

## Examples

### Obtaining access token from Puzzel Id

This example shows how to obtain an access token from Puzzel Id to use with the SCIM API using OAuth2.

#### Request

```
POST https://{platform-area-name}.puzzel.com/id/connect/token
2Content-Type: application/x-www-form-urlencoded
3Accept: application/json
4Host:
5
6grant_type=client_credentials&
7scope=iam-provisioning.contribute&
8client_id=<client-id>&
9client_secret=<secret>
```

#### Note

*Notice that line-breaks have been inserted for readability in the request above.*

#### Response

```
{
2 "access_token": "<jwt>",
3 "expires_in": 86400,
4 "token_type": "Bearer",
5 "scope": "iam-provisioning.contribute"
6}
```

The JWT in the

access\_token

part of the response should be used when accessing the SCIM API.

### Query users

SCIM offers rich filtering capabilities to query user data.

#### Get all users

```
GET https://<base-address>/scim/Users
```

The response includes data for all users for the specified customer.

#### Get specific user

```
GET https://<base-address>/scim/Users/{user-id}
```

Returns data for a single user, where

{user-id}

is the unique identifier for the user.

## Filter users by user name

The filter query parameter can be used to filter resources.

```
GET https://<base-address>/scim/Users/?filter=userName eq 'alice@example.com'
```

The request above returns data for the user with user name

alice@example.com

### Note

SCIM offers a rich syntax for filtering, see <https://www.rfc-editor.org/rfc/rfc7644#section-3.4.2.2>. However, be aware that Puzzel's SCIM implementation only supports a subset of the filter expressions and operators.

Currently eq is the only supported comparison operator.

## Get specific attributes for users

Sometimes only a subset of the attributes for a resource is needed. To avoid unnecessary data transfer and simplify the response, the attributes query parameter can be used to restrict which attributes are included. Similarly, the excludeAttributes query parameter can be used to specifically exclude a set of attributes.

```
GET https://<base-address>/scim/Users/?attributes=id,externalId&filter=userName eq 'alice@example.com'
```

Returns only the id and externalID attributes for the user with user name alice@example.com.

## Create users

The following gives an example of creating a user.

```
POST https://<base-address>/scim/Users
2Content-Type: application/json
3
4{
5  "schemas": [
6    "urn:ietf:params:scim:schemas:core:2.0:User",
7    "urn:ietf:params:scim:schemas:extension:puzzel:2.0:User"
8  ],
9  "userName": "bea.oproblem@example.invalid",
10 "active": true,
11 "displayName": "Bea O'Problem",
12 "externalId": "e5c2ce33-b081-4a73-9dbc-2a669e7948ad",
13 "name": {
14   "formatted": "Bea O'Problem",
15   "familyName": "O'Problem",
```

```

16   "givenName": "Bea"
17   },
18   "emails": [
19     {
20       "Primary": true,
21       "type": "work",
22       "value": "bea.oproblem@very.invalid"
23     }
24   ],
25   "preferredLanguage": "no-nb",
26   "urn:ietf:params:scim:schemas:extension:puzzel:2.0:User": {
27     "customerid": "<customer-id>",
28     "contactCentreSolutions": [
29       {
30         "value": "<solution-id>",
31         "type": "main",
32         "primary": "true",
33         "customerid": "<customer-id>",
34         "platform": "PCC",
35         "userName": "bea",
36         "userGroupName": "Agents",
37         "createUserGroupIfNotExists": true
38       }
39     ]
40   }
41}

```

The response includes the same data as in the request with the addition of the unique user identifier attribute and meta attributes.

## Update users by patching or replacing

### Replacement

The following request shows how to replace a user. Notice that compared with the request to create a user the unique user identifier is required in this request.

```

POST https://<base-address>/scim/Users/{user-id}
Content-Type: application/json
3
4{
5  "schemas": [
6    "urn:ietf:params:scim:schemas:core:2.0:User",
7    "urn:ietf:params:scim:schemas:extension:puzzel:2.0:User"
8  ],
9  "id": "<user-id>",
10 "userName": "bea.oproblem@example.com",
11 "active": true,
12 "displayName": "Bea O'Problem",
13 "externalId": "e5c2ce33-b081-4a73-9dbc-2a669e7948ad",
14 "name": {
15   "formatted": "Bea O'Problem",
16   "familyName": "O'Problem",
17   "givenName": "Bea"
18 },
19 "emails": [
20   {
21     "Primary": true,
22     "type": "work",
23     "value": "bea.oproblem@example.com"
24   }
25 ],
26 "preferredLanguage": "no-nb",
27 "urn:ietf:params:scim:schemas:extension:puzzel:2.0:User": {
28   "customerid": "<customer-id>",
29   "contactCentreSolutions": [
30     {

```

```

31     "value": "<solution-id>",
32     "type": "main",
33     "primary": "true",
34     "customerId": "<customer-id>",
35     "platform": "PCC",
36     "userName": "bea",
37     "userGroupName": "Agents",
38     "createUserGroupIfNotExists": true
39   }
40 ]
41 }
42}

```

## Patching - change primary solution

Below is an example to change the primary contact centre solution for a user to the solution with type "main"

```

PATCH https://<base-address>/scim/Users/{user-id}
2Content-Type: application/json
3
4{
5  "schemas": [
6    "urn:ietf:params:scim:api:messages:2.0:PatchOp",
7    "urn:ietf:params:scim:schemas:extension:puzzel:2.0:User"
8  ],
9  "Operations": [
10   {
11     "op": "replace",
12     "path": "urn:ietf:params:scim:schemas:extension:puzzel:2.0:User.contactCentreSolutions[platform eq 'PCC' and type eq 'main'].primary",
13     "value": true
14   }
15 ]
16}

```

## Patching - change username

The following request shows how to change the user name for a user and change the customer id to 13000

```

PATCH https://<base-address>/scim/Users/{user-id}
2Content-Type: application/json
3
4{
5  "schemas": [
6    "urn:ietf:params:scim:api:messages:2.0:PatchOp"
7  ],
8  "Operations": [
9    {
10     "op": "replace",
11     "path": "userName",
12     "value": "bea@oproblem.invalid"
13   },
14   {
15     "op": "replace",
16     "path": "title",
17     "value": "Director"
18   }
19 ]
20}

```

## Delete users

Users can be deleted by using the DELETE method as shown in the example below

DELETE https://<base-address>/scim/Users/{user-id}

Returns only the id and ext

#### Note

Notice that users are hard-deleted.