

# SCIM User type examples

## Operations

This page shows the operations that can be performed for the user type object

### List all

#### Request

```
GET http://<your-domain>/soffid/webservice/scim2/v1/UserType
```

#### Response 200 OK

```
{
  "schemas": [
    "urn:ietf:params:scim:api:messages:2.0:ListResponse"
  ],
  "totalResults": 3,
  "startIndex": 1,
  "Resources": [
    {
      "meta": {
        "location": "http://soffid.pat.lab:8080/soffid/webservice/scim2/v1/UserType/29",
        "resourceType": "UserType"
      },
      "unmanaged": false,
      "schemas": [
        "urn:soffid:com:soffid:iam:api:UserType"
      ],
      "name": "S",
      "description": "SSO account (USE IT)",
      "id": 29
    },
  ],
}
```

```

{
  "meta": {
    "location": "http://soffid.pat.lab:8080/soffid/webservice/scim2/v1/UserType/31",
    "resourceType": "UserType"
  },
  "unmanaged": false,
  "schemas": [
    "urn:soffid:com.soffid.iam.api.UserType"
  ],
  "name": "E",
  "description": "External user",
  "id": 31
},
{
  "meta": {
    "location": "http://soffid.pat.lab:8080/soffid/webservice/scim2/v1/UserType/33",
    "resourceType": "UserType"
  },
  "unmanaged": false,
  "schemas": [
    "urn:soffid:com.soffid.iam.api.UserType"
  ],
  "name": "I",
  "description": "Internal user",
  "id": 33
}
]
}

```

## List by filter

List all user types with a filter expression.

It is allowed to use pagination and sort the information, for more information visit the [Sorting and Pagination](#) information.

## Request

```
GET http://<your-domain>/soffid/webservice/scim2/v1/UserType?filter=description co sso
```

## Response 200 OK

```
{
  "schemas": [
    "urn:ietf:params:scim:api:messages:2.0:ListResponse"
  ],
  "totalResults": 1,
  "startIndex": 1,
  "Resources": [
    {
      "meta": {
        "location": "http://soffid.pat.lab:8080/soffid/webservice/scim2/v1/UserType/29",
        "resourceType": "UserType"
      },
      "unmanaged": false,
      "schemas": [
        "urn:soffid:com:soffid:iam:api:UserType"
      ],
      "name": "S",
      "description": "SSO account (USE IT)",
      "id": 29
    }
  ]
}
```

## Query by id

Query a user type by its id (primary key).

### Request

```
GET http://<your-domain>/soffid/webservice/scim2/v1/UserType/33
```

## Response 200 OK

```
{
  "meta": {
```

```
    "location": "http://soffid.pat.lab:8080/soffid/webservice/scim2/v1/UserType/33",
    "resourceType": "UserType"
  },
  "unmanaged": false,
  "schemas": [
    "urn:soffid:com.soffid.iam.api.UserType"
  ],
  "name": "I",
  "description": "Internal user",
  "id": 33
}
```

# Create

## Request

POST http://<your-domain>/soffid/webservice/scim2/v1/UserType

### JSON

```
{
  "schemas": [
    "urn:soffid:com.soffid.iam.api.UserType"
  ],
  "name": "O",
  "description": "Other user",
  "unmanaged": false
}
```

## Response 201 Created

```
{
  "meta": {
    "location": "http://soffid.pat.lab:8080/soffid/webservice/scim2/v1/UserType/1976718",
    "resourceType": "UserType"
  },
  "unmanaged": false,
  "schemas": [
    "urn:soffid:com.soffid.iam.api.UserType"
  ]
}
```

```
],
"name": "O",
"description": "Other user",
"id": 1976718
}
```

## Update partial

Only attributes with changes will be updated, the other will maintain the same value.

### Request

```
PATCH http://<your-domain>/soffid/webservice/scim2/v1/UserType/1976718
```

### JSON

```
{
  "Operations": [
    {
      "op": "replace",
      "path": "name",
      "value": "OT"
    }
  ]
}
```

### Response 200 OK

```
{
  "meta": {
    "location": "http://soffid.pat.lab:8080/soffid/webservice/scim2/v1/UserType/1976718",
    "resourceType": "UserType"
  },
  "unmanaged": false,
  "schemas": [
    "urn:soffid:com.soffid.iam.api.UserType"
  ],
  "name": "OT",
  "description": "Other user",
}
```

```
"id": 1976718
}
```

# Update all

This operation replaces all values in the roole.

- Note that the attribute id is required to confirm that the resource "...UserType/<id>" is the same that the JSON user type.
- Note that all the attributes not included in the request will be cleared in the user type and their data will be lost.
- Note that not all the attributes are updatable, for example, tag meta, avoid these tags.

For more information see [Resource data model page](#)

## Request

```
PUT http://<your-domain>/soffid/webservice/scim2/v1/UserType/1976718
```

## JSON

```
{
  "schemas": [
    "urn:soffid:com.soffid.iam.api.OUType"
  ],
  "id": 1975535,
  "name": "OY"
}
```

## Response 200 OK

```
{
  "meta": {
    "location": "http://soffid.pat.lab:8080/soffid/webservice/scim2/v1/UserType/1976718",
    "resourceType": "UserType"
  },
  "unmanaged": false,
  "schemas": [
    "urn:soffid:com.soffid.iam.api.UserType"
  ],
  "name": "OY",
}
```

```
"id": 1976718  
}
```

# Delete

Please note after this delete, the user type has to be created again to use it in the next examples.

## Request

```
DELETE http://<your-omain>/soffid/webservice/scim2/v1/UserType/1976718
```

## Response 204 No Content

```
204 No Content
```

# Error response

For more information about error response visit  
<https://bookstack.soffid.com/link/116#bkmrk-error-response>

Revision #11

Created 11 May 2021 09:45:58 by pgarcia@soffid.com

Updated 21 June 2023 09:33:50 by pgarcia@soffid.com