



# OpenShift Container Platform 4.17

## Authorization APIs

Reference guide for authorization APIs



# OpenShift Container Platform 4.17 Authorization APIs

---

Reference guide for authorization APIs

## Legal Notice

Copyright © 2024 Red Hat, Inc.

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

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

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

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

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

Linux<sup>®</sup> is the registered trademark of Linus Torvalds in the United States and other countries.

Java<sup>®</sup> is a registered trademark of Oracle and/or its affiliates.

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

MySQL<sup>®</sup> is a registered trademark of MySQL AB in the United States, the European Union and other countries.

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

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

All other trademarks are the property of their respective owners.

## Abstract

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

## Table of Contents

<b>CHAPTER 1. AUTHORIZATION APIS</b> .....	<b>5</b>
1.1. LOCALRESOURCEACCESSREVIEW [AUTHORIZATION.OPENSIFT.IO/V1]	5
1.2. LOCALSUBJECTACCESSREVIEW [AUTHORIZATION.OPENSIFT.IO/V1]	5
1.3. RESOURCEACCESSREVIEW [AUTHORIZATION.OPENSIFT.IO/V1]	5
1.4. SELFSUBJECTRULESREVIEW [AUTHORIZATION.OPENSIFT.IO/V1]	5
1.5. SUBJECTACCESSREVIEW [AUTHORIZATION.OPENSIFT.IO/V1]	6
1.6. SUBJECTRULESREVIEW [AUTHORIZATION.OPENSIFT.IO/V1]	6
1.7. SELFSUBJECTREVIEW [AUTHENTICATION.K8S.IO/V1]	6
1.8. TOKENREQUEST [AUTHENTICATION.K8S.IO/V1]	6
1.9. TOKENREVIEW [AUTHENTICATION.K8S.IO/V1]	6
1.10. LOCALSUBJECTACCESSREVIEW [AUTHORIZATION.K8S.IO/V1]	6
1.11. SELFSUBJECTACCESSREVIEW [AUTHORIZATION.K8S.IO/V1]	7
1.12. SELFSUBJECTRULESREVIEW [AUTHORIZATION.K8S.IO/V1]	7
1.13. SUBJECTACCESSREVIEW [AUTHORIZATION.K8S.IO/V1]	7
<b>CHAPTER 2. LOCALRESOURCEACCESSREVIEW [AUTHORIZATION.OPENSIFT.IO/V1]</b> .....	<b>8</b>
2.1. SPECIFICATION	8
2.2. API ENDPOINTS	9
2.2.1. /apis/authorization.openshift.io/v1/namespaces/{namespace}/localresourceaccessreviews	10
<b>CHAPTER 3. LOCALSUBJECTACCESSREVIEW [AUTHORIZATION.OPENSIFT.IO/V1]</b> .....	<b>12</b>
3.1. SPECIFICATION	12
3.2. API ENDPOINTS	14
3.2.1. /apis/authorization.openshift.io/v1/namespaces/{namespace}/localsubjectaccessreviews	14
<b>CHAPTER 4. RESOURCEACCESSREVIEW [AUTHORIZATION.OPENSIFT.IO/V1]</b> .....	<b>16</b>
4.1. SPECIFICATION	16
4.2. API ENDPOINTS	17
4.2.1. /apis/authorization.openshift.io/v1/resourceaccessreviews	18
<b>CHAPTER 5. SELFSUBJECTRULESREVIEW [AUTHORIZATION.OPENSIFT.IO/V1]</b> .....	<b>20</b>
5.1. SPECIFICATION	20
5.1.1. .spec	21
5.1.2. .status	21
5.1.3. .status.rules	22
5.1.4. .status.rules[]	22
5.2. API ENDPOINTS	23
5.2.1. /apis/authorization.openshift.io/v1/namespaces/{namespace}/selfsubjectrulesreviews	23
<b>CHAPTER 6. SUBJECTACCESSREVIEW [AUTHORIZATION.OPENSIFT.IO/V1]</b> .....	<b>25</b>
6.1. SPECIFICATION	25
6.2. API ENDPOINTS	27
6.2.1. /apis/authorization.openshift.io/v1/subjectaccessreviews	27
<b>CHAPTER 7. SUBJECTRULESREVIEW [AUTHORIZATION.OPENSIFT.IO/V1]</b> .....	<b>29</b>
7.1. SPECIFICATION	29
7.1.1. .spec	30
7.1.2. .status	30
7.1.3. .status.rules	31
7.1.4. .status.rules[]	31
7.2. API ENDPOINTS	32
7.2.1. /apis/authorization.openshift.io/v1/namespaces/{namespace}/subjectrulesreviews	32

<b>CHAPTER 8. SELFSUBJECTREVIEW [AUTHENTICATION.K8S.IO/V1]</b> .....	<b>35</b>
8.1. SPECIFICATION	35
8.1.1. .status	35
8.1.2. .status.userInfo	36
8.1.3. .status.userInfo.extra	36
8.2. API ENDPOINTS	36
8.2.1. /apis/authentication.k8s.io/v1/selfsubjectreviews	37
<b>CHAPTER 9. TOKENREQUEST [AUTHENTICATION.K8S.IO/V1]</b> .....	<b>39</b>
9.1. SPECIFICATION	39
9.1.1. .spec	40
9.1.2. .spec.boundObjectRef	40
9.1.3. .status	41
9.2. API ENDPOINTS	41
9.2.1. /api/v1/namespaces/{namespace}/serviceaccounts/{name}/token	41
<b>CHAPTER 10. TOKENREVIEW [AUTHENTICATION.K8S.IO/V1]</b> .....	<b>44</b>
10.1. SPECIFICATION	44
10.1.1. .spec	45
10.1.2. .status	45
10.1.3. .status.user	46
10.1.4. .status.user.extra	47
10.2. API ENDPOINTS	47
10.2.1. /apis/oauth.openshift.io/v1/tokenreviews	47
10.2.2. /apis/authentication.k8s.io/v1/tokenreviews	48
<b>CHAPTER 11. LOCALSUBJECTACCESSREVIEW [AUTHORIZATION.K8S.IO/V1]</b> .....	<b>51</b>
11.1. SPECIFICATION	51
11.1.1. .spec	52
11.1.2. .spec.extra	53
11.1.3. .spec.nonResourceAttributes	53
11.1.4. .spec.resourceAttributes	53
11.1.5. .status	54
11.2. API ENDPOINTS	55
11.2.1. /apis/authorization.k8s.io/v1/namespaces/{namespace}/localsubjectaccessreviews	55
<b>CHAPTER 12. SELFSUBJECTACCESSREVIEW [AUTHORIZATION.K8S.IO/V1]</b> .....	<b>57</b>
12.1. SPECIFICATION	57
12.1.1. .spec	58
12.1.2. .spec.nonResourceAttributes	58
12.1.3. .spec.resourceAttributes	59
12.1.4. .status	60
12.2. API ENDPOINTS	60
12.2.1. /apis/authorization.k8s.io/v1/selfsubjectaccessreviews	60
<b>CHAPTER 13. SELFSUBJECTRULESREVIEW [AUTHORIZATION.K8S.IO/V1]</b> .....	<b>63</b>
13.1. SPECIFICATION	63
13.1.1. .spec	64
13.1.2. .status	64
13.1.3. .status.nonResourceRules	65
13.1.4. .status.nonResourceRules[]	66
13.1.5. .status.resourceRules	66
13.1.6. .status.resourceRules[]	66
13.2. API ENDPOINTS	67

---

13.2.1. /apis/authorization.k8s.io/v1/selfsubjectrulesreviews	67
<b>CHAPTER 14. SUBJECTACCESSREVIEW [AUTHORIZATION.K8S.IO/V1]</b> .....	<b>69</b>
14.1. SPECIFICATION	69
14.1.1. .spec	70
14.1.2. .spec.extra	71
14.1.3. .spec.nonResourceAttributes	71
14.1.4. .spec.resourceAttributes	71
14.1.5. .status	72
14.2. API ENDPOINTS	73
14.2.1. /apis/authorization.k8s.io/v1/subjectaccessreviews	73





# CHAPTER 1. AUTHORIZATION APIS

## 1.1. LOCALRESOURCEACCESSREVIEW [AUTHORIZATION.OPENSIFT.IO/V1]

### Description

LocalResourceAccessReview is a means to request a list of which users and groups are authorized to perform the action specified by spec in a particular namespace

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. LOCALSUBJECTACCESSREVIEW [AUTHORIZATION.OPENSIFT.IO/V1]

### Description

LocalSubjectAccessReview is an object for requesting information about whether a user or group can perform an action in a particular namespace

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. RESOURCEACCESSREVIEW [AUTHORIZATION.OPENSIFT.IO/V1]

### Description

ResourceAccessReview is a means to request a list of which users and groups are authorized to perform the action specified by spec

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. SELFSUBJECTRULESREVIEW [AUTHORIZATION.OPENSIFT.IO/V1]

### Description

SelfSubjectRulesReview is a resource you can create to determine which actions you can perform in a namespace

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. SUBJECTACCESSREVIEW [AUTHORIZATION.OPENSIFT.IO/V1]

### Description

SubjectAccessReview is an object for requesting information about whether a user or group can perform an action  
Compatibility level 1: Stable within a major release for a minimum of 12 months or 3 minor releases (whichever is longer).

### Type

**object**

## 1.6. SUBJECTRULESREVIEW [AUTHORIZATION.OPENSIFT.IO/V1]

### Description

SubjectRulesReview is a resource you can create to determine which actions another user can perform in a namespace  
Compatibility level 1: Stable within a major release for a minimum of 12 months or 3 minor releases (whichever is longer).

### Type

**object**

## 1.7. SELFSUBJECTREVIEW [AUTHENTICATION.K8S.IO/V1]

### Description

SelfSubjectReview contains the user information that the kube-apiserver has about the user making this request. When using impersonation, users will receive the user info of the user being impersonated. If impersonation or request header authentication is used, any extra keys will have their case ignored and returned as lowercase.

### Type

**object**

## 1.8. TOKENREQUEST [AUTHENTICATION.K8S.IO/V1]

### Description

TokenRequest requests a token for a given service account.

### Type

**object**

## 1.9. TOKENREVIEW [AUTHENTICATION.K8S.IO/V1]

### Description

TokenReview attempts to authenticate a token to a known user. Note: TokenReview requests may be cached by the webhook token authenticator plugin in the kube-apiserver.

### Type

**object**

## 1.10. LOCALSUBJECTACCESSREVIEW [AUTHORIZATION.K8S.IO/V1]

**Description**

LocalSubjectAccessReview checks whether or not a user or group can perform an action in a given namespace. Having a namespace scoped resource makes it much easier to grant namespace scoped policy that includes permissions checking.

**Type**

**object**

## 1.11. SELFSUBJECTACCESSREVIEW [AUTHORIZATION.K8S.IO/V1]

**Description**

SelfSubjectAccessReview checks whether or the current user can perform an action. Not filling in a spec.namespace means "in all namespaces". Self is a special case, because users should always be able to check whether they can perform an action

**Type**

**object**

## 1.12. SELFSUBJECTRULESREVIEW [AUTHORIZATION.K8S.IO/V1]

**Description**

SelfSubjectRulesReview enumerates the set of actions the current user can perform within a namespace. The returned list of actions may be incomplete depending on the server's authorization mode, and any errors experienced during the evaluation. SelfSubjectRulesReview should be used by UIs to show/hide actions, or to quickly let an end user reason about their permissions. It should NOT Be used by external systems to drive authorization decisions as this raises confused deputy, cache lifetime/revocation, and correctness concerns. SubjectAccessReview, and LocalAccessReview are the correct way to defer authorization decisions to the API server.

**Type**

**object**

## 1.13. SUBJECTACCESSREVIEW [AUTHORIZATION.K8S.IO/V1]

**Description**

SubjectAccessReview checks whether or not a user or group can perform an action.

**Type**

**object**

## CHAPTER 2. LOCALRESOURCEACCESSREVIEW [AUTHORIZATION.OPENSIFT.IO/V1]

### Description

LocalResourceAccessReview is a means to request a list of which users and groups are authorized to perform the action specified by spec in a particular namespace

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

### Type

**object**

### Required

- **namespace**
- **verb**
- **resourceAPIGroup**
- **resourceAPIVersion**
- **resource**
- **resourceName**
- **path**
- **isNonResourceURL**

## 2.1. SPECIFICATION

Property	Type	Description
<b>apiVersion</b>	<b>string</b>	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: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources</a>
<b>content</b>	<b>RawExtension</b>	Content is the actual content of the request for create and update
<b>isNonResourceURL</b>	<b>boolean</b>	IsNonResourceURL is true if this is a request for a non-resource URL (outside of the resource hierarchy)

Property	Type	Description
<b>kind</b>	<b>string</b>	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: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds</a>
<b>namespace</b>	<b>string</b>	Namespace is the namespace of the action being requested. Currently, there is no distinction between no namespace and all namespaces
<b>path</b>	<b>string</b>	Path is the path of a non resource URL
<b>resource</b>	<b>string</b>	Resource is one of the existing resource types
<b>resourceAPIGroup</b>	<b>string</b>	Group is the API group of the resource Serialized as resourceAPIGroup to avoid confusion with the 'groups' field when inlined
<b>resourceAPIVersion</b>	<b>string</b>	Version is the API version of the resource Serialized as resourceAPIVersion to avoid confusion with TypeMeta.apiVersion and ObjectMeta.resourceVersion when inlined
<b>resourceName</b>	<b>string</b>	ResourceName is the name of the resource being requested for a "get" or deleted for a "delete"
<b>verb</b>	<b>string</b>	Verb is one of: get, list, watch, create, update, delete

## 2.2. API ENDPOINTS

The following API endpoints are available:

- `/apis/authorization.openshift.io/v1/namespaces/{namespace}/localresourceaccessreviews`
  - **POST**: create a LocalResourceAccessReview

## 2.2.1. `/apis/authorization.openshift.io/v1/namespaces/{namespace}/localresourceacc`

Table 2.1. Global query parameters

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

### HTTP method

**POST**

### Description

create a LocalResourceAccessReview

Table 2.2. Body parameters

Parameter	Type	Description
<b>body</b>	<a href="#">LocalResourceAccessReview</a> schema	

Table 2.3. HTTP responses

HTTP code	Reponse body
200 - OK	<b>LocalResourceAccessReview</b> schema
201 - Created	<b>LocalResourceAccessReview</b> schema
202 - Accepted	<b>LocalResourceAccessReview</b> schema
401 - Unauthorized	Empty

## CHAPTER 3. LOCALSUBJECTACCESSREVIEW [AUTHORIZATION.OPENSIFT.IO/V1]

### Description

LocalSubjectAccessReview is an object for requesting information about whether a user or group can perform an action in a particular namespace

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

### Type

**object**

### Required

- **namespace**
- **verb**
- **resourceAPIGroup**
- **resourceAPIVersion**
- **resource**
- **resourceName**
- **path**
- **isNonResourceURL**
- **user**
- **groups**
- **scopes**

## 3.1. SPECIFICATION

Property	Type	Description
<b>apiVersion</b>	<b>string</b>	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: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources</a>



Property	Type	Description
<b>content</b>	<b>RawExtension</b>	Content is the actual content of the request for create and update
<b>groups</b>	<b>array (string)</b>	Groups is optional. Groups is the list of groups to which the User belongs.
<b>isNonResourceURL</b>	<b>boolean</b>	IsNonResourceURL is true if this is a request for a non-resource URL (outside of the resource hierarchy)
<b>kind</b>	<b>string</b>	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: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds</a>
<b>namespace</b>	<b>string</b>	Namespace is the namespace of the action being requested. Currently, there is no distinction between no namespace and all namespaces
<b>path</b>	<b>string</b>	Path is the path of a non resource URL
<b>resource</b>	<b>string</b>	Resource is one of the existing resource types
<b>resourceAPIGroup</b>	<b>string</b>	Group is the API group of the resource Serialized as resourceAPIGroup to avoid confusion with the 'groups' field when inlined
<b>resourceAPIVersion</b>	<b>string</b>	Version is the API version of the resource Serialized as resourceAPIVersion to avoid confusion with TypeMeta.apiVersion and ObjectMeta.resourceVersion when inlined

Property	Type	Description
<b>resourceName</b>	<b>string</b>	ResourceName is the name of the resource being requested for a "get" or deleted for a "delete"
<b>scopes</b>	<b>array (string)</b>	Scopes to use for the evaluation. Empty means "use the unscoped (full) permissions of the user/groups". Nil for a self-SAR, means "use the scopes on this request". Nil for a regular SAR, means the same as empty.
<b>user</b>	<b>string</b>	User is optional. If both User and Groups are empty, the current authenticated user is used.
<b>verb</b>	<b>string</b>	Verb is one of: get, list, watch, create, update, delete

## 3.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/authorization.openshift.io/v1/namespaces/{namespace}/localsubjectaccessreviews**
  - **POST**: create a LocalSubjectAccessReview

### 3.2.1. /apis/authorization.openshift.io/v1/namespaces/{namespace}/localsubjectacce

Table 3.1. Global query parameters

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

Parameter	Type	Description
<b>fieldValidation</b>	<b>string</b>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: <ul style="list-style-type: none"> <li>- 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.</li> <li>- Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+</li> <li>- 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.</li> </ul>

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

create a LocalSubjectAccessReview

**Table 3.2. Body parameters**

Parameter	Type	Description
<b>body</b>	<b>LocalSubjectAccessReview</b> schema	

**Table 3.3. HTTP responses**

HTTP code	Response body
200 - OK	<b>LocalSubjectAccessReview</b> schema
201 - Created	<b>LocalSubjectAccessReview</b> schema
202 - Accepted	<b>LocalSubjectAccessReview</b> schema
401 - Unauthorized	Empty

## CHAPTER 4. RESOURCEACCESSREVIEW [AUTHORIZATION.OPENSIFT.IO/V1]

### Description

ResourceAccessReview is a means to request a list of which users and groups are authorized to perform the action specified by spec

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

### Type

**object**

### Required

- **namespace**
- **verb**
- **resourceAPIGroup**
- **resourceAPIVersion**
- **resource**
- **resourceName**
- **path**
- **isNonResourceURL**

## 4.1. SPECIFICATION

Property	Type	Description
<b>apiVersion</b>	<b>string</b>	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: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources</a>
<b>content</b>	<b>RawExtension</b>	Content is the actual content of the request for create and update
<b>isNonResourceURL</b>	<b>boolean</b>	IsNonResourceURL is true if this is a request for a non-resource URL (outside of the resource hierarchy)

Property	Type	Description
<b>kind</b>	<b>string</b>	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: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds</a>
<b>namespace</b>	<b>string</b>	Namespace is the namespace of the action being requested. Currently, there is no distinction between no namespace and all namespaces
<b>path</b>	<b>string</b>	Path is the path of a non resource URL
<b>resource</b>	<b>string</b>	Resource is one of the existing resource types
<b>resourceAPIGroup</b>	<b>string</b>	Group is the API group of the resource Serialized as resourceAPIGroup to avoid confusion with the 'groups' field when inlined
<b>resourceAPIVersion</b>	<b>string</b>	Version is the API version of the resource Serialized as resourceAPIVersion to avoid confusion with TypeMeta.apiVersion and ObjectMeta.resourceVersion when inlined
<b>resourceName</b>	<b>string</b>	ResourceName is the name of the resource being requested for a "get" or deleted for a "delete"
<b>verb</b>	<b>string</b>	Verb is one of: get, list, watch, create, update, delete

## 4.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/authorization.openshift.io/v1/resourceaccessreviews**

- **POST**: create a ResourceAccessReview

#### 4.2.1. /apis/authorization.openshift.io/v1/resourceaccessreviews

Table 4.1. Global query parameters

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

#### HTTP method

**POST**

#### Description

create a ResourceAccessReview

Table 4.2. Body parameters

Parameter	Type	Description
<b>body</b>	<a href="#">ResourceAccessReview</a> schema	

Table 4.3. HTTP responses

HTTP code	Reponse body
200 - OK	<b>ResourceAccessReview</b> schema
201 - Created	<b>ResourceAccessReview</b> schema
202 - Accepted	<b>ResourceAccessReview</b> schema
401 - Unauthorized	Empty

## CHAPTER 5. SELFSUBJECTRULESREVIEW [AUTHORIZATION.OPENSIFT.IO/V1]

### Description

SelfSubjectRulesReview is a resource you can create to determine which actions you can perform in a namespace

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

### Type

**object**

### Required

- **spec**

## 5.1. SPECIFICATION

Property	Type	Description
<b>apiVersion</b>	<b>string</b>	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: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources</a>
<b>kind</b>	<b>string</b>	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: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds</a>
<b>spec</b>	<b>object</b>	SelfSubjectRulesReviewSpec adds information about how to conduct the check
<b>status</b>	<b>object</b>	SubjectRulesReviewStatus is contains the result of a rules check



### 5.1.1. .spec

#### Description

SelfSubjectRulesReviewSpec adds information about how to conduct the check

#### Type

**object**

#### Required

- **scopes**

Property	Type	Description
<b>scopes</b>	<b>array (string)</b>	Scopes to use for the evaluation. Empty means "use the unscoped (full) permissions of the user/groups". Nil means "use the scopes on this request".

### 5.1.2. .status

#### Description

SubjectRulesReviewStatus is contains the result of a rules check

#### Type

**object**

#### Required

- **rules**

Property	Type	Description
<b>evaluationError</b>	<b>string</b>	EvaluationError can appear in combination with Rules. It means some error happened during evaluation that may have prevented additional rules from being populated.
<b>rules</b>	<b>array</b>	Rules is the list of rules (no particular sort) that are allowed for the subject
<b>rules[]</b>	<b>object</b>	PolicyRule holds information that describes a policy rule, but does not contain information about who the rule applies to or which namespace the rule applies to.

### 5.1.3. .status.rules

#### Description

Rules is the list of rules (no particular sort) that are allowed for the subject

#### Type

**array**

### 5.1.4. .status.rules[]

#### Description

PolicyRule holds information that describes a policy rule, but does not contain information about who the rule applies to or which namespace the rule applies to.

#### Type

**object**

#### Required

- **verbs**
- **resources**

Property	Type	Description
<b>apiGroups</b>	<b>array (string)</b>	APIGroups is the name of the APIGroup that contains the resources. If this field is empty, then both kubernetes and origin API groups are assumed. That means that if an action is requested against one of the enumerated resources in either the kubernetes or the origin API group, the request will be allowed
<b>attributeRestrictions</b>	<b>RawExtension</b>	AttributeRestrictions will vary depending on what the Authorizer/AuthorizationAttribute Builder pair supports. If the Authorizer does not recognize how to handle the AttributeRestrictions, the Authorizer should report an error.

Property	Type	Description
<b>nonResourceURLs</b>	<b>array (string)</b>	NonResourceURLsSlice is a set of partial urls that a user should have access to. *s are allowed, but only as the full, final step in the path This name is intentionally different than the internal type so that the DefaultConvert works nicely and because the ordering may be different.
<b>resourceNames</b>	<b>array (string)</b>	ResourceNames is an optional white list of names that the rule applies to. An empty set means that everything is allowed.
<b>resources</b>	<b>array (string)</b>	Resources is a list of resources this rule applies to. ResourceAll represents all resources.
<b>verbs</b>	<b>array (string)</b>	Verbs is a list of Verbs that apply to ALL the ResourceKinds and AttributeRestrictions contained in this rule. VerbAll represents all kinds.

## 5.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/authorization.openshift.io/v1/namespaces/{namespace}/selfsubjectrulesreviews**
  - **POST**: create a SelfSubjectRulesReview

### 5.2.1. /apis/authorization.openshift.io/v1/namespaces/{namespace}/selfsubjectrulesr

Table 5.1. Global query parameters

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

Parameter	Type	Description
<b>fieldValidation</b>	<b>string</b>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: <ul style="list-style-type: none"> <li>- 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.</li> <li>- Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+</li> <li>- 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.</li> </ul>

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

create a SelfSubjectRulesReview

Table 5.2. Body parameters

Parameter	Type	Description
<b>body</b>	<b>SelfSubjectRulesReview</b> schema	

Table 5.3. HTTP responses

HTTP code	Response body
200 - OK	<b>SelfSubjectRulesReview</b> schema
201 - Created	<b>SelfSubjectRulesReview</b> schema
202 - Accepted	<b>SelfSubjectRulesReview</b> schema
401 - Unauthorized	Empty

## CHAPTER 6. SUBJECTACCESSREVIEW [AUTHORIZATION.OPENSIFT.IO/V1]

### Description

SubjectAccessReview is an object for requesting information about whether a user or group can perform an action

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

### Type

**object**

### Required

- **namespace**
- **verb**
- **resourceAPIGroup**
- **resourceAPIVersion**
- **resource**
- **resourceName**
- **path**
- **isNonResourceURL**
- **user**
- **groups**
- **scopes**

## 6.1. SPECIFICATION

Property	Type	Description
<b>apiVersion</b>	<b>string</b>	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: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources</a>

Property	Type	Description
<b>content</b>	<b>RawExtension</b>	Content is the actual content of the request for create and update
<b>groups</b>	<b>array (string)</b>	GroupsSlice is optional. Groups is the list of groups to which the User belongs.
<b>isNonResourceURL</b>	<b>boolean</b>	IsNonResourceURL is true if this is a request for a non-resource URL (outside of the resource hierarchy)
<b>kind</b>	<b>string</b>	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: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds</a>
<b>namespace</b>	<b>string</b>	Namespace is the namespace of the action being requested. Currently, there is no distinction between no namespace and all namespaces
<b>path</b>	<b>string</b>	Path is the path of a non resource URL
<b>resource</b>	<b>string</b>	Resource is one of the existing resource types
<b>resourceAPIGroup</b>	<b>string</b>	Group is the API group of the resource Serialized as resourceAPIGroup to avoid confusion with the 'groups' field when inlined
<b>resourceAPIVersion</b>	<b>string</b>	Version is the API version of the resource Serialized as resourceAPIVersion to avoid confusion with TypeMeta.apiVersion and ObjectMeta.resourceVersion when inlined

Property	Type	Description
<b>resourceName</b>	<b>string</b>	ResourceName is the name of the resource being requested for a "get" or deleted for a "delete"
<b>scopes</b>	<b>array (string)</b>	Scopes to use for the evaluation. Empty means "use the unscoped (full) permissions of the user/groups". Nil for a self-SAR, means "use the scopes on this request". Nil for a regular SAR, means the same as empty.
<b>user</b>	<b>string</b>	User is optional. If both User and Groups are empty, the current authenticated user is used.
<b>verb</b>	<b>string</b>	Verb is one of: get, list, watch, create, update, delete

## 6.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/authorization.openshift.io/v1/subjectaccessreviews**
  - **POST**: create a SubjectAccessReview

### 6.2.1. /apis/authorization.openshift.io/v1/subjectaccessreviews

Table 6.1. Global query parameters

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

Parameter	Type	Description
<b>fieldValidation</b>	<b>string</b>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: <ul style="list-style-type: none"> <li>- 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.</li> <li>- Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+</li> <li>- 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.</li> </ul>

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

create a SubjectAccessReview

**Table 6.2. Body parameters**

Parameter	Type	Description
<b>body</b>	<b>SubjectAccessReview</b> schema	

**Table 6.3. HTTP responses**

HTTP code	Response body
200 - OK	<b>SubjectAccessReview</b> schema
201 - Created	<b>SubjectAccessReview</b> schema
202 - Accepted	<b>SubjectAccessReview</b> schema
401 - Unauthorized	Empty



## CHAPTER 7. SUBJECTRULESREVIEW [AUTHORIZATION.OPENSIFT.IO/V1]

### Description

SubjectRulesReview is a resource you can create to determine which actions another user can perform in a namespace

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

### Type

**object**

### Required

- **spec**

## 7.1. SPECIFICATION

Property	Type	Description
<b>apiVersion</b>	<b>string</b>	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: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources</a>
<b>kind</b>	<b>string</b>	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: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds</a>
<b>spec</b>	<b>object</b>	SubjectRulesReviewSpec adds information about how to conduct the check
<b>status</b>	<b>object</b>	SubjectRulesReviewStatus is contains the result of a rules check

### 7.1.1. .spec

#### Description

SubjectRulesReviewSpec adds information about how to conduct the check

#### Type

**object**

#### Required

- **user**
- **groups**
- **scopes**

Property	Type	Description
<b>groups</b>	<b>array (string)</b>	Groups is optional. Groups is the list of groups to which the User belongs. At least one of User and Groups must be specified.
<b>scopes</b>	<b>array (string)</b>	Scopes to use for the evaluation. Empty means "use the unscoped (full) permissions of the user/groups".
<b>user</b>	<b>string</b>	User is optional. At least one of User and Groups must be specified.

### 7.1.2. .status

#### Description

SubjectRulesReviewStatus is contains the result of a rules check

#### Type

**object**

#### Required

- **rules**

Property	Type	Description
<b>evaluationError</b>	<b>string</b>	EvaluationError can appear in combination with Rules. It means some error happened during evaluation that may have prevented additional rules from being populated.

Property	Type	Description
<b>rules</b>	<b>array</b>	Rules is the list of rules (no particular sort) that are allowed for the subject
<b>rules[]</b>	<b>object</b>	PolicyRule holds information that describes a policy rule, but does not contain information about who the rule applies to or which namespace the rule applies to.

### 7.1.3. .status.rules

#### Description

Rules is the list of rules (no particular sort) that are allowed for the subject

#### Type

**array**

### 7.1.4. .status.rules[]

#### Description

PolicyRule holds information that describes a policy rule, but does not contain information about who the rule applies to or which namespace the rule applies to.

#### Type

**object**

#### Required

- **verbs**
- **resources**

Property	Type	Description
<b>apiGroups</b>	<b>array (string)</b>	APIGroups is the name of the APIGroup that contains the resources. If this field is empty, then both kubernetes and origin API groups are assumed. That means that if an action is requested against one of the enumerated resources in either the kubernetes or the origin API group, the request will be allowed

Property	Type	Description
<b>attributeRestrictions</b>	<b>RawExtension</b>	AttributeRestrictions will vary depending on what the Authorizer/AuthorizationAttribute Builder pair supports. If the Authorizer does not recognize how to handle the AttributeRestrictions, the Authorizer should report an error.
<b>nonResourceURLs</b>	<b>array (string)</b>	NonResourceURLsSlice is a set of partial urls that a user should have access to. *s are allowed, but only as the full, final step in the path. This name is intentionally different than the internal type so that the DefaultConvert works nicely and because the ordering may be different.
<b>resourceNames</b>	<b>array (string)</b>	ResourceNames is an optional white list of names that the rule applies to. An empty set means that everything is allowed.
<b>resources</b>	<b>array (string)</b>	Resources is a list of resources this rule applies to. ResourceAll represents all resources.
<b>verbs</b>	<b>array (string)</b>	Verbs is a list of Verbs that apply to ALL the ResourceKinds and AttributeRestrictions contained in this rule. VerbAll represents all kinds.

## 7.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/authorization.openshift.io/v1/namespaces/{namespace}/subjectrulesreviews**
  - **POST**: create a SubjectRulesReview

### 7.2.1. /apis/authorization.openshift.io/v1/namespaces/{namespace}/subjectrulesreviews

Table 7.1. Global query parameters

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

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

create a SubjectRulesReview

**Table 7.2. Body parameters**

Parameter	Type	Description
<b>body</b>	<b>SubjectRulesReview</b> schema	

**Table 7.3. HTTP responses**

HTTP code	Response body
200 - OK	<b>SubjectRulesReview</b> schema
201 - Created	<b>SubjectRulesReview</b> schema

HTTP code	Response body
202 - Accepted	<b>SubjectRulesReview</b> schema
401 - Unauthorized	Empty

## CHAPTER 8. SELFSUBJECTREVIEW [AUTHENTICATION.K8S.IO/V1]

### Description

SelfSubjectReview contains the user information that the kube-apiserver has about the user making this request. When using impersonation, users will receive the user info of the user being impersonated. If impersonation or request header authentication is used, any extra keys will have their case ignored and returned as lowercase.

### Type

**object**

## 8.1. SPECIFICATION

Property	Type	Description
<b>apiVersion</b>	<b>string</b>	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: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources</a>
<b>kind</b>	<b>string</b>	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: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds</a>
<b>metadata</b>	<b>ObjectMeta</b>	Standard object's metadata. More info: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata</a>
<b>status</b>	<b>object</b>	SelfSubjectReviewStatus is filled by the kube-apiserver and sent back to a user.

### 8.1.1. .status

## Description

SelfSubjectReviewStatus is filled by the kube-apiserver and sent back to a user.

## Type

**object**

Property	Type	Description
<b>userInfo</b>	<b>object</b>	UserInfo holds the information about the user needed to implement the user.Info interface.

### 8.1.2. .status.userInfo

## Description

UserInfo holds the information about the user needed to implement the user.Info interface.

## Type

**object**

Property	Type	Description
<b>extra</b>	<b>object</b>	Any additional information provided by the authenticator.
<b>extra{}</b>	<b>array (string)</b>	
<b>groups</b>	<b>array (string)</b>	The names of groups this user is a part of.
<b>uid</b>	<b>string</b>	A unique value that identifies this user across time. If this user is deleted and another user by the same name is added, they will have different UIDs.
<b>username</b>	<b>string</b>	The name that uniquely identifies this user among all active users.

### 8.1.3. .status.userInfo.extra

## Description

Any additional information provided by the authenticator.

## Type

**object**

## 8.2. API ENDPOINTS

The following API endpoints are available:



- `/apis/authentication.k8s.io/v1/selfsubjectreviews`
  - **POST**: create a SelfSubjectReview

### 8.2.1. `/apis/authentication.k8s.io/v1/selfsubjectreviews`

Table 8.1. Global query parameters

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

#### HTTP method

**POST**

#### Description

create a SelfSubjectReview

Table 8.2. Body parameters

Parameter	Type	Description
<b>body</b>	<b>SelfSubjectReview</b> schema	

Table 8.3. HTTP responses

HTTP code	Reponse body
200 - OK	<b>SelfSubjectReview</b> schema
201 - Created	<b>SelfSubjectReview</b> schema
202 - Accepted	<b>SelfSubjectReview</b> schema
401 - Unauthorized	Empty

## CHAPTER 9. TOKENREQUEST [AUTHENTICATION.K8S.IO/V1]

### Description

TokenRequest requests a token for a given service account.

### Type

**object**

### Required

- **spec**

## 9.1. SPECIFICATION

Property	Type	Description
<b>apiVersion</b>	<b>string</b>	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: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources</a>
<b>kind</b>	<b>string</b>	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: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds</a>
<b>metadata</b>	<b>ObjectMeta</b>	Standard object's metadata. More info: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata</a>
<b>spec</b>	<b>object</b>	TokenRequestSpec contains client provided parameters of a token request.
<b>status</b>	<b>object</b>	TokenRequestStatus is the result of a token request.

### 9.1.1. .spec

#### Description

TokenRequestSpec contains client provided parameters of a token request.

#### Type

**object**

#### Required

- **audiences**

Property	Type	Description
<b>audiences</b>	<b>array (string)</b>	Audiences are the intended audiences of the token. A recipient of a token must identify themselves with an identifier in the list of audiences of the token, and otherwise should reject the token. A token issued for multiple audiences may be used to authenticate against any of the audiences listed but implies a high degree of trust between the target audiences.
<b>boundObjectRef</b>	<b>object</b>	BoundObjectReference is a reference to an object that a token is bound to.
<b>expirationSeconds</b>	<b>integer</b>	ExpirationSeconds is the requested duration of validity of the request. The token issuer may return a token with a different validity duration so a client needs to check the 'expiration' field in a response.

### 9.1.2. .spec.boundObjectRef

#### Description

BoundObjectReference is a reference to an object that a token is bound to.

#### Type

**object**

Property	Type	Description
<b>apiVersion</b>	<b>string</b>	API version of the referent.

Property	Type	Description
<b>kind</b>	<b>string</b>	Kind of the referent. Valid kinds are 'Pod' and 'Secret'.
<b>name</b>	<b>string</b>	Name of the referent.
<b>uid</b>	<b>string</b>	UID of the referent.

### 9.1.3. .status

#### Description

TokenRequestStatus is the result of a token request.

#### Type

**object**

#### Required

- **token**
- **expirationTimestamp**

Property	Type	Description
<b>expirationTimestamp</b>	<b>Time</b>	ExpirationTimestamp is the time of expiration of the returned token.
<b>token</b>	<b>string</b>	Token is the opaque bearer token.

## 9.2. API ENDPOINTS

The following API endpoints are available:

- **/api/v1/namespaces/{namespace}/serviceaccounts/{name}/token**
  - **POST**: create token of a ServiceAccount

### 9.2.1. /api/v1/namespaces/{namespace}/serviceaccounts/{name}/token

Table 9.1. Global path parameters

Parameter	Type	Description
<b>name</b>	<b>string</b>	name of the TokenRequest

Table 9.2. Global query parameters

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

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

create token of a ServiceAccount

Table 9.3. Body parameters

Parameter	Type	Description
<b>body</b>	<b>TokenRequest</b> schema	

Table 9.4. HTTP responses

HTTP code	Response body
200 - OK	<b>TokenRequest</b> schema
201 - Created	<b>TokenRequest</b> schema

HTTP code	Reponse body
202 - Accepted	<b>TokenRequest</b> schema
401 - Unauthorized	Empty

## CHAPTER 10. TOKENREVIEW [AUTHENTICATION.K8S.IO/V1]

### Description

TokenReview attempts to authenticate a token to a known user. Note: TokenReview requests may be cached by the webhook token authenticator plugin in the kube-apiserver.

### Type

**object**

### Required

- **spec**

## 10.1. SPECIFICATION

Property	Type	Description
<b>apiVersion</b>	<b>string</b>	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: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources</a>
<b>kind</b>	<b>string</b>	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: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds</a>
<b>metadata</b>	<b>ObjectMeta</b>	Standard object's metadata. More info: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata</a>
<b>spec</b>	<b>object</b>	TokenReviewSpec is a description of the token authentication request.



Property	Type	Description
<b>status</b>	<b>object</b>	TokenReviewStatus is the result of the token authentication request.

### 10.1.1. .spec

#### Description

TokenReviewSpec is a description of the token authentication request.

#### Type

**object**

Property	Type	Description
<b>audiences</b>	<b>array (string)</b>	Audiences is a list of the identifiers that the resource server presented with the token identifies as. Audience-aware token authenticators will verify that the token was intended for at least one of the audiences in this list. If no audiences are provided, the audience will default to the audience of the Kubernetes apiserver.
<b>token</b>	<b>string</b>	Token is the opaque bearer token.

### 10.1.2. .status

#### Description

TokenReviewStatus is the result of the token authentication request.

#### Type

**object**

Property	Type	Description
----------	------	-------------

Property	Type	Description
<b>audiences</b>	<b>array (string)</b>	Audiences are audience identifiers chosen by the authenticator that are compatible with both the TokenReview and token. An identifier is any identifier in the intersection of the TokenReviewSpec audiences and the token's audiences. A client of the TokenReview API that sets the spec.audiences field should validate that a compatible audience identifier is returned in the status.audiences field to ensure that the TokenReview server is audience aware. If a TokenReview returns an empty status.audience field where status.authenticated is "true", the token is valid against the audience of the Kubernetes API server.
<b>authenticated</b>	<b>boolean</b>	Authenticated indicates that the token was associated with a known user.
<b>error</b>	<b>string</b>	Error indicates that the token couldn't be checked
<b>user</b>	<b>object</b>	UserInfo holds the information about the user needed to implement the user.Info interface.

### 10.1.3. .status.user

#### Description

UserInfo holds the information about the user needed to implement the user.Info interface.

#### Type

**object**

Property	Type	Description
<b>extra</b>	<b>object</b>	Any additional information provided by the authenticator.
<b>extra{}</b>	<b>array (string)</b>	

Property	Type	Description
<b>groups</b>	<b>array (string)</b>	The names of groups this user is a part of.
<b>uid</b>	<b>string</b>	A unique value that identifies this user across time. If this user is deleted and another user by the same name is added, they will have different UIDs.
<b>username</b>	<b>string</b>	The name that uniquely identifies this user among all active users.

#### 10.1.4. .status.user.extra

##### Description

Any additional information provided by the authenticator.

##### Type

**object**

## 10.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/oauth.openshift.io/v1/tokenreviews**
  - **POST**: create a TokenReview
- **/apis/authentication.k8s.io/v1/tokenreviews**
  - **POST**: create a TokenReview

### 10.2.1. /apis/oauth.openshift.io/v1/tokenreviews

Table 10.1. Global query parameters

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

Parameter	Type	Description
<b>fieldValidation</b>	<b>string</b>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: <ul style="list-style-type: none"> <li>- 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.</li> <li>- Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+</li> <li>- 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.</li> </ul>

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

create a TokenReview

Table 10.2. Body parameters

Parameter	Type	Description
<b>body</b>	<b>TokenReview</b> schema	

Table 10.3. HTTP responses

HTTP code	Response body
200 - OK	<b>TokenReview</b> schema
201 - Created	<b>TokenReview</b> schema
202 - Accepted	<b>TokenReview</b> schema
401 - Unauthorized	Empty

**10.2.2. /apis/authentication.k8s.io/v1/tokenreviews**

Table 10.4. Global query parameters

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

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

create a TokenReview

**Table 10.5. Body parameters**

Parameter	Type	Description
<b>body</b>	<b>TokenReview</b> schema	

**Table 10.6. HTTP responses**

HTTP code	Response body
200 - OK	<b>TokenReview</b> schema
201 - Created	<b>TokenReview</b> schema
202 - Accepted	<b>TokenReview</b> schema

HTTP code	Reponse body
401 - Unauthorized	Empty

# CHAPTER 11. LOCALSUBJECTACCESSREVIEW [AUTHORIZATION.K8S.IO/V1]

## Description

LocalSubjectAccessReview checks whether or not a user or group can perform an action in a given namespace. Having a namespace scoped resource makes it much easier to grant namespace scoped policy that includes permissions checking.

## Type

**object**

## Required

- **spec**

## 11.1. SPECIFICATION

Property	Type	Description
<b>apiVersion</b>	<b>string</b>	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: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources</a>
<b>kind</b>	<b>string</b>	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: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds</a>
<b>metadata</b>	<b>ObjectMeta</b>	Standard list metadata. More info: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata</a>

Property	Type	Description
<b>spec</b>	<b>object</b>	SubjectAccessReviewSpec is a description of the access request. Exactly one of ResourceAuthorizationAttributes and NonResourceAuthorizationAttributes must be set
<b>status</b>	<b>object</b>	SubjectAccessReviewStatus

### 11.1.1. .spec

#### Description

SubjectAccessReviewSpec is a description of the access request. Exactly one of ResourceAuthorizationAttributes and NonResourceAuthorizationAttributes must be set

#### Type

**object**

Property	Type	Description
<b>extra</b>	<b>object</b>	Extra corresponds to the user.Info.GetExtra() method from the authenticator. Since that is input to the authorizer it needs a reflection here.
<b>extra{}</b>	<b>array (string)</b>	
<b>groups</b>	<b>array (string)</b>	Groups is the groups you're testing for.
<b>nonResourceAttributes</b>	<b>object</b>	NonResourceAttributes includes the authorization attributes available for non-resource requests to the Authorizer interface
<b>resourceAttributes</b>	<b>object</b>	ResourceAttributes includes the authorization attributes available for resource requests to the Authorizer interface
<b>uid</b>	<b>string</b>	UID information about the requesting user.



Property	Type	Description
<b>user</b>	<b>string</b>	User is the user you're testing for. If you specify "User" but not "Groups", then is it interpreted as "What if User were not a member of any groups"

### 11.1.2. .spec.extra

#### Description

Extra corresponds to the `user.Info.GetExtra()` method from the authenticator. Since that is input to the authorizer it needs a reflection here.

#### Type

**object**

### 11.1.3. .spec.nonResourceAttributes

#### Description

`NonResourceAttributes` includes the authorization attributes available for non-resource requests to the Authorizer interface

#### Type

**object**

Property	Type	Description
<b>path</b>	<b>string</b>	Path is the URL path of the request
<b>verb</b>	<b>string</b>	Verb is the standard HTTP verb

### 11.1.4. .spec.resourceAttributes

#### Description

`ResourceAttributes` includes the authorization attributes available for resource requests to the Authorizer interface

#### Type

**object**

Property	Type	Description
<b>group</b>	<b>string</b>	Group is the API Group of the Resource. "*" means all.

Property	Type	Description
<b>name</b>	<b>string</b>	Name is the name of the resource being requested for a "get" or deleted for a "delete". "" (empty) means all.
<b>namespace</b>	<b>string</b>	Namespace is the namespace of the action being requested. Currently, there is no distinction between no namespace and all namespaces "" (empty) is defaulted for LocalSubjectAccessReviews "" (empty) is empty for cluster-scoped resources "" (empty) means "all" for namespace scoped resources from a SubjectAccessReview or SelfSubjectAccessReview
<b>resource</b>	<b>string</b>	Resource is one of the existing resource types. "*" means all.
<b>subresource</b>	<b>string</b>	Subresource is one of the existing resource types. "" means none.
<b>verb</b>	<b>string</b>	Verb is a kubernetes resource API verb, like: get, list, watch, create, update, delete, proxy. "*" means all.
<b>version</b>	<b>string</b>	Version is the API Version of the Resource. "*" means all.

### 11.1.5. .status

#### Description

SubjectAccessReviewStatus

#### Type

**object**

#### Required

- **allowed**

Property	Type	Description
<b>allowed</b>	<b>boolean</b>	Allowed is required. True if the action would be allowed, false otherwise.
<b>denied</b>	<b>boolean</b>	Denied is optional. True if the action would be denied, otherwise false. If both allowed is false and denied is false, then the authorizer has no opinion on whether to authorize the action. Denied may not be true if Allowed is true.
<b>evaluationError</b>	<b>string</b>	EvaluationError is an indication that some error occurred during the authorization check. It is entirely possible to get an error and be able to continue determine authorization status in spite of it. For instance, RBAC can be missing a role, but enough roles are still present and bound to reason about the request.
<b>reason</b>	<b>string</b>	Reason is optional. It indicates why a request was allowed or denied.

## 11.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/authorization.k8s.io/v1/namespaces/{namespace}/localsubjectaccessreviews**
  - **POST**: create a LocalSubjectAccessReview

### 11.2.1. /apis/authorization.k8s.io/v1/namespaces/{namespace}/localsubjectaccessrevi

Table 11.1. Global query parameters

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

Parameter	Type	Description
<b>fieldValidation</b>	<b>string</b>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: <ul style="list-style-type: none"> <li>- 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.</li> <li>- Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+</li> <li>- 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.</li> </ul>

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

create a LocalSubjectAccessReview

Table 11.2. Body parameters

Parameter	Type	Description
<b>body</b>	<b>LocalSubjectAccessReview</b> schema	

Table 11.3. HTTP responses

HTTP code	Response body
200 - OK	<b>LocalSubjectAccessReview</b> schema
201 - Created	<b>LocalSubjectAccessReview</b> schema
202 - Accepted	<b>LocalSubjectAccessReview</b> schema
401 - Unauthorized	Empty

## CHAPTER 12. SELFSUBJECTACCESSREVIEW [AUTHORIZATION.K8S.IO/V1]

### Description

SelfSubjectAccessReview checks whether or the current user can perform an action. Not filling in a spec.namespace means "in all namespaces". Self is a special case, because users should always be able to check whether they can perform an action

### Type

**object**

### Required

- **spec**

## 12.1. SPECIFICATION

Property	Type	Description
<b>apiVersion</b>	<b>string</b>	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: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources</a>
<b>kind</b>	<b>string</b>	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: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds</a>
<b>metadata</b>	<b>ObjectMeta</b>	Standard list metadata. More info: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata</a>

Property	Type	Description
<b>spec</b>	<b>object</b>	SelfSubjectAccessReviewSpec is a description of the access request. Exactly one of ResourceAuthorizationAttributes and NonResourceAuthorizationAttributes must be set
<b>status</b>	<b>object</b>	SubjectAccessReviewStatus

### 12.1.1. .spec

#### Description

SelfSubjectAccessReviewSpec is a description of the access request. Exactly one of ResourceAuthorizationAttributes and NonResourceAuthorizationAttributes must be set

#### Type

**object**

Property	Type	Description
<b>nonResourceAttributes</b>	<b>object</b>	NonResourceAttributes includes the authorization attributes available for non-resource requests to the Authorizer interface
<b>resourceAttributes</b>	<b>object</b>	ResourceAttributes includes the authorization attributes available for resource requests to the Authorizer interface

### 12.1.2. .spec.nonResourceAttributes

#### Description

NonResourceAttributes includes the authorization attributes available for non-resource requests to the Authorizer interface

#### Type

**object**

Property	Type	Description
<b>path</b>	<b>string</b>	Path is the URL path of the request

Property	Type	Description
<b>verb</b>	<b>string</b>	Verb is the standard HTTP verb

### 12.1.3. .spec.resourceAttributes

#### Description

ResourceAttributes includes the authorization attributes available for resource requests to the Authorizer interface

#### Type

**object**

Property	Type	Description
<b>group</b>	<b>string</b>	Group is the API Group of the Resource. "*" means all.
<b>name</b>	<b>string</b>	Name is the name of the resource being requested for a "get" or deleted for a "delete". "" (empty) means all.
<b>namespace</b>	<b>string</b>	Namespace is the namespace of the action being requested. Currently, there is no distinction between no namespace and all namespaces "" (empty) is defaulted for LocalSubjectAccessReviews "" (empty) is empty for cluster-scoped resources "" (empty) means "all" for namespace scoped resources from a SubjectAccessReview or SelfSubjectAccessReview
<b>resource</b>	<b>string</b>	Resource is one of the existing resource types. "*" means all.
<b>subresource</b>	<b>string</b>	Subresource is one of the existing resource types. "" means none.
<b>verb</b>	<b>string</b>	Verb is a kubernetes resource API verb, like: get, list, watch, create, update, delete, proxy. "*" means all.
<b>version</b>	<b>string</b>	Version is the API Version of the Resource. "*" means all.

## 12.1.4. .status

### Description

SubjectAccessReviewStatus

### Type

**object**

### Required

- **allowed**

Property	Type	Description
<b>allowed</b>	<b>boolean</b>	Allowed is required. True if the action would be allowed, false otherwise.
<b>denied</b>	<b>boolean</b>	Denied is optional. True if the action would be denied, otherwise false. If both allowed is false and denied is false, then the authorizer has no opinion on whether to authorize the action. Denied may not be true if Allowed is true.
<b>evaluationError</b>	<b>string</b>	EvaluationError is an indication that some error occurred during the authorization check. It is entirely possible to get an error and be able to continue determine authorization status in spite of it. For instance, RBAC can be missing a role, but enough roles are still present and bound to reason about the request.
<b>reason</b>	<b>string</b>	Reason is optional. It indicates why a request was allowed or denied.

## 12.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/authorization.k8s.io/v1/selfsubjectaccessreviews**
  - **POST**: create a SelfSubjectAccessReview

### 12.2.1. /apis/authorization.k8s.io/v1/selfsubjectaccessreviews

Table 12.1. Global query parameters



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

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

create a SelfSubjectAccessReview

**Table 12.2. Body parameters**

Parameter	Type	Description
<b>body</b>	<b>SelfSubjectAccessReview</b> schema	

**Table 12.3. HTTP responses**

HTTP code	Response body
200 - OK	<b>SelfSubjectAccessReview</b> schema
201 - Created	<b>SelfSubjectAccessReview</b> schema
202 - Accepted	<b>SelfSubjectAccessReview</b> schema

HTTP code	Reponse body
401 - Unauthorized	Empty

## CHAPTER 13. SELFSUBJECTRULESREVIEW [AUTHORIZATION.K8S.IO/V1]

### Description

SelfSubjectRulesReview enumerates the set of actions the current user can perform within a namespace. The returned list of actions may be incomplete depending on the server's authorization mode, and any errors experienced during the evaluation. SelfSubjectRulesReview should be used by UIs to show/hide actions, or to quickly let an end user reason about their permissions. It should NOT Be used by external systems to drive authorization decisions as this raises confused deputy, cache lifetime/revocation, and correctness concerns. SubjectAccessReview, and LocalAccessReview are the correct way to defer authorization decisions to the API server.

### Type

**object**

### Required

- **spec**

## 13.1. SPECIFICATION

Property	Type	Description
<b>apiVersion</b>	<b>string</b>	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: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources</a>
<b>kind</b>	<b>string</b>	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: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds</a>
<b>metadata</b>	<b>ObjectMeta</b>	Standard list metadata. More info: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata</a>

Property	Type	Description
<b>spec</b>	<b>object</b>	SelfSubjectRulesReviewSpec defines the specification for SelfSubjectRulesReview.
<b>status</b>	<b>object</b>	SubjectRulesReviewStatus contains the result of a rules check. This check can be incomplete depending on the set of authorizers the server is configured with and any errors experienced during evaluation. Because authorization rules are additive, if a rule appears in a list it's safe to assume the subject has that permission, even if that list is incomplete.

### 13.1.1. .spec

#### Description

SelfSubjectRulesReviewSpec defines the specification for SelfSubjectRulesReview.

#### Type

**object**

Property	Type	Description
<b>namespace</b>	<b>string</b>	Namespace to evaluate rules for. Required.

### 13.1.2. .status

#### Description

SubjectRulesReviewStatus contains the result of a rules check. This check can be incomplete depending on the set of authorizers the server is configured with and any errors experienced during evaluation. Because authorization rules are additive, if a rule appears in a list it's safe to assume the subject has that permission, even if that list is incomplete.

#### Type

**object**

#### Required

- **resourceRules**
- **nonResourceRules**
- **incomplete**

Property	Type	Description
<b>evaluationError</b>	<b>string</b>	EvaluationError can appear in combination with Rules. It indicates an error occurred during rule evaluation, such as an authorizer that doesn't support rule evaluation, and that ResourceRules and/or NonResourceRules may be incomplete.
<b>incomplete</b>	<b>boolean</b>	Incomplete is true when the rules returned by this call are incomplete. This is most commonly encountered when an authorizer, such as an external authorizer, doesn't support rules evaluation.
<b>nonResourceRules</b>	<b>array</b>	NonResourceRules is the list of actions the subject is allowed to perform on non-resources. The list ordering isn't significant, may contain duplicates, and possibly be incomplete.
<b>nonResourceRules[]</b>	<b>object</b>	NonResourceRule holds information that describes a rule for the non-resource
<b>resourceRules</b>	<b>array</b>	ResourceRules is the list of actions the subject is allowed to perform on resources. The list ordering isn't significant, may contain duplicates, and possibly be incomplete.
<b>resourceRules[]</b>	<b>object</b>	ResourceRule is the list of actions the subject is allowed to perform on resources. The list ordering isn't significant, may contain duplicates, and possibly be incomplete.

### 13.1.3. .status.nonResourceRules

#### Description

NonResourceRules is the list of actions the subject is allowed to perform on non-resources. The list ordering isn't significant, may contain duplicates, and possibly be incomplete.

#### Type

**array****13.1.4. .status.nonResourceRules[]****Description**

NonResourceRule holds information that describes a rule for the non-resource

**Type**

**object**

**Required**

- **verbs**

Property	Type	Description
<b>nonResourceURLs</b>	<b>array (string)</b>	NonResourceURLs is a set of partial urls that a user should have access to. <b>s are allowed, but only as the full, final step in the path.</b> "" means all.
<b>verbs</b>	<b>array (string)</b>	Verb is a list of kubernetes non-resource API verbs, like: get, post, put, delete, patch, head, options. "*" means all.

**13.1.5. .status.resourceRules****Description**

ResourceRules is the list of actions the subject is allowed to perform on resources. The list ordering isn't significant, may contain duplicates, and possibly be incomplete.

**Type**

**array**

**13.1.6. .status.resourceRules[]****Description**

ResourceRule is the list of actions the subject is allowed to perform on resources. The list ordering isn't significant, may contain duplicates, and possibly be incomplete.

**Type**

**object**

**Required**

- **verbs**

Property	Type	Description
<b>apiGroups</b>	<b>array (string)</b>	APIGroups is the name of the APIGroup that contains the resources. If multiple API groups are specified, any action requested against one of the enumerated resources in any API group will be allowed. "*" means all.
<b>resourceNames</b>	<b>array (string)</b>	ResourceNames is an optional white list of names that the rule applies to. An empty set means that everything is allowed. "*" means all.
<b>resources</b>	<b>array (string)</b>	Resources is a list of resources this rule applies to. "" <b>means all in the specified apiGroups.</b> "/foo" represents the subresource 'foo' for all resources in the specified apiGroups.
<b>verbs</b>	<b>array (string)</b>	Verb is a list of kubernetes resource API verbs, like: get, list, watch, create, update, delete, proxy. "*" means all.

## 13.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/authorization.k8s.io/v1/selfsubjectrulesreviews**
  - **POST:** create a SelfSubjectRulesReview

### 13.2.1. /apis/authorization.k8s.io/v1/selfsubjectrulesreviews

Table 13.1. Global query parameters

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

Parameter	Type	Description
<b>fieldValidation</b>	<b>string</b>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: <ul style="list-style-type: none"> <li>- 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.</li> <li>- Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+</li> <li>- 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.</li> </ul>

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

create a SelfSubjectRulesReview

**Table 13.2. Body parameters**

Parameter	Type	Description
<b>body</b>	<a href="#">SelfSubjectRulesReview</a> schema	

**Table 13.3. HTTP responses**

HTTP code	Response body
200 - OK	<a href="#">SelfSubjectRulesReview</a> schema
201 - Created	<a href="#">SelfSubjectRulesReview</a> schema
202 - Accepted	<a href="#">SelfSubjectRulesReview</a> schema
401 - Unauthorized	Empty



## CHAPTER 14. SUBJECTACCESSREVIEW [AUTHORIZATION.K8S.IO/V1]

### Description

SubjectAccessReview checks whether or not a user or group can perform an action.

### Type

**object**

### Required

- **spec**

## 14.1. SPECIFICATION

Property	Type	Description
<b>apiVersion</b>	<b>string</b>	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: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources</a>
<b>kind</b>	<b>string</b>	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: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds</a>
<b>metadata</b>	<b>ObjectMeta</b>	Standard list metadata. More info: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata</a>

Property	Type	Description
<b>spec</b>	<b>object</b>	SubjectAccessReviewSpec is a description of the access request. Exactly one of ResourceAuthorizationAttributes and NonResourceAuthorizationAttributes must be set
<b>status</b>	<b>object</b>	SubjectAccessReviewStatus

### 14.1.1.1 .spec

#### Description

SubjectAccessReviewSpec is a description of the access request. Exactly one of ResourceAuthorizationAttributes and NonResourceAuthorizationAttributes must be set

#### Type

**object**

Property	Type	Description
<b>extra</b>	<b>object</b>	Extra corresponds to the user.Info.GetExtra() method from the authenticator. Since that is input to the authorizer it needs a reflection here.
<b>extra{}</b>	<b>array (string)</b>	
<b>groups</b>	<b>array (string)</b>	Groups is the groups you're testing for.
<b>nonResourceAttributes</b>	<b>object</b>	NonResourceAttributes includes the authorization attributes available for non-resource requests to the Authorizer interface
<b>resourceAttributes</b>	<b>object</b>	ResourceAttributes includes the authorization attributes available for resource requests to the Authorizer interface
<b>uid</b>	<b>string</b>	UID information about the requesting user.

Property	Type	Description
<b>user</b>	<b>string</b>	User is the user you're testing for. If you specify "User" but not "Groups", then is it interpreted as "What if User were not a member of any groups"

### 14.1.2. .spec.extra

#### Description

Extra corresponds to the `user.Info.GetExtra()` method from the authenticator. Since that is input to the authorizer it needs a reflection here.

#### Type

**object**

### 14.1.3. .spec.nonResourceAttributes

#### Description

NonResourceAttributes includes the authorization attributes available for non-resource requests to the Authorizer interface

#### Type

**object**

Property	Type	Description
<b>path</b>	<b>string</b>	Path is the URL path of the request
<b>verb</b>	<b>string</b>	Verb is the standard HTTP verb

### 14.1.4. .spec.resourceAttributes

#### Description

ResourceAttributes includes the authorization attributes available for resource requests to the Authorizer interface

#### Type

**object**

Property	Type	Description
<b>group</b>	<b>string</b>	Group is the API Group of the Resource. "*" means all.

Property	Type	Description
<b>name</b>	<b>string</b>	Name is the name of the resource being requested for a "get" or deleted for a "delete". "" (empty) means all.
<b>namespace</b>	<b>string</b>	Namespace is the namespace of the action being requested. Currently, there is no distinction between no namespace and all namespaces "" (empty) is defaulted for LocalSubjectAccessReviews "" (empty) is empty for cluster-scoped resources "" (empty) means "all" for namespace scoped resources from a SubjectAccessReview or SelfSubjectAccessReview
<b>resource</b>	<b>string</b>	Resource is one of the existing resource types. "*" means all.
<b>subresource</b>	<b>string</b>	Subresource is one of the existing resource types. "" means none.
<b>verb</b>	<b>string</b>	Verb is a kubernetes resource API verb, like: get, list, watch, create, update, delete, proxy. "*" means all.
<b>version</b>	<b>string</b>	Version is the API Version of the Resource. "*" means all.

### 14.1.5. .status

#### Description

SubjectAccessReviewStatus

#### Type

**object**

#### Required

- **allowed**

Property	Type	Description
----------	------	-------------

Property	Type	Description
<b>allowed</b>	<b>boolean</b>	Allowed is required. True if the action would be allowed, false otherwise.
<b>denied</b>	<b>boolean</b>	Denied is optional. True if the action would be denied, otherwise false. If both allowed is false and denied is false, then the authorizer has no opinion on whether to authorize the action. Denied may not be true if Allowed is true.
<b>evaluationError</b>	<b>string</b>	EvaluationError is an indication that some error occurred during the authorization check. It is entirely possible to get an error and be able to continue determine authorization status in spite of it. For instance, RBAC can be missing a role, but enough roles are still present and bound to reason about the request.
<b>reason</b>	<b>string</b>	Reason is optional. It indicates why a request was allowed or denied.

## 14.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/authorization.k8s.io/v1/subjectaccessreviews**
  - **POST**: create a SubjectAccessReview

### 14.2.1. /apis/authorization.k8s.io/v1/subjectaccessreviews

Table 14.1. Global query parameters

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

Parameter	Type	Description
<b>fieldValidation</b>	<b>string</b>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: <ul style="list-style-type: none"> <li>- 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.</li> <li>- Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+</li> <li>- 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.</li> </ul>

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

create a SubjectAccessReview

**Table 14.2. Body parameters**

Parameter	Type	Description
<b>body</b>	<b>SubjectAccessReview</b> w schema	

**Table 14.3. HTTP responses**

HTTP code	Response body
200 - OK	<b>SubjectAccessReview</b> schema
201 - Created	<b>SubjectAccessReview</b> schema
202 - Accepted	<b>SubjectAccessReview</b> schema
401 - Unauthorized	Empty

