



# Technische Referenz

*System for Cross-Domain Identity  
Management im Kontext  
Provisionierung*

*Februar 2024*

# Inhaltsverzeichnis

Vorwort.....	3
Leserkreis.....	3
Bezugsdokumente.....	3
Typografische Konventionen.....	3
Symbol Konventionen.....	3
Einführung.....	5
Funktionsweise.....	5
Begrifflichkeit.....	5
Modell.....	6
Repräsentation eines Benutzers.....	7
Repräsentation einer Gruppe.....	7
Schema.....	8
Attributdefinition.....	8
Datentypen.....	9
Operationen.....	10
Request Methoden.....	10
Request Header.....	10
Suchen (Search).....	10
Request.....	10
Response.....	11
Abrufen (WellKnown).....	12
Request.....	12
Response.....	13
Erzeugen (Create).....	14
Request.....	14
Response.....	15
Ersetzen (Replace).....	15
Request.....	15
Response.....	16
Ändern (Update).....	16
Request.....	16
Parameter.....	16
Response.....	17
Löschen (Delete).....	17
Request.....	17
Response.....	17
@artus.....	18
Verwaltung von Anwendern.....	18
Abfrage von Benutzerkonten.....	18
Anfragen.....	18
Request Header.....	18
Parameter.....	18
Antwort.....	19
Resource.....	20
Metadaten.....	21
Erweiterung.....	22
Datenmodell.....	22
Schema.....	22
Endpunkte.....	22

Operationen.....	23
Benutzer.....	23
Gruppen.....	23
Tenants.....	23
Berechtigung zuweisen.....	24
Berechtigung entziehen.....	24

# Vorwort

## Leserkreis

Dieses Dokument wendet sich an Personen, die sich mit der Administration von Ressourcen, sowie Teams, die sich mit der Integration von Zielsystemen, in Oracle Identity Governance befassen.

## Bezugsdokumente

SCIM 2.0 wurde im September 2015 als RFC7642, RFC7643 und RFC7644 durch IETF veröffentlicht.

- [RFC7642 - SCIM: Definitions, Overview, Concepts, and Requirements](#)  
In diesem Dokument werden die Szenarien und Anwendungsfälle des Systems für Cross-Domain Identity Management (SCIM) aufgeführt.
- [RFC7643 - SCIM: Core Schema](#)  
Das Core Schema definiert ein plattformneutrales Schema und und Modell zur Erweiterung für die Darstellung von Benutzern und Gruppen.
- [RFC7644 - SCIM: Protocol](#)  
Das SCIM-Protokoll ist ein REST-Protokoll auf Anwendungsebene zur Bereitstellung und Verwaltung von Identitätsdaten.
- [RFC4627- MediaType application/json](#)  
Der Medientyp application/json für JavaScript Object Notation (JSON).

## Typografische Konventionen

Die folgenden typografischen Konventionen werden in diesem Dokument verwendet:

Konvention	Bedeutung
<b>Fettdruck</b>	Fettdruck kennzeichnet Elemente der grafischen Benutzeroberfläche, die einer Aktion zugeordnet sind, oder Begriffe, die im Text oder im Glossar definiert sind.
<i>Kursiv</i>	Kursivschrift kennzeichnet Buchtitel, Hervorhebungen oder Platzhalter, für die Sie bestimmte Werte angeben.
Monospace	Monospace in einem Absatz kennzeichnet Befehle, URLs, Code-Beispiele, Text, der auf dem Bildschirm angezeigt wird, oder Text, den Sie eingeben.

## Symbol Konventionen

In diesem Dokument werden die folgenden Konventionen für Symbole verwendet.

Konvention	Bedeutung
[ ]	Enthält optionale Argumente und Befehlsoptionen.
{   }	Enthält eine Reihe von Auswahlmöglichkeiten für eine erforderliche Befehlsoption.
\${ }	Referenziert eine Variable.
-	Verbindet gleichzeitig mehrere Tastenanschläge.
+	Verbindet mehrere aufeinanderfolgende Tastenanschläge.
>	Zeigt die Auswahl eines Menüpunkts in der grafischen Benutzeroberfläche an.

# Einführung

Die System for Cross-Domain Identity Management (SCIM)-Spezifikation (ursprünglich Simple Cloud Identity Management) soll die Verwaltung von Benutzeridentitäten in Anwendungen und Diensten erleichtern. Die Spezifikation zielt darauf ab, mit definierten Schemata die Bereitstellungen von Benutzerkonten aufzubauen. Dabei wurde besonderer Wert auf die Einfachheit der Entwicklung und Integration gelegt wird und gleichzeitig bestehende Modelle zur Authentifizierung, Autorisierung und dem Datenschutz angewendet. Ziel ist es, die Kosten und Komplexität von Vorgängen der Benutzerverwaltung durch die Bereitstellung eines allgemeinen Benutzerschemas und entsprechenden Erweiterung sowie durch die Bindung von Dokumenten zu reduzieren, um Muster für den Austausch dieses Schemas mithilfe von Standardprotokollen bereitzustellen.

## Funktionsweise

SCIM ist ein Protokoll auf der Grundlage von REST und JSON, das eine Client- und eine Server-Rolle definiert. Der Client ist in der Regel ein Identity Provider (IdP) wie Oracle Identity Governance mit einer Datenbank von Benutzeridentitäten. Der Service Provider (SP) ist in der Regel eine Applikation wie Box oder Slack, die eine Teilmenge an Informationen von diesen Identitäten benötigt. Wenn beim IdP Änderungen wie das Erstellen, Aktualisieren und Löschen von Identitäten vorgenommen werden, werden sie automatisch über das SCIM-Protokoll mit dem SP synchronisiert. Der IdP kann auch Identitäten vom SP übernehmen, um seinen Datenbestand zu ergänzen und falsche Werte auf SP-Seite zu erkennen, die zu Sicherheitslücken führen könnten. Für den Endanwender bedeutet dies, dass er nahtlosen Zugriff auf die Anwendungen hat, die ihm zugewiesen sind, wobei diese über aktuelle Profile und Berechtigungen verfügen.

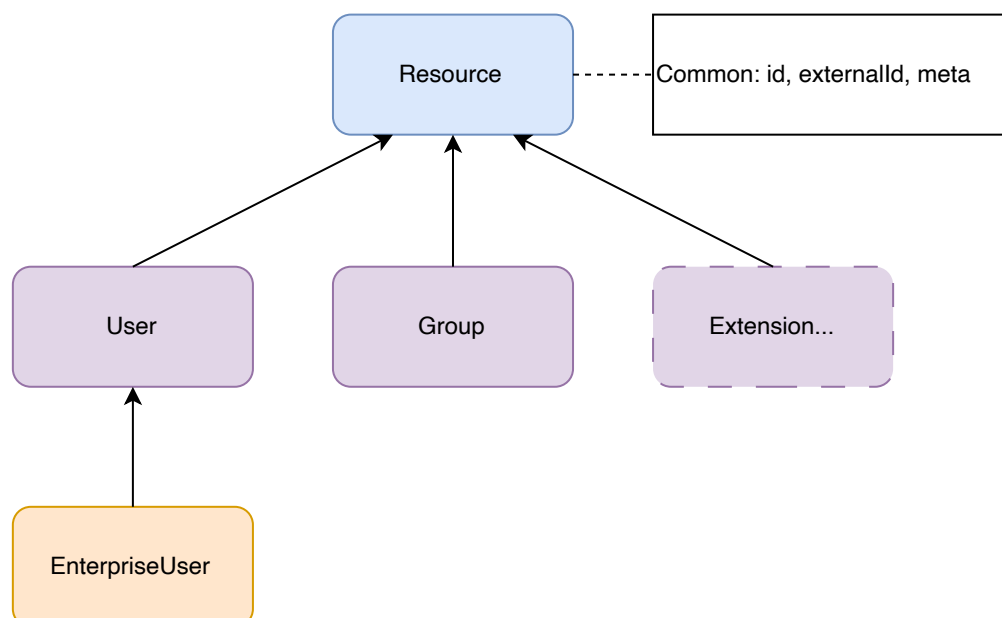
## Begrifflichkeit

- **Service Provider**  
Eine REST-fähige Anwendung, die Informationen zu Identitäten via SCIM Protokoll zur Verfügung stellt.
- **Client**  
Eine Anwendung, die über SCIM Identitätsdaten verwaltet, die ein Service Provider vorhält (schickt diesem SCIM HTTP Anfragen).
- **Ressource**  
SCIM 2.0 basiert auf einem Objektmodell. Als gemeinsamer Nenner gilt die Ressource, von der alle anderen Objekte abgeleitet werden.
- **Resource Type**  
Die Art der Ressource.  
Spezifiziert den Namen von Ressourcen, Endpunkten, URL, Schemas, und andere
- **Schema**  
Sammlung von Attributdefinitionen, die den Inhalt von Ressourcen beschreiben, z.B.  
"urn:ietf:params:scim:schemas:core:2.0:User"

# Modell

SCIM 2.0 basiert auf einem Objektmodell. Als gemeinsamer Nenner gilt die Ressource, von der alle anderen Objekte abgeleitet werden.

- Resource ist ein JSON Object
- Es besteht aus einem oder mehreren Attributen.
- Diese Attribute können zu einem oder mehreren Schemas gehören



Entsprechend der Vererbung ist **User** in SCIM eine **Resource**.

Jedes Benutzerprofil in einem System verfügt über einen Namen, eine Adresse, Telefonnummern, E-Mail-Adressen usw. Diese gemeinsamen Attribute gelten für jedes Benutzerprofil, das von einem beliebigen System verwaltet wird. Eine Sammlung solcher allgemeiner Benutzerattribute wird als Benutzerressource bezeichnet. Es kann jedoch Situationen geben, in denen die Verwaltung system-/anwendungsspezifischer Attribute für Benutzer erforderlich ist (z. B. Benutzer, die Geräte für die Anwendung anmelden). Aufgrund dieser Tatsache führt SCIM den **EnterpriseUser** als Erweiterung für die Benutzerressource ein.

**EnterpriseUser** ist eine Erweiterung des User-Attributs, das eine Sammlung anwendungsspezifischer Attribute enthält.

Darüber hinaus sind „id“, „externalId“ und „meta“ drei allgemeine Attribute, die in jeder SCIM-Ressource enthalten sind, mit Ausnahme der Ressourcen wie „ServiceProviderConfig“ und „ResourceType“, die zur Ermittlung von SCIM-Diensteanbietern und anderen mit der Erkennung der speziellen Eigenschaften von Diensteanbietern verbundenen Ressourcen verwendet werden.

Attribute	Bedeutung
id	Eine eindeutige Kennung für eine SCIM-Ressource, die vom SCIM-Diensteanbieter definiert wird. Es wird vom SCIM-Diensteanbieter beim Erstellen der Ressource generiert. Es handelt sich um ein <b>ERFORDERLICHES</b> Attribut.

Attribute	Bedeutung
externalId	ine Zeichenfolge, die eine Kennung für die Ressource darstellt, die vom SCIM-Client generiert wird. Mit anderen Worten: Kennung der ursprünglichen Quelle, aus der die Benutzerdaten stammen (z. B. Datenbank-ID, aus der der Benutzer vom SCIM-Client entnommen wird). Es handelt sich um ein <b>OPTIONAL</b> -Attribut.
meta	Allgemeine Metadaten für eine Ressource. Sie werden vom SCIM-Dienstanbieter beim Erstellen der Ressource generiert. Zum Beispiel: Zeitstempel der Erstellung, Zeitstempel der letzten Änderung, Standort der Ressource. Es handelt sich um ein <b>ERFORDERLICHES</b> Attribut.

## Repräsentation eines Benutzers

Dies ist ein Beispiel dafür, wie Benutzerdaten als SCIM-Objekt in JSON kodiert werden können.

```
{ "schemas": [
  "urn:ietf:params:scim:schemas:core:2.0:User"
],
  "id": "2819c223-7f76-453a-919d-413861904646",
  "externalId": "4711",
  "meta": {
    "resourceType": "User",
    "created": "2011-08-01T18:29:49.793Z",
    "lastModified": "2011-08-01T18:29:49.793Z",
    "location": "https://service-provider/v2/Users/2819c223...",
    "version": "W\\/\\"f250dd84f0671c3\\""}
  },
  "name": {
    "formatted": "Mr. Alfons Zitterbacke, III",
    "familyName": "Zitterbacke",
    "givenName": "Alfons",
    "honorificPrefix": "Mr.",
    "honorificSuffix": "III"
  },
  "userName": "azitterbacke",
  "phoneNumbers": [
    { "primary": true,
      "type": "work",
      "value": "555-555-8377"
    }
  ],
  "emails": [
    { "primary": true,
      "type": "work",
      "value": "azitterbacke@somewhere.com"
    }
  ]
}
```

## Repräsentation einer Gruppe

Zusätzlich zu den Benutzern umfasst SCIM auch die Definitionen von Gruppen. Gruppen werden verwendet, um die Organisationsstruktur bereitgestellter Ressourcen zu modellieren. Gruppen können Benutzer oder andere Gruppen enthalten.



```

{ "schemas": [
  "urn:ietf:params:scim:schemas:core:2.0:Group"
],
  "id"      : "e9e30dba-f08f-4109-8486-d5c6a331660a"
, "displayName" : "Clerk"
, "members" : [
  { "value" : "2819c223-7f76-453a-919d-413861904646"
    , "$ref" : "https://service-provider/v2/Users/2819c223..."
    , "display" : " Alfons Zitterbacke"
  }
  , { "value" : "902c246b-6245-4190-8e05-00816be7344a"
    , "$ref" : "https://service-provider/v2/Users/902c246b..."
    , "display" : "Agathe Musterfrau"
  }
],
  "meta": {
    "resourceType" : "Group",
    "created"      : "2010-01-23T04:56:22Z",
    "lastModified" : "2011-05-13T04:42:34Z",
    "version"      : "W\//"3694e05e9dff592\"",
    "location"     : "\"https://service-provider/v2/Groups/e9e30dba..."
  }
}

```

## Schema

Schema ist eine Sammlung von Attributdefinitionen, die den Inhalt einer gesamten oder eines Teils der Ressource beschreiben.

Jede Darstellung von Daten ist über JSON definiert.

## Attributdefinition

Attributdefinition bedeutet, wie Attribute definiert werden. SCIM-Attributdefinitionen enthalten die folgenden Details:

Type	Bedeutung
Singular Attribute	Attribute einer Ressource, das 0..1 Werte enthalten kann, z.B. „displayName“.
Multi-valued Attribute	Attribute einer Ressource, das 0..n Werte enthalten kann, z.B.: „emails“.
Simple Attribute	Attribute einer Ressource (singular oder multi-valued), dessen Werte einen einfachen Datentyp wie z.B. „String“ hat.
Complex Attribute	Attribute einer Ressource (singular oder multi-valued), dessen Werte aus einer Sammlung von einem oder mehrere Simple Attributes besteht, z.B.: „adresses“.
Sub-Attribute	Ein Simple Attribute, das in einem Complex Attribute enthalten ist.

**Beispiel:**

```
{
  "name"      : "userName"
, "type"      : "string"
, "multiValued" : false
, "description" : "Unique identifier for the User, typically used by ..."
, "required"   : true
, "caseExact"  : false
, "mutability" : "readWrite"
, "returned"   : "default"
, "uniqueness" : "server"
}
```

## Datentypen

Es gibt eine festgelegte Anzahl von Datentypen. SCIM-Erweiterungen sollen keine neue Datentypen definieren.

Es werden konform zu RFC4627 werden folgende Datentypen spezifiziert:

Type	Bedeutung
String	0..n Unicode Zeichen.
Boolean	Literale „true“ oder „false“
Decimal	Reale Zahl mit mindestens einer Stelle hinter dem Dezimalpunkt, z.B. „3.4“ [=3,4]
Integer	Dezimalzahl ohne gebrochenen Anteil.
DateTime	Datum/Zeit-Angabe im Format: 2008-01-23T04:56:22Z
Binary	Beliebige Binäre Daten, base64-kodiert
Reference (auch \$ref)	Eine Referenz zu einer SCIM-Ressource als absolute oder relative URI, z.B.:
Complex	Attribute einer Ressource (singular oder multi-valued), dessen Werte aus einer Sammlung von einem oder mehrere Simple Attributes besteht.

Wenn nicht anders spezifiziert gelten folgende Standards für Attribut-Datentypen:

- optional ("required=false")
- case insensitive ("caseExact=false")
- veränderbar ("mutability=readWrite")
- werden bei Anfragen zurückgegeben ("returned=default")
- sind nicht eindeutig ("uniqueness=none")
- sind vom Datentyp String

# Operationen

Für die Manipulation von Ressourcen stellt SCIM eine REST-API mit einer umfangreichen, aber einfachen Reihe von Operationen bereit, die alles vom Patchen eines bestimmten Attributs für einen bestimmten Benutzer bis hin zur Durchführung umfangreicher Massenaktualisierungen unterstützen:

## Request Methoden

Operation	Methode	Beispiel
Search	GET	<code>https://{service-provider}/{v}/{resource}?filter={attribute}{op}{value}&amp;sortBy={attributeName}&amp;sortOrder={ascending descending}</code>
WellKnown	GET	<code>https://{service-provider}/{v}/{resource}/{id}</code>
Create	POST	<code>https://{service-provider}/{v}/{resource}</code>
Replace	PUT	<code>https://{service-provider}/{v}/{resource}/{id}</code>
Update	PATCH	<code>https://{service-provider}/{v}/{resource}/{id}</code>
Delete	DELETE	<code>https://{service-provider}/{v}/{resource}/{id}</code>

## Request Header

Name	Type	Wert
accept	String	application/scim+json
content-type	String	<code>https://{service-provider}/{v}/{resource}/{id}</code>
authorization	String	Bearer mit einem entsprechend autorisiertem Token

## Suchen (Search)

### Request

```
GET /Users?startIndex=1&count=10
Accept: application/scim+json
Content-Type: application/scim+json
Authorization: Bearer h480djs93hd8.....
```

### Parameter

Name	Type	Erforderlich	Beschreibung
startIndex	Integer	nein	Der 1-basierte Index des ersten

Name	Type	Erforderlich	Beschreibung
			zurückgegebenen Ergebnisses der Abfrage.  Ein Wert kleiner als 1 wird als 1 interpretiert.
count	Integer	nein	Gibt die gewünschte maximale Anzahl von Ergebnissen der Anfrage pro Batch an.  Ein negativer Wert wird als 0 interpretiert.  Der Wert 0 gibt an, dass keine Ergebnisse zurückgegeben werden sollen.
filter	String	nein	Die Filterkriterien, die auf die Abfrage angewendet werden sollen.
sortBy	String	nein	Die auf die Suche anzuwendenden Sortierkriterien.
sortOrder	String	nein	Die Reihenfolge, in der die Sortierung durchgeführt wird, z. B. aufsteigende oder absteigende Sortierung.
attributes	String List	nein	Der Wert dieses Parameters ist eine durch Kommas getrennte Liste von Attributnamen einer Ressource, ausgedrückt in der Standardnotation von Attributen (z. B. id, userName).
excludedAttributes	String List	nein	Der Wert dieses Parameters ist eine durch Kommas getrennte Liste von Attributnamen einer Ressource, ausgedrückt in der Standardnotation von Attributen (z. B. id, userName).

## Response

```

HTTP/1.1 200 OK
{ "schemas" : [
  "urn:ietf:params:scim:api:messages:2.0:ListResponse"
]
, "totalResults" : 2
, "startIndex" : 1
, "itemsPerPage" : 100
, "Resources" : [
  { "id" : "2819c223-7f76-453a-919d-413861904646"
  , "userName" : "azitterbacke"
  }
  ...
  , { "id" : "c75ad752-64ae-4823-840d-ffa80929976c",
    "userName" : "amusterfrau"
  }
]
}

```

# Abrufen (WellKnown)

## Request

```
GET /Users/{id}
Accept: application/scim+json
Content-Type: application/scim+json
Authorization: Bearer h480djs93hd8.....
```

## Parameter

Name	Type	Erforderlich	Beschreibung
attributes	String List	nein	<p>Der Wert dieses Parameters ist eine durch Kommas getrennte Liste von Attributnamen einer Ressource, ausgedrückt in der Standardnotation von Attributen (z. B. id, userName).</p> <p>Ein Wert kleiner als 1 wird als 1 interpretiert.</p>
excludedAttributes	String List	nein	<p>Der Wert dieses Parameters ist eine durch Kommas getrennte Liste von Attributnamen einer Ressource, ausgedrückt in der Standardnotation von Attributen (z. B. id, userName).</p>

## Response

```
HTTP/1.1 200 OK
{ "schemas" : [
  "urn:ietf:params:scim:schemas:core:2.0:User"
]
, "id"      : "2819c223-7f76-453a-919d-413861904646"
, "meta" : {
  "resourceType" : "User"
, "version"      : "99"
, "created"      : "2011-08-01T21:32:44.882Z"
, "lastModified" : "2011-08-01T21:32:44.882Z"
, "location"     : "https://.../scim/v2/Users/2819c223-..."
}
, "userName": "azitterbacke"
, "active"   : true
, "name" : {
  "formatted" : "Zitterbacke, Alfons"
, "familyName" : "Zitterbacke"
, "givenName"  : "Alfons"
}
, "emails": [
  { "primary": true
  , "type"    : "work"
  , "value"   : "alfons.zitterbacke@somewhere.com",
  }
]
, "phoneNumbers": [
  { "primary": true
  , "type"    : "work"
  , "value"   : "+49 nn ...",
  }
, { "type"    : "fax"
  , "value"   : "+49 nn ...",
  }
]
}
```

# Erzeugen (Create)

## Request

```
POST: /Users
Accept: application/scim+json
Content-Type: application/scim+json
Authorization: Bearer h480djs93hd8.....
Content-Length: ...
{ "schemas" : [
    "urn:ietf:params:scim:schemas:core:2.0:User"
]
, "userName" : "azitterbacke"
, "name" : {
    "familyName" : "Zitterbacke"
  , "givenName" : "Alfons"
}
, "emails": [
    { "primary" : true
    , "type" : "work"
    , "value" : "alfons.zitterbacke@somewhere.com"
    }
]
, "phoneNumbers": [
    { "primary" : true
    , "type" : "work"
    , "value" : "+49 nn ..."
    }
  , { "type" : "fax"
    , "value" : "+49 nn ..."
    }
]
}
```

## *Parameter*

Dieser Aufruf erlaubt keine weiteren Parameter.

## Response

```
HTTP/1.1 200 OK
{ "schemas" : [
  "urn:ietf:params:scim:schemas:core:2.0:User"
]
, "id"      : "2819c223-7f76-453a-919d-413861904646"
, "meta" : {
  "resourceType" : "User"
, "version"      : "99"
, "created"      : "2011-08-01T21:32:44.882Z"
, "lastModified" : "2011-08-01T21:32:44.882Z"
, "location"     : "https://.../scim/v2/Users/2819c223-..."
}
, "userName": "azitterbacke"
, "active"   : true
, "name" : {
  "formatted" : "Zitterbacke, Alfons"
, "familyName" : "Zitterbacke"
, "givenName"  : "Alfons"
}
, "emails": [
  { "primary": true
  , "type"    : "work"
  , "value"   : "alfons.zitterbacke@somewhere.com",
  }
]
, "phoneNumbers": [
  { "primary": true
  , "type"    : "work"
  , "value"   : "+49 nn ...",
  }
, { "type"    : "fax"
  , "value"   : "+49 nn ...",
  }
]
}
```

## Ersetzen (Replace)

### Request

```
PUT /Users/2819c223-7f76-453a-919d-413861904646
Accept: application/scim+json
Content-Type: application/scim+json
Authorization: Bearer h480djs93hd8.....
Content-Length: ...
{ "schemas" : [
  "urn:ietf:params:scim:schemas:core:2.0:User"
]
, "id"      : "2819c223-7f76-453a-919d-413861904646"
, "emails" : [
  { "primary" : true
  , "type"    : "work"
  , "value"   : "alfons.zitterbacke@nowhere.com"
  }
]
}
```

### Parameter

Dieser Aufruf erlaubt keine weiteren Parameter.



## Response

```
HTTP/1.1 200 OK
{ "schemas" : [
  "urn:ietf:params:scim:schemas:core:2.0:User"
]
, "id"      : "2819c223-7f76-453a-919d-413861904646"
, "meta" : {
  "resourceType" : "User"
, "version"      : "99"
, "created"      : "2011-08-01T21:32:44.882Z"
, "lastModified" : "2011-08-01T21:32:44.882Z"
, "location"     : "https://.../scim/v2/Users/2819c223-..."
}
, "userName": "azitterbacke"
, "active"   : true
, "name" : {
  "formatted" : "Zitterbacke, Alfons"
, "familyName" : "Zitterbacke"
, "givenName"  : "Alfons"
}
, "emails": [
  { "primary": true
  , "type"    : "work"
  , "value"   : "alfons.zitterbacke@nowhere.com",
  }
]
, "phoneNumbers": [
  { "primary": true
  , "type"    : "work"
  , "value"   : "+49 nn ...",
  }
, { "type"    : "fax"
  , "value"   : "+49 nn ...",
  }
]
}
```

## Ändern (Update)

### Request

```
PATCH /Users/2819c223-7f76-453a-919d-413861904646
Accept: application/scim+json
Content-Type: application/scim+json
Authorization: Bearer h480djs93hd8.....
{ "schemas"      : ["urn:ietf:params:scim:api:messages:2.0:PatchOp"]
, "Operations"   : [
  { "op"         : "replace"
  , "path"       : "name.givenName"
  , "value"      : "Alfons"
  }
, { "op"         : "replace"
  , "path"       : "emails[type eq \"work\"].value"
  , "value"      : "alfons.zitterbacke@somewhere.com"
  }
]
}
```

### Parameter

Dieser Aufruf erlaubt keine weiteren Parameter.

## Response

```
HTTP/1.1 204 No Content
```

## Löschen (Delete)

### Request

```
DELETE /Users/2819c223-7f76-453a-919d-413861904646  
Accept: application/scim+json  
Content-Type: application/scim+json  
Authorization: Bearer h480djs93hd8...
```

### *Parameter*

Dieser Aufruf erlaubt keine weiteren Parameter.

## Response

```
HTTP/1.1 204 No Content
```

## Verwaltung von Anwendern

### Abfrage von Benutzerkonten

#### Anfragen

Beispiel:

```
curl --request GET \
  --location 'https://<service-host>:<service-port>/scim/v2/Users' \
  --header 'Accept: application/scim+json' \
  --header 'Content-Type: application/scim+json' \
  --header 'Authorization: Bearer <your-token>'
```

#### Request Header

Name	Type	Wert
accept	String	application/scim+json
content-type	String	https://{service-provider}/{v}/{resource}/{id}
authorization	String	Bearer mit einem entsprechend autorisiertem Token

#### Parameter

Name	Type	Erforderlich	Beschreibung
startIndex	Integer	nein	Der 1-basierte Index des ersten zurückgegebenen Ergebnisses der Abfrage.  Ein Wert kleiner als 1 wird als 1 interpretiert.
count	Integer	nein	Gibt die gewünschte maximale Anzahl von Ergebnissen der Anfrage pro Batch an.  Ein negativer Wert wird als 0 interpretiert.  Der Wert 0 gibt an, dass keine Ergebnisse zurückgegeben werden sollen.
filter	String	nein	Die Filterkriterien, die auf die Abfrage angewendet werden sollen.
sortBy	String	nein	Die auf die Suche anzuwendenden Sortierkriterien.
sortOrder	String	nein	Die Reihenfolge, in der die Sortierung durchgeführt wird, z. B. aufsteigende oder absteigende

Name	Type	Erforderlich	Beschreibung
Sortierung.			
attributes	String List	nein	Der Wert dieses Parameters ist eine durch Kommas getrennte Liste von Attributnamen einer Ressource, ausgedrückt in der Standardnotation von Attributen (z. B. id, userName).
excludedAttributes	String List	nein	Der Wert dieses Parameters ist eine durch Kommas getrennte Liste von Attributnamen einer Ressource, ausgedrückt in der Standardnotation von Attributen (z. B. id, userName).

## Antwort

Bei erfolgreichem Vorgang wird der Statuscode 200 zurückgegeben.

### Beispiel:

```

HTTP/1.1 200 OK
Content-Type: application/scim+json
{ "schemas" : [
  "urn:ietf:params:scim:api:messages:2.0:ListResponse"
]
, "totalResults"      : 65768
, "startIndex"       : 1
, "itemsPerPage"     : 500
, "Resources"        : [
  { "id"              : "2819c223-7f76-453a-919d-413861904646"
  , "userName"        : "azitterbacke"
  , "name"            : {
    "givenName"       : "Alfons"
    , "familyName"    : "Zitterbacke"
  }
  , "emails"          : [
    { "primary"       : true
    , "type"           : work
    , "value"          : ""
    }
  ]
  , "phoneNumbers"    : [
    { "primary"       : true
    , "type"           : work
    , "value"          : ""
    }
    , { "type"         : fax
    , "value"          : ""
    }
  ]
  ...
}
...
, { "id"              : "c75ad752-64ae-4823-840d-ffa80929976c"
  , "userName"        : "amusterfrau"
  ...
}
]
}

```

Die ausgegebenen Daten haben das folgende Format:

Name	Type	Erforderlich	Beschreibung
totalResults	Integer	ja	Nicht negative Ganzzahl.  Gibt die Gesamtzahl der Ergebnisse an, die mit der Abfrage übereinstimmen.
startIndex	Integer	ja	Der 1-basierte Index des ersten Ergebnisses im aktuellen Satz von Abfrageergebnissen.
itemsPerPage	Integer	ja	Nicht negative Ganzzahl. Gibt die Anzahl der Ergebnisse der Abfrage an, die auf einer Seite zurückgegeben werden.
schemas	String List	ja	Ist immer urn:ietf:params:scim:api:messages:2.0:ListResponse
Resources	Resource array	ja	Das Array von Ressourcen des aufgefüllten Ergebnissatzes.

## Resource

Name	Type	Erforderlich	Beschreibung
Id	String	ja	
userName	String	ja	
name	Complex	ja	
emails	E-Mail Resource array		
phoneNumbers	Telefonnummer Resource array		

## Name

Name	Type	Erforderlich	Beschreibung
familyName	String	ja	Nachname der Identität, der das Benutzerkonto gehört.
givenName	String	ja	Vorname der Identität, der das

Benutzerkonto gehört.
-----------------------

Email

Name	Type	Erforderlich	Beschreibung
primary	Boolean	ja	
type	String	ja	Kanonischer Werte, der die Verwendung der e-Mail Adresse definiert.  Gültig ist: work
value	String	Ja	Die e-mail Adresse.

Telefonnummern

Name	Type	Erforderlich	Beschreibung
primary	Boolean	ja	
type	String	ja	Kanonischer Werte, der die Verwendung der Telefonnummer definiert.  Gültig sind: work   fax
value	String	Ja	Die e Telefonnummer Adresse.

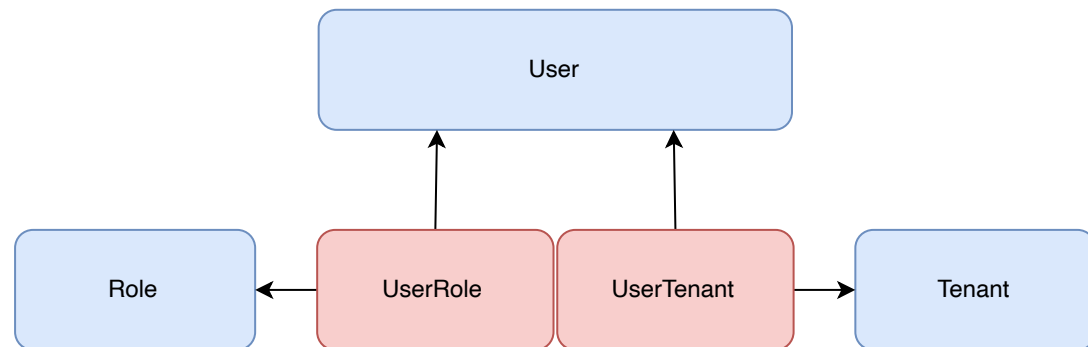
Metadaten

Name	Type	Erforderlich	Beschreibung
version	String		

# Erweiterung

## Datenmodell

Folgendes theoretisches Datenmodell, soll durch entsprechende Erweiterung des SCIM-Core Schemas einem hypothetischen Prozess zu Bereitstellung von Benutzerkonten und den damit verbundenen Zuweisungen und Entzug von Berechtigungsobjekten zugrunde gelegt werden.



## Schema

Um die Entität „Tenant“ innerhalb der Prozesse der Benutzerverwaltung wird eine Schemaerweiterung deklariert.

Somit exponiert der Service Provider folgende Beziehung zwischen dem Datenmodell und dem erforderlichen SCHIM-Schema:

Entität	Schema
Users	urn:ietf:params:scim:schemas:core:2.0:User
Role	urn:ietf:params:scim:schemas:core:2.0:Group
Tenant	urn:ietf:params:scim:schemas:extension:p20:1.0:Tenant

## Endpunkte

Folgende Endpunkte sind hierfür zu implementieren:

Endpunkt	Bedeutung
/Users	Exponiert die Standard-Operationen zur Suche und dem Anlegen, Ändern und Löschen von Benutzerkonten.
/Groups	Exponiert die Standard-Operationen zur Suche und der Zuweisung bzw. dem Entzug von Berechtigungsobjekten in Relation zu Benutzerkonten.
/Tenants	Exponiert die erweiterten Operationen zur Suche und der Zuweisung bzw. dem Entzug von Berechtigungsobjekten in Relation zu Benutzerkonten.

# Operationen

## Benutzer

- Das im Modell spezifizierte Entität „User“ entspricht dem im RFC7643 definiertem Schema-Objekt „User“.
- Die Operationen zur Suche von Benutzerkonten entsprechen den Standards gemäß [RFC7644 - SCIM: Protocol Abschnitt 3.4](#).
- Die Operationen zum Anlegen, Ändern und Löschen von Benutzerkonten werden gemäß den Standards [RFC7644 - SCIM: Protocol Abschnitt 3.3](#), [RFC7644 - SCIM: Protocol Abschnitt 3.5.1](#) und [RFC7644 - SCIM: Protocol Abschnitt 3.6](#) durch den entsprechenden Endpunkt exponiert.

## Gruppen

- Das im Modell spezifizierte Entität „Role“ entspricht dem im RFC7643 definiertem Schema-Objekt „Group“.
- Die Operationen zur Suche diese Berechtigungsobjekts entsprechen den Standards gemäß [RFC7644 - SCIM: Protocol Abschnitt 3.4](#).
- Die Operationen zum Anlegen und Löschen diese Berechtigungsobjekts gemäß den Standards [RFC7644 - SCIM: Protocol Abschnitt 3.3](#), [RFC7644 - SCIM: Protocol Abschnitt 3.5.1](#) und [RFC7644 - SCIM: Protocol Abschnitt 3.6](#) werden NICHT durch den entsprechenden Endpunkt exponiert.
- Die Operation Ändern diese Berechtigungsobjekts beschränkt sich auf die Zuweisung und den Entzug diese Berechtigungsobjekts in Relation zu Benutzerkonten und werden gemäß Standard [RFC7644 - SCIM: Protocol Abschnitt 3.5.2](#) exponiert.
- Die Zuweisung und der Entzug diese Berechtigungsobjekts erfolgt über den Endpunkt „/Groups“.

## Tenants

- Das im Modell spezifizierte Entität „Tenant“ wird konform zu RFC7643 als Schemaerweiterung „Tenant“.exponiert.
- Die Operationen zur Suche diese Berechtigungsobjekts entsprechen den Standards gemäß [RFC7644 - SCIM: Protocol Abschnitt 3.4](#).
- Die Operationen zum Anlegen und Löschen diese Berechtigungsobjekts gemäß den Standards [RFC7644 - SCIM: Protocol Abschnitt 3.3](#), [RFC7644 - SCIM: Protocol Abschnitt 3.5.1](#) und [RFC7644 - SCIM: Protocol Abschnitt 3.6](#) werden NICHT durch den entsprechenden Endpunkt exponiert.
- Die Operation Ändern diese Berechtigungsobjekts beschränkt sich auf die Zuweisung und den Entzug diese Berechtigungsobjekts in Relation zu Benutzerkonten und werden gemäß Standard [RFC7644 - SCIM: Protocol Abschnitt 3.5.2](#) exponiert.
- Die Zuweisung und der Entzug diese Berechtigungsobjekts erfolgt über den Endpunkt „/Tenants“.



## Berechtigung zuweisen

```
PATCH /Tenants/{id}
Accept: application/scim+json
Content-Type: application/scim+json
Authorization: Bearer h480djs93hd8.....
{ "schemas"      : ["urn:ietf:params:scim:api:messages:2.0:PatchOp"]
, "Operations"   : [
  { "op"         : "add"
  , "path"       : "roles"
  , "value"      : [
    { "type"     : "User"
    , "value"    : 5
    , "scope"    : "uid.generate"
    }
    , { "type"     : "User"
    , "value"    : 5
    , "scope"    : "uid.register"
    }
  ]
  }
]
```

## Berechtigung entziehen

```
PATCH /Tenants/{id}
Accept: application/scim+json
Content-Type: application/scim+json
Authorization: Bearer h480djs93hd8.....
{ "schemas"      : ["urn:ietf:params:scim:api:messages:2.0:PatchOp"]
, "Operations"   : [
  { "op"         : "remove"
  , "path"       : "roles[value eq 5 and scopes eq \"uid.generate\"]"
  }
  , { "op"         : "remove"
  , "path"       : "roles[value eq 5 and scopes eq \"uid.register\"]"
  }
]
```