



# Connecting to Lt with SCIM 2.0

---

[Introduction](#)

[Setting Up Lt](#)

[Setting up your Identity Provider](#)

[Mappings](#)

[Provisioning](#)

[Deprovisioning](#)

[Appendix](#)

[Schemas](#)

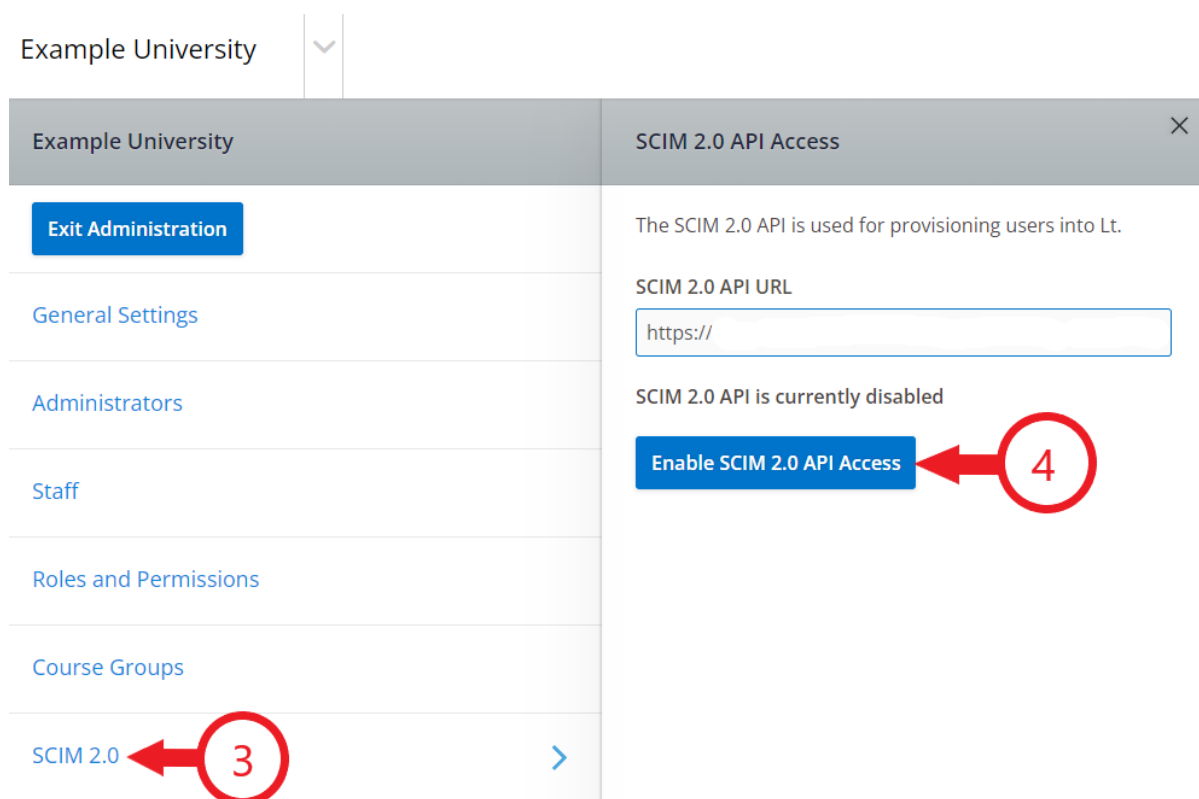
# Introduction

Lt supports using SCIM 2.0 to manage the users in your Lt instance. You can use SCIM to control access by staff, administrators and students. In particular SCIM can be used to provision users for [SSO](#) and deprovision users to prevent access via SSO and regular login. Lt is a SCIM Service Provider implementing the SCIM HTTP API , and connects to your SCIM Provisioning Client. Lt does not act as a SCIM Client and initiate requests.

## Setting Up Lt

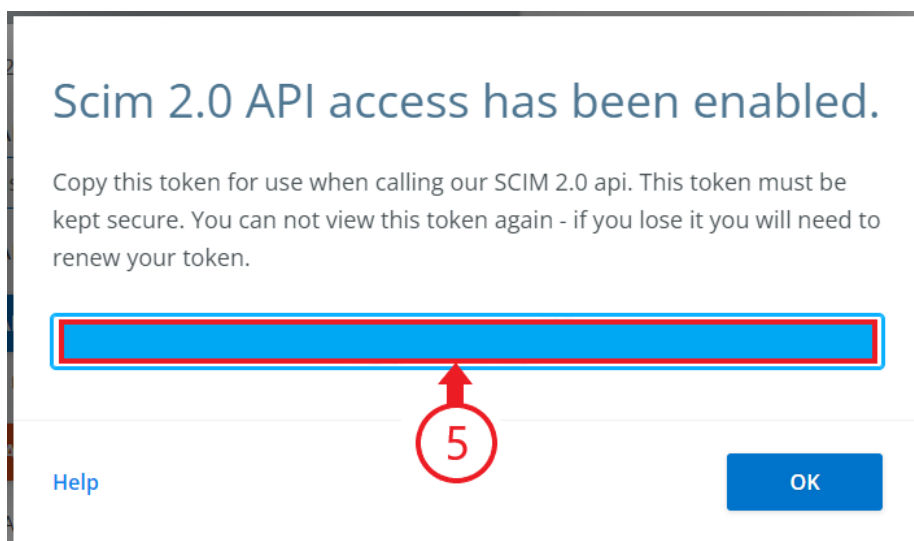
You will first need to enable SCIM in your Lt Instance.

1. As Instance Administrator, log in to your Lt instance at [ltlogin.com](https://ltlogin.com).
2. Select **Instance Administration**.
3. Select **SCIM 2.0**.
4. Select **Enable SCIM 2.0 API Access**.

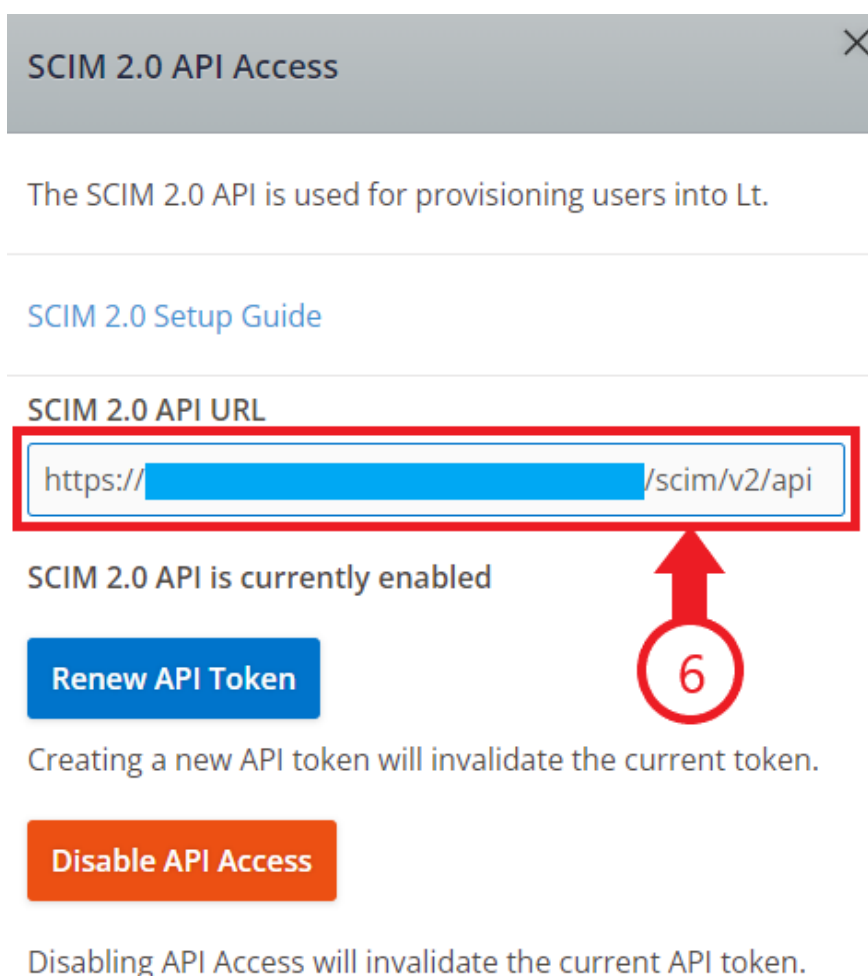


5. Copy the API token to a secure place.

**Note:** You cannot view this API token again so be sure to store it in a secure place. If you need to renew it later you will invalidate any existing SCIM connections.



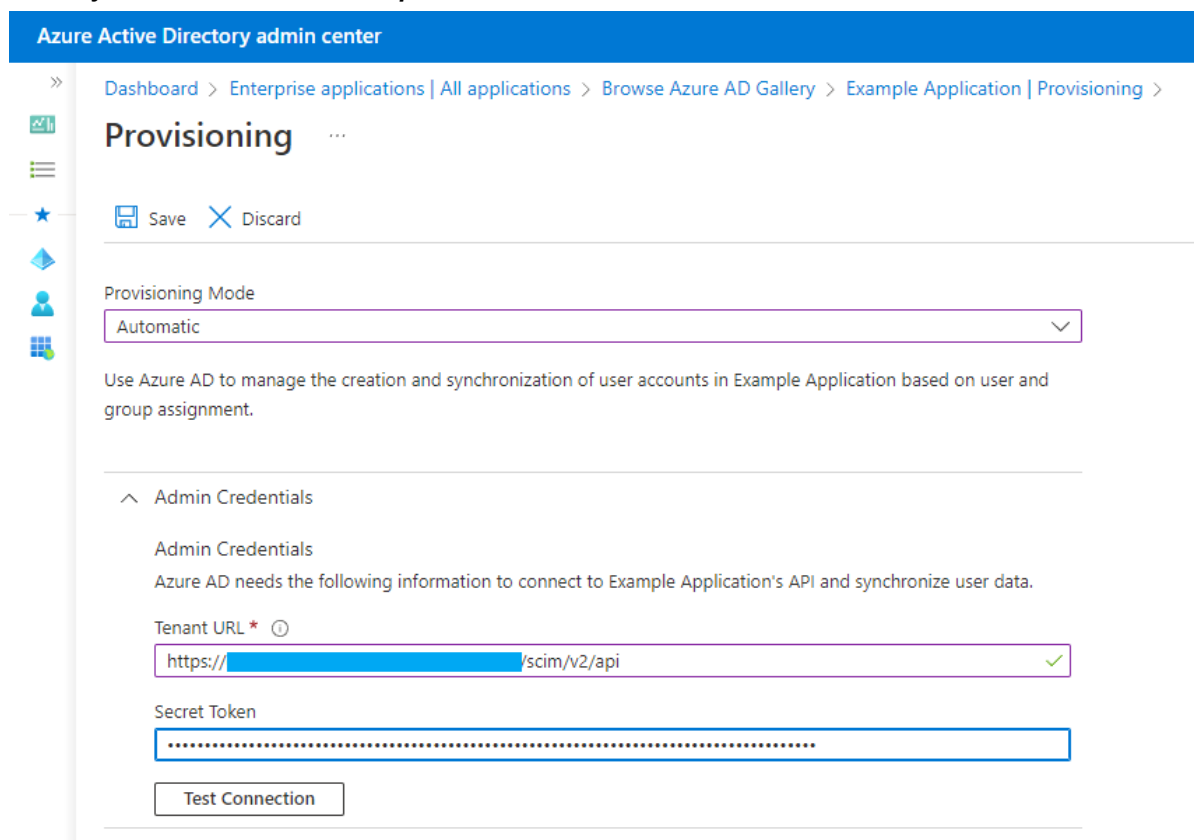
6. Copy the API URL for your instance. You will need it to setup your identity provider.



# Setting up your Identity Provider

In the Provisioning settings of your SCIM Provisioning Client setup a new connection inputting the URL and API token recorded above.

**Microsoft Azure AD shown as example**



Azure Active Directory admin center

Dashboard > Enterprise applications | All applications > Browse Azure AD Gallery > Example Application | Provisioning >

## Provisioning

Save Discard

Provisioning Mode

Automatic

Use Azure AD to manage the creation and synchronization of user accounts in Example Application based on user and group assignment.

Admin Credentials

Admin Credentials

Azure AD needs the following information to connect to Example Application's API and synchronize user data.

Tenant URL \*

https://.../scim/v2/api

Secret Token

.....

Test Connection

Lt uses only a limited subset of user resource attributes. You must adjust the attribute mappings appropriately. Any additional attributes may cause your provisioning to fail.

## Mappings

- **userName** - **\*Required\*** This must be the users email address. This is used to identify the user in Lt and populate the email attribute
- **active** - Whether the user is enabled (True) or disabled (False), users that are deleted are marked as disabled
- **name**
  - **name.givenName** - Useful when searching for a user using SCIM, not currently used in Lt
  - **name.familyName** - Useful when searching for a user using SCIM, not currently used in Lt

- **externalId** - optional identifier for the user as specified by the provisioning client. You can optionally configure SAML to use this for matching users in Lt.

 **Note:** Lt does not currently support the SCIM Groups resource.

## Attribute Mapping ...

 Save  Discard

Name

Provision Azure Active Directory Users

Enabled

Yes

No

Source Object

User

Source Object Scope

[All records](#)

Source Object

urn:ietf:params:scim:schemas:extension:enterprise:2.0:User

Target Object Actions



Create



Update



Delete

Attribute Mappings

Attribute mappings define how attributes are synchronized between Azure Active Directory and customappsso

Azure Active Directory Attribute	customappsso Attri...	Matching preceden...	Remove
userPrincipalName	userName	1	<button>Delete</button>
Switch([IsSoftDeleted], , "False", "True", "True", "False")	active		<button>Delete</button>
givenName	name.givenName		<button>Delete</button>
surname	name.familyName		<button>Delete</button>
mailNickname	externalId		<button>Delete</button>

[Add New Mapping](#)

*Microsoft Azure AD shown as example*

## Provisioning

Provisioning a user from your SCIM Client will allow them to log in using SAML SSO and allow you to disable access for any users. However, any users added will also need to be added as a student or staff to a course, or as an administrator. If you are not using SAML SSO, the user will need to be added as a student, staff, or admin and accept an email invite to set their login credentials.

## Deprovisioning

When you remove a user from your SCIM Client, the next time the SCIM Client synchronises its users those users will be marked as **active: False**. Those users will no longer be able to log in to Lt until they are re-enabled using SCIM. A deprovisioned user will not be counted against your active student number, will be shown as deleted in courses and within the admin interface, and will not be able to be restored from within Lt.

We do not delete the account from our database. It will be retained for as long as your instance remains active.

 **Note:** There is no way to re-enable users without using SCIM.

# Appendix

## Schemas

User schema:

---

```
{
  "schemas": [
    "urn:ietf:params:scim:schemas:core:2.0:Schema"
  ],
  "id": "urn:ietf:params:scim:schemas:core:2.0:User",
  "name": "User",
  "description": "User Account",
  "attributes": [
    {
      "name": "id",
      "type": "string",
      "description": "Unique identifier for the SCIM Resource as defined by the Service
Provider.",
      "mutability": "readOnly",
      "returned": "always",
      "uniqueness": "server",
      "multiValued": false,
      "required": true,
      "caseExact": true
    },
    {
      "name": "externalId",
      "type": "string",
      "description": "A String that is an identifier for the resource as defined by the
provisioning client. The service provider MUST always interpret the externalId as scoped
to the provisioning domain.",
      "mutability": "readWrite",
```

```

    "returned": "default",
    "uniqueness": "none",
    "multiValued": false,
    "required": false,
    "caseExact": true
  },
  {
    "name": "userName",
    "type": "string",
    "description": "A service provider's unique identifier for the user, typically\nused by
the user to directly authenticate to the service provider. Each User MUST include a
non-empty userName value. This identifier\nMUST be unique across the service
provider's entire set of Users.",
    "mutability": "readWrite",
    "returned": "default",
    "uniqueness": "server",
    "multiValued": false,
    "required": true,
    "caseExact": false,
    "pattern": "^(?!\\s*$).+"
  },
  {
    "name": "name",
    "type": "complex",
    "description": "The components of the user's real name. Providers MAY return just
the full name as a single string in the\nformatted sub-attribute, or they MAY return just
the individual component attributes using the other sub-attributes, or they MAY\nreturn
both. If both variants are returned, they SHOULD be describing the same name, with
the formatted name indicating how the\ncomponent attributes should be combined.",
    "mutability": "readWrite",
    "returned": "default",
    "uniqueness": "none",
    "multiValued": false,
    "required": false,
    "caseExact": false,
    "subAttributes": [
      {
        "name": "formatted",
        "type": "string",

```



```

    "description": "The full name, including all middle names, titles, and suffixes as
appropriate, formatted for display\n(e.g., 'Ms. Barbara J Jensen, III').",
    "mutability": "readOnly",
    "returned": "default",
    "uniqueness": "none",
    "multiValued": false,
    "required": false,
    "caseExact": false
  },
  {
    "name": "familyName",
    "type": "string",
    "description": "The family name of the User, or last name in most Western
languages (e.g., 'Jensen' given the full\nname 'Ms. Barbara J Jensen, III').",
    "mutability": "readWrite",
    "returned": "default",
    "uniqueness": "none",
    "multiValued": false,
    "required": false,
    "caseExact": false
  },
  {
    "name": "givenName",
    "type": "string",
    "description": "The given name of the User, or first name in most Western
languages (e.g., 'Barbara' given the\nfull name 'Ms. Barbara J Jensen, III').",
    "mutability": "readWrite",
    "returned": "default",
    "uniqueness": "none",
    "multiValued": false,
    "required": false,
    "caseExact": false
  }
]
},
{
  "name": "active",
  "type": "boolean",
  "description": "A Boolean value indicating the User's administrative status.",
  "mutability": "readWrite",

```

```

    "returned": "default",
    "uniqueness": "none",
    "multiValued": false,
    "required": false,
    "caseExact": false
  },
  {
    "name": "emails",
    "type": "complex",
    "description": "Email addresses for the user. The value SHOULD be canonicalized by
the service provider, e.g.,\n'bjensen@example.com' instead of
'bjensen@EXAMPLE.COM'. Canonical type values of 'work', 'home', and 'other'.",
    "mutability": "readOnly",
    "returned": "default",
    "uniqueness": "none",
    "multiValued": true,
    "required": false,
    "caseExact": false,
    "subAttributes": [
      {
        "name": "value",
        "type": "string",
        "description": "Email addresses for the user. The value SHOULD be canonicalized
by the service provider, e.g.,\n'bjensen@example.com' instead of
'bjensen@EXAMPLE.COM'. Canonical type values of 'work', 'home', and 'other'.",
        "mutability": "readOnly",
        "returned": "default",
        "uniqueness": "none",
        "multiValued": false,
        "required": false,
        "caseExact": false
      },
      {
        "name": "primary",
        "type": "boolean",
        "description": "A Boolean value indicating the 'primary' or preferred attribute value
for this attribute, e.g., the preferred mailing address or primary email address. The
primary attribute value 'true' MUST appear no more than once.",
        "mutability": "readOnly",
        "returned": "default",

```

```
"uniqueness": "none",
"multiValued": false,
"required": false,
"caseExact": false
}
]
},
"meta": {
  "resourceType": "Schema",
  "created": "2019-10-18T14:51:11+02:00",
  "lastModified": "2019-10-18T14:51:11+02:00",
  "location": "/Schemas/User"
}
}
```

---