to realize:
  - add: allows you to add a new value to an attribute.
  - remove: allows you to delete the value of an attribute.
  - replace: allows you to replace (update) the value of an attribute.

> More information about the operations on https://www.rfc-editor.org/rfc/rfc6902

- path: to indicate the attribute on which the operation is to be performed.
- value: the new value for the attribute.

Delete

A HTTP Delete request is used to delete a resource.

```
http://<your-domain>/soffid/webservice/scim2/v1/{Resource}/{id}
```

- id: is the identifier of a specific resource

Request

In the PUT and PATCH methods, a JSON stream with the data model is required (please see this format in the following link: Resource data model).

Response

The response format will be represented as a SCIM JSON response, but all the keys in the response will depend on the method requested and the result of the operation.

HTTP Status

The most commons responses

Successful Responses

200	OK

201	Created
204	No Content

Error Response

400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

For instance, when you search by id but no resource is found, only a 404 HTTP code is included in the response (the body is empty, no JSON is provided).

User cases:

- Search by id but no resource is found (404 code).
- Update all, the id is not found (404 code).
- Update partial, the id is not found (404 code).
- Delete, the id is not found (404 code).
- A "/<resource>" (in the URL) not exist (404 code).
- Other errors (404 or 500 code).

> More detail about SCIM JSON error

SCIM JSON Response

SCIM JSON list

For instance, when a list of resources is requested, this is the JSON output format:

Note, to simplify the JSON output every resource has been replaced by {...}

```
{
  "totalResults": 3,
  "schemas": [
    "urn:ietf:params:scim:api:messages:2.0:ListResponse"
  ],
  "resources": [
    {...},
    {...},
    {...}
  ]
}
```

This is the description of this type of response:

Attribute	Description
totalResults	Number of the resources returned in the response
schemas	Defined by SCIM protocl. Always: "urn:ietf:params:scim:api:messages:2.0:ListResponse"
resources	List of resources returned

User cases:

- A list all operation (200 code).
- A search by filter operation (200 code).
- The delete operation (204 code).

SCIM JSON resource

For instance, when a resource by id is requested, this is the JSON format:

Note, to simplify the JSON output every resource has been replaced by {...}

```
{
  "id": 11345
  "organizational": true,
  ...
}
```

In this case, the JSON stream of the resource is included directly in the response.

User cases:

- Search by id operations (200 code).
- Successful create operations (201 code).
- Successful complete update operations (200 code).
- Successful partial update operations (200 code).

SCIM JSON error

For instance, if an attempt to delete a resource is made, but this resource is not found the following JSON response will be obtained:

```
{
  "schemas": [
    "urn:ietf:params:scim:api:messages:2.0:Error"
  ],
  "detail": "User 1234 not found",
  "status": "404"
}
```

This is the description of this type of response:

Attribute	Description
schemas	Defined by SCIM protocl. Always: "urn:ietf:params:scim:api:messages:2.0:Error"
detail	Returns the description on the validation, problem, error, etc
status	Is the HTTP status, that is the same that the HTTP code of the HTTP response

User cases:

- When you try to delete a resource but it's not found (404 code).
- When you try to delete a group, the solution is to disable it by PATCH (500 code).
- Generic errors (500 code).

Revision #33
Created 3 May 2021 09:12:40
Updated 3 February 2026 15:44:18 by Sion Vives