OpenShift Container Platform 4.13

OAuth APIs

Reference guide for Oauth APIs

Last Updated: 2025-05-15

OpenShift Container Platform 4.13 OAuth APIs

Reference guide for Oauth APIs

Legal Notice

Abstract

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

Table of Contents

CHAPTER 1. OAUTH APIS

1.1. OAUTHACCESSTOKEN [OAUTH.OPENSHIFT.IO/V1]

Description

OAuthAccessToken describes an OAuth access token. The name of a token must be prefixed with a **sha256~** string, must not contain "/" or "%" characters and must be at least 32 characters long.
The name of the token is constructed from the actual token by sha256-hashing it and using URL-safe unpadded base64-encoding (as described in RFC4648) on the hashed result.

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. OAUTHAUTHORIZETOKEN [OAUTH.OPENSHIFT.IO/V1]

Description

OAuthAuthorizeToken describes an OAuth authorization token
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. OAUTHCLIENTAUTHORIZATION [OAUTH.OPENSHIFT.IO/V1]

Description

OAuthClientAuthorization describes an authorization created by an OAuth client
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. OAUTHCLIENT [OAUTH.OPENSHIFT.IO/V1]

Description

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

Type

object

1.5. USEROAUTHACCESSTOKEN [OAUTH.OPENSHIFT.IO/V1]

Description

UserOAuthAccessToken is a virtual resource to mirror OAuthAccessTokens to the user the access token was issued for

Type

object

CHAPTER 2. OAUTHACCESSTOKEN [OAUTH.OPENSHIFT.IO/V1]

Description

OAuthAccessToken describes an OAuth access token. The name of a token must be prefixed with a **sha256~** string, must not contain "/" or "%" characters and must be at least 32 characters long. The name of the token is constructed from the actual token by sha256-hashing it and using URL-safe unpadded base64-encoding (as described in RFC4648) on the hashed result.

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

Type

object

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
authorizeToken	**string**	AuthorizeToken contains the token that authorized this token
clientName	**string**	ClientName references the client that created this token.
expiresIn	**integer**	ExpiresIn is the seconds from CreationTime before this token expires.
inactivityTimeoutSeconds	**integer**	InactivityTimeoutSeconds is the value in seconds, from the CreationTimestamp, after which this token can no longer be used. The value is automatically incremented when the token is used.

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**	
redirectURI	**string**	RedirectURI is the redirection associated with the token.
refreshToken	**string**	RefreshToken is the value by which this token can be renewed. Can be blank.
scopes	**array (string)**	Scopes is an array of the requested scopes.
userName	**string**	UserName is the user name associated with this token
userUID	**string**	UserUID is the unique UID associated with this token

2.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/oauth.openshift.io/v1/oauthaccesstokens**
  - **DELETE**: delete collection of OAuthAccessToken
  - **GET**: list or watch objects of kind OAuthAccessToken
  - **POST**: create an OAuthAccessToken
- **/apis/oauth.openshift.io/v1/watch/oauthaccesstokens**
  - **GET**: watch individual changes to a list of OAuthAccessToken. deprecated: use the 'watch' parameter with a list operation instead.
- **/apis/oauth.openshift.io/v1/oauthaccesstokens/{name}**
  - **DELETE**: delete an OAuthAccessToken

  - **GET**: read the specified OAuthAccessToken
  - **PATCH**: partially update the specified OAuthAccessToken
  - **PUT**: replace the specified OAuthAccessToken

- **/apis/oauth.openshift.io/v1/watch/oauthaccesstokens/{name}**
  - **GET**: watch changes to an object of kind OAuthAccessToken. deprecated: use the 'watch' parameter with a list operation instead, filtered to a single item with the 'fieldSelector' parameter.

2.2.1. /apis/oauth.openshift.io/v1/oauthaccesstokens

Table 2.1. Global query parameters

Parameter	Type	Description
pretty	**string**	If 'true', then the output is pretty printed.

HTTP method

DELETE

Description

delete collection of OAuthAccessToken

Table 2.2. Query parameters

Parameter	Type	Description

Parameter	Type	Description
continue	**string**	The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server, the server will respond with a 410 ResourceExpired error together with a continue token. If the client needs a consistent list, it must restart their list without the continue field. Otherwise, the client may send another list request with the token received with the 410 error, the server will respond with a list starting from the next key, but from the latest snapshot, which is inconsistent from the previous list results - objects that are created, modified, or deleted after the first list request will be included in the response, as long as their keys are after the "next key". This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications.
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
fieldSelector	**string**	A selector to restrict the list of returned objects by their fields. Defaults to everything.
gracePeriodSeconds	**integer**	The duration in seconds before the object should be deleted. Value must be non-negative integer. The value zero indicates delete immediately. If this value is nil, the default grace period for the specified type will be used. Defaults to a per object value if not specified. zero means delete immediately.
labelSelector	**string**	A selector to restrict the list of returned objects by their labels. Defaults to everything.

Parameter	Type	Description
limit	**integer**	limit is a maximum number of responses to return for a list call. If more items exist, the server will set the **continue** field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned.
orphanDependents	**boolean**	Deprecated: please use the PropagationPolicy, this field will be deprecated in 1.7. Should the dependent objects be orphaned. If true/false, the "orphan" finalizer will be added to/removed from the object’s finalizers list. Either this field or PropagationPolicy may be set, but not both.
propagationPolicy	**string**	Whether and how garbage collection will be performed. Either this field or OrphanDependents may be set, but not both. The default policy is decided by the existing finalizer set in the metadata.finalizers and the resource-specific default policy. Acceptable values are: 'Orphan' - orphan the dependents; 'Background' - allow the garbage collector to delete the dependents in the background; 'Foreground' - a cascading policy that deletes all dependents in the foreground.

Parameter	Type	Description
resourceVersion	**string**	resourceVersion sets a constraint on what resource versions a request may be served from. See https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions for details. Defaults to unset
resourceVersionMatch	**string**	resourceVersionMatch determines how resourceVersion is applied to list calls. It is highly recommended that resourceVersionMatch be set for list calls where resourceVersion is set See https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions for details. Defaults to unset
timeoutSeconds	**integer**	Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity.

Table 2.3. Body parameters

Parameter	Type	Description
body	**DeleteOptions** schema	

Table 2.4. HTTP responses

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

HTTP method

GET

Description

list or watch objects of kind OAuthAccessToken

Table 2.5. Query parameters

Parameter	Type	Description

Parameter	Type	Description
allowWatchBookmarks	**boolean**	allowWatchBookmarks requests watch events with type "BOOKMARK". Servers that do not implement bookmarks may ignore this flag and bookmarks are sent at the server's discretion. Clients should not assume bookmarks are returned at any specific interval, nor may they assume the server will send any BOOKMARK event during a session. If this is not a watch, this field is ignored.
continue	**string**	The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server, the server will respond with a 410 ResourceExpired error together with a continue token. If the client needs a consistent list, it must restart their list without the continue field. Otherwise, the client may send another list request with the token received with the 410 error, the server will respond with a list starting from the next key, but from the latest snapshot, which is inconsistent from the previous list results - objects that are created, modified, or deleted after the first list request will be included in the response, as long as their keys are after the "next key". This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications.
fieldSelector	**string**	A selector to restrict the list of returned objects by their fields. Defaults to everything.
labelSelector	**string**	A selector to restrict the list of returned objects by their labels. Defaults to everything.

Parameter	Type	Description
limit	**integer**	limit is a maximum number of responses to return for a list call. If more items exist, the server will set the **continue** field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned.
resourceVersion	**string**	resourceVersion sets a constraint on what resource versions a request may be served from. See https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions for details. Defaults to unset
resourceVersionMatch	**string**	resourceVersionMatch determines how resourceVersion is applied to list calls. It is highly recommended that resourceVersionMatch be set for list calls where resourceVersion is set See https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions for details. Defaults to unset
timeoutSeconds	**integer**	Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity.

Parameter	Type	Description
watch	**boolean**	Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion.

Table 2.6. HTTP responses

HTTP code	Reponse body
200 - OK	**OAuthAccessTokenList** schema
401 - Unauthorized	Empty

HTTP method

POST

Description

create an OAuthAccessToken

Table 2.7. 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
fieldManager	**string**	fieldManager is a name associated with the actor or entity that is making these changes. The value must be less than or 128 characters long, and only contain printable characters, as defined by https://golang.org/pkg/unicode/#IsPrint.

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, provided that the **ServerSideFieldValidation** feature gate is also enabled. 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 and is the default behavior when the **ServerSideFieldValidation** feature gate is disabled. - 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 when the **ServerSideFieldValidation** feature gate is enabled. - 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 2.8. Body parameters

Parameter	Type	Description
body	**OAuthAccessToken** schema	

Table 2.9. HTTP responses

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

2.2.2. /apis/oauth.openshift.io/v1/watch/oauthaccesstokens

Table 2.10. Global query parameters

Parameter	Type	Description
allowWatchBookmarks	**boolean**	allowWatchBookmarks requests watch events with type "BOOKMARK". Servers that do not implement bookmarks may ignore this flag and bookmarks are sent at the server's discretion. Clients should not assume bookmarks are returned at any specific interval, nor may they assume the server will send any BOOKMARK event during a session. If this is not a watch, this field is ignored.
continue	**string**	The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server, the server will respond with a 410 ResourceExpired error together with a continue token. If the client needs a consistent list, it must restart their list without the continue field. Otherwise, the client may send another list request with the token received with the 410 error, the server will respond with a list starting from the next key, but from the latest snapshot, which is inconsistent from the previous list results - objects that are created, modified, or deleted after the first list request will be included in the response, as long as their keys are after the "next key". This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications.
fieldSelector	**string**	A selector to restrict the list of returned objects by their fields. Defaults to everything.
labelSelector	**string**	A selector to restrict the list of returned objects by their labels. Defaults to everything.

Parameter	Type	Description
limit	**integer**	limit is a maximum number of responses to return for a list call. If more items exist, the server will set the **continue** field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned.
pretty	**string**	If 'true', then the output is pretty printed.
resourceVersion	**string**	resourceVersion sets a constraint on what resource versions a request may be served from. See https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions for details. Defaults to unset
resourceVersionMatch	**string**	resourceVersionMatch determines how resourceVersion is applied to list calls. It is highly recommended that resourceVersionMatch be set for list calls where resourceVersion is set See https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions for details. Defaults to unset
timeoutSeconds	**integer**	Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity.

Parameter	Type	Description
watch	**boolean**	Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion.

HTTP method

GET

Description

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

Table 2.11. HTTP responses

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

2.2.3. /apis/oauth.openshift.io/v1/oauthaccesstokens/{name}

Table 2.12. Global path parameters

Parameter	Type	Description
name	**string**	name of the OAuthAccessToken

Table 2.13. Global query parameters

Parameter	Type	Description
pretty	**string**	If 'true', then the output is pretty printed.

HTTP method

DELETE

Description

delete an OAuthAccessToken

Table 2.14. Query parameters

Parameter	Type	Description

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
gracePeriodSeconds	**integer**	The duration in seconds before the object should be deleted. Value must be non-negative integer. The value zero indicates delete immediately. If this value is nil, the default grace period for the specified type will be used. Defaults to a per object value if not specified. zero means delete immediately.
orphanDependents	**boolean**	Deprecated: please use the PropagationPolicy, this field will be deprecated in 1.7. Should the dependent objects be orphaned. If true/false, the "orphan" finalizer will be added to/removed from the object's finalizers list. Either this field or PropagationPolicy may be set, but not both.
propagationPolicy	**string**	Whether and how garbage collection will be performed. Either this field or OrphanDependents may be set, but not both. The default policy is decided by the existing finalizer set in the metadata.finalizers and the resource-specific default policy. Acceptable values are: 'Orphan' - orphan the dependents; 'Background' - allow the garbage collector to delete the dependents in the background; 'Foreground' - a cascading policy that deletes all dependents in the foreground.

Table 2.15. Body parameters

Parameter	Type	Description
body	**DeleteOptions** schema	

Table 2.16. HTTP responses

HTTP code	Reponse body
200 - OK	**OAuthAccessToken** schema
202 - Accepted	**OAuthAccessToken** schema

HTTP code	Reponse body
401 - Unauthorized	Empty

HTTP method

GET

Description

read the specified OAuthAccessToken

Table 2.17. HTTP responses

HTTP code	Reponse body
200 - OK	**OAuthAccessToken** schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update the specified OAuthAccessToken

Table 2.18. 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
fieldManager	**string**	fieldManager is a name associated with the actor or entity that is making these changes. The value must be less than or 128 characters long, and only contain printable characters, as defined by https://golang.org/pkg/unicode/#IsPrint. This field is required for apply requests (application/apply-patch) but optional for non-apply patch types (JsonPatch, MergePatch, StrategicMergePatch).

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, provided that the **ServerSideFieldValidation** feature gate is also enabled. 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 and is the default behavior when the **ServerSideFieldValidation** feature gate is disabled. - 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 when the **ServerSideFieldValidation** feature gate is enabled. - 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.
force	**boolean**	Force is going to "force" Apply requests. It means user will re-acquire conflicting fields owned by other people. Force flag must be unset for non-apply patch requests.

Table 2.19. Body parameters

Parameter	Type	Description
body	**Patch** schema	

Table 2.20. HTTP responses

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

HTTP method

PUT

Description

replace the specified OAuthAccessToken

Table 2.21. 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
fieldManager	**string**	fieldManager is a name associated with the actor or entity that is making these changes. The value must be less than or 128 characters long, and only contain printable characters, as defined by https://golang.org/pkg/unicode/#IsPrint.
fieldValidation	**string**	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields, provided that the **ServerSideFieldValidation** feature gate is also enabled. 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 and is the default behavior when the **ServerSideFieldValidation** feature gate is disabled. - 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 when the **ServerSideFieldValidation** feature gate is enabled. - 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 2.22. Body parameters

Parameter	Type	Description
body	**OAuthAccessToken** schema	

Table 2.23. HTTP responses

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

2.2.4. /apis/oauth.openshift.io/v1/watch/oauthaccesstokens/{name}

Table 2.24. Global path parameters

Parameter	Type	Description
name	**string**	name of the OAuthAccessToken

Table 2.25. Global query parameters

Parameter	Type	Description
allowWatchBookmarks	**boolean**	allowWatchBookmarks requests watch events with type "BOOKMARK". Servers that do not implement bookmarks may ignore this flag and bookmarks are sent at the server's discretion. Clients should not assume bookmarks are returned at any specific interval, nor may they assume the server will send any BOOKMARK event during a session. If this is not a watch, this field is ignored.

Parameter	Type	Description
continue	**string**	The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server, the server will respond with a 410 ResourceExpired error together with a continue token. If the client needs a consistent list, it must restart their list without the continue field. Otherwise, the client may send another list request with the token received with the 410 error, the server will respond with a list starting from the next key, but from the latest snapshot, which is inconsistent from the previous list results - objects that are created, modified, or deleted after the first list request will be included in the response, as long as their keys are after the "next key". This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications.
fieldSelector	**string**	A selector to restrict the list of returned objects by their fields. Defaults to everything.
labelSelector	**string**	A selector to restrict the list of returned objects by their labels. Defaults to everything.

Parameter	Type	Description
limit	**integer**	limit is a maximum number of responses to return for a list call. If more items exist, the server will set the **continue** field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned.
pretty	**string**	If 'true', then the output is pretty printed.
resourceVersion	**string**	resourceVersion sets a constraint on what resource versions a request may be served from. See https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions for details. Defaults to unset
resourceVersionMatch	**string**	resourceVersionMatch determines how resourceVersion is applied to list calls. It is highly recommended that resourceVersionMatch be set for list calls where resourceVersion is set See https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions for details. Defaults to unset
timeoutSeconds	**integer**	Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity.

Parameter	Type	Description
watch	**boolean**	Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion.

HTTP method

GET

Description

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

Table 2.26. HTTP responses

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

CHAPTER 3. OAUTHAUTHORIZETOKEN [OAUTH.OPENSHIFT.IO/V1]

Description

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

Type

object

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
clientName	**string**	ClientName references the client that created this token.
codeChallenge	**string**	CodeChallenge is the optional code_challenge associated with this authorization code, as described in rfc7636
codeChallengeMethod	**string**	CodeChallengeMethod is the optional code_challenge_method associated with this authorization code, as described in rfc7636
expiresIn	**integer**	ExpiresIn is the seconds from CreationTime before this token expires.

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**	
redirectURI	**string**	RedirectURI is the redirection associated with the token.
scopes	**array (string)**	Scopes is an array of the requested scopes.
state	**string**	State data from request
userName	**string**	UserName is the user name associated with this token
userUID	**string**	UserUID is the unique UID associated with this token. UserUID and UserName must both match for this token to be valid.

3.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/oauth.openshift.io/v1/oauthauthorizetokens**
  - **DELETE**: delete collection of OAuthAuthorizeToken
  - **GET**: list or watch objects of kind OAuthAuthorizeToken
  - **POST**: create an OAuthAuthorizeToken
- **/apis/oauth.openshift.io/v1/watch/oauthauthorizetokens**
  - **GET**: watch individual changes to a list of OAuthAuthorizeToken. deprecated: use the 'watch' parameter with a list operation instead.
- **/apis/oauth.openshift.io/v1/oauthauthorizetokens/{name}**
  - **DELETE**: delete an OAuthAuthorizeToken

- **DELETE**: delete an OAuthAuthorizeToken
- **GET**: read the specified OAuthAuthorizeToken
- **PATCH**: partially update the specified OAuthAuthorizeToken
- **PUT**: replace the specified OAuthAuthorizeToken

- **/apis/oauth.openshift.io/v1/watch/oauthauthorizetokens/{name}**
  - **GET**: watch changes to an object of kind OAuthAuthorizeToken. deprecated: use the 'watch' parameter with a list operation instead, filtered to a single item with the 'fieldSelector' parameter.

3.2.1. /apis/oauth.openshift.io/v1/oauthauthorizetokens

Table 3.1. Global query parameters

Parameter	Type	Description
pretty	**string**	If 'true', then the output is pretty printed.

HTTP method

DELETE

Description

delete collection of OAuthAuthorizeToken

Table 3.2. Query parameters

Parameter	Type	Description

Parameter	Type	Description
continue	**string**	The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server, the server will respond with a 410 ResourceExpired error together with a continue token. If the client needs a consistent list, it must restart their list without the continue field. Otherwise, the client may send another list request with the token received with the 410 error, the server will respond with a list starting from the next key, but from the latest snapshot, which is inconsistent from the previous list results - objects that are created, modified, or deleted after the first list request will be included in the response, as long as their keys are after the "next key". This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications.
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
fieldSelector	**string**	A selector to restrict the list of returned objects by their fields. Defaults to everything.
gracePeriodSeconds	**integer**	The duration in seconds before the object should be deleted. Value must be non-negative integer. The value zero indicates delete immediately. If this value is nil, the default grace period for the specified type will be used. Defaults to a per object value if not specified. zero means delete immediately.
labelSelector	**string**	A selector to restrict the list of returned objects by their labels. Defaults to everything.

Parameter	Type	Description
limit	**integer**	limit is a maximum number of responses to return for a list call. If more items exist, the server will set the **continue** field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned.
orphanDependents	**boolean**	Deprecated: please use the PropagationPolicy, this field will be deprecated in 1.7. Should the dependent objects be orphaned. If true/false, the "orphan" finalizer will be added to/removed from the object's finalizers list. Either this field or PropagationPolicy may be set, but not both.
propagationPolicy	**string**	Whether and how garbage collection will be performed. Either this field or OrphanDependents may be set, but not both. The default policy is decided by the existing finalizer set in the metadata.finalizers and the resource-specific default policy. Acceptable values are: 'Orphan' - orphan the dependents; 'Background' - allow the garbage collector to delete the dependents in the background; 'Foreground' - a cascading policy that deletes all dependents in the foreground.

Parameter	Type	Description
resourceVersion	**string**	resourceVersion sets a constraint on what resource versions a request may be served from. See https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions for details. Defaults to unset
resourceVersionMatch	**string**	resourceVersionMatch determines how resourceVersion is applied to list calls. It is highly recommended that resourceVersionMatch be set for list calls where resourceVersion is set See https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions for details. Defaults to unset
timeoutSeconds	**integer**	Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity.

Table 3.3. Body parameters

Parameter	Type	Description
body	**DeleteOptions** schema	

Table 3.4. HTTP responses

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

HTTP method

GET

Description

list or watch objects of kind OAuthAuthorizeToken

Table 3.5. Query parameters

Parameter	Type	Description

Parameter	Type	Description
allowWatchBookmarks	**boolean**	allowWatchBookmarks requests watch events with type "BOOKMARK". Servers that do not implement bookmarks may ignore this flag and bookmarks are sent at the server's discretion. Clients should not assume bookmarks are returned at any specific interval, nor may they assume the server will send any BOOKMARK event during a session. If this is not a watch, this field is ignored.
continue	**string**	The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server, the server will respond with a 410 ResourceExpired error together with a continue token. If the client needs a consistent list, it must restart their list without the continue field. Otherwise, the client may send another list request with the token received with the 410 error, the server will respond with a list starting from the next key, but from the latest snapshot, which is inconsistent from the previous list results - objects that are created, modified, or deleted after the first list request will be included in the response, as long as their keys are after the "next key". This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications.
fieldSelector	**string**	A selector to restrict the list of returned objects by their fields. Defaults to everything.
labelSelector	**string**	A selector to restrict the list of returned objects by their labels. Defaults to everything.

Parameter	Type	Description
limit	**integer**	limit is a maximum number of responses to return for a list call. If more items exist, the server will set the **continue** field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned.
resourceVersion	**string**	resourceVersion sets a constraint on what resource versions a request may be served from. See https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions for details. Defaults to unset
resourceVersionMatch	**string**	resourceVersionMatch determines how resourceVersion is applied to list calls. It is highly recommended that resourceVersionMatch be set for list calls where resourceVersion is set See https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions for details. Defaults to unset
timeoutSeconds	**integer**	Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity.

Parameter	Type	Description
watch	**boolean**	Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion.

Table 3.6. HTTP responses

HTTP code	Reponse body
200 - OK	**OAuthAuthorizeTokenList** schema
401 - Unauthorized	Empty

HTTP method

POST

Description

create an OAuthAuthorizeToken

Table 3.7. 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
fieldManager	**string**	fieldManager is a name associated with the actor or entity that is making these changes. The value must be less than or 128 characters long, and only contain printable characters, as defined by https://golang.org/pkg/unicode/#IsPrint.

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, provided that the **ServerSideFieldValidation** feature gate is also enabled. 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 and is the default behavior when the **ServerSideFieldValidation** feature gate is disabled. - 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 when the **ServerSideFieldValidation** feature gate is enabled. - 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 3.8. Body parameters

Parameter	Type	Description
body	**OAuthAuthorizeToken** schema	

Table 3.9. HTTP responses

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

3.2.2. /apis/oauth.openshift.io/v1/watch/oauthauthorizetokens

Table 3.10. Global query parameters

Parameter	Type	Description
allowWatchBookmarks	**boolean**	allowWatchBookmarks requests watch events with type "BOOKMARK". Servers that do not implement bookmarks may ignore this flag and bookmarks are sent at the server's discretion. Clients should not assume bookmarks are returned at any specific interval, nor may they assume the server will send any BOOKMARK event during a session. If this is not a watch, this field is ignored.
continue	**string**	The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server, the server will respond with a 410 ResourceExpired error together with a continue token. If the client needs a consistent list, it must restart their list without the continue field. Otherwise, the client may send another list request with the token received with the 410 error, the server will respond with a list starting from the next key, but from the latest snapshot, which is inconsistent from the previous list results - objects that are created, modified, or deleted after the first list request will be included in the response, as long as their keys are after the "next key". This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications.
fieldSelector	**string**	A selector to restrict the list of returned objects by their fields. Defaults to everything.
labelSelector	**string**	A selector to restrict the list of returned objects by their labels. Defaults to everything.

Parameter	Type	Description
limit	**integer**	limit is a maximum number of responses to return for a list call. If more items exist, the server will set the **continue** field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned.
pretty	**string**	If 'true', then the output is pretty printed.
resourceVersion	**string**	resourceVersion sets a constraint on what resource versions a request may be served from. See https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions for details. Defaults to unset
resourceVersionMatch	**string**	resourceVersionMatch determines how resourceVersion is applied to list calls. It is highly recommended that resourceVersionMatch be set for list calls where resourceVersion is set See https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions for details. Defaults to unset
timeoutSeconds	**integer**	Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity.

Parameter	Type	Description
watch	**boolean**	Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion.

HTTP method

GET

Description

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

Table 3.11. HTTP responses

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

3.2.3. /apis/oauth.openshift.io/v1/oauthauthorizetokens/{name}

Table 3.12. Global path parameters

Parameter	Type	Description
name	**string**	name of the OAuthAuthorizeToken

Table 3.13. Global query parameters

Parameter	Type	Description
pretty	**string**	If 'true', then the output is pretty printed.

HTTP method

DELETE

Description

delete an OAuthAuthorizeToken

Table 3.14. Query parameters

Parameter	Type	Description

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
gracePeriodSeconds	**integer**	The duration in seconds before the object should be deleted. Value must be non-negative integer. The value zero indicates delete immediately. If this value is nil, the default grace period for the specified type will be used. Defaults to a per object value if not specified. zero means delete immediately.
orphanDependents	**boolean**	Deprecated: please use the PropagationPolicy, this field will be deprecated in 1.7. Should the dependent objects be orphaned. If true/false, the "orphan" finalizer will be added to/removed from the object's finalizers list. Either this field or PropagationPolicy may be set, but not both.
propagationPolicy	**string**	Whether and how garbage collection will be performed. Either this field or OrphanDependents may be set, but not both. The default policy is decided by the existing finalizer set in the metadata.finalizers and the resource-specific default policy. Acceptable values are: 'Orphan' - orphan the dependents; 'Background' - allow the garbage collector to delete the dependents in the background; 'Foreground' - a cascading policy that deletes all dependents in the foreground.

Table 3.15. Body parameters

Parameter	Type	Description
body	**DeleteOptions** schema	

Table 3.16. HTTP responses

HTTP code	Reponse body
200 - OK	**OAuthAuthorizeToken** schema
202 - Accepted	**OAuthAuthorizeToken** schema

HTTP code	Reponse body
401 - Unauthorized	Empty

HTTP method

GET

Description

read the specified OAuthAuthorizeToken

Table 3.17. HTTP responses

HTTP code	Reponse body
200 - OK	**OAuthAuthorizeToken** schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update the specified OAuthAuthorizeToken

Table 3.18. 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
fieldManager	**string**	fieldManager is a name associated with the actor or entity that is making these changes. The value must be less than or 128 characters long, and only contain printable characters, as defined by https://golang.org/pkg/unicode/#IsPrint. This field is required for apply requests (application/apply-patch) but optional for non-apply patch types (JsonPatch, MergePatch, StrategicMergePatch).

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, provided that the **ServerSideFieldValidation** feature gate is also enabled. 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 and is the default behavior when the **ServerSideFieldValidation** feature gate is disabled. - 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 when the **ServerSideFieldValidation** feature gate is enabled. - 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.
force	**boolean**	Force is going to "force" Apply requests. It means user will re-acquire conflicting fields owned by other people. Force flag must be unset for non-apply patch requests.

Table 3.19. Body parameters

Parameter	Type	Description
body	**Patch** schema	

Table 3.20. HTTP responses

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

HTTP method

PUT

Description

replace the specified OAuthAuthorizeToken

Table 3.21. 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
fieldManager	**string**	fieldManager is a name associated with the actor or entity that is making these changes. The value must be less than or 128 characters long, and only contain printable characters, as defined by https://golang.org/pkg/unicode/#IsPrint.
fieldValidation	**string**	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields, provided that the **ServerSideFieldValidation** feature gate is also enabled. 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 and is the default behavior when the **ServerSideFieldValidation** feature gate is disabled. - 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 when the **ServerSideFieldValidation** feature gate is enabled. - 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 3.22. Body parameters

Parameter	Type	Description
body	**OAuthAuthorizeToken** schema	

Table 3.23. HTTP responses

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

3.2.4. /apis/oauth.openshift.io/v1/watch/oauthauthorizetokens/{name}

Table 3.24. Global path parameters

Parameter	Type	Description
name	**string**	name of the OAuthAuthorizeToken

Table 3.25. Global query parameters

Parameter	Type	Description
allowWatchBookmarks	**boolean**	allowWatchBookmarks requests watch events with type "BOOKMARK". Servers that do not implement bookmarks may ignore this flag and bookmarks are sent at the server’s discretion. Clients should not assume bookmarks are returned at any specific interval, nor may they assume the server will send any BOOKMARK event during a session. If this is not a watch, this field is ignored.

Parameter	Type	Description
continue	**string**	The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server, the server will respond with a 410 ResourceExpired error together with a continue token. If the client needs a consistent list, it must restart their list without the continue field. Otherwise, the client may send another list request with the token received with the 410 error, the server will respond with a list starting from the next key, but from the latest snapshot, which is inconsistent from the previous list results - objects that are created, modified, or deleted after the first list request will be included in the response, as long as their keys are after the "next key". This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications.
fieldSelector	**string**	A selector to restrict the list of returned objects by their fields. Defaults to everything.
labelSelector	**string**	A selector to restrict the list of returned objects by their labels. Defaults to everything.

Parameter	Type	Description
limit	**integer**	limit is a maximum number of responses to return for a list call. If more items exist, the server will set the **continue** field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned.
pretty	**string**	If 'true', then the output is pretty printed.
resourceVersion	**string**	resourceVersion sets a constraint on what resource versions a request may be served from. See https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions for details. Defaults to unset
resourceVersionMatch	**string**	resourceVersionMatch determines how resourceVersion is applied to list calls. It is highly recommended that resourceVersionMatch be set for list calls where resourceVersion is set See https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions for details. Defaults to unset
timeoutSeconds	**integer**	Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity.

Parameter	Type	Description
watch	**boolean**	Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion.

HTTP method

GET

Description

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

Table 3.26. HTTP responses

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

CHAPTER 4. OAUTHCLIENTAUTHORIZATION [OAUTH.OPENSHIFT.IO/V1]

Description

OAuthClientAuthorization describes an authorization created by an OAuth client
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
clientName	**string**	ClientName references the client that created this authorization
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**	
scopes	**array (string)**	Scopes is an array of the granted scopes.
userName	**string**	UserName is the user name that authorized this client

Property	Type	Description
userUID	**string**	UserUID is the unique UID associated with this authorization. UserUID and UserName must both match for this authorization to be valid.

4.2. API ENDPOINTS

The following API endpoints are available:

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

4.2.1. /apis/oauth.openshift.io/v1/oauthclientauthorizations

Table 4.1. Global query parameters

Parameter	Type	Description
pretty	**string**	If 'true', then the output is pretty printed.

HTTP method

DELETE

Description

delete collection of OAuthClientAuthorization

Table 4.2. Query parameters

Parameter	Type	Description
continue	**string**	The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server, the server will respond with a 410 ResourceExpired error together with a continue token. If the client needs a consistent list, it must restart their list without the continue field. Otherwise, the client may send another list request with the token received with the 410 error, the server will respond with a list starting from the next key, but from the latest snapshot, which is inconsistent from the previous list results - objects that are created, modified, or deleted after the first list request will be included in the response, as long as their keys are after the "next key". This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications.
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
fieldSelector	**string**	A selector to restrict the list of returned objects by their fields. Defaults to everything.
gracePeriodSeconds	**integer**	The duration in seconds before the object should be deleted. Value must be non-negative integer. The value zero indicates delete immediately. If this value is nil, the default grace period for the specified type will be used. Defaults to a per object value if not specified. zero means delete immediately.
labelSelector	**string**	A selector to restrict the list of returned objects by their labels. Defaults to everything.

Parameter	Type	Description
limit	**integer**	limit is a maximum number of responses to return for a list call. If more items exist, the server will set the **continue** field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned.
orphanDependents	**boolean**	Deprecated: please use the PropagationPolicy, this field will be deprecated in 1.7. Should the dependent objects be orphaned. If true/false, the "orphan" finalizer will be added to/removed from the object's finalizers list. Either this field or PropagationPolicy may be set, but not both.
propagationPolicy	**string**	Whether and how garbage collection will be performed. Either this field or OrphanDependents may be set, but not both. The default policy is decided by the existing finalizer set in the metadata.finalizers and the resource-specific default policy. Acceptable values are: 'Orphan' - orphan the dependents; 'Background' - allow the garbage collector to delete the dependents in the background; 'Foreground' - a cascading policy that deletes all dependents in the foreground.

Parameter	Type	Description
resourceVersion	**string**	resourceVersion sets a constraint on what resource versions a request may be served from. See https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions for details. Defaults to unset
resourceVersionMatch	**string**	resourceVersionMatch determines how resourceVersion is applied to list calls. It is highly recommended that resourceVersionMatch be set for list calls where resourceVersion is set See https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions for details. Defaults to unset
timeoutSeconds	**integer**	Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity.

Table 4.3. Body parameters

Parameter	Type	Description
body	**DeleteOptions** schema	

Table 4.4. HTTP responses

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

HTTP method

GET

Description

list or watch objects of kind OAuthClientAuthorization

Table 4.5. Query parameters

Parameter	Type	Description
allowWatchBookmarks	**boolean**	allowWatchBookmarks requests watch events with type "BOOKMARK". Servers that do not implement bookmarks may ignore this flag and bookmarks are sent at the server's discretion. Clients should not assume bookmarks are returned at any specific interval, nor may they assume the server will send any BOOKMARK event during a session. If this is not a watch, this field is ignored.
continue	**string**	The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server, the server will respond with a 410 ResourceExpired error together with a continue token. If the client needs a consistent list, it must restart their list without the continue field. Otherwise, the client may send another list request with the token received with the 410 error, the server will respond with a list starting from the next key, but from the latest snapshot, which is inconsistent from the previous list results - objects that are created, modified, or deleted after the first list request will be included in the response, as long as their keys are after the "next key". This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications.
fieldSelector	**string**	A selector to restrict the list of returned objects by their fields. Defaults to everything.
labelSelector	**string**	A selector to restrict the list of returned objects by their labels. Defaults to everything.

Parameter	Type	Description
limit	**integer**	limit is a maximum number of responses to return for a list call. If more items exist, the server will set the **continue** field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned.
resourceVersion	**string**	resourceVersion sets a constraint on what resource versions a request may be served from. See https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions for details. Defaults to unset
resourceVersionMatch	**string**	resourceVersionMatch determines how resourceVersion is applied to list calls. It is highly recommended that resourceVersionMatch be set for list calls where resourceVersion is set See https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions for details. Defaults to unset
timeoutSeconds	**integer**	Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity.

Parameter	Type	Description
watch	**boolean**	Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion.

Table 4.6. HTTP responses

HTTP code	Reponse body
200 - OK	**OAuthClientAuthorizationList** schema
401 - Unauthorized	Empty

HTTP method

POST

Description

create an OAuthClientAuthorization

Table 4.7. 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
fieldManager	**string**	fieldManager is a name associated with the actor or entity that is making these changes. The value must be less than or 128 characters long, and only contain printable characters, as defined by https://golang.org/pkg/unicode/#IsPrint.

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, provided that the **ServerSideFieldValidation** feature gate is also enabled. 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 and is the default behavior when the **ServerSideFieldValidation** feature gate is disabled. - 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 when the **ServerSideFieldValidation** feature gate is enabled. - 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 4.8. Body parameters

Parameter	Type	Description
body	**OAuthClientAuthorization** schema	

Table 4.9. HTTP responses

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

4.2.2. /apis/oauth.openshift.io/v1/watch/oauthclientauthorizations

Table 4.10. Global query parameters

Parameter	Type	Description
allowWatchBookmarks	**boolean**	allowWatchBookmarks requests watch events with type "BOOKMARK". Servers that do not implement bookmarks may ignore this flag and bookmarks are sent at the server's discretion. Clients should not assume bookmarks are returned at any specific interval, nor may they assume the server will send any BOOKMARK event during a session. If this is not a watch, this field is ignored.
continue	**string**	The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server, the server will respond with a 410 ResourceExpired error together with a continue token. If the client needs a consistent list, it must restart their list without the continue field. Otherwise, the client may send another list request with the token received with the 410 error, the server will respond with a list starting from the next key, but from the latest snapshot, which is inconsistent from the previous list results - objects that are created, modified, or deleted after the first list request will be included in the response, as long as their keys are after the "next key". This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications.
fieldSelector	**string**	A selector to restrict the list of returned objects by their fields. Defaults to everything.
labelSelector	**string**	A selector to restrict the list of returned objects by their labels. Defaults to everything.

Parameter	Type	Description
limit	**integer**	limit is a maximum number of responses to return for a list call. If more items exist, the server will set the **continue** field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned.
pretty	**string**	If 'true', then the output is pretty printed.
resourceVersion	**string**	resourceVersion sets a constraint on what resource versions a request may be served from. See https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions for details. Defaults to unset
resourceVersionMatch	**string**	resourceVersionMatch determines how resourceVersion is applied to list calls. It is highly recommended that resourceVersionMatch be set for list calls where resourceVersion is set See https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions for details. Defaults to unset
timeoutSeconds	**integer**	Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity.

Parameter	Type	Description
watch	**boolean**	Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion.

HTTP method

GET

Description

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

Table 4.11. HTTP responses

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

4.2.3. /apis/oauth.openshift.io/v1/oauthclientauthorizations/{name}

Table 4.12. Global path parameters

Parameter	Type	Description
name	**string**	name of the OAuthClientAuthorization

Table 4.13. Global query parameters

Parameter	Type	Description
pretty	**string**	If 'true', then the output is pretty printed.

HTTP method

DELETE

Description

delete an OAuthClientAuthorization

Table 4.14. Query parameters

Parameter	Type	Description

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
gracePeriodSeconds	**integer**	The duration in seconds before the object should be deleted. Value must be non-negative integer. The value zero indicates delete immediately. If this value is nil, the default grace period for the specified type will be used. Defaults to a per object value if not specified. zero means delete immediately.
orphanDependents	**boolean**	Deprecated: please use the PropagationPolicy, this field will be deprecated in 1.7. Should the dependent objects be orphaned. If true/false, the "orphan" finalizer will be added to/removed from the object's finalizers list. Either this field or PropagationPolicy may be set, but not both.
propagationPolicy	**string**	Whether and how garbage collection will be performed. Either this field or OrphanDependents may be set, but not both. The default policy is decided by the existing finalizer set in the metadata.finalizers and the resource-specific default policy. Acceptable values are: 'Orphan' - orphan the dependents; 'Background' - allow the garbage collector to delete the dependents in the background; 'Foreground' - a cascading policy that deletes all dependents in the foreground.

Table 4.15. Body parameters

Parameter	Type	Description
body	**DeleteOptions** schema	

Table 4.16. HTTP responses

HTTP code	Reponse body
200 - OK	**Status** schema
202 - Accepted	**Status** schema

HTTP code	Reponse body
401 - Unauthorized	Empty

HTTP method

GET

Description

read the specified OAuthClientAuthorization

Table 4.17. HTTP responses

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

HTTP method

PATCH

Description

partially update the specified OAuthClientAuthorization

Table 4.18. 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
fieldManager	**string**	fieldManager is a name associated with the actor or entity that is making these changes. The value must be less than or 128 characters long, and only contain printable characters, as defined by https://golang.org/pkg/unicode/#IsPrint. This field is required for apply requests (application/apply-patch) but optional for non-apply patch types (JsonPatch, MergePatch, StrategicMergePatch).

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, provided that the **ServerSideFieldValidation** feature gate is also enabled. 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 and is the default behavior when the **ServerSideFieldValidation** feature gate is disabled. - 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 when the **ServerSideFieldValidation** feature gate is enabled. - 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.
force	**boolean**	Force is going to "force" Apply requests. It means user will re-acquire conflicting fields owned by other people. Force flag must be unset for non-apply patch requests.

Table 4.19. Body parameters

Parameter	Type	Description
body	**Patch** schema	

Table 4.20. HTTP responses

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

HTTP method

PUT

Description

replace the specified OAuthClientAuthorization

Table 4.21. 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
fieldManager	**string**	fieldManager is a name associated with the actor or entity that is making these changes. The value must be less than or 128 characters long, and only contain printable characters, as defined by https://golang.org/pkg/unicode/#IsPrint.
fieldValidation	**string**	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields, provided that the **ServerSideFieldValidation** feature gate is also enabled. 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 and is the default behavior when the **ServerSideFieldValidation** feature gate is disabled. - 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 when the **ServerSideFieldValidation** feature gate is enabled. - 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 4.22. Body parameters

Parameter	Type	Description
body	**OAuthClientAuthorization** schema	

Table 4.23. HTTP responses

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

4.2.4. /apis/oauth.openshift.io/v1/watch/oauthclientauthorizations/{name}

Table 4.24. Global path parameters

Parameter	Type	Description
name	**string**	name of the OAuthClientAuthorization

Table 4.25. Global query parameters

Parameter	Type	Description
allowWatchBookmarks	**boolean**	allowWatchBookmarks requests watch events with type "BOOKMARK". Servers that do not implement bookmarks may ignore this flag and bookmarks are sent at the server's discretion. Clients should not assume bookmarks are returned at any specific interval, nor may they assume the server will send any BOOKMARK event during a session. If this is not a watch, this field is ignored.

Parameter	Type	Description
continue	**string**	The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server, the server will respond with a 410 ResourceExpired error together with a continue token. If the client needs a consistent list, it must restart their list without the continue field. Otherwise, the client may send another list request with the token received with the 410 error, the server will respond with a list starting from the next key, but from the latest snapshot, which is inconsistent from the previous list results - objects that are created, modified, or deleted after the first list request will be included in the response, as long as their keys are after the "next key". This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications.
fieldSelector	**string**	A selector to restrict the list of returned objects by their fields. Defaults to everything.
labelSelector	**string**	A selector to restrict the list of returned objects by their labels. Defaults to everything.

Parameter	Type	Description
limit	**integer**	limit is a maximum number of responses to return for a list call. If more items exist, the server will set the **continue** field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned.
pretty	**string**	If 'true', then the output is pretty printed.
resourceVersion	**string**	resourceVersion sets a constraint on what resource versions a request may be served from. See https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions for details. Defaults to unset
resourceVersionMatch	**string**	resourceVersionMatch determines how resourceVersion is applied to list calls. It is highly recommended that resourceVersionMatch be set for list calls where resourceVersion is set See https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions for details. Defaults to unset
timeoutSeconds	**integer**	Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity.

Parameter	Type	Description
watch	**boolean**	Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion.

HTTP method

GET

Description

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

Table 4.26. HTTP responses

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

CHAPTER 5. OAUTHCLIENT [OAUTH.OPENSHIFT.IO/V1]

Description

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

Type

object

5.1. SPECIFICATION

Property	Type	Description
accessTokenInactivityTimeoutSeconds	**integer**	AccessTokenInactivityTimeoutSeconds overrides the default token inactivity timeout for tokens granted to this client. The value represents the maximum amount of time that can occur between consecutive uses of the token. Tokens become invalid if they are not used within this temporal window. The user will need to acquire a new token to regain access once a token times out. This value needs to be set only if the default set in configuration is not appropriate for this client. Valid values are: - 0: Tokens for this client never time out - X: Tokens time out if there is no activity for X seconds The current minimum allowed value for X is 300 (5 minutes) WARNING: existing tokens' timeout will not be affected (lowered) by changing this value
accessTokenMaxAgeSeconds	**integer**	AccessTokenMaxAgeSeconds overrides the default access token max age for tokens granted to this client. 0 means no expiration.
additionalSecrets	**array (string)**	AdditionalSecrets holds other secrets that may be used to identify the client. This is useful for rotation and for service account token validation

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
grantMethod	**string**	GrantMethod is a required field which determines how to handle grants for this client. Valid grant handling methods are: - auto: always approves grant requests, useful for trusted clients - prompt: prompts the end user for approval of grant requests, useful for third-party clients
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**	
redirectURIs	**array (string)**	RedirectURIs is the valid redirection URIs associated with a client
respondWithChallenges	**boolean**	RespondWithChallenges indicates whether the client wants authentication needed responses made in the form of challenges instead of redirects

Property	Type	Description
scopeRestrictions	**array**	ScopeRestrictions describes which scopes this client can request. Each requested scope is checked against each restriction. If any restriction matches, then the scope is allowed. If no restriction matches, then the scope is denied.
scopeRestrictions[]	**object**	ScopeRestriction describe one restriction on scopes. Exactly one option must be non-nil.
secret	**string**	Secret is the unique secret associated with a client

5.1.1. .scopeRestrictions

Description

ScopeRestrictions describes which scopes this client can request. Each requested scope is checked against each restriction. If any restriction matches, then the scope is allowed. If no restriction matches, then the scope is denied.

Type

array

5.1.2. .scopeRestrictions[]

Description

ScopeRestriction describe one restriction on scopes. Exactly one option must be non-nil.

Type

object

Property	Type	Description
clusterRole	**object**	ClusterRoleScopeRestriction describes restrictions on cluster role scopes
literals	**array (string)**	ExactValues means the scope has to match a particular set of strings exactly

5.1.3. .scopeRestrictions[].clusterRole

Description

ClusterRoleScopeRestriction describes restrictions on cluster role scopes

Type

object

Required

- **roleNames**
- **namespaces**
- **allowEscalation**

Property	Type	Description
allowEscalation	**boolean**	AllowEscalation indicates whether you can request roles and their escalating resources
namespaces	**array (string)**	Namespaces is the list of namespaces that can be referenced. * means any of them (including *)
roleNames	**array (string)**	RoleNames is the list of cluster roles that can referenced. * means anything

5.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/oauth.openshift.io/v1/oauthclients**
  - **DELETE**: delete collection of OAuthClient
  - **GET**: list or watch objects of kind OAuthClient
  - **POST**: create an OAuthClient
- **/apis/oauth.openshift.io/v1/watch/oauthclients**
  - **GET**: watch individual changes to a list of OAuthClient. deprecated: use the 'watch' parameter with a list operation instead.
- **/apis/oauth.openshift.io/v1/oauthclients/{name}**
  - **DELETE**: delete an OAuthClient
  - **GET**: read the specified OAuthClient
  - **PATCH**: partially update the specified OAuthClient
  - **PUT**: replace the specified OAuthClient

- **/apis/oauth.openshift.io/v1/watch/oauthclients/{name}**
  - **GET**: watch changes to an object of kind OAuthClient. deprecated: use the 'watch' parameter with a list operation instead, filtered to a single item with the 'fieldSelector' parameter.

5.2.1. /apis/oauth.openshift.io/v1/oauthclients

Table 5.1. Global query parameters

Parameter	Type	Description
pretty	**string**	If 'true', then the output is pretty printed.

HTTP method

DELETE

Description

delete collection of OAuthClient

Table 5.2. Query parameters

Parameter	Type	Description
continue	**string**	The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server, the server will respond with a 410 ResourceExpired error together with a continue token. If the client needs a consistent list, it must restart their list without the continue field. Otherwise, the client may send another list request with the token received with the 410 error, the server will respond with a list starting from the next key, but from the latest snapshot, which is inconsistent from the previous list results - objects that are created, modified, or deleted after the first list request will be included in the response, as long as their keys are after the "next key". This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications.

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
fieldSelector	**string**	A selector to restrict the list of returned objects by their fields. Defaults to everything.
gracePeriodSeconds	**integer**	The duration in seconds before the object should be deleted. Value must be non-negative integer. The value zero indicates delete immediately. If this value is nil, the default grace period for the specified type will be used. Defaults to a per object value if not specified. zero means delete immediately.
labelSelector	**string**	A selector to restrict the list of returned objects by their labels. Defaults to everything.
limit	**integer**	limit is a maximum number of responses to return for a list call. If more items exist, the server will set the **continue** field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned.

Parameter	Type	Description
orphanDependents	**boolean**	Deprecated: please use the PropagationPolicy, this field will be deprecated in 1.7. Should the dependent objects be orphaned. If true/false, the "orphan" finalizer will be added to/removed from the object's finalizers list. Either this field or PropagationPolicy may be set, but not both.
propagationPolicy	**string**	Whether and how garbage collection will be performed. Either this field or OrphanDependents may be set, but not both. The default policy is decided by the existing finalizer set in the metadata.finalizers and the resource-specific default policy. Acceptable values are: 'Orphan' - orphan the dependents; 'Background' - allow the garbage collector to delete the dependents in the background; 'Foreground' - a cascading policy that deletes all dependents in the foreground.
resourceVersion	**string**	resourceVersion sets a constraint on what resource versions a request may be served from. See https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions for details. Defaults to unset
resourceVersionMatch	**string**	resourceVersionMatch determines how resourceVersion is applied to list calls. It is highly recommended that resourceVersionMatch be set for list calls where resourceVersion is set See https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions for details. Defaults to unset
timeoutSeconds	**integer**	Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity.

Table 5.3. Body parameters

Parameter	Type	Description
body	**DeleteOptions** schema	

Table 5.4. HTTP responses

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

HTTP method

GET

Description

list or watch objects of kind OAuthClient

Table 5.5. Query parameters

Parameter	Type	Description
allowWatchBookmarks	**boolean**	allowWatchBookmarks requests watch events with type "BOOKMARK". Servers that do not implement bookmarks may ignore this flag and bookmarks are sent at the server’s discretion. Clients should not assume bookmarks are returned at any specific interval, nor may they assume the server will send any BOOKMARK event during a session. If this is not a watch, this field is ignored.
continue	**string**	The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server, the server will respond with a 410 ResourceExpired error together with a continue token. If the client needs a consistent list, it must restart their list without the continue field. Otherwise, the client may send another list request with the token received with the 410 error, the server will respond with a list starting from the next key, but from the latest snapshot, which is inconsistent from the previous list results - objects that are created, modified, or deleted after the first list request will be included in the response, as long as their keys are after the "next key". This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications.

Parameter	Type	Description
fieldSelector	**string**	A selector to restrict the list of returned objects by their fields. Defaults to everything.
labelSelector	**string**	A selector to restrict the list of returned objects by their labels. Defaults to everything.
limit	**integer**	limit is a maximum number of responses to return for a list call. If more items exist, the server will set the **continue** field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned.
resourceVersion	**string**	resourceVersion sets a constraint on what resource versions a request may be served from. See https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions for details. Defaults to unset
resourceVersionMatch	**string**	resourceVersionMatch determines how resourceVersion is applied to list calls. It is highly recommended that resourceVersionMatch be set for list calls where resourceVersion is set See https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions for details. Defaults to unset

Parameter	Type	Description
timeoutSeconds	**integer**	Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity.
watch	**boolean**	Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion.

Table 5.6. HTTP responses

HTTP code	Reponse body
200 - OK	**OAuthClientList** schema
401 - Unauthorized	Empty

HTTP method

POST

Description

create an OAuthClient

Table 5.7. 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
fieldManager	**string**	fieldManager is a name associated with the actor or entity that is making these changes. The value must be less than or 128 characters long, and only contain printable characters, as defined by https://golang.org/pkg/unicode/#IsPrint.

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, provided that the **ServerSideFieldValidation** feature gate is also enabled. 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 and is the default behavior when the **ServerSideFieldValidation** feature gate is disabled. - 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 when the **ServerSideFieldValidation** feature gate is enabled. - 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 5.8. Body parameters

Parameter	Type	Description
body	**OAuthClient** schema	

Table 5.9. HTTP responses

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

5.2.2. /apis/oauth.openshift.io/v1/watch/oauthclients

Table 5.10. Global query parameters

Parameter	Type	Description
allowWatchBookmarks	**boolean**	allowWatchBookmarks requests watch events with type "BOOKMARK". Servers that do not implement bookmarks may ignore this flag and bookmarks are sent at the server's discretion. Clients should not assume bookmarks are returned at any specific interval, nor may they assume the server will send any BOOKMARK event during a session. If this is not a watch, this field is ignored.
continue	**string**	The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server, the server will respond with a 410 ResourceExpired error together with a continue token. If the client needs a consistent list, it must restart their list without the continue field. Otherwise, the client may send another list request with the token received with the 410 error, the server will respond with a list starting from the next key, but from the latest snapshot, which is inconsistent from the previous list results - objects that are created, modified, or deleted after the first list request will be included in the response, as long as their keys are after the "next key". This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications.
fieldSelector	**string**	A selector to restrict the list of returned objects by their fields. Defaults to everything.
labelSelector	**string**	A selector to restrict the list of returned objects by their labels. Defaults to everything.

Parameter	Type	Description
limit	**integer**	limit is a maximum number of responses to return for a list call. If more items exist, the server will set the **continue** field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned.
pretty	**string**	If 'true', then the output is pretty printed.
resourceVersion	**string**	resourceVersion sets a constraint on what resource versions a request may be served from. See https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions for details. Defaults to unset
resourceVersionMatch	**string**	resourceVersionMatch determines how resourceVersion is applied to list calls. It is highly recommended that resourceVersionMatch be set for list calls where resourceVersion is set See https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions for details. Defaults to unset
timeoutSeconds	**integer**	Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity.

Parameter	Type	Description
watch	**boolean**	Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion.

HTTP method

GET

Description

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

Table 5.11. HTTP responses

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

5.2.3. /apis/oauth.openshift.io/v1/oauthclients/{name}

Table 5.12. Global path parameters

Parameter	Type	Description
name	**string**	name of the OAuthClient

Table 5.13. Global query parameters

Parameter	Type	Description
pretty	**string**	If 'true', then the output is pretty printed.

HTTP method

DELETE

Description

delete an OAuthClient

Table 5.14. Query parameters

Parameter	Type	Description

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
gracePeriodSeconds	**integer**	The duration in seconds before the object should be deleted. Value must be non-negative integer. The value zero indicates delete immediately. If this value is nil, the default grace period for the specified type will be used. Defaults to a per object value if not specified. zero means delete immediately.
orphanDependents	**boolean**	Deprecated: please use the PropagationPolicy, this field will be deprecated in 1.7. Should the dependent objects be orphaned. If true/false, the "orphan" finalizer will be added to/removed from the object's finalizers list. Either this field or PropagationPolicy may be set, but not both.
propagationPolicy	**string**	Whether and how garbage collection will be performed. Either this field or OrphanDependents may be set, but not both. The default policy is decided by the existing finalizer set in the metadata.finalizers and the resource-specific default policy. Acceptable values are: 'Orphan' - orphan the dependents; 'Background' - allow the garbage collector to delete the dependents in the background; 'Foreground' - a cascading policy that deletes all dependents in the foreground.

Table 5.15. Body parameters

Parameter	Type	Description
body	**DeleteOptions** schema	

Table 5.16. HTTP responses

HTTP code	Reponse body
200 - OK	**Status** schema
202 - Accepted	**Status** schema

HTTP code	Reponse body
401 - Unauthorized	Empty

HTTP method

GET

Description

read the specified OAuthClient

Table 5.17. HTTP responses

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

HTTP method

PATCH

Description

partially update the specified OAuthClient

Table 5.18. 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
fieldManager	**string**	fieldManager is a name associated with the actor or entity that is making these changes. The value must be less than or 128 characters long, and only contain printable characters, as defined by https://golang.org/pkg/unicode/#IsPrint. This field is required for apply requests (application/apply-patch) but optional for non-apply patch types (JsonPatch, MergePatch, StrategicMergePatch).

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, provided that the **ServerSideFieldValidation** feature gate is also enabled. 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 and is the default behavior when the **ServerSideFieldValidation** feature gate is disabled. - 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 when the **ServerSideFieldValidation** feature gate is enabled. - 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.
force	**boolean**	Force is going to "force" Apply requests. It means user will re-acquire conflicting fields owned by other people. Force flag must be unset for non-apply patch requests.

Table 5.19. Body parameters

Parameter	Type	Description
body	**Patch** schema	

Table 5.20. HTTP responses

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

HTTP method

PUT

Description

replace the specified OAuthClient

Table 5.21. 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
fieldManager	**string**	fieldManager is a name associated with the actor or entity that is making these changes. The value must be less than or 128 characters long, and only contain printable characters, as defined by https://golang.org/pkg/unicode/#IsPrint.
fieldValidation	**string**	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields, provided that the **ServerSideFieldValidation** feature gate is also enabled. 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 and is the default behavior when the **ServerSideFieldValidation** feature gate is disabled. - 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 when the **ServerSideFieldValidation** feature gate is enabled. - 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 5.22. Body parameters

Parameter	Type	Description
body	**OAuthClient** schema	

Table 5.23. HTTP responses

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

5.2.4. /apis/oauth.openshift.io/v1/watch/oauthclients/{name}

Table 5.24. Global path parameters

Parameter	Type	Description
name	**string**	name of the OAuthClient

Table 5.25. Global query parameters

Parameter	Type	Description
allowWatchBookmarks	**boolean**	allowWatchBookmarks requests watch events with type "BOOKMARK". Servers that do not implement bookmarks may ignore this flag and bookmarks are sent at the server’s discretion. Clients should not assume bookmarks are returned at any specific interval, nor may they assume the server will send any BOOKMARK event during a session. If this is not a watch, this field is ignored.

Parameter	Type	Description
continue	**string**	The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server, the server will respond with a 410 ResourceExpired error together with a continue token. If the client needs a consistent list, it must restart their list without the continue field. Otherwise, the client may send another list request with the token received with the 410 error, the server will respond with a list starting from the next key, but from the latest snapshot, which is inconsistent from the previous list results - objects that are created, modified, or deleted after the first list request will be included in the response, as long as their keys are after the "next key". This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications.
fieldSelector	**string**	A selector to restrict the list of returned objects by their fields. Defaults to everything.
labelSelector	**string**	A selector to restrict the list of returned objects by their labels. Defaults to everything.

Parameter	Type	Description
limit	**integer**	limit is a maximum number of responses to return for a list call. If more items exist, the server will set the **continue** field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned.
pretty	**string**	If 'true', then the output is pretty printed.
resourceVersion	**string**	resourceVersion sets a constraint on what resource versions a request may be served from. See https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions for details. Defaults to unset
resourceVersionMatch	**string**	resourceVersionMatch determines how resourceVersion is applied to list calls. It is highly recommended that resourceVersionMatch be set for list calls where resourceVersion is set See https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions for details. Defaults to unset
timeoutSeconds	**integer**	Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity.

Parameter	Type	Description
watch	**boolean**	Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion.

HTTP method

GET

Description

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

Table 5.26. HTTP responses

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

CHAPTER 6. USEROAUTHACCESSTOKEN [OAUTH.OPENSHIFT.IO/V1]

Description

UserOAuthAccessToken is a virtual resource to mirror OAuthAccessTokens to the user the access token was issued for

Type

object

6.1. 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
authorizeToken	**string**	AuthorizeToken contains the token that authorized this token
clientName	**string**	ClientName references the client that created this token.
expiresIn	**integer**	ExpiresIn is the seconds from CreationTime before this token expires.
inactivityTimeoutSeconds	**integer**	InactivityTimeoutSeconds is the value in seconds, from the CreationTimestamp, after which this token can no longer be used. The value is automatically incremented when the token is used.

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**	
redirectURI	**string**	RedirectURI is the redirection associated with the token.
refreshToken	**string**	RefreshToken is the value by which this token can be renewed. Can be blank.
scopes	**array (string)**	Scopes is an array of the requested scopes.
userName	**string**	UserName is the user name associated with this token
userUID	**string**	UserUID is the unique UID associated with this token

6.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/oauth.openshift.io/v1/useroauthaccesstokens**
  - **GET**: list or watch objects of kind UserOAuthAccessToken
- **/apis/oauth.openshift.io/v1/watch/useroauthaccesstokens**
  - **GET**: watch individual changes to a list of UserOAuthAccessToken. deprecated: use the 'watch' parameter with a list operation instead.
- **/apis/oauth.openshift.io/v1/useroauthaccesstokens/{name}**
  - **DELETE**: delete an UserOAuthAccessToken
  - **GET**: read the specified UserOAuthAccessToken

- **/apis/oauth.openshift.io/v1/watch/useroauthaccesstokens/{name}**
  - **GET**: watch changes to an object of kind UserOAuthAccessToken. deprecated: use the 'watch' parameter with a list operation instead, filtered to a single item with the 'fieldSelector' parameter.

6.2.1. /apis/oauth.openshift.io/v1/useroauthaccesstokens

Table 6.1. Global query parameters

Parameter	Type	Description
allowWatchBookmarks	**boolean**	allowWatchBookmarks requests watch events with type "BOOKMARK". Servers that do not implement bookmarks may ignore this flag and bookmarks are sent at the server’s discretion. Clients should not assume bookmarks are returned at any specific interval, nor may they assume the server will send any BOOKMARK event during a session. If this is not a watch, this field is ignored.
continue	**string**	The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server, the server will respond with a 410 ResourceExpired error together with a continue token. If the client needs a consistent list, it must restart their list without the continue field. Otherwise, the client may send another list request with the token received with the 410 error, the server will respond with a list starting from the next key, but from the latest snapshot, which is inconsistent from the previous list results - objects that are created, modified, or deleted after the first list request will be included in the response, as long as their keys are after the "next key". This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications.
fieldSelector	**string**	A selector to restrict the list of returned objects by their fields. Defaults to everything.
labelSelector	**string**	A selector to restrict the list of returned objects by their labels. Defaults to everything.

Parameter	Type	Description
limit	**integer**	limit is a maximum number of responses to return for a list call. If more items exist, the server will set the **continue** field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned.
pretty	**string**	If 'true', then the output is pretty printed.
resourceVersion	**string**	resourceVersion sets a constraint on what resource versions a request may be served from. See https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions for details. Defaults to unset
resourceVersionMatch	**string**	resourceVersionMatch determines how resourceVersion is applied to list calls. It is highly recommended that resourceVersionMatch be set for list calls where resourceVersion is set See https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions for details. Defaults to unset
timeoutSeconds	**integer**	Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity.

Parameter	Type	Description
watch	**boolean**	Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion.

HTTP method

GET

Description

list or watch objects of kind UserOAuthAccessToken

Table 6.2. HTTP responses

HTTP code	Reponse body
200 - OK	**UserOAuthAccessTokenList** schema
401 - Unauthorized	Empty

6.2.2. /apis/oauth.openshift.io/v1/watch/useroauthaccesstokens

Table 6.3. Global query parameters

Parameter	Type	Description
allowWatchBookmarks	**boolean**	allowWatchBookmarks requests watch events with type "BOOKMARK". Servers that do not implement bookmarks may ignore this flag and bookmarks are sent at the server’s discretion. Clients should not assume bookmarks are returned at any specific interval, nor may they assume the server will send any BOOKMARK event during a session. If this is not a watch, this field is ignored.

Parameter	Type	Description
continue	**string**	The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server, the server will respond with a 410 ResourceExpired error together with a continue token. If the client needs a consistent list, it must restart their list without the continue field. Otherwise, the client may send another list request with the token received with the 410 error, the server will respond with a list starting from the next key, but from the latest snapshot, which is inconsistent from the previous list results - objects that are created, modified, or deleted after the first list request will be included in the response, as long as their keys are after the "next key". This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications.
fieldSelector	**string**	A selector to restrict the list of returned objects by their fields. Defaults to everything.
labelSelector	**string**	A selector to restrict the list of returned objects by their labels. Defaults to everything.

Parameter	Type	Description
limit	**integer**	limit is a maximum number of responses to return for a list call. If more items exist, the server will set the **continue** field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned.
pretty	**string**	If 'true', then the output is pretty printed.
resourceVersion	**string**	resourceVersion sets a constraint on what resource versions a request may be served from. See https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions for details. Defaults to unset
resourceVersionMatch	**string**	resourceVersionMatch determines how resourceVersion is applied to list calls. It is highly recommended that resourceVersionMatch be set for list calls where resourceVersion is set See https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions for details. Defaults to unset
timeoutSeconds	**integer**	Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity.

Parameter	Type	Description
watch	**boolean**	Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion.

HTTP method

GET

Description

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

Table 6.4. HTTP responses

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

6.2.3. /apis/oauth.openshift.io/v1/useroauthaccesstokens/{name}

Table 6.5. Global path parameters

Parameter	Type	Description
name	**string**	name of the UserOAuthAccessToken

Table 6.6. Global query parameters

Parameter	Type	Description
pretty	**string**	If 'true', then the output is pretty printed.

HTTP method

DELETE

Description

delete an UserOAuthAccessToken

Table 6.7. Query parameters

Parameter	Type	Description

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
gracePeriodSeconds	**integer**	The duration in seconds before the object should be deleted. Value must be non-negative integer. The value zero indicates delete immediately. If this value is nil, the default grace period for the specified type will be used. Defaults to a per object value if not specified. zero means delete immediately.
orphanDependents	**boolean**	Deprecated: please use the PropagationPolicy, this field will be deprecated in 1.7. Should the dependent objects be orphaned. If true/false, the "orphan" finalizer will be added to/removed from the object's finalizers list. Either this field or PropagationPolicy may be set, but not both.
propagationPolicy	**string**	Whether and how garbage collection will be performed. Either this field or OrphanDependents may be set, but not both. The default policy is decided by the existing finalizer set in the metadata.finalizers and the resource-specific default policy. Acceptable values are: 'Orphan' - orphan the dependents; 'Background' - allow the garbage collector to delete the dependents in the background; 'Foreground' - a cascading policy that deletes all dependents in the foreground.

Table 6.8. Body parameters

Parameter	Type	Description
body	**DeleteOptions** schema	

Table 6.9. HTTP responses

HTTP code	Reponse body
200 - OK	**Status** schema
202 - Accepted	**Status** schema

HTTP code	Reponse body
401 - Unauthorized	Empty

HTTP method

GET

Description

read the specified UserOAuthAccessToken

Table 6.10. HTTP responses

HTTP code	Reponse body
200 - OK	**UserOAuthAccessToken** schema
401 - Unauthorized	Empty

6.2.4. /apis/oauth.openshift.io/v1/watch/useroauthaccesstokens/{name}

Table 6.11. Global path parameters

Parameter	Type	Description
name	**string**	name of the UserOAuthAccessToken

Table 6.12. Global query parameters

Parameter	Type	Description
allowWatchBookmarks	**boolean**	allowWatchBookmarks requests watch events with type "BOOKMARK". Servers that do not implement bookmarks may ignore this flag and bookmarks are sent at the server’s discretion. Clients should not assume bookmarks are returned at any specific interval, nor may they assume the server will send any BOOKMARK event during a session. If this is not a watch, this field is ignored.

Parameter	Type	Description
continue	**string**	The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server, the server will respond with a 410 ResourceExpired error together with a continue token. If the client needs a consistent list, it must restart their list without the continue field. Otherwise, the client may send another list request with the token received with the 410 error, the server will respond with a list starting from the next key, but from the latest snapshot, which is inconsistent from the previous list results - objects that are created, modified, or deleted after the first list request will be included in the response, as long as their keys are after the "next key". This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications.
fieldSelector	**string**	A selector to restrict the list of returned objects by their fields. Defaults to everything.
labelSelector	**string**	A selector to restrict the list of returned objects by their labels. Defaults to everything.

Parameter	Type	Description
limit	**integer**	limit is a maximum number of responses to return for a list call. If more items exist, the server will set the **continue** field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned.
pretty	**string**	If 'true', then the output is pretty printed.
resourceVersion	**string**	resourceVersion sets a constraint on what resource versions a request may be served from. See https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions for details. Defaults to unset
resourceVersionMatch	**string**	resourceVersionMatch determines how resourceVersion is applied to list calls. It is highly recommended that resourceVersionMatch be set for list calls where resourceVersion is set See https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions for details. Defaults to unset

Parameter	Type	Description
timeoutSeconds	**integer**	Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity.
watch	**boolean**	Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion.

HTTP method

GET

Description

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

Table 6.13. HTTP responses

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