IdentityIQ®

Introduction

Welcome to the SailPoint IdentityIQ API! The IdentityIQ API provides access to the IdentityIQ platform, allowing new opportunities for expanded innovation. The IdentityIQ API is standards- based, built upon the RESTful SCIM 2.0 specification. You can use our API to access IdentityIQ API endpoints, which allows you to programmatically interact with objects within IdentityIQ.
If you are looking for a SCIM connector, SailPoint offers both a SCIM 1.1 connector and a SCIM 2.0 connector. Please see Compass for more details on connectivity.

Getting Started

1. Read the IdentityIQ API Terms of Use
2. If you are unfamiliar with the SCIM 2.0 specification, or need a refresher, we suggest you start here: SCIM Overview.
3. Ensure you have IdentityIQ 7.0 Patch 2 or later versions installed.
4. Read our documentation. All you have to do is keep scrolling!
5. Participate on the forums. Ask questions, read about requested and upcoming functionality, and provide assistance to others.
6. Send us feedback! We want to hear from you.

Endpoint Availability

Endpoint	IdentityIQ Version
Core Schema	7.0 Patch 2 and later
/Users	7.0 Patch 2 and later
/Applications	7.1 and later
/Accounts	7.1 and later
/Entitlements	7.1 and later
/Roles	7.1 and later
/PolicyViolations	7.2 and later
/CheckedPolicyViolations	7.2 and later

Endpoint	IdentityIQ Version
/Workflows	7.2 and later
/LaunchedWorkflows	7.2 and later
/TaskResults	7.2 and later

SCIM Protocol

SCIM stands for System for Cross-Domain Identity Management, and it is an HTTP-based protocol that makes managing identities in multi-domain scenarios easier to support through a standardized RESTful API service. It provides a platform neutral schema and extension model for representing users, groups and other resource types in JSON format.

The core schema consists of five resource types, as described below:

Resource Type	Endpoint	Description	Supported Methods
User	/Users	Various attributes to describe the identity	GET, PUT, POST, DELETE
Group	/Groups	Currently not supported	N/A
Service Provider Configuration	/ServiceProviderConfig	Provides SCIM specification features and implementation details to clients	GET
Resource Types	/ResourceTypes	Defines attributes and metadata on supported resource types	GET
Schemas	/Schemas	Schema definitions and attributes for all schemas	GET

Supported HTTP Methods

GET POST PUT DELETE

Authentication

Basic Authentication

Beginning in IdentityIQ version 7.0, Patch 2, Basic Authentication is used to allow access to the API. Basic authentication is a simple technique for enforcing access controls to API resoureces because it doesn’t require session IDs, cookies, or login pages but instead uses standard fields in the HTTP header. For more information on Basic authentication, please see https://tools.ietf.org/html/rfc1945#section-11 and https://www.ietf.org/rfc/rfc2617.txt. Support for Basic Authentication will continue to exist in future releases.

OAuth 2.0

OAuth 2.0 Authentication is supported in IdentityIQ versions 7.1 and later. Versions prior to 7.1 only support Basic Authentication.

After configuring an OAuth2 API Credential you can access the token endpoint using your favorite client.

Sample Request

```
curl --user clientid:secret -d grant_type=client_credentials
"http://localhost:8080/identityiq/oauth2/token"
```

Sample Response

```
"expires_in":1200,"token_type":"bearer","access_token":"bHRiYWVUVk5ERzFrSjdzUHNFNUllWFFjM1NOTHZVbW0uODFyVkZlVC8rcnB1bVpGNHBVZ2grWWMrdVA0bk9idjJwMUhuTE83
```

Using the access_token value you can then make requests to any SCIM endpoint using Authorization: Bearer in the header.

Sample SCIM endpoint request

```
curl -H "Authorization: Bearer bHRiYWVUVk5ERzFrSjdzUHNFNUllWFFjM1NOTHZVbW0uODFyVkZlVC8rcnB1bVpGNHBVZ2grWWMrdVA0bk9idjJwMUhuTE83QzR3MUJWb2pCc2VFbU5LTnZXa
```

SCIM - Core Schema

/ServiceProviderConfig

Sample Request

```
curl
"http://localhost:8080/iiq/scim/v2/ServiceProviderConfig"
```

Sample Response (JSON)

```
{
"filter": {
"maxResults": 1000,
"supported": true
},
"patch": {
"supported": false
},
"authenticationSchemes": [
{
"documentationUri": "https://community.sailpoint.com/community/identityiq/product-downloads",
"name": "HTTP Basic",
"description": "Authentication Scheme using the Http Basic Standard",
"specUri": "http://www.ietf.org/rfc/rfc2617.txt",
"type": "httpbasic"
}
],
"documentationUri": "https://community.sailpoint.com/community/identityiq/product-downloads",
"meta": {
"location": "http://localhost:8080/iiq/scim/v2/ServiceProviderConfig",
"resourceType": "ServiceProviderConfig"
},
"schemas": [
"urn:ietf:params:scim:schemas:core:2.0:ServiceProviderConfig"
],
"etag": {
"supported": true
},
"sort": {
"supported": true
},
"bulk": {
"maxPayloadSize": 0,
"maxOperations": 0,
"supported": false
},
"changePassword": {
"supported": true
}
}
```

The SCIM 2.0 protocol provides a schema that represents the service provider's configuration. The service provider configuration gives the developer SCIM 2.0 specifications and additional implementation details in a standardized format. It is recommended that first time users make a call to /ServiceProviderConfig before using other endpoints. /ServiceProviderConfig is read-only and does not require any authentication.

HTTP REQUEST

```
GET http://example.com/identityIQ/scim/v2/ServiceProviderConfig
```

RESPONSE FORMAT

Parameter	Description	Default Value	Configurable
documentationURI	An HTTP-addressable URL pointing to the service provider's human-consumable help documentation.	https://community.sailpoint.com/community/identityiq/product-downloads	No
patch	A complex type that specifies whether the operation is supported. 'supported' is a boolean value indicating support for patch.	False	No
bulk	A complete type that specifies bulk configuration options. 'supported' is a boolean value indicating support for bulk. maxOperations and maxPayloadSize further specify bulk constraints	False	No
filter	A complex type that specifies filter options. 'supported' is a boolean value indicating support for filter. maxResults is an integer specifying the maximum # of results returned.	True - 1000 Max	Yes

Parameter	Description	Default Value	Configurable
changePassword	A complex type that specifies password change options. 'supported' is a boolean value indicating support for password change.	True	No
sort	A complex type that specifies sort options. 'supported' is a boolean value indicating support for sort.	True	No
etag	A complex type that specifies ETag options. 'supported' is a boolean value indicating support for ETag.	True	No
authenticationSchemes (multi-valued)	A multi-valued complex type that specifies supported authentication scheme properties.The following sub-attributes have been defined: type, name, description, specUri, and documentationUri.	HTTP Basic (OAuth 2.0 Support coming)	No

/Schemas

Sample Request

```
curl -u "<username>:<password>"
"http://localhost:8080/iiq/scim/v2/Schemas"
```

Sample Response (JSON)

```
{
"meta": {
"location": "",
"version": "",
"resourceType": "Schema"
},
"schemas": [
"urn:ietf:params:scim:schemas:core:2.0:Schema"
],
"name": "SailPoint User",
"description": "Additional attributes of the SailPoint User",
"attributes": [
{
"uniqueness": "none",
"name": "entitlements",
"description": "extended attribute description",
"mutability": "readOnly",
"type": "complex",
"multiValued": true,
"caseExact": false,
"returned": "default",
"required": false,
"subAttributes": [
{
"uniqueness": "none",
"name": "value",
"description": "The value of the entitlement",
"mutability": "readOnly",
"type": "string",
"multiValued": false,
"caseExact": false,
"returned": "request",
"required": false
}
]
}]
}
```

The /Schemas endpoint specifies defined attributes and characteristics of the core SCIM schema and all subsequent extended schemas. /Schemas is read-only.

HTTP REQUEST

`GET http://example.com/identityIQ/scim/v2/Schemas`

RESPONSE FORMAT

Parameter	Description	Default Value
id	Unique URI of the schema	Varies
name	Human-readable name of schema (e.g. “User”)	Varies
description	Human-readable description	Varies
attributes	Complex type with many sub-attributes including name, type, subAttributes, multiValued, description, required, and others. See section 7 of rfc7643 for full details.	

/ResourceTypes

Sample Request

```
curl -u "<user>:<password>"
"http://localhost:8080/identityiq/scim/v2/ResourceTypes"
```

Sample Response (JSON)

```
{
"totalResults": 1,
"schemas": [
"urn:ietf:params:scim:api:messages:2.0:ListResponse"
],
"Resources": [
{
"schema": "urn:ietf:params:scim:schemas:core:2.0:User",
"endpoint": "/Users",
"meta": {
"location": "http://localhost:8080/iiq/scim/v2/ResourceTypes/User",
"resourceType": "ResourceType"
},
"schemas": [
"urn:ietf:params:scim:schemas:core:2.0:ResourceType"
],
"name": "User",
"description": "User Account",
"schemaExtensions": [
{
"schema": "urn:ietf:params:scim:schemas:extension:enterprise:2.0:User",
"required": true
},
{
"schema": "urn:ietf:params:scim:schemas:sailpoint:1.0:User",
"required": true
}
],
"id": "User"
}
]
}
```

The /ResourceType endpoint provides metadata and details for endpoints. This includes information such as an resource ID, name, description, endpoint, base URI, schemas and schema extensions. /ResourceTypes is read-only.

HTTP REQUEST

```
GET http://example.com/identityIQ/scim/v2/ResourceTypes
```

RESPONSE FORMAT

Parameter	Description
id	Resource type’s server unique id
name	Name of resource type
description	Human-readable description of resource type
endpoint	The HTTP-addressable endpoint
schema	The primary/base schema URI. URI MUST match the id associated with the Schema resource.
schemaExtensions	A list of URIs for the resource types extentions.

Identity (/users)

The Identity endpoint allows implementors full read, write, and delete capabilities to the Identities within IdentityIQ. The Identity resource has most SCIM parameters and has been extended to support parameters that are specific to IdentityIQ. The API supports getting one or more identities, updating an identity, setting a password, deleting an identity and many other usecases. Please continue reading below for additional information.

RESPONSE FORMAT

Parameter	Description	Schema (SCIM or Extended)
id	Unique alpha-numeric ID of the identity	SCIM
externalId	An identifier for the resource as defined by the provisioning client	SCIM
userName	Unique identifier for the User typically used by the user to directly authenticate to the service provider	SCIM

Parameter	Description	Schema (SCIM or Extended)
name	Supports formatted, familyName (last name), middleName, honorificPrefix, honorificSuffix, and given (first) name	SCIM
displayName	The user’s display name	SCIM
active	Boolean status of identity. (Active/Inactive)	SCIM
password	The User’s clear text password. This attribute is intended to be used as a means to specify an initial password when creating a new User or to reset an existing User’s password.	SCIM
emails	Email address of user	SCIM
entitlements	Entitlements on source system. Entitlements are not returned by default.	Extended
roles	Role(s) of the user	Extended
capabilities	Users’ capabilities. I.E. System Administrator	Extended
riskScore	Composite risk score of a user	Extended
isManager	Is this user a manager?	Extended
lastRefresh	When was the last time this user was refreshed?	Extended
manager	The user’s manager, referencing the 'id’ attribute of another User	Extended

Get All Identities

Sample Request

```
curl -u "<user>:<password>"
"http://localhost:8080/iiq/scim/v2/Users"
```

Sample Response (in JSON)

```
{
"totalResults": 2,
"startIndex": 1,
"schemas": [
"urn:ietf:params:scim:api:messages:2.0:ListResponse"
],
"Resources": [
{
"urn:ietf:params:scim:schemas:sailpoint:1.0:User": {
"entitlements": [],
"capabilities": [
"SystemAdministrator"
],
"roles": [],
"isManager": false
},
"emails": [
{
"type": "work",
"value": "spadmin@sailpointdemo.com",
"primary": "true"
}
],
"displayName": "The Administrator",
"meta": {
"created": "2016-01-29T14:43:09.165-06:00",
"location": "http://localhost:8080/iiq/scim/v2/Users/2c908cbf528f1fd001528f200feb00fc",
"lastModified": "2016-02-18T16:00:26.165-06:00",
"version": "W/\"1455832826165\"",
"resourceType": "User"
},
"schemas": [
"urn:ietf:params:scim:schemas:sailpoint:1.0:User",
"urn:ietf:params:scim:schemas:core:2.0:User",
"urn:ietf:params:scim:schemas:extension:enterprise:2.0:User"
],
"name": {
"formatted": "The Administrator",
"familyName": "Administrator",
"givenName": "The"
},
"active": true,
"id": "2c908cbf528f1fd001528f200feb00fc",
"userName": "spadmin",
"urn:ietf:params:scim:schemas:extension:enterprise:2.0:User": {
"manager": {}
}
},
{
"urn:ietf:params:scim:schemas:sailpoint:1.0:User": {
"entitlements": [],
"capabilities": [
"SCIMExecutor"
],
"roles": [],
"isManager": false
},
"emails": [
{
"type": "work",
"value": "spadmin@sailpointdemo.com",
"primary": "true"
}
],
}
```

This endpoint retrieves all identities. As a performance consideration, roles and entitlements are not returned by default and must be requested explicitly.

HTTP REQUEST

`GET http://localhost:8080/iiq/scim/v2/Users/`

Get a Single Identity

Sample Request

```
curl -u "<user>:<password>"
"http://localhost:8080/iiq/scim/v2/Users/<id>"
```

This endpoint retrieves a specific identity, where ID in the request is the ID of the identity.

HTTP REQUEST

`GET http://localhost:8080/iiq/scim/v2/Users/<ID>`

Get Identity with Roles & Entitlements

Sample Request

```
curl -u "<user>:<password>"
"http://localhost:8080/identityiq/scim/v2/Users/andy.dwyer?attributes=userName,urn:ietf:params:scim:schemas:extension:enterprise:2.0:User:manager,urn:iet
```

This endpoint retrieves a specific identity and its role and entitlement information.

HTTP REQUEST

`http://localhost:8080/identityiq/scim/v2/Users/andy.dwyer?attributes=userName,urn:ietf:params:scim:schemas:extension:enterprise:2.0:User:manager,urn:ietf:params:scim:schemas:sailpoint:1.0:User:entitlements,urn:ietf:params:scim:schemas:sailpoint:1.0:User:roles`

Filter Identities

Sample Request

```
curl -u "<user>:<password>"
"http://localhost:8080/iiq/scim/v2/Users?filter=urn:ietf:params:scim:schemas:sailpoint:1.0:User:capabilities eq "SCIMExecutor"&sortBy=displayName"
```

This endpoint retrieves identities that meet the filter criteria as specified in the request.

HTTP REQUEST

`GET http://localhost:8080/iiq/scim/v2/Users?filter=urn:ietf:params:scim:schemas:sailpoint:1.0:User:capabilities eq "SCIMExecutor"&sortBy=displayName`

Create an Identity

Sample Request

```
curl -X POST -u "<user>:<password>" -H "Content-Type: application/scim+json"
 -d ' {
"userName": "mouseRat",
"name": {
"familyName":"Dwyer",
"givenName":"Andy",
"displayName":"Andy Dwyer"},
"active": true,
"password": "xyzzy",
"capabilities": "SystemAdministrator"
} ' "http://localhost:8080/iiq/scim/v2/Users/"
```

This request creates a single, new identity using the parameters passed in the request.

HTTP REQUEST

`POST http://localhost:8080/iiq/scim/v2/Users/`

Edit an Identity

Sample Request

```
curl -X PUT -u "<user>:<password>"
-d '    {
"urn:ietf:params:scim:schemas:sailpoint:1.0:User": {
},
"emails": [
{
"type": "work",
"value": "spadmin@sailpointdemo.com",
"primary": "true"
}
],
"schemas": [
"urn:ietf:params:scim:schemas:sailpoint:1.0:User",
"urn:ietf:params:scim:schemas:core:2.0:User",
"urn:ietf:params:scim:schemas:extension:enterprise:2.0:User"
],
"name": {
"formatted": "Andy Dwyer",
"familyName": "Dwyer",
"givenName": "Andrew"
},
"active": true,
"userName": "mouseRat",
"password": "xyzzy"
}  ' "http://localhost:8080/iiq/scim/v2/Users/2c909180534353fe0153574354ea0104"
```

Sample Response (in JSON)

```
{
"urn:ietf:params:scim:schemas:sailpoint:1.0:User":{
"entitlements":[
],
"capabilities":[
],
"roles":[
],
"isManager":false
},
"emails":[
{
"type":"work",
"value":"spadmin@sailpointdemo.com",
"primary":"true"
}
],
"displayName":"Andrew Dywer",
"meta":{
"created":"2016-03-08T11:25:43.786-06:00",
"location":"http://localhost:8080/iiq/scim/v2/Users/2c909180534353fe0153574354ea0104",
"lastModified":"2016-03-08T12:04:40.111-06:00",
"version":"W/\"1457460280111\"",
"resourceType":"User"
},
"schemas":[
"urn:ietf:params:scim:schemas:sailpoint:1.0:User",
"urn:ietf:params:scim:schemas:core:2.0:User",
"urn:ietf:params:scim:schemas:extension:enterprise:2.0:User"
],
"name":{
"formatted":"Andrew Dywer",
"familyName":"Dywer",
"givenName":"Andrew"
},
"active":true,
"id":"2c909180534353fe0153574354ea0104",
"userName":"mouseRat",
"urn:ietf:params:scim:schemas:extension:enterprise:2.0:User":{
"manager":{
}
}
}
```

This request makes one or more changes on an existing Identity. This can be used for actions like updating a user’s name or email address, or changing a password. In the example on the right, the first name of user “Andy Dwyer” is changed from “Andy” to “Andrew”.

HTTP REQUEST

`PUT http://localhost:8080/iiq/scim/v2/Users/<ID>`

Delete an Identity

Sample Request

```
curl -X DELETE -u "<user>:<password>"
-H "Content-Type: application/scim+json"
"http://localhost:8080/iiq/scim/v2/Users/2c909180534353fe0153574354ea0104"
```

This endpoint deletes a single identity. To delete an identity, the authenticated SCIM user must have DELETE rights, and must pass the identity ID of the identity to be deleted.

HTTP REQUEST

`DELETE http://localhost:8080/iiq/scim/v2/Users/<ID>`

Applications (/applications)

The Applications endpoint allows implementors to get information for a single application. When making a request to the application endpoint, the application ID must be included.

RESPONSE FORMAT

Parameter	Description
id	Unique application identifier
schema	Schema definition
applicationSchemas	Type, value, and $ref
name	Name of the application
features	Supported features of the application
owner	Identity ID of the application owner
type	Application type
meta	Application metadata, such as date created, location, resource type, and version

Get a Single Application

This request is used to get application details when creating, editing, or deleting an account for an identity. To get application information using this request, the application id MUST be included.

Sample Request

```
curl -u "<user>:<password>"
"http://localhost:8080/iiq/scim/v2/Applications/<applicationID>"
```

Sample Response (in JSON)

```
{
"id": "2c9084ee5571ab87015571ac44810319",
"schemas": [
"urn:ietf:params:scim:schemas:sailpoint:1.0:Application"
],
"identAttr": {},
"applicationSchemas": [
{
"value": "2c9084ee5571ab87015571ac4482031b",
"$ref": "http://localhost:8080/iiq/scim/v2/Schemas/urn:ietf:params:scim:schemas:sailpoint:1.0:Application:Schema:2c9084ee5571ab87015571ac4482031b",
"type": "account"
}
],
"name": "HR_Employees",
"features": [
"DIRECT_PERMISSIONS",
"NO_RANDOM_ACCESS",
"DISCOVER_SCHEMA"
],
"owner": {
"value": "2c9084ee5571ab87015571ac426d0316",
"$ref": "http://localhost:8080/iiq/scim/v2/Users/2c9084ee5571ab87015571ac426d0316",
"displayName": "HR_Employees App Owners"
},
"type": "Delimited File Parsing Connector",
"meta": {
"lastModified": "2016-06-21T01:42:49.362-05:00",
"created": "2016-06-21T01:36:03.074-05:00",
"location": "http://localhost:8080/iiq/scim/v2/Applications/2c9084ee5571ab87015571ac44810319",
"resourceType": "Application",
"version": "W/\"1466491369362\""
}
}
```

HTTP REQUEST

```
GET http://localhost:8080/iiq/scim/v2/Applications/<applicationID>
```

Accounts (/accounts)

The Accounts resource allows for retrieving, updating, and deleting of accounts on target systems.

Parameter	Description
id	The unique identifier for the Account object associated with IdentityIQ
nativeIdentity	The Account unique identifier associated with the native application

Parameter	Description
displayName	The name of the Account, suitable for display to end-users
instance	The instance identifier of the Account
uuid	The UUID of the Account
password	The password of the account, used in creating or changing an account password. Write-only, and never returned
lastRefresh	The last refresh date of the Account
lastTargetAggregation	The date aggregation was last targeted of the Account
manuallyCorrelated	Flag to indicate this account has been manually correlated in the UI
hasEntitlements	Flag to indicate this account has one or more entitlement attributes
identity	The corresponding User object of the Account
application	The corresponding Application object of the Account
inactive	The status of the account

Get All Accounts

The request retrieves all accounts for all identities within IdentityIQ.

Sample Request

```
curl –u "<user>:<password>"
"http://localhost:8080/iiq/scim/v2/Accounts"
```

Sample Response (in JSON)

```
{
"id": "2c9084ee5576d46f015576d4a7620003",
"identity": {
"value": "2c9084ee5576d164015576d271be05f4",
"$ref": "http://localhost:8080/iiq/scim/v2/Users/2c9084ee5576d164015576d271be05f4",
"displayName": "James Smith"
},
"hasEntitlements": false,
"schemas": [
"urn:ietf:params:scim:schemas:sailpoint:1.0:Account",
"urn:ietf:params:scim:schemas:sailpoint:1.0:Application:Schema:2c9084ee5576d164015576d20b60031b"
],
"manuallyCorrelated": false,
"application": {
"value": "2c9084ee5576d164015576d20b5f0319",
"$ref": "http://localhost:8080/iiq/scim/v2/Applications/2c9084ee5576d164015576d20b5f0319",
"displayName": "HR_Employees"
},
"nativeIdentity": "1a",
"urn:ietf:params:scim:schemas:sailpoint:1.0:Application:Schema:2c9084ee5576d164015576d20b60031b": {
"employeeId": "1a",
"region": "Americas",
"lastName": "Smith",
"email": "James.Smith@demoexample.com",
"location": "Austin",
"department": "Executive Management",
"managerId": "NULL",
"costcenter": [
"R03",
"L07",
"L08",
"L09"
],
"inactiveIdentity": "FALSE",
"fullName": "James.Smith",
"firstName": "James"
},
"lastRefresh": "2016-06-22T01:38:15.917-05:00",
"displayName": "James.Smith",
"meta": {
"location": "http://localhost:8080/iiq/scim/v2/Accounts/2c9084ee5576d46f015576d4a7620003",
"resourceType": "Account",
"version": "W/\"1466577495995\""
}
}
```

HTTP REQUEST

```
GET http://localhost:8080/iiq/scim/v2/Accounts
```

Get a Single Account

This request returns a single account. To retrieve account information on a single account, simply add the account ID to the end of the request.

Sample Request

```
curl -u "<user>:<password>"
"http://localhost:8080/iiq/scim/v2/Accounts/2c9084ee5576d46f015576d4a7620003"
```

Sample Response (in JSON)

```
{
"id": "2c9084ee5576d46f015576d4a7620003",
"identity": {
"value": "2c9084ee5576d164015576d271be05f4",
"$ref": "http://localhost:8080/iiq/scim/v2/Users/2c9084ee5576d164015576d271be05f4",
"displayName": "James Smith"
},
"hasEntitlements": false,
"schemas": [
"urn:ietf:params:scim:schemas:sailpoint:1.0:Account",
"urn:ietf:params:scim:schemas:sailpoint:1.0:Application:Schema:2c9084ee5576d164015576d20b60031b"
],
"manuallyCorrelated": false,
"application": {
"value": "2c9084ee5576d164015576d20b5f0319",
"$ref": "http://localhost:8080/iiq/scim/v2/Applications/2c9084ee5576d164015576d20b5f0319",
"displayName": "HR_Employees"
},
"nativeIdentity": "1a",
"urn:ietf:params:scim:schemas:sailpoint:1.0:Application:Schema:2c9084ee5576d164015576d20b60031b": {
"employeeId": "1a",
"region": "Americas",
"lastName": "Smith",
"email": "James.Smith@demoexample.com",
"location": "Austin",
"department": "Executive Management",
"managerId": "NULL",
"costcenter": [
"R03",
"L07",
"L08",
"L09"
],
"inactiveIdentity": "FALSE",
"fullName": "James.Smith",
"firstName": "James"
},
"lastRefresh": "2016-06-22T01:38:15.917-05:00",
"displayName": "James.Smith",
"meta": {
"location": "http://localhost:8080/iiq/scim/v2/Accounts/2c9084ee5576d46f015576d4a7620003",
"resourceType": "Account",
"version": "W/\"1466577495995\""
}
}
```

HTTP REQUEST

`GET http://localhost:8080/iiq/scim/v2/Accounts/accountID`

Filter Accounts

This request retrieves accounts that meet the filter criteria. The following fields are filterable or searchable: displayName, lastRefresh, nativeIdentity, uuid, lastTargetAgg, identity, and application. Search on application schema specific attributes is not supported.

Sample Request

```
curl -u "<user>:<password>"
"http://localhost:8080/iiq/scim/v2/Accounts?attributes=displayName&filter=(displayName co "Smith")"
```

Sample Response (in JSON)

```
{
"schemas": [
"urn:ietf:params:scim:api:messages:2.0:ListResponse"
],
"startIndex": 1,
"totalResults": 5,
"Resources": [
{
"id": "2c9084ee5576d46f015576d62cce0fdd",
"schemas": [
"urn:ietf:params:scim:schemas:sailpoint:1.0:Account"
],
"nativeIdentity": "CN=James Smith,OU=Austin,OU=Americas,OU=DemoData,DC=test,DC=sailpoint,DC=com",
"displayName": "James Smith",
"meta": {
"location": "http://localhost:8080/iiq/scim/v2/Accounts/2c9084ee5576d46f015576d62cce0fdd",
"resourceType": "Account",
"version": "W/\"1466577595603\""
}
},
{
"id": "2c9084ee5576d46f015576d4a7620003",
"schemas": [
"urn:ietf:params:scim:schemas:sailpoint:1.0:Account"
],
"nativeIdentity": "1a",
"displayName": "James.Smith",
"meta": {
"location": "http://localhost:8080/iiq/scim/v2/Accounts/2c9084ee5576d46f015576d4a7620003",
"resourceType": "Account",
"version": "W/\"1466577495995\""
}
},
{
"id": "2c9084ee5576d46f015576d5f31d0e98",
"schemas": [
"urn:ietf:params:scim:schemas:sailpoint:1.0:Account"
],
"nativeIdentity": "CN=James Smith,OU=Austin,OU=Americas,OU=DemoData,DC=test,DC=sailpoint,DC=com",
"displayName": "James.Smith",
"meta": {
"location": "http://localhost:8080/iiq/scim/v2/Accounts/2c9084ee5576d46f015576d5f31d0e98",
"resourceType": "Account",
"version": "W/\"1466577648646\""
}
},
{
"id": "2c9084ee5576d46f015576d6689a1103",
"schemas": [
"urn:ietf:params:scim:schemas:sailpoint:1.0:Account"
],
"nativeIdentity": "1a",
"displayName": "James.Smith",
"meta": {
"location": "http://localhost:8080/iiq/scim/v2/Accounts/2c9084ee5576d46f015576d6689a1103",
"resourceType": "Account",
"version": "W/\"1466577648646\""
}
},
{
"id": "2c9084ee5576d46f015576d525d408ad",
"schemas": [
"urn:ietf:params:scim:schemas:sailpoint:1.0:Account"
],
"nativeIdentity": "100",
"displayName": "JamesSmith",
"meta": {
"location": "http://localhost:8080/iiq/scim/v2/Accounts/2c9084ee5576d46f015576d525d408ad",
"resourceType": "Account",
"version": "W/\"1466577648646\""
}
}
]
}
```

HTTP REQUEST

```
http://localhost:8080/iiq/scim/v2/Accounts?attributes=displayName&filter=(displayName co "Smith")
```

Create an Account

This request is a basic request that creates an Active Directory account. Account creation depends greatly on the application schema, so requests must be modified accordingly.

** Sample Request**

```
curl -X POST -u "<user>:<password>" -H "Content-Type: application/scim+json"
{
  "identity": {
    "value": "2c9091cb5512cd85015512d0071f001f"
  },
  "application": {
    "value": "2c9091cb5512cd85015512ce25150004",
  },
  "nativeIdentity": "CN=James3 Smith3,OU=OrganzationalGroup2,OU=OrganzationalGroup1,OU=JamesSmith,DC=test,DC=sailpoint,DC=com",
  "displayName": "James.Smith",
"urn:ietf:params:scim:schemas:sailpoint:1.0:Application:Schema:2c9091cb5512cd85015512ce25240006": {
    "sn": "Smith3",
    "cn": "James3 Smith3",
    "department": "Accounting",
    "sAMAccountName": "James.Smith",
    "givenName": "James3",
    "distinguishedName": "CN=James3 Smith3,OU=OrganzationalGroup2,OU=OrganzationalGroup1,OU=duketest,DC=test,DC=sailpoint,DC=com"
  }
}
 "http://localhost:8080/ideiq/scim/v2/Accounts"
```

HTTP REQUEST

`POST http://localhost:8080/iiq/scim/v2/Accounts`

Edit an Account

This request is a basic request that edits an Active Directory account. In this example, the identity’s account name is being changed from “James.Smith” to “James.Smith.New” Account creation depends greatly on the application schema, so requests must be modified accordingly.

** Sample Request**

```
 curl -X POST -u "<user>:<password>" -H "Content-Type: application/scim+json"
{
  "identity": {
    "value": "2c9091cb5512cd85015512d0071f001f"
  },
  "application": {
    "value": "2c9091cb5512cd85015512ce25150004",
  },
  "nativeIdentity": "CN=James3 Smith3,OU=OrganzationalGroup2,OU=OrganzationalGroup1,OU=JamesSmith,DC=test,DC=sailpoint,DC=com",
  "displayName": "James.Smith",
"urn:ietf:params:scim:schemas:sailpoint:1.0:Application:Schema:2c9091cb5512cd85015512ce25240006": {
    "sn": "Smith3",
    "cn": "James3 Smith3",
    "department": "Accounting",
    "sAMAccountName": "James.Smith",
    "givenName": "James3",
    "lastName": "Smith3",
    "distinguishedName": "CN=James3 Smith3,OU=OrganzationalGroup2,OU=OrganzationalGroup1,OU=duketest,DC=test,DC=sailpoint,DC=com"
  }
}
"http://localhost:8080/idiq/scim/v2/Accounts"
```

HTTP REQUEST

`POST http://localhost:8080/iiq/scim/v2/Accounts/<accountID>`

Enable/ Disable Account

This request is used to enable or disable an account. In this example, Mary Johnson’s account status is changed from active to inactive, effectively disabling her account. To do this, the Active attribute is changed from true to false with a PUT request.

** Sample Request**

```
 curl -X PUT -u "<user>:<password>" -H "Content-Type: application/scim_json"
{
"lastRefresh": "2015-04-22T16:30:17.760-05:00",
"displayName": "Mary.Johnson",
"active": false,
"manuallyCorrelated": false,
"nativeIdentity": "1a2a",
"application": {
```

```
"displayName": "HR_Employees",
"value": "2c9084ce4ce3093a014ce30966270026",
"$ref": "http://moonraker.test.sailpoint.com:8082/identityiq/scim/v2/Applications/2c9084ce4ce3093a014ce30966270026"
},
"identity": {
"displayName": "Mary Johnson",
"value": "2c9084ce4ce3093a014ce3096878002a",
"$ref": "http://moonraker.test.sailpoint.com:8082/identityiq/scim/v2/Users/2c9084ce4ce3093a014ce3096878002a"
},
"meta": {
"created": "2015-04-22T16:30:17.760-05:00",
"location": "http://moonraker.test.sailpoint.com:8082/identityiq/scim/v2/Accounts/2c9084ce4ce309ba014ce309e1200005",
"lastModified": "2015-04-22T16:30:17.776-05:00",
"version": "W/\"1429738217776\"",
"resourceType": "Account"
},
"schemas": [
"urn:ietf:params:scim:schemas:sailpoint:1.0:Account",
"urn:ietf:params:scim:schemas:sailpoint:1.0:Application:Schema:2c9084ce4ce3093a014ce30966270027"
],
"urn:ietf:params:scim:schemas:sailpoint:1.0:Application:Schema:2c9084ce4ce3093a014ce30966270027": {
"inactiveIdentity": "FALSE",
"firstName": "Mary",
"lastName": "Johnson",
"jobtitle": "Global Infrastructure Manager",
"fullName": "Mary.Johnson",
"employeeId": "1a2a",
"location": "Austin",
"managerId": "1a",
"department": "Regional Operations",
"region": "Americas",
"costcenter": [
"R03",
"L07"
],
"email": "Mary.Johnson@demoexample.com"
},
"hasEntitlements": false,
"id": "2c9084ce4ce309ba014ce309e1200005",
"locked": false
}
```

HTTP REQUEST

`PUT https://localhost:8080/iiq/scim/v2/Accounts/<ID>`

Change Account Password

This request is used to change or reset a password for an account. In this example, Mary Johnson’s account password is changed to “passwordSwordfish”. Please note, the password attribute is not returned in the GET response, so the attribute must be added to the body.

** Sample Request**

```
curl -X PUT -u "<user>:<password>" -H "Content-Type: application/scim_json"
{
"password": "passwordSwordfish",
"lastRefresh": "2015-04-22T16:30:17.760-05:00",
"displayName": "Mary.Johnson",
"active": false,
"manuallyCorrelated": false,
"nativeIdentity": "1a2a",
"application": {
"displayName": "HR_Employees",
"value": "2c9084ce4ce3093a014ce30966270026",
"$ref": "http://moonraker.test.sailpoint.com:8082/identityiq/scim/v2/Applications/2c9084ce4ce3093a014ce30966270026"
},
"identity": {
"displayName": "Mary Johnson",
"value": "2c9084ce4ce3093a014ce3096878002a",
"$ref": "http://moonraker.test.sailpoint.com:8082/identityiq/scim/v2/Users/2c9084ce4ce3093a014ce3096878002a"
},
"meta": {
"created": "2015-04-22T16:30:17.760-05:00",
"location": "http://moonraker.test.sailpoint.com:8082/identityiq/scim/v2/Accounts/2c9084ce4ce309ba014ce309e1200005",
"lastModified": "2015-04-22T16:30:17.776-05:00",
"version": "W/\"1429738217776\"",
"resourceType": "Account"
},
"schemas": [
"urn:ietf:params:scim:schemas:sailpoint:1.0:Account",
"urn:ietf:params:scim:schemas:sailpoint:1.0:Application:Schema:2c9084ce4ce3093a014ce30966270027"
],
"urn:ietf:params:scim:schemas:sailpoint:1.0:Application:Schema:2c9084ce4ce3093a014ce30966270027": {
"inactiveIdentity": "FALSE",
"firstName": "Mary",
"lastName": "Johnson",
"jobtitle": "Global Infrastructure Manager",
"fullName": "Mary.Johnson",
"employeeId": "1a2a",
"location": "Austin",
"managerId": "1a",
"department": "Regional Operations",
"region": "Americas",
"costcenter": [
"R03",
"L07"
],
"email": "Mary.Johnson@demoexample.com"
},
"hasEntitlements": false,
"id": "2c9084ce4ce309ba014ce309e1200005",
"locked": false
}
```

HTTP REQUEST

`PUT https://localhost:8080/iiq/scim/v2/Accounts/<ID>`

Delete Account

This request is used to delete a valid account on a target application for a given identity. In this example, Adam Kennedy’s Active Directory account is deleted, preventing Adam from accessing the application in the future.

** Sample Request**

```
curl -X DELETE – u "<user>:<password>" -H "Content-Type: application/scim_json"
"http://localhost:8080/iiq/scim/v2/Accounts/2c9084ee56c0905f0156c090b4d800b8"
```

HTTP REQUEST

`DELETE http://localhost:8080/iiq/scim/v2/Accounts/<ID>`

Entitlements (/Entitlements)

The Entitlement resource allows for getting entitlements within IdentityIQ.

Parameter	Description
id	The unique identifier for the entitlement object associated with IdentityIQ
application	The application where the entitlement resides
attribute	The type of entitlement attribute
type	The type of entitlement
entitleAuth	
owner	Entitlement Owner
requestable	Flag to determine if an entitlement is requestable
aggregated	Flag to determine if entitlement is aggregated
value	Group name for the entitlement resource with the attribute of “memberOf”
lastRefresh	Date of last refresh
lastTargetAggregation	Date of last target aggregation
active	Status of the entitlement

Get All Entitlements

The request retrieves all entitlements within IdentityIQ.

Sample Request

```
curl -u "<user>:<password>"
"http://localhost:8080/iiq/scim/v2/Entitlements"
```

Sample Response (in JSON)

```
{
"schemas": [
"urn:ietf:params:scim:schemas:sailpoint:1.0:Entitlement"
],
"application": {
"value": "2c9084ee586e9dcc01586e9ed9c6032a",
"$ref": "http://blackbeard.test.sailpoint.com:8081/identityiq/scim/v2/Applications/2c9084ee586e9dcc01586e9ed9c6032a",
"displayName": "Active_Directory"
},
"attribute": "accessLog",
"type": "Permission",
"entitleAuth": "None",
"meta": {
"lastModified": "2016-11-21T10:34:42.301-06:00",
"created": "2016-11-16T13:31:04.018-06:00",
"location": "http://blackbeard.test.sailpoint.com:8081/identityiq/scim/v2/Entitlements/2c9084ee586e9dcc01586e9f00d20487",
"resourceType": "Entitlement",
"version": "W/\"1479746082301\""
},
"descriptions": [
{
"locale": "en_GB",
"value": "<strong>**дccᴇssLᴏg**</strong> tдrgᴇt frɪᴇndlŸ dᴇscrɪptɪᴏn"
},
{
"locale": "en_US",
"value": "<strong>**accessLog**</strong> <em>target friendly description</em>"
}
],
"id": "2c9084ee586e9dcc01586e9f00d20487",
"requestable": true,
"owner": {
"value": "2c9084ee586e9dcc01586e9ed8b80329",
"$ref": "http://blackbeard.test.sailpoint.com:8081/identityiq/scim/v2/Users/2c9084ee586e9dcc01586e9ed8b80329",
"displayName": "Mary Johnson"
},
"aggregated": false,
"reviewer": {},
"displayableName": "accessLog"
}
```

HTTP REQUEST

`GET http://localhost:8080/iiq/scim/v2/Entitlements`

Get Single Entitlement

The request retrieves a single entitlement specified by the entitlementID.

Sample Request

```
curl -u "<user>:<password>"
"http://localhost:8080/iiq/scim/v2/Entitlements/<EntitlementID>"
```

Sample Response (in JSON)

```
{
"schemas": [
"urn:ietf:params:scim:schemas:sailpoint:1.0:Entitlement"
],
"application": {
"value": "2c9084ee586e9dcc01586e9ed9c6032a",
"$ref": "http://blackbeard.test.sailpoint.com:8081/identityiq/scim/v2/Applications/2c9084ee586e9dcc01586e9ed9c6032a",
"displayName": "Active_Directory"
},
"attribute": "accessLog",
"type": "Permission",
"entitleAuth": "None",
"meta": {
"lastModified": "2016-11-21T10:34:42.301-06:00",
"created": "2016-11-16T13:31:04.018-06:00",
"location": "http://blackbeard.test.sailpoint.com:8081/identityiq/scim/v2/Entitlements/2c9084ee586e9dcc01586e9f00d20487",
"resourceType": "Entitlement",
"version": "W/\"1479746082301\""
},
"descriptions": [
{
"locale": "en_GB",
"value": "<strong>**дccᴇssLоg**</strong> tдrgᴇt frɪᴇndlŸ dᴇscrɪptɪоn"
},
{
"locale": "en_US",
"value": "<strong>**accessLog**</strong> <em>target friendly description</em>"
}
],
"id": "2c9084ee586e9dcc01586e9f00d20487",
"requestable": true,
"owner": {
"value": "2c9084ee586e9dcc01586e9ed8b80329",
"$ref": "http://blackbeard.test.sailpoint.com:8081/identityiq/scim/v2/Users/2c9084ee586e9dcc01586e9ed8b80329",
"displayName": "Mary Johnson"
},
"aggregated": false,
"reviewer": {},
"displayableName": "accessLog"
}
```

HTTP REQUEST

`GET http://localhost:8080/iiq/scim/v2/Entitlements/<EntitlementID>`

Roles (/Roles)

The Roles resource allows for getting roles within IdentityIQ.

Parameter	Description
id	The unique identifier for the role object associated with IdentityIQ
name	The name of the role
displayableName	The displayable name of the role
owner	Role owner information with additional sub-attributes
type	The type of role with additional sub-attributes
descriptions	Role descriptions
active	Active status of role
activationDate	Date the role became active
deactivationDate	Date the role become inactive

Get All Roles

The request retrieves all roles within IdentityIQ.

Sample Request

```
curl –u "<user>:<password>"
"http://localhost:8080/iiq/scim/v2/Roles”
```

Sample Response (in JSON)

```
{
"schemas": [
"urn:ietf:params:scim:api:messages:2.0:ListResponse"
],
"startIndex": 1,
"totalResults": 240,
"Resources": [
{
"id": "2c9084ee586e9dcc01586e9eeafe034d",
"schemas": [
"urn:ietf:params:scim:schemas:sailpoint:1.0:Role"
],
"identAttr": {},
"name": "User - IT",
"owner": {
"value": "2c9084ee586e9dcc01586e9eeaf3034c",
"$ref": "http://localhost:8080/iiq/scim/v2/Users/2c9084ee586e9dcc01586e9eeaf3034c",
"displayName": "Dennis Barnes"
},
"active": true,
"displayableName": "User - IT",
"permits": [],
"type": {
"assignmentSelector": false,
"iiq": false,
"name": "it",
"autoAssignment": false,
"permits": false,
"displayName": "IT",
"manualAssignment": false,
"requirements": false
},
"requirements": [],
"inheritance": [],
"descriptions": [
{
"locale": "en_US",
"value": "Has a user account on the company database."
}
],
"meta": {
"lastModified": "2016-11-16T13:36:15.730-06:00",
"created": "2016-11-16T13:30:58.430-06:00",
"location": "http://localhost:8080/iiq/scim/v2/Roles/2c9084ee586e9dcc01586e9eeafe034d",
"resourceType": "Role",
"version": "W/\"1479324975730\""
}
}
}
```

HTTP REQUEST

`GET http://localhost:8080/iiq/scim/v2/Roles`

Get Single Role

The request retrieves a single role specified by the roleID.

Sample Request

```
curl -u "<user>:<password>"
"http://localhost:8080/iiq/scim/v2/Roles/<RoleID>"
```

Sample Response (in JSON)

```
{
"schemas": [
"urn:ietf:params:scim:api:messages:2.0:ListResponse"
],
"startIndex": 1,
"totalResults": 240,
"Resources": [
{
"id": "2c9084ee586e9dcc01586e9eeafe034d",
"schemas": [
"urn:ietf:params:scim:schemas:sailpoint:1.0:Role"
],
"identAttr": {},
"name": "User - IT",
"owner": {
"value": "2c9084ee586e9dcc01586e9eeaf3034c",
"$ref": "http://localhost:8080/iiq/scim/v2/Users/2c9084ee586e9dcc01586e9eeaf3034c",
"displayName": "Dennis Barnes"
},
"active": true,
"displayableName": "User - IT",
"permits": [],
"type": {
"assignmentSelector": false,
"iiq": false,
"name": "it",
"autoAssignment": false,
"permits": false,
"displayName": "IT",
"manualAssignment": false,
"requirements": false
},
"requirements": [],
"inheritance": [],
"descriptions": [
{
"locale": "en_US",
"value": "Has a user account on the company database."
}
],
"meta": {
"lastModified": "2016-11-16T13:36:15.730-06:00",
"created": "2016-11-16T13:30:58.430-06:00",
"location": "http://localhost:8080/iiq/scim/v2/Roles/2c9084ee586e9dcc01586e9eeafe034d",
"resourceType": "Role",
"version": "W/\"1479324975730\""
}
}
}
```

HTTP REQUEST

```
GET http://localhost:8080/iiq/scim/v2/Roles/RoleID>
```

Policy Violations (/PolicyViolations)

The policy violation resource within IdentityIQ allows you to programmatically retrieve existing policy violations. You can choose to filter your results by one or more attributes.

Parameter	Description
id	The unique identifier for the policy violation object associated with IdentityIQ
identity	The identity effected by the violation
policyName	The name of the policy in violation
constraintName	The name of the policy contrainst
status	The status of the policy violation
description	Description of the violation
owner	Owner of the policy violation

Get All Policy Violations

The request retrieves all policy violations within IdentityIQ.

Sample Request

```
curl –u "<user>:<password>"
"http://localhost:8080/iiq/scim/v2/PolicyViolations”
```

Sample Response (in JSON)

```
{
"schemas": [
"urn:ietf:params:scim:api:messages:2.0:ListResponse"
],
"startIndex": 1,
"totalResults": 1,
"Resources": [
{
"id": "2c9084ee5cf2ff4b015cf301b2861498",
"identity": {
"value": "2c9084ee5cf2fc59015cf2fce63a0334",
"$ref": "http://blackbeard.test.sailpoint.com:8081/identityiq/scim/v2/Users/2c9084ee5cf2fc59015cf2fce63a0334",
"displayName": "Mary Johnson"
},
"schemas": [
"urn:ietf:params:scim:schemas:sailpoint:1.0:PolicyViolation"
],
"policyName": "Advanced Entitlement Policy with Details",
"constraintName": "System Administration Violation",
"status": "Open",
"description": "Active_Directory': groupmbr'='UnixAdministration'' --- conflicts with --- Active_Directory': groupmbr'='WindowsAdministration'' ",
"owner": {
"value": "2c9084ee5cf2fc59015cf2fd49fd0613",
"$ref": "http://blackbeard.test.sailpoint.com:8081/identityiq/scim/v2/Users/2c9084ee5cf2fc59015cf2fd49fd0613",
"displayName": "James Smith"
},
"meta": {
"lastModified": "2017-06-29T03:41:53.433-05:00",
"created": "2017-06-29T03:39:53.734-05:00",
"location": "http://blackbeard.test.sailpoint.com:8081/identityiq/scim/v2/PolicyViolations/2c9084ee5cf2ff4b015cf301b2861498",
"version": "W/\"1498725713433\"",
"resourceType": "PolicyViolation"
}
}
}
```

HTTP REQUEST

`GET http://localhost:8080/iiq/scim/v2/PolicyViolations`

Check for Potential Policy Violations (/CheckedPolicyViolations)

The checked policy violation resource within IdentityIQ allows you to check for a potential policy violation before an action is performed.

Parameter	Description

policies	Multivalue string of poliucy names to check, if empty do all active policies
id	The id of the identity. Either id or userName is required
userName	the identity name to check for violations
plan	the provisioning plan applied to the object
plan value	the value of the plan
plan type	Either application/xml or application/sailpoint.object.ProvisioningPlan+json

Check Account Request for Potential Policy Violation

The request checks an account request for a potential policy violation. This request can be a role or entitlement with a provisioning plan type of XML or JSON.

Sample Request

```
curl -X POST -u "<user>:<password>"
{
"identity": "Ryan.Russell",
"plan": {
"value": "{accounts=[{op=Modify, instance=null, application=Active_Directory, attributes=[{op=Add, name=groupmbr, value=UnixAdministration}], nativeIdent
"type": "application/sailpoint.object.ProvisioningPlan+json"
},
"policies": ["SOD Policy", "Entitlement Policy", "RandomPolicyNotExisting"]
}
"http://localhost:8080/iiq/scim/v2/CheckedPolicyViolations”
```

Sample Response (in JSON)

```
{
"violations": [
{
"policyName": "SOD Policy",
"constraintName": "IT SOD-117",
"description": "Security design should not be combined with administrative permissions.",
"leftBundles": [
"Security Architect - IT"
],
"policyType": "SOD",
"entitlements": [],
"rightBundles": [
"Unix Administrator - IT"
]
}
],
"identity": "Ryan.Russell",
"schemas": [
"urn:ietf:params:scim:schemas:sailpoint:1.0:CheckedPolicyViolation"
],
"plan": {
"value": "{accounts=[{op=Modify, instance=null, application=Active_Directory, attributes=[{op=Add, name=groupmbr, value=UnixAdministration}], nativeIdent
"type": "application/sailpoint.object.ProvisioningPlan+json"
},
"policies": [
"SOD Policy",
"Entitlement Policy",
"RandomPolicyNotExisting"
],
"meta": {
"resourceType": "CheckedPolicyViolation"
```

```
 }
 }
```

HTTP REQUEST

`POST http://localhost:8080/iiq/scim/v2/CheckedPolicyViolations`

Workflows (/Workflows)

The workflows resource within IdentityIQ allows you to perform various workflow related activities, from listing available workflows, launching a workflow, and retrieving the status of a workflow.

Parameter	Description
id	ID of the workflow object
name	Name of the workflow
description	Description of the workflow
type	Type of workflow
handler	Workflow handler

Get all workflows

This request retrieves all workflows within IdentityIQ

Sample Request

```
curl -u "<user>:<password>"
"http://localhost:8080/iiq/scim/v2/Workflows”
```

Sample Response (in JSON)

```
{
"schemas": [
"urn:ietf:params:scim:api:messages:2.0:ListResponse"
],
"startIndex": 1,
"totalResults": 41,
"Resources": [
{
"id": "2c9084ee5cf81e11015cf81e5156017b",
"schemas": [
"urn:ietf:params:scim:schemas:sailpoint:1.0:Workflow"
],
"name": "Do Provisioning Forms",
"type": "Subprocess",
"meta": {
"created": "2017-06-30T03:29:15.478-05:00",
"location": "http://blackbeard.test.sailpoint.com:8081/identityiq/scim/v2/Workflows/2c9084ee5cf81e11015cf81e5156017b",
"version": "W/\"1498811355478\"",
"resourceType": "Workflow"
}
}]}
```

HTTP REQUEST

```
GET http://localhost:8080/iiq/scim/v2/Workflows
```

Launch a Workflow

This request launches a workflow.

Sample Request

```
curl -x POST -u "<user>:<password>"
{
"schemas": [
"urn:ietf:params:scim:schemas:sailpoint:1.0:LaunchedWorkflow",
"urn:ietf:params:scim:schemas:sailpoint:1.0:TaskResult"
],
"urn:ietf:params:scim:schemas:sailpoint:1.0:LaunchedWorkflow": {
"workflowName": "LCM Manage Passwords",
"input": [
{
"key": "plan",
"value": "<ProvisioningPlan nativeIdentity=\"Ernest.Wagner\" targetIntegration=\"ADDirectDemodata\">\n  <AccountRequest application=\"ADDirectDemodata\"
"type": "application\/xml"
},
{
"key": "targetName",
"value": "Ernest.Wagner"
},
{
"key": "targetClass",
"value": "Identity"
},
{
"key": "identityName",
"value": "Ernest.Wagner"
},
{
"key": "flow",
"value": "PasswordRequest"
}
]
}
}
"http://localhost:8080/iiq/scim/v2/LaunchedWorkflows”
```

Sample Response (in JSON)

```
{
"urn:ietf:params:scim:schemas:sailpoint:1.0:LaunchedWorkflow": {
"input": [
{}
],
"workflowName": "LCM Manage Passwords",
"identityRequestId": "0000000001",
"retries": 0,
"output": [
{
"value": "<ProvisioningProject identity=\"Ernest.Wagner\">\n  <Attributes>\n    <Map>\n      <entry key=\"disableRetryRequest\">\n        <value>\n
"type": "application/xml",
"key": "project"
},
{
"value": "0000000001",
"key": "identityRequestId"
},
{
"value": "0",
"type": "application/int",
"key": "retries"
},
{
"value": "2c9084ee5cf825e8015cf9fca3c10ffd",
"key": "workflowCaseId"
},
{
"value": "<WorkflowSummary step=\"end\"/>\n",
"type": "application/xml",
"key": "workflowSummary"
}
],
"workflowCaseId": "2c9084ee5cf825e8015cf9fca3c10ffd",
"workflowSummary": "<WorkflowSummary step=\"end\"/>\n"
},
"launched": "2017-06-30T12:11:42.421-05:00",
"schemas": [
"urn:ietf:params:scim:schemas:sailpoint:1.0:LaunchedWorkflow",
"urn:ietf:params:scim:schemas:sailpoint:1.0:TaskResult"
],
"taskDefinition": "Workflow Launcher",
"targetClass": "Identity",
"targetName": "Ernest.Wagner",
"type": "LCM",
"pendingSignoffs": 0,
"meta": {
"lastModified": "2017-06-30T12:11:43.910-05:00",
"created": "2017-06-30T12:11:42.782-05:00",
"location": "http://blackbeard.test.sailpoint.com:8081/identityiq/scim/v2/TaskResults/2c9084ee5cf825e8015cf9fca3be0ffc",
"version": "W/\"1498842703910\"",
"resourceType": "LaunchedWorkflow"
},
"messages": [],
"id": "2c9084ee5cf825e8015cf9fca3be0ffc",
"completionStatus": "Success",
"launcher": "James.Smith",
"partitioned": false,
"verified": "2017-06-30T12:11:43.675-05:00",
"terminated": false,
"name": "LCM Manage Passwords",
"attributes": [
{
"value": "<ProvisioningProject identity=\"Ernest.Wagner\">\n  <Attributes>\n    <Map>\n      <entry key=\"disableRetryRequest\">\n        <value>\n
"key": "project"
},
{
"value": "0000000001",
"key": "identityRequestId"
},
{
"value": "0",
"key": "retries"
},
{
"value": "2c9084ee5cf825e8015cf9fca3c10ffd",
"key": "workflowCaseId"
},
{
"value": "<WorkflowSummary step=\"end\"/>\n",
"key": "workflowSummary"
}
],
"completed": "2017-06-30T12:11:43.909-05:00"
}
```

HTTP REQUEST

```
POST http://localhost:8080/iiq/scim/v2/LaunchedWorkflows
```

Get all Launched Workflows

This request retrieves all launched workflows within IdentityIQ

Sample Request

```
curl –u "<user>:<password>"
"http://localhost:8080/iiq/scim/v2/LaunchedWorkflows”
```

Sample Response (in JSON)

```
{
"schemas": [
"urn:ietf:params:scim:api:messages:2.0:ListResponse"
],
"startIndex": 1,
"totalResults": 1,
"Resources": [
{
"urn:ietf:params:scim:schemas:sailpoint:1.0:LaunchedWorkflow": {
"input": [
{}
],
"workflowName": "LCM Manage Passwords",
"identityRequestId": "0000000001",
"retries": 0,
"output": [
{
"value": "<ProvisioningProject identity=\"Ernest.Wagner\">\n  <Attributes>\n    <Map>\n      <entry key=\"disableRetryRequest\">\n        <value>\n
"type": "application/xml",
"key": "project"
},
{
"value": "0000000001",
"key": "identityRequestId"
},
{
"value": "0",
"type": "application/int",
"key": "retries"
},
{
"value": "2c9084ee5cf825e8015cf9fca3c10ffd",
"key": "workflowCaseId"
},
{
"value": "<WorkflowSummary step=\"end\"/>\n",
"type": "application/xml",
"key": "workflowSummary"
}
],
"workflowCaseId": "2c9084ee5cf825e8015cf9fca3c10ffd",
"workflowSummary": "<WorkflowSummary step=\"end\"/>\n"
},
"launched": "2017-06-30T12:11:42.421-05:00",
"schemas": [
"urn:ietf:params:scim:schemas:sailpoint:1.0:LaunchedWorkflow",
"urn:ietf:params:scim:schemas:sailpoint:1.0:TaskResult"
],
"taskDefinition": "Workflow Launcher",
"targetClass": "Identity",
"targetName": "Ernest.Wagner",
"type": "LCM",
"pendingSignoffs": 0,
"meta": {
"lastModified": "2017-06-30T12:11:43.910-05:00",
"created": "2017-06-30T12:11:42.782-05:00",
"location": "http://blackbeard.test.sailpoint.com:8081/identityiq/scim/v2/TaskResults/2c9084ee5cf825e8015cf9fca3be0ffc",
"version": "W/\"1498842703910\"",
"resourceType": "LaunchedWorkflow"
},
"messages": [],
"id": "2c9084ee5cf825e8015cf9fca3be0ffc",
"completionStatus": "Success",
"launcher": "James.Smith",
"partitioned": false,
"verified": "2017-06-30T12:11:43.675-05:00",
"terminated": false,
"name": "LCM Manage Passwords",
"attributes": [
{
"value": "<ProvisioningProject identity=\"Ernest.Wagner\">\n  <Attributes>\n    <Map>\n      <entry key=\"disableRetryRequest\">\n        <value>\n
"key": "project"
},
{
"value": "0000000001",
"key": "identityRequestId"
},
{
"value": "0",
"key": "retries"
},
{
"value": "2c9084ee5cf825e8015cf9fca3c10ffd",
"key": "workflowCaseId"
},
{
"value": "<WorkflowSummary step=\"end\"/>\n",
"key": "workflowSummary"
}
],
"completed": "2017-06-30T12:11:43.909-05:00"
}
]
}
```

HTTP REQUEST

`GET http://localhost:8080/iiq/scim/v2/LaunchedWorkflows`

Get a Workflow Status

This request retrieves the status of a launched workflow by WorkflowCaseID

Sample Request

```
curl -u "<user>:<password>"
"http://localhost:8080/iiq/scim/v2/LaunchedWorkflows/[ID]”
```

Sample Response (in JSON)

```
{
"urn:ietf:params:scim:schemas:sailpoint:1.0:LaunchedWorkflow": {
"input": [
{}
],
"workflowName": "LCM Manage Passwords",
"identityRequestId": "0000000001",
"retries": 0,
"output": [
{
"value": "<ProvisioningProject identity=\"Ernest.Wagner\">\n  <Attributes>\n    <Map>\n      <entry key=\"disableRetryRequest\">\n        <value>\n
"type": "application/xml",
"key": "project"
},
{
"value": "0000000001",
"key": "identityRequestId"
},
{
"value": "0",
"type": "application/int",
"key": "retries"
},
{
"value": "2c9084ee5cf825e8015cf9fca3c10ffd",
"key": "workflowCaseId"
},
{
"value": "<WorkflowSummary step=\"end\"/>\n",
"type": "application/xml",
"key": "workflowSummary"
}
],
"workflowCaseId": "2c9084ee5cf825e8015cf9fca3c10ffd",
"workflowSummary": "<WorkflowSummary step=\"end\"/>\n"
},
"launched": "2017-06-30T12:11:42.421-05:00",
"schemas": [
"urn:ietf:params:scim:schemas:sailpoint:1.0:LaunchedWorkflow",
"urn:ietf:params:scim:schemas:sailpoint:1.0:TaskResult"
],
"taskDefinition": "Workflow Launcher",
"targetClass": "Identity",
"targetName": "Ernest.Wagner",
"type": "LCM",
"pendingSignoffs": 0,
"meta": {
"lastModified": "2017-06-30T12:11:43.910-05:00",
"created": "2017-06-30T12:11:42.782-05:00",
"location": "http://blackbeard.test.sailpoint.com:8081/identityiq/scim/v2/TaskResults/2c9084ee5cf825e8015cf9fca3be0ffc",
"version": "W/\"1498842703910\"",
"resourceType": "LaunchedWorkflow"
},
"messages": [],
"id": "2c9084ee5cf825e8015cf9fca3be0ffc",
"completionStatus": "Success",
"launcher": "James.Smith",
"partitioned": false,
"verified": "2017-06-30T12:11:43.675-05:00",
"terminated": false,
"name": "LCM Manage Passwords",
"attributes": [
{
"value": "<ProvisioningProject identity=\"Ernest.Wagner\">\n  <Attributes>\n    <Map>\n      <entry key=\"disableRetryRequest\">\n        <value>\n
"key": "project"
},
{
"value": "0000000001",
"key": "identityRequestId"
},
{
"value": "0",
"key": "retries"
},
{
"value": "2c9084ee5cf825e8015cf9fca3c10ffd",
"key": "workflowCaseId"
},
{
"value": "<WorkflowSummary step=\"end\"/>\n",
"key": "workflowSummary"
}
],
"completed": "2017-06-30T12:11:43.909-05:00"
}
```

HTTP REQUEST

`GET http://localhost:8080/iiq/scim/v2/LaunchedWorkflows/[ID]`

Task Results (/TaskResults)

The task results resource within IdentityIQ allows you to get/list one or all task results.

Parameter	Description
id	ID of the task result object
name	Name of the task result
targetName	Name of the target
completed	Date task completed
expiration	Date the task expired
verified	Date the task was verified

Get all Task Results

This request retrieves all task results within IdentityIQ

Sample Request

```
curl -u "<user>:<password>"
"http://localhost:8080/iiq/scim/v2/TaskResults”
```

Sample Response (in JSON)

```
{
"schemas": [
"urn:ietf:params:scim:api:messages:2.0:ListResponse"
],
"startIndex": 1,
"totalResults": 1,
"Resources": [
{
"progress": "Demodata Effective Access Indexing: Running",
"launched": "2017-06-30T03:32:21.167-05:00",
"schemas": [
"urn:ietf:params:scim:schemas:sailpoint:1.0:TaskResult"
],
"taskDefinition": "setupAllTask",
"host": "blackbeard",
"type": "Generic",
"pendingSignoffs": 0,
"meta": {
"lastModified": "2017-06-30T03:37:11.934-05:00",
"created": "2017-06-30T03:32:21.173-05:00",
"location": "http://blackbeard.test.sailpoint.com:8081/identityiq/scim/v2/TaskResults/2c9084ee5cf82100015cf82126b50002",
"version": "W/\"1498811831934\"",
"resourceType": "TaskResult"
},
"messages": [],
"id": "2c9084ee5cf82100015cf82126b50002",
"completionStatus": "Success",
"launcher": "spadmin",
"partitioned": false,
"terminated": false,
"name": "setupAllTask",
"attributes": [
{
"value": "Aggregate HR Authoritative: Starting\nAggregate HR Authoritative: Complete\n\nAggregate Correlated Applications: Starting\nAggregate Correlated
"key": "tasksRun"
}
],
"completed": "2017-06-30T03:37:11.930-05:00"
}]}
```

HTTP REQUEST

`GET http://localhost:8080/iiq/scim/v2/TaskResults`

Get Single Task Results

This request retrieves a single task results within IdentityIQ

Sample Request

```
curl -u "<user>:<password>"
"http://localhost:8080/iiq/scim/v2/TaskResults/[ID]”
```

Sample Response (in JSON)

```
{
"schemas": [
"urn:ietf:params:scim:api:messages:2.0:ListResponse"
],
"startIndex": 1,
"totalResults": 1,
"Resources": [
{
"progress": "Demodata Effective Access Indexing: Running",
"launched": "2017-06-30T03:32:21.167-05:00",
"schemas": [
"urn:ietf:params:scim:schemas:sailpoint:1.0:TaskResult"
],
"taskDefinition": "setupAllTask",
"host": "blackbeard",
"type": "Generic",
"pendingSignoffs": 0,
"meta": {
"lastModified": "2017-06-30T03:37:11.934-05:00",
"created": "2017-06-30T03:32:21.173-05:00",
"location": "http://blackbeard.test.sailpoint.com:8081/identityiq/scim/v2/TaskResults/2c9084ee5cf82100015cf82126b50002",
"version": "W/\"1498811831934\"",
"resourceType": "TaskResult"
},
"messages": [],
"id": "2c9084ee5cf82100015cf82126b50002",
"completionStatus": "Success",
"launcher": "spadmin",
"partitioned": false,
"terminated": false,
"name": "setupAllTask",
"attributes": [
{
"value": "Aggregate HR Authoritative: Starting\nAggregate HR Authoritative: Complete\n\nAggregate Correlated Applications: Starting\nAggregate Correlated
"key": "tasksRun"
}
],
"completed": "2017-06-30T03:37:11.930-05:00"
}]}
```

HTTP REQUEST

```
GET http://localhost:8080/iiq/scim/v2/TaskResults/[ID]
```

HTTP Status Codes

The IdentityIQ API uses the following HTTP Status codes:

HTTP Status Code	Meaning
200	OK - Success
204	No Response
307	Temporary Redirect
308	Permanent Redirect
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found – The specified resource could not be found
409	Conflict – The specified version number does not match the resource’s latest version number
412	Precondition Failed – Failed to update
413	Payload Too Large
500	Internal Server Error – We had a problem with our server. Try again later.
501	Not Implemented – Service provider does not support the requested operation
503	Service Unavailable – We’re temporarily offline for maintenance. Please try again later.