

# User

## Session Token

### HTTP REQUEST: GET

Lists the user that generated the API token, and the time when this token was generated

```
GET /api/1.0/sessiontoken
```

#### RESPONSES

Code	Description
200	Success.

200 Reponse Body

```
{
  "status": {
    "i18n_message": "response.ok",
    "message": "OK"
  },
  "response": {
    "user_id": "admin1234",
    "generated_at": "2018-06-22T11:42:20.996111502-07:00"
  }
}
```

## Create And List

### HTTP REQUEST: GET

Lists all users for the Organization

```
GET /api/1.0/org/{orgId}/users
```

#### URI PARAMETERS

Name	Type	Description
------	------	-------------

orgId

string

The Organization ID (UUID) for which you are querying.

**RESPONSES**

Code	Description
200	Success.
401	This error will occur if calling user does not have access right for the API.
404	This error will occur if the org identified by orgId does not exist.
500	A server error occurred. Check the logs for more details.

200 Reponse Body

```

{
  "response": [
    {
      "user_id": "568a4ab9-f7a4-4ac2-5162-9a496453d161",
      "auth_username": "abc",
      "name": "abc",
      "super_user": true,
      "api_super_user": true,
      "email": "abc@atscale.com"
    },
    {
      "user_id": "b609f4e2-10fa-4dde-425f-114ae7e182e5",
      "auth_username": "xyz",
      "name": "xyz",
      "super_user": false,
      "api_super_user": false,
      "email": "xyz@atscale.com"
    }
  ],
  "status": {
    "i18n_message": "response.ok",
    "message": "OK"
  }
}

```

## HTTP REQUEST: POST

Create a new user

**POST** /api/1.0/org/{orgId}/users

### URI PARAMETERS

Name	Type	Description
orgId	string	The Organization ID (UUID) for which you are querying.

### RESPONSES

Code	Description
------	-------------

200	Success.
401	This error will occur if calling user does not have access right for the API.
404	This error will occur if the org identified by orgId does not exist.
500	A server error occurred. Check the logs for more details.

#### Request Body

```
{
  "username": "abcid",
  "confirm_password": "abc",
  "password": "abc",
  "email": "abc@atscale.com",
  "name": "abcname",
  "roles": [
    "designcenter_user"
  ],
  "picture": "BINHEX ENCODED GIF/PNG/JPG"
}
```

#### 200 Reponse Body

```
{
  "status": {
    "i18n_message": "response.ok",
    "message": "OK"
  },
  "response": "User abcname successfully created"
}
```

## Read, Update & Delete

### HTTP REQUEST: GET

Get user details for the user

```
GET /api/1.0/org/{orgId}/users/{userId}
```

#### URI PARAMETERS

Name	Type	Description
orgId	string	The Organization ID (UUID) for which you are querying.
userId	string	The User ID of the user for which you are querying.

## RESPONSES

Code	Description
200	Success.
401	This error will occur if calling user does not have access right for the API.
404	This error will occur if the org identified by orgId does not exist, or the user identified by userId does not exist.
500	A server error occurred. Check the logs for more details.

### 200 Reponse Body

```
{
  "status": {
    "i18n_message": "response.ok",
    "message": "OK"
  },
  "response": {
    "user": {
      "user_id": "83cc05ae-7ebd-4c69-5f37-b39e718a7d80",
      "name": "testuser_dc",
      "email": "testuser@atscale.com",
      "auth_username": "testuser_dc",
      "super_user": false,
      "api_super_user": false,
      "roles": null
    },
    "organization": {
      "id": "default",
      "name": "default"
    }
  }
}
```

## HTTP REQUEST: POST

Edit the user details for the user

```
POST /api/1.0/org/{orgId}/users/{userId}
```

### URI PARAMETERS

Name	Type	Description
orgId	string	The Organization ID (UUID) for which you are querying.
userId	string	The User ID of the user for which you are querying.

### RESPONSES

Code	Description
------	-------------

200

Success.

## Request Body

```
{
  "email": "abc@atscale.com",
  "name": "abc user",
  "roles": [
    "designcenter_user"
  ],
  "picture": "BINHEX ENCODED GIF/PNG/JPG",
  "super_user": true,
  "api_super_user": false
}
```

## 200 Reponse Body

```
{
  "status": {
    "i18n_message": "response.ok",
    "message": "OK"
  },
  "response": "User abc user successfully updated"
}
```

## HTTP REQUEST: DELETE

Deletes the user

```
DELETE/api/1.0/org/{orgId}/users/{userId}
```

### URI PARAMETERS

Name	Type	Description
orgId	string	The Organization ID (UUID) for which you are querying.
userId	string	The User ID of the user for which you are querying.

### RESPONSES

Code	Description
------	-------------

200	Success.
401	This error will occur if calling user does not have access right for the API.
404	This error will occur if the org identified by orgId does not exist, or the user identified by userId does not exist.
500	A server error occurred. Check the logs for more details.

#### 200 Reponse Body

```
{
  "status": {
    "i18n_message": "response.ok",
    "message": "OK"
  },
  "response": "User abc user deleted succesfully"
}
```

## Get By Username

### HTTP REQUEST: GET

Lists the user details for the user

```
GET /api/1.0/org/{orgId}/username/{username}
```

#### URI PARAMETERS

Name	Type	Description
orgId	string	The Organization ID (UUID) for which you are querying.
username	string	The username of the user for which you are querying.

#### RESPONSES

Code	Description
------	-------------



200	Success.
401	This error will occur if calling user does not have access right for the API.
404	This error will occur if the org identified by orgId does not exist, or the user identified by username does not exist.
500	A server error occurred. Check the logs for more details.

#### 200 Reponse Body

```
{
  "status": {
    "i18n_message": "response.ok",
    "message": "OK"
  },
  "response": {
    "user": {
      "user_id": "83cc05ae-7ebd-4c69-5f37-b39e718a7d80",
      "name": "testuser_dc",
      "email": "testuser@atscale.com",
      "auth_username": "testuser_dc",
      "super_user": false,
      "api_super_user": false,
      "roles": null
    },
    "organization": {
      "id": "default",
      "name": "default"
    }
  }
}
```

## Profile Delete Picture

### HTTP REQUEST: DELETE

Removes the Profile Picture

**DELETE** /api/1.0/org/{orgId}/users/{userId}/picture**URI PARAMETERS**

Name	Type	Description
orgId	string	The Organization ID (UUID) for which you are querying.
userId	string	The userId of the user for which you are querying.

**RESPONSES**

Code	Description
200	Success.
401	This error will occur if calling user does not have access right for the API.
404	This error will occur if the org identified by orgId does not exist, or the user identified by userId does not exist.
500	A server error occurred. Check the logs for more details.

## 200 Reponse Body

```
{
  "status": {
    "i18n_message": "response.ok",
    "message": "OK"
  },
  "response": {
    "raw_json": "{\"lastOrg\":\"default\",\"logincount\":1}"
  }
}
```

**Profile Read****HTTP REQUEST: GET**

## Get the User Profile

**GET** /api/1.0/org/{orgId}/users/profile/{userId}

## URI PARAMETERS

Name	Type	Description
orgId	string	The Organization ID (UUID) for which you are querying.
userId	string	The User ID (UUID) for which you are querying.

## RESPONSES

Code	Description
200	Success.

## 200 Reponse Body

```
{
  "status": {
    "i18n_message": "response.ok",
    "message": "OK"
  },
  "response": {
    "raw_json": "{\"lastOrg\":\"default\",\"logincount\":1}"
  }
}
```