



OpenShift Container Platform 4.20

User and group APIs

Reference guide for user and group APIs

OpenShift Container Platform 4.20 User and group APIs

Reference guide for user and group APIs

Legal Notice

Copyright © 2025 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux[®] is the registered trademark of Linus Torvalds in the United States and other countries.

Java[®] is a registered trademark of Oracle and/or its affiliates.

XFS[®] is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL[®] is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js[®] is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack[®] Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

Abstract

This document describes the OpenShift Container Platform user and group API objects and their detailed specifications.

Table of Contents

CHAPTER 1. USER AND GROUP APIS	3
1.1. GROUP [USER.OPENSIFT.IO/V1]	3
1.2. IDENTITY [USER.OPENSIFT.IO/V1]	3
1.3. USERIDENTITYMAPPING [USER.OPENSIFT.IO/V1]	3
1.4. USER [USER.OPENSIFT.IO/V1]	3
CHAPTER 2. GROUP [USER.OPENSIFT.IO/V1]	4
2.1. SPECIFICATION	4
2.2. API ENDPOINTS	5
2.2.1. /apis/user.openshift.io/v1/groups	5
2.2.2. /apis/user.openshift.io/v1/watch/groups	7
2.2.3. /apis/user.openshift.io/v1/groups/{name}	8
2.2.4. /apis/user.openshift.io/v1/watch/groups/{name}	11
CHAPTER 3. IDENTITY [USER.OPENSIFT.IO/V1]	12
3.1. SPECIFICATION	12
3.2. API ENDPOINTS	13
3.2.1. /apis/user.openshift.io/v1/identities	14
3.2.2. /apis/user.openshift.io/v1/watch/identities	15
3.2.3. /apis/user.openshift.io/v1/identities/{name}	16
3.2.4. /apis/user.openshift.io/v1/watch/identities/{name}	19
CHAPTER 4. USERIDENTITYMAPPING [USER.OPENSIFT.IO/V1]	20
4.1. SPECIFICATION	20
4.2. API ENDPOINTS	21
4.2.1. /apis/user.openshift.io/v1/useridentitymappings	21
4.2.2. /apis/user.openshift.io/v1/useridentitymappings/{name}	22
CHAPTER 5. USER [USER.OPENSIFT.IO/V1]	26
5.1. SPECIFICATION	26
5.2. API ENDPOINTS	27
5.2.1. /apis/user.openshift.io/v1/users	28
5.2.2. /apis/user.openshift.io/v1/watch/users	29
5.2.3. /apis/user.openshift.io/v1/users/{name}	30
5.2.4. /apis/user.openshift.io/v1/watch/users/{name}	33

CHAPTER 1. USER AND GROUP APIS

1.1. GROUP [USER.OPENSIFT.IO/V1]

Description

Group represents a referenceable set of Users

Compatibility level 1: Stable within a major release for a minimum of 12 months or 3 minor releases (whichever is longer).

Type

object

1.2. IDENTITY [USER.OPENSIFT.IO/V1]

Description

Identity records a successful authentication of a user with an identity provider. The information about the source of authentication is stored on the identity, and the identity is then associated with a single user object. Multiple identities can reference a single user. Information retrieved from the authentication provider is stored in the extra field using a schema determined by the provider.

Compatibility level 1: Stable within a major release for a minimum of 12 months or 3 minor releases (whichever is longer).

Type

object

1.3. USERIDENTITYMAPPING [USER.OPENSIFT.IO/V1]

Description

UserIdentityMapping maps a user to an identity

Compatibility level 1: Stable within a major release for a minimum of 12 months or 3 minor releases (whichever is longer).

Type

object

1.4. USER [USER.OPENSIFT.IO/V1]

Description

Upon log in, every user of the system receives a User and Identity resource. Administrators may directly manipulate the attributes of the users for their own tracking, or set groups via the API. The user name is unique and is chosen based on the value provided by the identity provider - if a user already exists with the incoming name, the user name may have a number appended to it depending on the configuration of the system.

Compatibility level 1: Stable within a major release for a minimum of 12 months or 3 minor releases (whichever is longer).

Type

object

CHAPTER 2. GROUP [USER.OPENSIFT.IO/V1]

Description

Group represents a referenceable set of Users

Compatibility level 1: Stable within a major release for a minimum of 12 months or 3 minor releases (whichever is longer).

Type

object

Required

- **users**

2.1. SPECIFICATION

Property	Type	Description
apiVersion	string	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
kind	string	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ObjectMeta	metadata is the standard object's metadata. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata
users	array (string)	users is the list of users in this group.

2.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/user.openshift.io/v1/groups**
 - **DELETE**: delete collection of Group
 - **GET**: list or watch objects of kind Group
 - **POST**: create a Group
- **/apis/user.openshift.io/v1/watch/groups**
 - **GET**: watch individual changes to a list of Group. deprecated: use the 'watch' parameter with a list operation instead.
- **/apis/user.openshift.io/v1/groups/{name}**
 - **DELETE**: delete a Group
 - **GET**: read the specified Group
 - **PATCH**: partially update the specified Group
 - **PUT**: replace the specified Group
- **/apis/user.openshift.io/v1/watch/groups/{name}**
 - **GET**: watch changes to an object of kind Group. deprecated: use the 'watch' parameter with a list operation instead, filtered to a single item with the 'fieldSelector' parameter.

2.2.1. /apis/user.openshift.io/v1/groups

HTTP method

DELETE

Description

delete collection of Group

Table 2.1. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

Table 2.2. HTTP responses

HTTP code	Reponse body
200 - OK	Status schema
401 - Unauthorized	Empty

HTTP method**GET****Description**

list or watch objects of kind Group

Table 2.3. HTTP responses

HTTP code	Reponse body
200 - OK	GroupList schema
401 - Unauthorized	Empty

HTTP method**POST****Description**

create a Group

Table 2.4. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

Parameter	Type	Description
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Table 2.5. Body parameters

Parameter	Type	Description
body	Group schema	

Table 2.6. HTTP responses

HTTP code	Response body
200 - OK	Group schema
201 - Created	Group schema
202 - Accepted	Group schema
401 - Unauthorized	Empty

2.2.2. /apis/user.openshift.io/v1/watch/groups

HTTP method

GET

Description

watch individual changes to a list of Group. deprecated: use the 'watch' parameter with a list operation instead.

Table 2.7. HTTP responses

HTTP code	Reponse body
200 - OK	WatchEvent schema
401 - Unauthorized	Empty

2.2.3. /apis/user.openshift.io/v1/groups/{name}

Table 2.8. Global path parameters

Parameter	Type	Description
name	string	name of the Group

HTTP method

DELETE

Description

delete a Group

Table 2.9. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

Table 2.10. HTTP responses

HTTP code	Reponse body
200 - OK	Status schema
202 - Accepted	Status schema
401 - Unauthorized	Empty

HTTP method

GET

Description

read the specified Group

Table 2.11. HTTP responses

HTTP code	Response body
200 - OK	Group schema
401 - Unauthorized	Empty

HTTP method**PATCH****Description**

partially update the specified Group

Table 2.12. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Table 2.13. HTTP responses

HTTP code	Response body
200 - OK	Group schema
201 - Created	Group schema

HTTP code	Response body
401 - Unauthorized	Empty

HTTP method**PUT****Description**

replace the specified Group

Table 2.14. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Table 2.15. Body parameters

Parameter	Type	Description
body	Group schema	

Table 2.16. HTTP responses

HTTP code	Reponse body
200 - OK	Group schema
201 - Created	Group schema
401 - Unauthorized	Empty

2.2.4. /apis/user.openshift.io/v1/watch/groups/{name}

Table 2.17. Global path parameters

Parameter	Type	Description
name	string	name of the Group

HTTP method

GET

Description

watch changes to an object of kind Group. deprecated: use the 'watch' parameter with a list operation instead, filtered to a single item with the 'fieldSelector' parameter.

Table 2.18. HTTP responses

HTTP code	Reponse body
200 - OK	WatchEvent schema
401 - Unauthorized	Empty

CHAPTER 3. IDENTITY [USER.OPENSIFT.IO/V1]

Description

Identity records a successful authentication of a user with an identity provider. The information about the source of authentication is stored on the identity, and the identity is then associated with a single user object. Multiple identities can reference a single user. Information retrieved from the authentication provider is stored in the extra field using a schema determined by the provider.

Compatibility level 1: Stable within a major release for a minimum of 12 months or 3 minor releases (whichever is longer).

Type

object

Required

- **providerName**
- **providerUserName**
- **user**

3.1. SPECIFICATION

Property	Type	Description
apiVersion	string	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
extra	object (string)	extra holds extra information about this identity
kind	string	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds

Property	Type	Description
metadata	ObjectMeta	metadata is the standard object's metadata. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata
providerName	string	providerName is the source of identity information
providerUserName	string	providerUserName uniquely represents this identity in the scope of the provider
user	ObjectReference	user is a reference to the user this identity is associated with Both Name and UID must be set

3.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/user.openshift.io/v1/identities**
 - **DELETE**: delete collection of Identity
 - **GET**: list or watch objects of kind Identity
 - **POST**: create an Identity
- **/apis/user.openshift.io/v1/watch/identities**
 - **GET**: watch individual changes to a list of Identity. deprecated: use the 'watch' parameter with a list operation instead.
- **/apis/user.openshift.io/v1/identities/{name}**
 - **DELETE**: delete an Identity
 - **GET**: read the specified Identity
 - **PATCH**: partially update the specified Identity
 - **PUT**: replace the specified Identity
- **/apis/user.openshift.io/v1/watch/identities/{name}**
 - **GET**: watch changes to an object of kind Identity. deprecated: use the 'watch' parameter with a list operation instead, filtered to a single item with the 'fieldSelector' parameter.

3.2.1. /apis/user.openshift.io/v1/identities

HTTP method

DELETE

Description

delete collection of Identity

Table 3.1. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

Table 3.2. HTTP responses

HTTP code	Reponse body
200 - OK	Status schema
401 - Unauthorized	Empty

HTTP method

GET

Description

list or watch objects of kind Identity

Table 3.3. HTTP responses

HTTP code	Reponse body
200 - OK	IdentityList schema
401 - Unauthorized	Empty

HTTP method

POST

Description

create an Identity

Table 3.4. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Table 3.5. Body parameters

Parameter	Type	Description
body	Identity schema	

Table 3.6. HTTP responses

HTTP code	Reponse body
200 - OK	Identity schema
201 - Created	Identity schema
202 - Accepted	Identity schema
401 - Unauthorized	Empty

3.2.2. /apis/user.openshift.io/v1/watch/identities

HTTP method

GET**Description**

watch individual changes to a list of Identity. deprecated: use the 'watch' parameter with a list operation instead.

Table 3.7. HTTP responses

HTTP code	Reponse body
200 - OK	WatchEvent schema
401 - Unauthorized	Empty

3.2.3. /apis/user.openshift.io/v1/identities/{name}**Table 3.8. Global path parameters**

Parameter	Type	Description
name	string	name of the Identity

HTTP method**DELETE****Description**

delete an Identity

Table 3.9. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

Table 3.10. HTTP responses

HTTP code	Reponse body
200 - OK	Status schema
202 - Accepted	Status schema
401 - Unauthorized	Empty

HTTP method

GET**Description**

read the specified Identity

Table 3.11. HTTP responses

HTTP code	Response body
200 - OK	Identity schema
401 - Unauthorized	Empty

HTTP method**PATCH****Description**

partially update the specified Identity

Table 3.12. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Table 3.13. HTTP responses

HTTP code	Response body
200 - OK	Identity schema
201 - Created	Identity schema
401 - Unauthorized	Empty

HTTP method**PUT****Description**

replace the specified Identity

Table 3.14. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Table 3.15. Body parameters

Parameter	Type	Description
body	Identity schema	

Table 3.16. HTTP responses

HTTP code	Reponse body
200 - OK	Identity schema
201 - Created	Identity schema
401 - Unauthorized	Empty

3.2.4. /apis/user.openshift.io/v1/watch/identities/{name}

Table 3.17. Global path parameters

Parameter	Type	Description
name	string	name of the Identity

HTTP method

GET

Description

watch changes to an object of kind Identity. deprecated: use the 'watch' parameter with a list operation instead, filtered to a single item with the 'fieldSelector' parameter.

Table 3.18. HTTP responses

HTTP code	Reponse body
200 - OK	WatchEvent schema
401 - Unauthorized	Empty

CHAPTER 4. USERIDENTITYMAPPING

[USER.OPENSIFT.IO/V1]

Description

UserIdentityMapping maps a user to an identity

Compatibility level 1: Stable within a major release for a minimum of 12 months or 3 minor releases (whichever is longer).

Type

object

4.1. SPECIFICATION

Property	Type	Description
apiVersion	string	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
identity	ObjectReference	identity is a reference to an identity
kind	string	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ObjectMeta	metadata is the standard object's metadata. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata
user	ObjectReference	user is a reference to a user

4.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/user.openshift.io/v1/useridentitymappings**
 - **POST**: create an UserIdentityMapping
- **/apis/user.openshift.io/v1/useridentitymappings/{name}**
 - **DELETE**: delete an UserIdentityMapping
 - **GET**: read the specified UserIdentityMapping
 - **PATCH**: partially update the specified UserIdentityMapping
 - **PUT**: replace the specified UserIdentityMapping

4.2.1. /apis/user.openshift.io/v1/useridentitymappings

Table 4.1. Global query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP method

POST

Description

create an `UserIdentityMapping`

Table 4.2. Body parameters

Parameter	Type	Description
body	UserIdentityMapping schema	

Table 4.3. HTTP responses

HTTP code	Response body
200 - OK	UserIdentityMapping schema
201 - Created	UserIdentityMapping schema
202 - Accepted	UserIdentityMapping schema
401 - Unauthorized	Empty

4.2.2. `/apis/user.openshift.io/v1/useridentitymappings/{name}`

Table 4.4. Global path parameters

Parameter	Type	Description
name	string	name of the <code>UserIdentityMapping</code>

HTTP method

DELETE

Description

delete an `UserIdentityMapping`

Table 4.5. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

Table 4.6. HTTP responses

HTTP code	Reponse body
200 - OK	Status schema
202 - Accepted	Status schema
401 - Unauthorized	Empty

HTTP method**GET****Description**

read the specified UserIdentityMapping

Table 4.7. HTTP responses

HTTP code	Reponse body
200 - OK	UserIdentityMapping schema
401 - Unauthorized	Empty

HTTP method**PATCH****Description**

partially update the specified UserIdentityMapping

Table 4.8. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

Parameter	Type	Description
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Table 4.9. HTTP responses

HTTP code	Response body
200 - OK	UserIdentityMapping schema
201 - Created	UserIdentityMapping schema
401 - Unauthorized	Empty

HTTP method**PUT****Description**

replace the specified UserIdentityMapping

Table 4.10. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

Parameter	Type	Description
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Table 4.11. Body parameters

Parameter	Type	Description
body	UserIdentityMapping schema	

Table 4.12. HTTP responses

HTTP code	Reponse body
200 - OK	UserIdentityMapping schema
201 - Created	UserIdentityMapping schema
401 - Unauthorized	Empty

CHAPTER 5. USER [USER.OPENSIFT.IO/V1]

Description

Upon log in, every user of the system receives a User and Identity resource. Administrators may directly manipulate the attributes of the users for their own tracking, or set groups via the API. The user name is unique and is chosen based on the value provided by the identity provider - if a user already exists with the incoming name, the user name may have a number appended to it depending on the configuration of the system.

Compatibility level 1: Stable within a major release for a minimum of 12 months or 3 minor releases (whichever is longer).

Type

object

Required

- **groups**

5.1. SPECIFICATION

Property	Type	Description
apiVersion	string	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
fullName	string	fullName is the full name of user
groups	array (string)	groups specifies group names this user is a member of. This field is deprecated and will be removed in a future release. Instead, create a Group object containing the name of this User.
identities	array (string)	identities are the identities associated with this user

Property	Type	Description
kind	string	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ObjectMeta	metadata is the standard object's metadata. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata

5.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/user.openshift.io/v1/users**
 - **DELETE**: delete collection of User
 - **GET**: list or watch objects of kind User
 - **POST**: create an User
- **/apis/user.openshift.io/v1/watch/users**
 - **GET**: watch individual changes to a list of User. deprecated: use the 'watch' parameter with a list operation instead.
- **/apis/user.openshift.io/v1/users/{name}**
 - **DELETE**: delete an User
 - **GET**: read the specified User
 - **PATCH**: partially update the specified User
 - **PUT**: replace the specified User
- **/apis/user.openshift.io/v1/watch/users/{name}**
 - **GET**: watch changes to an object of kind User. deprecated: use the 'watch' parameter with a list operation instead, filtered to a single item with the 'fieldSelector' parameter.

5.2.1. /apis/user.openshift.io/v1/users

HTTP method

DELETE

Description

delete collection of User

Table 5.1. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

Table 5.2. HTTP responses

HTTP code	Reponse body
200 - OK	Status schema
401 - Unauthorized	Empty

HTTP method

GET

Description

list or watch objects of kind User

Table 5.3. HTTP responses

HTTP code	Reponse body
200 - OK	UserList schema
401 - Unauthorized	Empty

HTTP method

POST

Description

create an User

Table 5.4. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Table 5.5. Body parameters

Parameter	Type	Description
body	User schema	

Table 5.6. HTTP responses

HTTP code	Response body
200 - OK	User schema
201 - Created	User schema
202 - Accepted	User schema
401 - Unauthorized	Empty

5.2.2. /apis/user.openshift.io/v1/watch/users

HTTP method**GET****Description**

watch individual changes to a list of User. deprecated: use the 'watch' parameter with a list operation instead.

Table 5.7. HTTP responses

HTTP code	Reponse body
200 - OK	WatchEvent schema
401 - Unauthorized	Empty

5.2.3. /apis/user.openshift.io/v1/users/{name}**Table 5.8. Global path parameters**

Parameter	Type	Description
name	string	name of the User

HTTP method**DELETE****Description**

delete an User

Table 5.9. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

Table 5.10. HTTP responses

HTTP code	Reponse body
200 - OK	Status schema
202 - Accepted	Status schema
401 - Unauthorized	Empty

HTTP method**GET****Description**

read the specified User

Table 5.11. HTTP responses

HTTP code	Reponse body
200 - OK	User schema
401 - Unauthorized	Empty

HTTP method**PATCH****Description**

partially update the specified User

Table 5.12. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Table 5.13. HTTP responses

HTTP code	Response body
200 - OK	User schema
201 - Created	User schema
401 - Unauthorized	Empty

HTTP method**PUT****Description**

replace the specified User

Table 5.14. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Table 5.15. Body parameters

Parameter	Type	Description
body	User schema	

Table 5.16. HTTP responses

HTTP code	Reponse body
200 - OK	User schema
201 - Created	User schema
401 - Unauthorized	Empty

5.2.4. /apis/user.openshift.io/v1/watch/users/{name}

Table 5.17. Global path parameters

Parameter	Type	Description
name	string	name of the User

HTTP method

GET

Description

watch changes to an object of kind User. deprecated: use the 'watch' parameter with a list operation instead, filtered to a single item with the 'fieldSelector' parameter.

Table 5.18. HTTP responses

HTTP code	Reponse body
200 - OK	WatchEvent schema
401 - Unauthorized	Empty