Modules - SCIM [scim] - System for Cross-domain Identity Management

https://wiki.czechidm.com/
2019/12/16 05:28
# Table of Contents

Modules - SCIM [scim] - System for Cross-domain Identity Management .................................. 1

**Installation**  ......................................................................................................................................... 1  
  Dependencies ....................................................................................................................................... 1  
  Maven .................................................................................................................................................. 1  
  Standalone module ............................................................................................................................... 2  

**Architecture**  ...................................................................................................................................... 2  

**API**  .................................................................................................................................................. 2  
  Security ............................................................................................................................................... 3  
  Permissions ........................................................................................................................................ 3  
  SCIM standard resources ...................................................................................................................... 3  
  Custom schemas, resources and extensions ......................................................................................... 4  
  Swagger ............................................................................................................................................... 5  

**Read more**  ........................................................................................................................................ 6  

**Admin guide**  .................................................................................................................................... 6  

**Devel tutorials**  ................................................................................................................................. 6  

CzechIdM Identity Manager - https://wiki.czechidm.com/
The System for Cross-domain Identity Management (SCIM) specification is designed to make managing user identities in cloud-based applications and services easier. The specification suite seeks to build upon experience with existing schemas and deployments, placing specific emphasis on simplicity of development and integration, while applying existing authentication, authorization, and privacy models. Its intent is to reduce the cost and complexity of user management operations by providing a common user schema and extension model, as well as binding documents to provide patterns for exchanging this schema using standard protocols. In essence: make it fast, cheap, and easy to move users in to, out of, and around the cloud.

Scim module exposes interface in CzechIdM by the SCIM 2.0 specification. Read more about SCIM model, operations and endpoints.

### Installation

### Dependencies

We are using third party library scim2-sdk-common under LGPLv2 license. This library contains very useful DTOs (scim standard resources) and filter parser. Rest endpoints are exposed by our devstack.

Module can be installed both ways:

#### Maven

This way is mainly for developer, when CzechIdM is opened in your favorite IDE. You can add maven dependency into idm-app pom.xml (or into other project specific (and used) module pom.xml):

```xml
<dependency>
   <groupId>eu.bcvolutions.idm</groupId>
   <artifactId>idm-scim-impl</artifactId>
   <version>1.0.1</version>
</dependency>
```

Third party dependencies will be included automatically.

Note: module has to be built locally - available in [gitlab](https://gitlab.com) or access to our [nexus](https://nexus) has to be granted and repository included in the same pom.xml:

```xml
<repository>
   <id>nexus</id>
   <name>Nexus private modules</name>
</repository>
```
Standalone module

Module artefact (idm-scim-api, idm-scim-impl) can be downloaded from nexus releases and copied into installed CzechIdM server libraries <server>/WEB-INF/lib.

Third party library scim2-sdk-common in version 2.1.3 has to be copied into CzechIdM server libraries <server>/WEB-INF/lib too. Library can be built from github or downloaded from a public maven repository (e.g. central.maven.org).

After module is installed, run the CzechIdm and enable module by GUI (Setting → Modules) or by application properties:

```bash
# enable scim module by default
idm.pub.scim.enabled=true
```

Architecture

Module is separated to two libraries - api and implementation. Api contains useful DTOs, which can be used for the client implementation:

```xml
<dependency>
  <groupId>eu.bcvsolutions.idm</groupId>
  <artifactId>idm-scim-api</artifactId>
  <version>1.0.1</version>
</dependency>
```

Note: The same nexus repository has to be configured and access has to be granted as above.

API

Scim base path is available at url <server>/api/v1/scim. To simplify interoperability, SCIM provides three end points to discover supported features and specific attribute details:

- <server>/idm/api/v1/scim/ResourceTypes - An endpoint used to discover the types of resources available.
- <server>/idm/api/v1/scim/Schemas - Introspect resources and attribute extensions.
- <server>/idm/api/v1/scim/ServiceProviderConfig - Specification compliance, authentication schemes, data models.

This endpoints are public - no authentication is needed.

Resources are exposed on endpoints by resource types (authentication is needed):
Security

Two authentication schemes is supported now:

- Basic authentication - specification is available [here](#)
- Token authentication - Authentication scheme using the CzechIdM authentication token in CIDMST header. Documentation and example is available in swagger [server]/swagger-ui.html#!/Authentication/loginUsingPOST (read more below).

Permissions

Module defines new permission group:

- SCIM_READ - read all resources
- SCIM_CREATE - create all resources
- SCIM_UPDATE - update all resources
- SCIM_DELETE - delete all resources

Standard CzechIdM authorization policies are not supported.

SCIM standard resources

Read more about SCIM model, operations and endpoints.

Standard scim schemas and resources [rfc7643](#) are implemented with a few limitations:

User resource unsupported properties, which cannot be saved in CzechIdM:

- nickName
- title
- preferredLanguage
- locale
- timeZone
- addresses
- ims
- photos
- roles
- entitlements
- x509Certificates
- name.middleName
- emails - only one email (primary) can be given
- phoneNumbers - only one phone number (primary) can be given

This attributes are not implemented on the CzechIdM side by default, so when client try to save this attributes, exception with filled unsupported attribute name will be thrown. Custom module extension can be created, when attributes should be supported and saved in CzechIdM (with
ScimUserService extension - override toDto and toResource methods and save attribute e.g. to custom extended attributes or into custom entity).

Group resource property displayName is mapped to CzechIdM attributes code and name. If displayName is changed, then both properties are changed (since 1.2.0 version, previous version modified name attribute only, code was unmodifiable).

Implemented filter and sort properties on standard scim resources:

- User - externalId, userName, name.familyName
- Group - externalId, displayName

Filter supports equals (eq) operator and AND clause only. When other operator or clause is used, then unsupported operation will be thrown. Pagination startIndex and count parameter can be used. startIndex parameter is he 1-based index of the first query result. Start index has to be the first index on the page (n * count + 1), exception with code FIND_START_INDEX_INVALID is thrown otherwise (CzechIdM can paginate by the whole page only). Filter parameter has to be url encoded.

Standard CzechIdM filter properties can be used too, this is not in SCIM standard - e.g. 
<server>/api/v1/scim/Users?username=testOne is alias to SCIM standard 
<server>/api/v1/scim/Users?filter%3DuserName%20eq%20%22testOne%22.

If CzechIdM resource implements Codeable interface (e.g. User, Group, TreeType), then resource can be addressed additively by code in url and used in filter properties. E.q. 
<server>/api/v1/scim/Users/testOne will get user resource. UUID identifier can be used too.

Bulk operation is not supported.

Attribute names and values are case sensitive.

**Custom schemas, resources and extensions**

Module adds custom schema urn:ietf:params:scim:schemas:CzechIdM:8.1 with resources:

- Contract - ~ IdmIdentityContract, implemented filter properties externalId, user, position, main, workPosition
- ContractGuarantee - ~ IdmContractGuarantee, implemented filter properties externalId, guarantee, contract
- UserGroup - ~ IdmIdentityRole, implemented filter properties externalId, guarantee, contract
- TreeType - ~ IdmTreeType, implemented filter properties externalId, code
- TreeNode - ~ IdmTreeNode, implemented filter properties externalId, code, parent, treeType
- Form - extended form definitions, attributes and values - just schema without resource
- FormDefinition - ~ IdmFormDefinition, implemented filter properties type, code. Readonly form definitions.
- FormAttribute - ~ IdmFormAttribute - just schema without resource
- FormValue - ~ IdmFormValue - just schema without resource

and extensions:
Extended form values - namespace

urn:ietf:params:scim:schemas:CzechIdM:8.1:Form. Extended form values can be added to standard and custom resources. Resources, which supports extended forms can be listed on schemes endpoint.

Available form definitions (and their attributes) for saving extended attribute values can be listed by FormDefinition resource type endpoint. Then extended form values can be saved together with resource (Resource has to implement FormableResource interface):

```json
{
    "schemas": [
        "urn:ietf:params:scim:schemas:CzechIdM:8.1:Form"
    ],
    "userName": "scimOne",
    "urn:ietf:params:scim:schemas:CzechIdM:8.1:Form": {
        "forms": [{
            "code": "default",
            "attributes": [{
                "code": "extArtrOne",
                "values": [{
                    "shortTextValue": "test"
                }]
            }
        ]
    }
}
```

As you can see, you don't need to know exact form definition and attributes uuid identifiers - code can be given instead, but uuid identifier can be used too (alias). When request with resource is sent (POST / PUT / PATCH), then uuid identifiers are returned for saved resources. Saved value identifier has to be used, when value has to be updated, otherwise value will be recreated (⇒ drop and create).

### Swagger

To expose swagger documentation endpoints, application property has to be configured:

```bash
## Swagger config
# enable swagger endpoint (can be disabled for development etc.)
springfox.documentation.swagger.enabled=true
```

Then swagger documentation will be available at url `<server>/swagger-ui.html`.

This property is configured in test and production profile by default.
Read more

Admin guide

- SCIM module: Forming groups of users

Devel tutorials

- SCIM module - Creating a user with group assigned
- SCIM module - save extended form value

From: https://wiki.czechidm.com/ - CzechIdM Identity Manager

Permanent link: https://wiki.czechidm.com/devel/documentation/modules_scim

Last update: 2019/08/16 09:17