AgileWorks R3

SCIM API ガイド

R3.3 第 1 版(2026/07/15)

ATLED

目次／索引

1. SCIM API について 4
1.1. SCIM API とは 4
1.2. SCIM API の主な機能 4
2. 共通仕様 5
2.1. 認証方式 5
2.2. SCIM API 一覧 5
2.3. リクエストヘッダ 5
2.4. ステータスコード 6
2.5. エラーレスポンス 6
3. ユーザーAPIs 7
3.1. ユーザー情報取得 7
3.2. 指定ユーザー情報取得 10
3.3. ユーザー情報作成 11
3.4. ユーザー情報更新 13
3.5. ユーザー情報一部更新 15
3.6. ユーザー情報削除 17
4. 組織管理 APIs 18
4.1. 組織情報取得 18
4.2. 指定組織情報取得 20
4.3. 組織情報作成 21
4.4. 組織情報更新 22
4.5. 組織情報一部更新 23
4.6. 組織情報削除 25
5. リソース種別 API 26
5.1. リソース情報取得 26
6. スキーマ定義 API 28
6.1. スキーマ定義取得 28
7. サービス定義 33
7.1. サービス定義取得 33

◆ 改版履歴

版数	年月日	改版内容
第 1 版	2026 年 07 月 15 日	第 1 版作成

1. SCIM APIについて

この章では、AgileWorks の SCIM API について説明します。

1.1. SCIM API とは

SCIM（System for Cross-domain Identity Management）API は、アイデンティティ管理を簡素化および標準化するためのプロトコルと API 仕様です。主な目的は、異なるアプリケーションやシステム間でユーザーアカウントやアイデンティティ情報を効率的に管理できるようにすることを目的としています。

1.2. SCIM API の主な機能

AgileWorks の SCIM API は、次の機能を提供します。

#	機能名	概要
1	ユーザー管理	ユーザーアカウントを作成、更新、削除、検索するための機能
2	組織管理	組織を作成、更新、削除、検索するための機能
3	リソース種別	各スキーマで使用可能なリソースとその使用目的を参照する
4	スキーマ定義	受け入れ可能なリソースを取得する
5	サービス定義	サービスの詳細を取得する

2. 共通仕様

2.1. 認証方式

認証は Web API と同様になります。
※別紙『AwSDK08-WebAPI 利用ガイド』の『2.1 認証方式』参照

2.2. SCIM API 一覧

【UsersEndpoint】

#	API 名称	説明
1	ユーザー情報取得	すべてのユーザーの情報を取得します。
2	指定ユーザー情報取得	指定のユーザーの情報を取得します。
3	ユーザー情報作成	ユーザー情報の新規登録を行います。
4	ユーザー情報更新	指定ユーザーの情報を更新します。
5	ユーザー情報一部更新	指定ユーザーの情報の一部を更新します。
6	ユーザー情報削除	指定ユーザーの削除を行います。

【GroupsEndpoint】

#	API 名称	説明
1	組織情報取得	すべての組織の情報を取得します。
2	指定組織情報取得	指定の組織の情報を取得します。
3	組織情報作成	組織情報の新規登録を行います。
4	組織情報更新	指定組織の情報を更新します。
5	組織情報一部更新	指定組織の情報の一部を更新します。
6	組織情報削除	指定組織の削除を行います。

【ResourceTypesEndpoint】

#	API 名称	説明
1	リソース情報取得	サポートされているリソースタイプやそれらのスキーマ情報に関する情報を取得します。

【SchemasEndpoint】

#	API 名称	説明
1	スキーマ情報取得	サポートされているスキーマ情報を取得します。

【ServiceProviderConfigsEndpoint】

#	API 名称	説明
1	サービス情報取得	SCIM サービスプロバイダの設定情報を取得します。

2.3. リクエストヘッダ

送信するリクエストに指定するリクエストヘッダを指定します。

#	ヘッダ	必須	説明	例
1	Authorization	○	HTTP 認証の認証資格証明	別紙『AwSDK08-WebAPI 利用ガイド』の『3.1 リクエストヘッダ』参照
2	Content-Type		リクエストの本文のメディア・タイプ。	application/json; charset=UTF-8

2.4. ステータスコード

API コールの後で、以下の HTTP ステータスコードが返されます。ステータスコードの説明は、特に断りがない限り、HTTP status code specification に準拠しています。

#	ステータスコード	説明
1	200 OK	リクエストが成功しました。
2	400 Bad Request	リクエストに問題があります。
3	401 Unauthorized	認証に失敗した際のエラーです。
4	403 Forbidden	リソースにアクセスする権限がありません。
5	404 Not Found	リクエストしたリソースが見つかりません。
6	500 Internal Server Error	サーバー内部でエラーが発生しました。
7	503 Service Unavailable	サービスが一時的に利用できません。

2.5. エラーレスポンス

エラー発生時は、以下の JSON データを含むレスポンスボディが返されます。

▼レスポンスボディ例（JSON 形式）

```
{
  "schemas": [
    "urn:ietf:params:scim:api:messages:2.0:Error"
  ],
  "status": 401,
  "detail": "ログイン ID またはパスワードに誤りがあります。"
}
```

▼ レスポンスボディ

#	項目名	型	内容	説明
1	schemas	文字列	スキーマ	
2	status	整数	ステータスコード	2.4 ステータスコード参照
3	detail	文字列	エラー詳細	

3. ユーザーAPIs

この章では、ユーザーアカウントを操作する API について説明します。
ユーザーアカウントを作成、更新、削除、検索することができます。

3.1. ユーザー情報取得

指定された条件を元にユーザー情報を取得します。

▼ メソッドタイプ
GET

▼ リクエスト URL 例

```
https://sample.atled.jp/AgileWorks/Broker/SCIM/Users
```

▼ クエリパラメータ

#	項目名	必須	型	内容	説明
1	filter		文字列	フィルター	ユーザーコード、ユーザー名称、ログイン ID、メールアドレスを絞り込む時に指定します。 詳細は下記「フィルター」を参照。 ※検索結果が 1000 件を超えた場合、1000 件まで応答する

・フィルター
クエリパラメータの「filter」にユーザーコード、ユーザー名称、ログイン ID、メールアドレスを
指定する事でユーザー情報の絞り込みができます。
使用出来る演算子は`eq(equals)` 等しいことを示す演算子のみになります。

【フィルター指定形式】

フィルターで条件を指定する場合は以下のフォーマットで指定します。
・ユーザーコード

```
externalId eq "{ユーザーコード}"
```

・ユーザー名称

```
displayName eq "{ユーザー名称}"
```

・ログイン ID

```
userName eq "{ログイン ID}"
```

・メールアドレス

```
emails eq "{メールアドレス}"
```

【フィルター指定例】
・ユーザーコードで絞り込みを行う場合

```
https://sample.atled.jp/AgileWorks/Broker/SCIM/Users?filter=externalId eq "user001"
```

▼**レスポンスボディ例**（JSON 形式）


```
{
  "schemas": [
     "urn:ietf:params:scim:api:messages:2.0:ListResponse"
  ],
  "totalResults": 1,
  "Resources": [
     {
         "id": 100767,
         "externalId": "user001",
         "displayName": "user001",
         "userName": "user001",
         "emails": [
             {
                 "value": "user001@atled.jp",
                 "primary": true
             }
         ],
         "active": true,
         "roles": [
             {
                 "value": 100804,
                 "display": "UR_001"
             }
         ],
         "groups": [
             {
                 "value": 100772,
                 "display": "org001"
             }
         ],
         "schemas": [
             "urn:ietf:params:scim:schemas:core:2.0:User"
         ]
     }
  ],…
}
```


▼ **レスポンスボディ**

#	項目名	型	内容	説明
1	schemas	文字列	スキーマ	"urn:ietf:params:scim:api:messages:2.0:ListResponse" 固定
2	totalResults	整数	検索結果件数	
3	Resources	配列	ユーザー情報	詳細は以下「ユーザー情報」を参照。

▼ **レスポンスボディ（ユーザ情報）**

#	項目名	型	内容	説明
1	id	整数	ユーザーID	
2	externalId	文字列	ユーザーコード	
3	displayName	文字列	ユーザー名称	
4	userName	文字列	ログイン ID	
5	emails	配列	メール情報	詳細は以下「メール情報」を参照。
6	active	論理	アカウント状態	true:ログイン可能 false:ログイン不可
7	roles	配列	ロール情報	詳細は以下「ロール情報」を参照。
8	groups	配列	グループ情報	詳細は以下「グループ情報」を参照。
9	schemas	文字列	スキーマ	"urn:ietf:params:scim:schemas:core:2.0:User" 固定

▼ **レスポンスボディ（メール情報）**

#	項目名	型	内容	説明
1	value	文字列	メールアドレス	
2	primary	論理	優先フラグ	true 固定

▼ **レスポンスボディ（ロール情報）**

#	項目名	型	内容	説明
1	value	整数	ロール ID	ユニバーサルロール
2	display	文字列	表示名	

▼ **レスポンスボディ（グループ情報）**

#	項目名	型	内容	説明
1	value	整数	グループ ID	組織
2	display	文字列	表示名	

3.2. 指定ユーザー情報取得

パスに指定したユーザー ID に紐づくユーザーの情報を取得します。

▼ メソッドタイプ
GET

▼ リクエスト URL 例

```
https://sample.atled.jp/AgileWorks/Broker/SCIM/Users/≪USER_ID≫
```

▼パスパラメーター

#	項目名	必須	型	内容	説明
1	≪USER_ID≫	○	整数	ユーザーID	

▼レスポンスボディ例（JSON 形式）

```
{
  "id": 100767,
  "externalId": "user001",
  "displayName": "user001",
  "userName": "user001",
  "emails": [
    {
      "value": "user001@atled.jp",
      "primary": true
    }
  ],
  "active": true,
  "roles": [
    {
      "value": 100804,
      "display": "UR_001"
    }
  ],
  "groups": [
    {
      "value": 100772,
      "display": "org001"
    }
  ],
  "schemas": [
    "urn:ietf:params:scim:schemas:core:2.0:User"
  ]
}
```

▼ レスポンスボディ
3.1 ＜レスポンスボディ（ユーザー情報）＞参照

3.3. ユーザー情報作成

リクエストボディに指定したデータを元にユーザー情報を作成します。

▼ メソッドタイプ
POST

▼ リクエスト URL 例

```
https://sample.atled.jp/AgileWorks/Broker/SCIM/Users
```

▼ リクエストボディ例（JSON 形式）

```
{
  "externalId": "user90001",
  "userName": "user90001",
  "displayName": "user90001",
  "password": "agileworks123",
  "active": true,
  "locale": "ja",
  "emails": [
    {
      "value": "user90001@aw.co.jp",
      "primary": true
    }
  ],
  "roles": [
    {
      "value": "100804",
      "display": "UR_001"
    }
  ],
  "groups": [
    {
      "value": "100772",
      "display": "org001"
    }
  ]
}
```

▼ リクエストボディ（ユーザー情報）

#	項目名	必須	型	内容	説明
1	externalId	○	文字列	ユーザーコード	
2	userName	○	文字列	ログイン ID	
3	displayName	○	文字列	ユーザー名称	
4	password		文字列	パスワード	
5	locale		文字列	言語	
6	active		論理	アカウント状態	true:ログイン可能 false:ログイン不可
7	emails		配列	メール情報	詳細は以下「メール情報」を参照。
8	roles		配列	ロール情報	詳細は以下「ロール情報」を参照。
9	groups		配列	グループ情報	詳細は以下「グループ情報」を参照。
10	schemas		文字列	スキーマ	"urn:ietf:params:scim:schemas:core:2.0:User" 固定

▼ リクエストボディ（メール情報）

#	項目名	必須	型	内容	説明
1	value		文字列	メールアドレス	
2	primary		論理	優先フラグ	

▼ リクエストボディ（ロール情報）

#	項目名	必須	型	内容	説明
1	value		整数	ロール ID	ユニバーサルロール
2	display		文字列	表示名	

▼ **リクエストボディ（グループ情報）**

#	項目名	必須	型	内容	説明
1	value		整数	グループ ID	組織
2	display		文字列	表示名	

▼**レスポンスボディ例**（JSON 形式）


```
{
  "id": 110017,
  "externalId": "user90001",
  "displayName": "user90001",
  "userName": "user90001",
  "emails": [
    {
      "value": "user90001@aw.co.jp",
      "primary": true
    }
  ],
  "active": true,
  "roles": [
    {
      "value": 100804,
      "display": "UR_001"
    }
  ],
  "groups": [
    {
      "value": 100772,
      "display": "org001"
    }
  ],
  "schemas": [
    "urn:ietf:params:scim:schemas:core:2.0:User"
  ]
}
```


▼ **レスポンスボディ**

3.1 ＜レスポンスボディ（ユーザー情報）＞参照

3.4. ユーザー情報更新

パスに指定したユーザーID に紐づくユーザの情報を更新します。
※リクエストボディで指定しなかった項目は、更新されません。

▼ メソッドタイプ
PUT

▼ リクエスト URL 例

```
https://sample.atled.jp/AgileWorks/Broker/SCIM/Users/≪USER_ID≫
```

▼パスパラメーター

#	項目名	必須	型	内容	説明
1	≪USER_ID≫	○	整数	ユーザーID	

▼ リクエストボディ例（JSON 形式）

```
{
  "externalId": "user90001",
  "userName": "user90001",
  "displayName": "user90001",
  "active": true,
  "locale": "ja",
  "emails": [
    {
      "value": "user90001@aw.co.jp",
      "primary": true
    }
  ],
  "roles": [
    {
      "value": "100804",
      "display": "UR_001"
    }
  ],
  "groups": [
    {
      "value": "100772",
      "display": "org001"
    }
  ]
}
```

▼ リクエストボディ（ユーザ情報）

#	項目名	必須	型	内容	説明
1	externalId	○	文字列	ユーザーコード	
2	userName	○	文字列	ログイン ID	
3	displayName	○	文字列	ユーザー名称	
4	locale		文字列	言語	
5	active		論理	アカウント状態	true:ログイン可能 false:ログイン不可
6	emails		配列	メール情報	詳細は以下「メール情報」を参照。
7	roles		配列	ロール情報	詳細は以下「ロール情報」を参照。
8	groups		配列	グループ情報	詳細は以下「グループ情報」を参照。
9	schemas		文字列	スキーマ	"urn:ietf:params:scim:schemas:core:2.0:User" 固定

▼ リクエストボディ（メール情報）

#	項目名	必須	型	内容	説明
1	value		文字列	メールアドレス	
2	primary		論理	優先フラグ	

▼ リクエストボディ（ロール情報）

#	項目名	必須	型	内容	説明
1	value		整数	ロール ID	ユニバーサルロール

2	display		文字列	表示名	

▼ リクエストボディ（グループ情報）

#	項目名	必須	型	内容	説明
1	value		整数	グループ ID	組織
2	display		文字列	表示名	

▼レスポンスボディ例（JSON 形式）

```
{
  "id": 110017,
  "externalId": "user90001",
  "displayName": "user90001",
  "userName": "user90001",
  "emails": [
    {
      "value": "user90001@aw.co.jp",
      "primary": true
    }
  ],
  "active": true,
  "roles": [
    {
      "value": 100804,
      "display": "UR_001"
    }
  ],
  "groups": [
    {
      "value": 100772,
      "display": "org001"
    }
  ],
  "schemas": [
    "urn:ietf:params:scim:schemas:core:2.0:User"
  ]
}
```

▼ レスポンスボディ

3.1 ＜レスポンスボディ（ユーザ情報）＞参照

3.5. ユーザー情報一部更新

パスに指定したユーザー ID に紐づくユーザー情報の指定項目のみを更新します。

▼ メソッドタイプ
PATCH

▼ リクエスト URL 例

```
https://sample.atled.jp/AgileWorks/Broker/SCIM/Users/≪USER_ID≫
```

▼パスパラメーター

#	項目名	必須	型	内容	説明
1	≪USER_ID≫	○	整数	ユーザーID	

▼ リクエストボディ例（JSON 形式）

```
{
  "schemas": [
    "urn:ietf:params:scim:api:messages:2.0:PatchOp"
  ],
  "Operations": [
    {
      "op": "Replace",
      "path": "emails[type eq \"work\"].value",
      "value": "user10003_UP2@kasasakiatled.onmicrosoft.com"
    },
    {
      "op": "Add",
      "path": "locale",
      "value": "ja"
    }
  ]
}
```

▼ リクエストボディ（ユーザー情報）

#	項目名	必須	型	内容	説明
1	schemas		文字列	スキーマ	"urn:ietf:params:scim:api:messages:2.0:PatchOp" 固定
2	Operations		配列	オペレーション	詳細は以下「オペレーション情報」を参照。

▼ リクエストボディ（オペレーション情報）

#	項目名	必須	型	内容	説明
1	op	○	文字列	操作タイプ	更新操作は以下が指定可能です。 （大文字小文字は区別しません） 追加(add) 置換(replace) 削除(remove) ※
2	path	○	文字列	変更対象属性	詳細は以下「path について」を参照。
3	value		文字列	新しい値	path で指定した属性の更新する値

※削除(remove)の操作はリソースの必須項目（externalId、displayName、userName）に対して行うことは出来ません。
必須指定の項目に削除(remove)の操作を行った場合はエラーレスポンスが返却されます。

▼ path について
指定可能な path 一覧と、更新方法は以下記載に従います。

#	path	AW 項目	op（操作タイプ）		
			Add	Replace	Remove
1	externalId	ユーザーコード	value で指定した値で更新します		指定出来ません
2	displayName	ユーザー名称	〃		〃
3	username	ログイン ID	〃		〃
4	password	パスワード	〃		変更しない
5	active	アカウントロック	〃		〃
6	emails	メールアドレス	〃		null 固定
7	locale	表示言語	〃		規定の言語に更新します
8	roles	ユニバーサルロール	value で指定したユニバーサルロールを追加します	value で指定したユニバーサルロールに置き換えします	ユーザーが所持している既存のユニバーサルロールを適用終了又は削除します※１
9	groups	所属組織	value で指定した所属組織を追加します	value で指定した所属組織に置き換えします	ユーザーが所属している既存の所属組織を適用終了又は削除します※１

※１　履歴モードの状態によって動作が変わります
　　無効：削除します／有効：適用終了します

▼レスポンスボディ例（JSON 形式）

```
{
  "id": 110017,
  "externalId": "user90001",
  "displayName": "user90001",
  "userName": "user90001",
  "emails": [
    {
      "value": "user90001@aw.co.jp",
      "primary": true
    }
  ],
  "active": true,
  "roles": [
    {
      "value": 100804,
      "display": "UR_001"
    }
  ],
  "groups": [
    {
      "value": 100772,
      "display": "org001"
    }
  ],
  "schemas": [
    "urn:ietf:params:scim:schemas:core:2.0:User"
  ]
}
```

▼ レスポンスボディ
3.1 ＜レスポンスボディ（ユーザ情報）＞参照

3.6. ユーザー情報削除

パスに指定したユーザー ID に紐づくユーザ情報を削除します。

▼ メソッドタイプ
DELETE

▼ リクエスト URL 例

```
https://sample.atled.jp/AgileWorks/Broker/SCIM/Users/≪USER_ID≫
```

▼パスパラメーター

#	項目名	必須	型	内容	説明
1	≪USER_ID≫	○	整数	ユーザーID	

▼ 削除方法について
ユーザー情報削除 API は、履歴モードの状態で削除動作が異なります

#	条件（履歴モード）	説明
1	無効の場合	削除します
2	有効の場合	適用終了します

▼ レスポンスボディ
レスポンスボディなし

4. 組織管理APIs

この章では、組織を操作する API について説明します。
組織を作成、更新、削除、検索することができます。

4.1. 組織情報取得

指定された条件を元に組織情報を取得します。

▼ メソッドタイプ
GET

▼ リクエスト URL 例

```
https://sample.atled.jp/AgileWorks/Broker/SCIM/Groups
```

▼クエリパラメーター

#	項目名	必須	型	内容	説明
1	filter		文字列	フィルター	組織コード、組織名称を絞り込む時に指定します。 詳細は下記「フィルター」を参照。

・フィルター
クエリパラメータの「filter」に組織コード、組織名称を指定する事で組織情報の絞り込みができます。
使用出来る演算子は`eq(equals)` 等しいことを示す演算子のみになります。

【フィルター指定形式】
フィルターで条件を指定する場合は以下のフォーマットで指定します。
・組織コード

```
externalId eq "{組織コード}"
```

・組織名称

```
displayName eq "{組織名称}"
```

【フィルター指定例】
・組織コードで絞り込みを行う場合

```
https://sample.atled.jp/AgileWorks/Broker/SCIM/Groups?filter=externalId eq "org001"
```

▼レスポンスボディ例（JSON 形式）

```
{
  "schemas": [
    "urn:ietf:params:scim:api:messages:2.0:ListResponse"
  ],
  "totalResults": 1,
  "Resources": [
    {
      "schemas": [
        "urn:ietf:params:scim:schemas:core:2.0:Group"
      ],
      "id": "100772",
      "externalId": "org001",
      "displayName": "org001"
    }
  ],…
}
```

▼ レスポンスボディ

#	項目名	型	内容	説明
1	schemas	文字列	スキーマ	"urn:ietf:params:scim:api:messages:2.0:ListResponse" 固定
2	totalResults	整数	検索結果件数	
3	Resources	配列	組織情報	詳細は以下「組織情報」を参照。

▼ **レスポンスボディ（組織情報）**

#	項目名	型	内容	説明
1	id	整数	組織 ID	
2	externalId	文字列	組織コード	
3	displayName	文字列	組織名称	
4	schemas	文字列	スキーマ	"urn:ietf:params:scim:schemas:core:2.0:Group" 固定

4.2. 指定組織情報取得

パスに指定した組織 ID に紐づく組織の情報を取得します。

▼ メソッドタイプ
GET

▼ リクエスト URL 例

```
https://sample.atled.jp/AgileWorks/Broker/SCIM/Groups/≪ORG_ID≫
```

▼パスパラメーター

#	項目名	必須	型	内容	説明
1	≪ORG_ID≫	○	整数	組織 ID	

▼レスポンスボディ例（JSON 形式）

```
{
  "schemas": [
    "urn:ietf:params:scim:schemas:core:2.0:Group"
  ],
  "id": "100772",
  "externalId": "org001",
  "displayName": "org001"
}
```

▼ レスポンスボディ
4.1 ＜レスポンスボディ（組織情報）＞参照

4.3. 組織情報作成

リクエストボディに指定したデータを元に組織情報を作成します。

▼ メソッドタイプ
POST

▼ リクエスト URL 例

```
https://sample.atled.jp/AgileWorks/Broker/SCIM/Groups
```

▼ リクエストボディ例（JSON 形式）

```
{
  "externalId": "org101",
  "displayName": "組織1O1"
}
```

▼ リクエストボディ（組織情報）

#	項目名	必須	型	内容	説明
1	externalId	○	文字列	組織コード	
2	displayName	○	文字列	組織名称	

▼レスポンスボディ例（JSON 形式）

```
{
  "schemas": [
    "urn:ietf:params:scim:schemas:core:2.0:Group"
  ],
  "id": "110023",
  "externalId": "org101",
  "displayName": "組織1O1"
}
```

▼ レスポンスボディ
4.1 ＜レスポンスボディ（組織情報）＞参照

4.4. 組織情報更新

パスに指定した組織 ID に紐づく組織の情報を更新します。

▼ メソッドタイプ
PUT

▼ リクエスト URL 例

```
https://sample.atled.jp/AgileWorks/Broker/SCIM/Groups/≪ORG_ID≫
```

▼パスパラメーター

#	項目名	必須	型	内容	説明
1	≪ORG_ID≫	○	整数	組織 ID	

▼ リクエストボディ例（JSON 形式）

```
{
  "externalId": "org101_UP",
  "displayName": "組織1O1_UP"
}
```

▼ リクエストボディ（組織情報）

#	項目名	必須	型	内容	説明
1	externalId		文字列	組織コード	
2	displayName		文字列	組織名称	

▼レスポンスボディ例（JSON 形式）

```
{
  "schemas": [
    "urn:ietf:params:scim:schemas:core:2.0:Group"
  ],
  "id": "110023",
  "externalId": " org101_UP ",
  "displayName": "組織1O1_UP "
}
```

▼ レスポンスボディ
4.1 ＜レスポンスボディ（組織情報）＞参照

4.5. 組織情報一部更新

パスに指定した組織 ID に紐づく組織情報の指定項目のみを更新します。

▼ メソッドタイプ
PATCH

▼ リクエスト URL 例

```
https://sample.atled.jp/AgileWorks/Broker/SCIM/Groups/≪ORG_ID≫
```

▼パスパラメーター

#	項目名	必須	型	内容	説明
1	≪ORG_ID≫	○	整数	組織 ID	

▼ リクエストボディ例（JSON 形式）

```
{
  "schemas": [
    "urn:ietf:params:scim:api:messages:2.0:PatchOp"
  ],
  "Operations": [
    {
      "op": "Replace",
      "path": "externalId",
      "value": "org101_UP2"
    },
    {
      "op": "Replace",
      "path": "displayName",
      "value": "組織1O1_UP2"
    }
  ]
}
```

▼ リクエストボディ（ユーザ情報）

#	項目名	必須	型	内容	説明
1	schemas		文字列	スキーマ	"urn:ietf:params:scim:api:messages:2.0:PatchOp" 固定
2	Operations		配列	オペレーション	詳細は以下「オペレーション情報」を参照。

▼ リクエストボディ（オペレーション情報）

#	項目名	必須	型	内容	説明
1	op		文字列	操作タイプ	更新操作は以下が指定可能です。 （大文字小文字は区別しません） 追加(add) 置換(replace) 削除(remove) ※
2	path		文字列	変更対象の属性のパス	詳細は以下「path について」を参照。
3	value		文字列	新しい値	

※削除(remove)の操作はリソースの必須項目（externalId、displayName）に対して行うことは出来ません。
必須指定の項目に削除(remove)の操作を行った場合はエラーレスポンスが返却されます。

▼ path について
指定可能な path 一覧と、更新方法は以下記載に従います。

#	path	AW 項目	op（操作タイプ）		
			Add	Replace	Remove
1	externalId	組織コード	value で指定した値で更新します		指定出来ません
2	displayName	組織名称	〃		〃

▼レスポンスボディ例（JSON 形式）


```
{
  "schemas": [
    "urn:ietf:params:scim:schemas:core:2.0:Group"
  ],
  "id": "110023",
  "externalId": " org101_UP 2",
  "displayName": "組織1O1_UP 2"
}
```


▼ レスポンスボディ

4.1 ＜レスポンスボディ（組織情報）＞参照

4.6. 組織情報削除

パスに指定した組織 ID に紐づく組織情報を削除します。

▼ メソッドタイプ
DELETE

▼ リクエスト URL 例

```
https://sample.atled.jp/AgileWorks/Broker/SCIM/Groups/≪ORG_ID≫
```

▼パスパラメーター

#	項目名	必須	型	内容	説明
1	≪ORG_ID≫	○	整数	組織 ID	

▼ 削除方法について
組織情報削除 API は、履歴モードの状態で削除動作が異なります

#	条件（履歴モード）	説明
1	無効の場合	削除します
2	有効の場合	適用終了します

▼ レスポンスボディ
レスポンスボディなし

5. リソース種別API

この章では、リソース・タイプの構成の取得 API について説明します。

5.1. リソース情報取得

SCIM サーバーがサポートするリソースタイプ（User、Group）に関する情報を提供するためのエンドポイントです。

▼ メソッドタイプ
GET

▼ リクエスト URL 例

```
https://sample.atled.jp/AgileWorks/Broker/SCIM/ResourceTypes
```

▼レスポンスボディ例（JSON 形式）

```
{
  "schemas": [
    "urn:ietf:params:scim:api:messages:2.0:ListResponse"
  ],
  "totalResults": 2,
  "Resources": [
    {
      "schemas": [
        "urn:ietf:params:scim:schemas:core:2.0:ResourceType"
      ],
      "id": "User",
      "name": "User",
      "endpoint": "/Users",
      "description": "User Account",
      "schema": "urn:ietf:params:scim:schemas:core:2.0:User",
      "schemaExtensions": [
        {
          "schema": "urn:ietf:params:scim:schemas:extension:enterprise:2.0:User",
          "required": true
        }
      ]
    },
    {
      "schemas": [
        "urn:ietf:params:scim:schemas:core:2.0:ResourceType"
      ],
      "id": "Group",
      "name": "Group",
      "endpoint": "/Groups",
      "description": "Group",
      "schema": "urn:ietf:params:scim:schemas:core:2.0:Group"
    }
  ]
}
```

▼ レスポンスボディ

#	項目名	型	内容	説明
1	schemas	文字列	応答時のスキーマタイプ	"urn:ietf:params:scim:api:messages:2.0:ListResponse" 固定
2	totalResults	整数	検索結果件数	リソースの件数
3	Resources	配列	リソース情報リスト	詳細は以下「リソース情報」を参照

▼ **レスポンスボディ（リソース情報）**

#	項目名	型	内容	説明	
				User の場合	Group の場合
1	id	整数	ID	“User”固定	"Group"固定
2	name	文字列	名称	“User”固定	"Group"固定
3	endpoint	文字列	エンドポイント	“/Users""固定	“/Groups”固定
4	description	文字列	説明	"User Account"固定	"Group"固定
5	schemaExtensions	配列	スキーマ拡張のリスト	詳細は以下「スキーマ拡張」を参照	なし
6	schema	文字列	スキーマ	"urn:ietf:params:scim:schemas:core:2.0:User"固定	"urn:ietf:params:scim:schemas:core:2.0:Group"固定
7	schemas	文字列	スキーマ	"urn:ietf:params:scim:schemas:core:2.0:ResourceType"固定	

▼ **レスポンスボディ（スキーマ拡張）**

#	項目名	型	内容	説明
1	schema	文字列	拡張スキーマの URI	"urn:ietf:params:scim:schemas:extension:enterprise:2.0:User " 固定
2	required	論理	スキーマ拡張フラグ	リソースタイプにスキーマ拡張が必要かどうかを示すフラグ。

6. スキーマ定義API

この章では、リソースのスキーマの取得 API について説明します。

6.1. スキーマ定義取得

SCIM サーバーがサポートするスキーマ（データモデル）を定義し、それに関連する属性やメタデータを提供するためのエンドポイントです。

▼ メソッドタイプ
GET

▼ リクエスト URL 例

```
https://sample.atled.jp/AgileWorks/Broker/SCIM/Schemas
```

▼レスポンスボディ例（JSON 形式）

```
{
  "schemas": [
    "urn:ietf:params:scim:api:messages:2.0:ListResponse"
  ],
  "totalResults": 2,
  "Resources": [
    {
      "id": "urn:ietf:params:scim:schemas:core:2.0:User",
      "name": "User",
      "description": "User",
      "attributes": [
        {
          "name": "id",
          "type": "integer",
          "multiValued": false,
          "description": "id",
          "mutability": "readOnly",
          "required": false,
          "caseExact": false,
          "uniqueness": "server",
          "returned": "always"
        },
        {
          "name": "externalId",
          "type": "string",
          "multiValued": false,
          "description": "externalId",
          "mutability": "readWrite",
          "required": true,
          "caseExact": false,
          "uniqueness": "server",
          "returned": "always"
        },
        {
          "name": "userName",
          "type": "string",
          "multiValued": false,
          "description": "userName",
          "mutability": "readWrite",
          "required": true,
          "caseExact": false,
          "uniqueness": "server",
          "returned": "always"
        },
        {
          "name": "displayName",
```

```
        "type": "string",
        "multiValued": false,
        "description": "displayName",
        "mutability": "readWrite",
        "required": true,
        "caseExact": false,
        "uniqueness": "none",
        "returned": "always"
      },
      {
        "name": "locale",
        "type": "string",
        "multiValued": false,
        "description": "locale",
        "mutability": "writeOnly",
        "required": false,
        "caseExact": false,
        "uniqueness": "none",
        "returned": "never"
      },
      {
        "name": "active",
        "type": "boolean",
        "multiValued": false,
        "description": "active",
        "mutability": "readWrite",
        "required": false,
        "caseExact": false,
        "uniqueness": "none",
        "returned": "always"
      },
      {
        "name": "password",
        "type": "string",
        "multiValued": false,
        "description": "password",
        "mutability": "writeOnly",
        "required": false,
        "caseExact": true,
        "uniqueness": "none",
        "returned": "never"
      },
      {
        "name": "emails",
        "type": "complex",
        "multiValued": false,
        "description": "emails",
        "mutability": "readWrite",
        "required": false,
        "caseExact": false,
        "uniqueness": "none",
        "returned": "always"
      },
      {
        "name": "emails.value",
        "type": "string",
        "multiValued": false,
        "description": "emails.value",
        "mutability": "readWrite",
        "required": false,
        "caseExact": false,
        "uniqueness": "none",
        "returned": "always"
      },
      {
        "name": "emails.primary",
        "type": "boolean",
        "multiValued": false,
```

```
          "description": "emails.primary",
          "mutability": "readWrite",
          "required": false,
          "caseExact": false,
          "uniqueness": "none",
          "returned": "always"
        },
        {
          "name": "groups",
          "type": "complex",
          "multiValued": false,
          "description": "groups",
          "mutability": "readWrite",
          "required": false,
          "caseExact": false,
          "uniqueness": "none",
          "returned": "always"
        },
        {
          "name": "groups.value",
          "type": "string",
          "multiValued": false,
          "description": "groups.value",
          "mutability": "readWrite",
          "required": false,
          "caseExact": false,
          "uniqueness": "server",
          "returned": "always"
        },
        {
          "name": "groups.display",
          "type": "string",
          "multiValued": false,
          "description": "groups.display",
          "mutability": "readWrite",
          "required": false,
          "caseExact": false,
          "uniqueness": "none",
          "returned": "always"
        },
        {
          "name": "roles",
          "type": "complex",
          "multiValued": false,
          "description": "roles",
          "mutability": "readWrite",
          "required": false,
          "caseExact": false,
          "uniqueness": "none",
          "returned": "always"
        },
        {
          "name": "roles.value",
          "type": "string",
          "multiValued": false,
          "description": "roles.value",
          "mutability": "readWrite",
          "required": false,
          "caseExact": false,
          "uniqueness": "server",
          "returned": "always"
        },
        {
          "name": "roles.display",
          "type": "string",
          "multiValued": false,
          "description": "roles.display",
          "mutability": "readWrite",
```

```
          "required": false,
          "caseExact": false,
          "uniqueness": "none",
          "returned": "always"
        }
      ]
    },
    {
      "id": "urn:ietf:params:scim:schemas:core:2.0:Group",
      "name": "Group",
      "description": "Group",
      "attributes": [
        {
          "name": "id",
          "type": "integer",
          "multiValued": false,
          "description": "id",
          "mutability": "readOnly",
          "required": false,
          "caseExact": false,
          "uniqueness": "server",
          "returned": "always"
        },
        {
          "name": "externalId",
          "type": "string",
          "multiValued": false,
          "description": "externalId",
          "mutability": "readWrite",
          "required": true,
          "caseExact": false,
          "uniqueness": "server",
          "returned": "always"
        },
        {
          "name": "displayName",
          "type": "string",
          "multiValued": false,
          "description": "displayName",
          "mutability": "readWrite",
          "required": true,
          "caseExact": false,
          "uniqueness": "none",
          "returned": "always"
        }
      ]
    }
  ]
}
```


▼ レスポンスボディ

#	項目名	型	内容	説明
1	schemas	文字列	応答時のスキーマタイプ	"urn:ietf:params:scim:api:messages:2.0:ListResponse" 固定
2	totalResults	整数	検索結果件数	リソースの件数
3	Resources	配列	リソース情報	詳細は以下「リソース情報」を参照。

▼ レスポンスボディ（リソース情報）

#	項目名	型	内容	説明	
				User の場合	Group の場合
1	id	整数	ID	“urn:ietf:params:scim:schemas:core:2.0:User”固定	“urn:ietf:params:scim:schemas:core:2.0:Group”固定
2	name	文字列	名称	“User”固定	"Group"固定
3	description	文字列	説明	“User”固定	"Group"固定
4	attributes	配列	属性	詳細は以下「属性情報」を参照。	

▼ レスポンスボディ（属性情報）

#	項目名	型	内容	説明
1	name	文字列	属性名称	属性の名前
2	description	文字列	説明	
3	multiValued	論理	属性タイプ	属性が複数の値を持つかどうかを示す属性
4	type	文字列	データ型	属性のタイプ（型）を示す属性 "boolean" "decimal" "integer" "dateTime" "reference" "complex"
5	required	論理	必須	属性が必須かどうかを指定するブール値
6	mutability	文字列	可変性	属性に関する変更可能性 mutable（変更可能） immutable（変更不可能） read-only（読み取り専用）
7	caseExact	論理	大文字/小文字の区別	属性が大文字と小文字を区別するかどうかを指定するブール値
8	returned	文字列	戻り値	属性が検索結果に含まれるかどうかを示す属性 always（常に返される） default（デフォルトで返される） never（決して返されない）
9	uniqueness	文字列	独自性	属性における一意性（uniqueness）の制約を定義するための属性 server（サーバー管理の一意性） global（グローバルー意性） none（一意性の制約なし）

▼リソース定義

①ユーザ

#	SCIM 属性 (name) (description)	属性タイプ (multiValued)	データ型 (type)	必須 (required)	可変性 (mutability)	大文字/小文字の区別 (caseExact)	戻り値 (returned)	独自性 (uniqueness)
1	id	FALSE	integer	FALSE	readOnly	FALSE	always	server
2	externalId	FALSE	string	TRUE	readWrite	FALSE	always	server
3	userName	FALSE	string	TRUE	readWrite	FALSE	always	server
4	displayName	FALSE	string	TRUE	readWrite	FALSE	always	none
5	locale	FALSE	string	FALSE	readOnly	FALSE	always	none
6	active	FALSE	boolean	FALSE	readWrite	FALSE	always	none
7	password	FALSE	string	FALSE	readOnly	TRUE	never	none
8	emails	FALSE	complex	FALSE	readWrite	FALSE	always	none
9	emails.value	FALSE	string	FALSE	readWrite	FALSE	always	none
10	emails.primary	FALSE	Boolean	FALSE	readWrite	FALSE	always	none
11	groups	FALSE	complex	FALSE	readWrite	FALSE	always	none
12	groups.value	FALSE	string	FALSE	readWrite	FALSE	always	server
13	groups.display	FALSE	string	FALSE	readWrite	FALSE	always	none
14	roles	FALSE	complex	FALSE	readWrite	FALSE	always	none
15	roles.value	FALSE	string	FALSE	readWrite	FALSE	always	server
16	roles.display	FALSE	string	FALSE	readWrite	FALSE	always	none

②グループ

#	SCIM 属性 (name) (description)	属性タイプ (multiValued)	データ型 (type)	必須 (required)	可変性 (mutability)	大文字/小文字の区別 (caseExact)	戻り値 (returned)	独自性 (uniqueness)
1	id	FALSE	integer	FALSE	readOnly	FALSE	always	server
2	externalId	FALSE	string	TRUE	readWrite	FALSE	always	server
3	displayName	FALSE	string	TRUE	readWrite	FALSE	always	none

7. サービス定義

この章では、サービス・プロバイダの構成の取得 API について説明します。

7.1. サービス定義取得

SCIM サーバがサポートするサービスプロバイダ構成情報に関する情報を提供するためのエンドポイントです。

▼ メソッドタイプ
GET

▼ リクエスト URL 例

```
https://sample.atled.jp/AgileWorks/Broker/SCIM/ServiceProviderConfigs
```

▼レスポンスボディ例（JSON 形式）

```
{
  "schemas": [
    "urn:ietf:params:scim:schemas:core:2.0:ServiceProviderConfig"
  ],
  "patch": {
    "supported": true
  },
  "bulk": {
    "supported": false,
    "maxOperations": 0,
    "maxPayloadSize": 0
  },
  "filter": {
    "supported": true,
    "maxResults": 1000
  },
  "changePassword": {
    "supported": true
  },
  "sort": {
    "supported": false
  },
  "etag": {
    "supported": false
  },
  "authenticationSchemes": [
    {
      "type": "oauth2",
      "name": "oauth2",
      "description": "oauth2",
      "specUri": "https://datatracker.ietf.org/doc/html/rfc6749",
      "primary": true
    },
    {
      "type": "httpbasic",
      "name": "httpbasic",
      "description": "httpbasic",
      "specUri": "https://datatracker.ietf.org/doc/html/rfc7617",
      "primary": false
    }
  ]
}
```

▼ **レスポンスボディ**

#	項目名	型	内容	説明
1	schemas	文字列	スキーマ情報	"urn:ietf:params:scim:schemas:core:2.0:ServiceProviderConfig " 固定
2	patch	構造体	パッチ	
3	supported	論理	サポート有無	“true”固定
4	bulk	構造体	バルク	
5	supported	論理	サポート有無	“false”固定
6	maxOperations	整数	最大操作数	バルクリクエスト内で許可される操作の最大数を示します。 0 固定
7	maxPayloadSize	整数	最大サイズ	バルクリクエストのペイロード（データのサイズ）に許可される最大サイズを示します。 0 固定
8	filter	構造体	フィルター	
9	supported	論理	サポート有無	“true”固定
10	maxResults	整数	最大検索数	1000 固定
11	changePassword	構造体	パスワード変更有無	
12	supported	論理	サポート有無	“true”固定
13	sort	構造体	ソート	
14	supported	論理	サポート有無	“false”固定
15	etag	構造体	etag	
16	supported	論理	サポート有無	“false”固定
17	authenticationSchemes	配列	認証スキーマ	詳細は以下「認証スキーマ情報」を参照。

▼ **レスポンスボディ（認証スキーマ情報）**

#	項目名	型	内容	説明	
				OAuth2 認証の場合	Basic 認証の場合
1	type	文字列	認証スキーマ種別	“oauth2”固定	“httpbasic”固定
2	name	文字列	認証名称	〃	〃
3	description	文字列	認証の説明	〃	〃
4	specUri	文字列	認証仕様 URI	“https://datatracker.ietf.org/doc/html/rfc6749”固定	“https://datatracker.ietf.org/doc/html/rfc7617”固定
5	primary	論理	優先	“TRUE”固定	“FALSE”固定