



OpenShift Container Platform 4.20

RBAC APIs

Reference guide for RBAC APIs

OpenShift Container Platform 4.20 RBAC APIs

Reference guide for RBAC APIs

Legal Notice

Copyright © 2025 Red Hat, Inc.

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

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

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

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

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

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

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

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

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

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

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

All other trademarks are the property of their respective owners.

Abstract

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

Table of Contents

CHAPTER 1. RBAC APIS	3
1.1. CLUSTERROLEBINDING [RBAC.AUTHORIZATION.K8S.IO/V1]	3
1.2. CLUSTERROLE [RBAC.AUTHORIZATION.K8S.IO/V1]	3
1.3. ROLEBINDING [RBAC.AUTHORIZATION.K8S.IO/V1]	3
1.4. ROLE [RBAC.AUTHORIZATION.K8S.IO/V1]	3
CHAPTER 2. CLUSTERROLEBINDING [RBAC.AUTHORIZATION.K8S.IO/V1]	4
2.1. SPECIFICATION	4
2.1.1. .roleRef	5
2.1.2. .subjects	5
2.1.3. .subjects[]	5
2.2. API ENDPOINTS	6
2.2.1. /apis/rbac.authorization.k8s.io/v1/clusterrolebindings	7
2.2.2. /apis/rbac.authorization.k8s.io/v1/watch/clusterrolebindings	9
2.2.3. /apis/rbac.authorization.k8s.io/v1/clusterrolebindings/{name}	9
2.2.4. /apis/rbac.authorization.k8s.io/v1/watch/clusterrolebindings/{name}	12
CHAPTER 3. CLUSTERROLE [RBAC.AUTHORIZATION.K8S.IO/V1]	14
3.1. SPECIFICATION	14
3.1.1. .aggregationRule	15
3.1.2. .rules	15
3.1.3. .rules[]	15
3.2. API ENDPOINTS	16
3.2.1. /apis/rbac.authorization.k8s.io/v1/clusterroles	17
3.2.2. /apis/rbac.authorization.k8s.io/v1/watch/clusterroles	19
3.2.3. /apis/rbac.authorization.k8s.io/v1/clusterroles/{name}	19
3.2.4. /apis/rbac.authorization.k8s.io/v1/watch/clusterroles/{name}	22
CHAPTER 4. ROLEBINDING [RBAC.AUTHORIZATION.K8S.IO/V1]	24
4.1. SPECIFICATION	24
4.1.1. .roleRef	25
4.1.2. .subjects	25
4.1.3. .subjects[]	25
4.2. API ENDPOINTS	26
4.2.1. /apis/rbac.authorization.k8s.io/v1/rolebindings	27
4.2.2. /apis/rbac.authorization.k8s.io/v1/watch/rolebindings	27
4.2.3. /apis/rbac.authorization.k8s.io/v1/namespaces/{namespace}/rolebindings	28
4.2.4. /apis/rbac.authorization.k8s.io/v1/watch/namespaces/{namespace}/rolebindings	30
4.2.5. /apis/rbac.authorization.k8s.io/v1/namespaces/{namespace}/rolebindings/{name}	30
4.2.6. /apis/rbac.authorization.k8s.io/v1/watch/namespaces/{namespace}/rolebindings/{name}	33
CHAPTER 5. ROLE [RBAC.AUTHORIZATION.K8S.IO/V1]	35
5.1. SPECIFICATION	35
5.1.1. .rules	35
5.1.2. .rules[]	36
5.2. API ENDPOINTS	37
5.2.1. /apis/rbac.authorization.k8s.io/v1/roles	37
5.2.2. /apis/rbac.authorization.k8s.io/v1/watch/roles	38
5.2.3. /apis/rbac.authorization.k8s.io/v1/namespaces/{namespace}/roles	38
5.2.4. /apis/rbac.authorization.k8s.io/v1/watch/namespaces/{namespace}/roles	40
5.2.5. /apis/rbac.authorization.k8s.io/v1/namespaces/{namespace}/roles/{name}	41
5.2.6. /apis/rbac.authorization.k8s.io/v1/watch/namespaces/{namespace}/roles/{name}	44

CHAPTER 1. RBAC APIS

1.1. CLUSTERROLEBINDING [RBAC.AUTHORIZATION.K8S.IO/V1]

Description

ClusterRoleBinding references a ClusterRole, but not contain it. It can reference a ClusterRole in the global namespace, and adds who information via Subject.

Type

object

1.2. CLUSTERROLE [RBAC.AUTHORIZATION.K8S.IO/V1]

Description

ClusterRole is a cluster level, logical grouping of PolicyRules that can be referenced as a unit by a RoleBinding or ClusterRoleBinding.

Type

object

1.3. ROLEBINDING [RBAC.AUTHORIZATION.K8S.IO/V1]

Description

RoleBinding references a role, but does not contain it. It can reference a Role in the same namespace or a ClusterRole in the global namespace. It adds who information via Subjects and namespace information by which namespace it exists in. RoleBindings in a given namespace only have effect in that namespace.

Type

object

1.4. ROLE [RBAC.AUTHORIZATION.K8S.IO/V1]

Description

Role is a namespaced, logical grouping of PolicyRules that can be referenced as a unit by a RoleBinding.

Type

object

CHAPTER 2. CLUSTERROLEBINDING

[RBAC.AUTHORIZATION.K8S.IO/V1]

Description

ClusterRoleBinding references a ClusterRole, but not contain it. It can reference a ClusterRole in the global namespace, and adds who information via Subject.

Type

object

Required

- **roleRef**

2.1. SPECIFICATION

Property	Type	Description
apiVersion	string	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
kind	string	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ObjectMeta	Standard object's metadata.
roleRef	object	RoleRef contains information that points to the role being used
subjects	array	Subjects holds references to the objects the role applies to.

Property	Type	Description
subjects[]	object	Subject contains a reference to the object or user identities a role binding applies to. This can either hold a direct API object reference, or a value for non-objects such as user and group names.

2.1.1. .roleRef

Description

RoleRef contains information that points to the role being used

Type

object

Required

- **apiGroup**
- **kind**
- **name**

Property	Type	Description
apiGroup	string	APIGroup is the group for the resource being referenced
kind	string	Kind is the type of resource being referenced
name	string	Name is the name of resource being referenced

2.1.2. .subjects

Description

Subjects holds references to the objects the role applies to.

Type

array

2.1.3. .subjects[]

Description

Subject contains a reference to the object or user identities a role binding applies to. This can either hold a direct API object reference, or a value for non-objects such as user and group names.

Type

object

Required

- **kind**
- **name**

Property	Type	Description
apiGroup	string	APIGroup holds the API group of the referenced subject. Defaults to "" for ServiceAccount subjects. Defaults to "rbac.authorization.k8s.io" for User and Group subjects.
kind	string	Kind of object being referenced. Values defined by this API group are "User", "Group", and "ServiceAccount". If the Authorizer does not recognize the kind value, the Authorizer should report an error.
name	string	Name of the object being referenced.
namespace	string	Namespace of the referenced object. If the object kind is non-namespace, such as "User" or "Group", and this value is not empty the Authorizer should report an error.

2.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/rbac.authorization.k8s.io/v1/clusterrolebindings**
 - **DELETE**: delete collection of ClusterRoleBinding
 - **GET**: list or watch objects of kind ClusterRoleBinding
 - **POST**: create a ClusterRoleBinding
- **/apis/rbac.authorization.k8s.io/v1/watch/clusterrolebindings**
 - **GET**: watch individual changes to a list of ClusterRoleBinding. deprecated: use the 'watch' parameter with a list operation instead.

- **/apis/rbac.authorization.k8s.io/v1/clusterrolebindings/{name}**
 - **DELETE**: delete a ClusterRoleBinding
 - **GET**: read the specified ClusterRoleBinding
 - **PATCH**: partially update the specified ClusterRoleBinding
 - **PUT**: replace the specified ClusterRoleBinding
- **/apis/rbac.authorization.k8s.io/v1/watch/clusterrolebindings/{name}**
 - **GET**: watch changes to an object of kind ClusterRoleBinding. deprecated: use the 'watch' parameter with a list operation instead, filtered to a single item with the 'fieldSelector' parameter.

2.2.1. /apis/rbac.authorization.k8s.io/v1/clusterrolebindings

HTTP method

DELETE

Description

delete collection of ClusterRoleBinding

Table 2.1. Query parameters

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

Table 2.2. HTTP responses

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

HTTP method

GET

Description

list or watch objects of kind ClusterRoleBinding

Table 2.3. HTTP responses

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

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

create a ClusterRoleBinding

Table 2.4. Query parameters

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

Table 2.5. Body parameters

Parameter	Type	Description
body	ClusterRoleBinding schema	

Table 2.6. HTTP responses

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

2.2.2. /apis/rbac.authorization.k8s.io/v1/watch/clusterrolebindings

HTTP method

GET

Description

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

Table 2.7. HTTP responses

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

2.2.3. /apis/rbac.authorization.k8s.io/v1/clusterrolebindings/{name}

Table 2.8. Global path parameters

Parameter	Type	Description
name	string	name of the ClusterRoleBinding

HTTP method

DELETE

Description

delete a ClusterRoleBinding

Table 2.9. Query parameters

Parameter	Type	Description
-----------	------	-------------

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

Table 2.10. HTTP responses

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

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

read the specified ClusterRoleBinding

Table 2.11. HTTP responses

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

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

partially update the specified ClusterRoleBinding

Table 2.12. Query parameters

Parameter	Type	Description
-----------	------	-------------

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

Table 2.13. HTTP responses

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

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

replace the specified ClusterRoleBinding

Table 2.14. Query parameters

Parameter	Type	Description
-----------	------	-------------

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

Table 2.15. Body parameters

Parameter	Type	Description
body	ClusterRoleBinding schema	

Table 2.16. HTTP responses

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

2.2.4. /apis/rbac.authorization.k8s.io/v1/watch/clusterrolebindings/{name}

Table 2.17. Global path parameters

Parameter	Type	Description
name	string	name of the ClusterRoleBinding

HTTP method

GET

Description

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

Table 2.18. HTTP responses

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

CHAPTER 3. CLUSTERROLE

[RBAC.AUTHORIZATION.K8S.IO/V1]

Description

ClusterRole is a cluster level, logical grouping of PolicyRules that can be referenced as a unit by a RoleBinding or ClusterRoleBinding.

Type

object

3.1. SPECIFICATION

Property	Type	Description
aggregationRule	object	AggregationRule describes how to locate ClusterRoles to aggregate into the ClusterRole
apiVersion	string	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
kind	string	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ObjectMeta	Standard object's metadata.
rules	array	Rules holds all the PolicyRules for this ClusterRole
rules[]	object	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.

3.1.1. .aggregationRule

Description

AggregationRule describes how to locate ClusterRoles to aggregate into the ClusterRole

Type

object

Property	Type	Description
clusterRoleSelectors	array (LabelSelector)	ClusterRoleSelectors holds a list of selectors which will be used to find ClusterRoles and create the rules. If any of the selectors match, then the ClusterRole's permissions will be added

3.1.2. .rules

Description

Rules holds all the PolicyRules for this ClusterRole

Type

array

3.1.3. .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**

Property	Type	Description
apiGroups	array (string)	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. "" represents the core API group and "*" represents all API groups.

Property	Type	Description
nonResourceURLs	array (string)	NonResourceURLs 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. Since non-resource URLs are not namespaced, this field is only applicable for ClusterRoles referenced from a ClusterRoleBinding. Rules can either apply to API resources (such as "pods" or "secrets") or non-resource URL paths (such as "/api"), but not both.
resourceNames	array (string)	ResourceNames is an optional white list of names that the rule applies to. An empty set means that everything is allowed.
resources	array (string)	Resources is a list of resources this rule applies to. '*' represents all resources.
verbs	array (string)	Verbs is a list of Verbs that apply to ALL the ResourceKinds contained in this rule. '*' represents all verbs.

3.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/rbac.authorization.k8s.io/v1/clusterroles**
 - **DELETE:** delete collection of ClusterRole
 - **GET:** list or watch objects of kind ClusterRole
 - **POST:** create a ClusterRole
- **/apis/rbac.authorization.k8s.io/v1/watch/clusterroles**
 - **GET:** watch individual changes to a list of ClusterRole. deprecated: use the 'watch' parameter with a list operation instead.
- **/apis/rbac.authorization.k8s.io/v1/clusterroles/{name}**
 - **DELETE:** delete a ClusterRole

- **GET**: read the specified ClusterRole
- **PATCH**: partially update the specified ClusterRole
- **PUT**: replace the specified ClusterRole
- **/apis/rbac.authorization.k8s.io/v1/watch/clusterroles/{name}**
 - **GET**: watch changes to an object of kind ClusterRole. deprecated: use the 'watch' parameter with a list operation instead, filtered to a single item with the 'fieldSelector' parameter.

3.2.1. /apis/rbac.authorization.k8s.io/v1/clusterroles

HTTP method

DELETE

Description

delete collection of ClusterRole

Table 3.1. Query parameters

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

Table 3.2. HTTP responses

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

HTTP method

GET

Description

list or watch objects of kind ClusterRole

Table 3.3. HTTP responses

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

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

create a ClusterRole

Table 3.4. Query parameters

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

Table 3.5. Body parameters

Parameter	Type	Description
body	ClusterRole schema	

Table 3.6. HTTP responses

HTTP code	Response body
200 - OK	ClusterRole schema
201 - Created	ClusterRole schema
202 - Accepted	ClusterRole schema

HTTP code	Reponse body
401 - Unauthorized	Empty

3.2.2. /apis/rbac.authorization.k8s.io/v1/watch/clusterroles

HTTP method

GET

Description

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

Table 3.7. HTTP responses

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

3.2.3. /apis/rbac.authorization.k8s.io/v1/clusterroles/{name}

Table 3.8. Global path parameters

Parameter	Type	Description
name	string	name of the ClusterRole

HTTP method

DELETE

Description

delete a ClusterRole

Table 3.9. Query parameters

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

Table 3.10. HTTP responses

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

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

read the specified ClusterRole

Table 3.11. HTTP responses

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

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

partially update the specified ClusterRole

Table 3.12. Query parameters

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

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

Table 3.13. HTTP responses

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

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

replace the specified ClusterRole

Table 3.14. Query parameters

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

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

Table 3.15. Body parameters

Parameter	Type	Description
body	ClusterRole schema	

Table 3.16. HTTP responses

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

3.2.4. /apis/rbac.authorization.k8s.io/v1/watch/clusterroles/{name}

Table 3.17. Global path parameters

Parameter	Type	Description
name	string	name of the ClusterRole

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

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

Table 3.18. HTTP responses

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

CHAPTER 4. ROLEBINDING

[RBAC.AUTHORIZATION.K8S.IO/V1]

Description

RoleBinding references a role, but does not contain it. It can reference a Role in the same namespace or a ClusterRole in the global namespace. It adds who information via Subjects and namespace information by which namespace it exists in. RoleBindings in a given namespace only have effect in that namespace.

Type

object

Required

- **roleRef**

4.1. SPECIFICATION

Property	Type	Description
apiVersion	string	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
kind	string	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ObjectMeta	Standard object's metadata.
roleRef	object	RoleRef contains information that points to the role being used
subjects	array	Subjects holds references to the objects the role applies to.

Property	Type	Description
subjects[]	object	Subject contains a reference to the object or user identities a role binding applies to. This can either hold a direct API object reference, or a value for non-objects such as user and group names.

4.1.1. .roleRef

Description

RoleRef contains information that points to the role being used

Type

object

Required

- **apiGroup**
- **kind**
- **name**

Property	Type	Description
apiGroup	string	APIGroup is the group for the resource being referenced
kind	string	Kind is the type of resource being referenced
name	string	Name is the name of resource being referenced

4.1.2. .subjects

Description

Subjects holds references to the objects the role applies to.

Type

array

4.1.3. .subjects[]

Description

Subject contains a reference to the object or user identities a role binding applies to. This can either hold a direct API object reference, or a value for non-objects such as user and group names.

Type

object

Required

- **kind**
- **name**

Property	Type	Description
apiGroup	string	APIGroup holds the API group of the referenced subject. Defaults to "" for ServiceAccount subjects. Defaults to "rbac.authorization.k8s.io" for User and Group subjects.
kind	string	Kind of object being referenced. Values defined by this API group are "User", "Group", and "ServiceAccount". If the Authorizer does not recognize the kind value, the Authorizer should report an error.
name	string	Name of the object being referenced.
namespace	string	Namespace of the referenced object. If the object kind is non-namespace, such as "User" or "Group", and this value is not empty the Authorizer should report an error.

4.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/rbac.authorization.k8s.io/v1/rolebindings**
 - **GET**: list or watch objects of kind RoleBinding
- **/apis/rbac.authorization.k8s.io/v1/watch/rolebindings**
 - **GET**: watch individual changes to a list of RoleBinding. deprecated: use the 'watch' parameter with a list operation instead.
- **/apis/rbac.authorization.k8s.io/v1/namespaces/{namespace}/rolebindings**
 - **DELETE**: delete collection of RoleBinding

- **GET**: list or watch objects of kind RoleBinding
- **POST**: create a RoleBinding
- **/apis/rbac.authorization.k8s.io/v1/watch/namespaces/{namespace}/rolebindings**
 - **GET**: watch individual changes to a list of RoleBinding. deprecated: use the 'watch' parameter with a list operation instead.
- **/apis/rbac.authorization.k8s.io/v1/namespaces/{namespace}/rolebindings/{name}**
 - **DELETE**: delete a RoleBinding
 - **GET**: read the specified RoleBinding
 - **PATCH**: partially update the specified RoleBinding
 - **PUT**: replace the specified RoleBinding
- **/apis/rbac.authorization.k8s.io/v1/watch/namespaces/{namespace}/rolebindings/{name}**
 - **GET**: watch changes to an object of kind RoleBinding. deprecated: use the 'watch' parameter with a list operation instead, filtered to a single item with the 'fieldSelector' parameter.

4.2.1. /apis/rbac.authorization.k8s.io/v1/rolebindings

HTTP method

GET

Description

list or watch objects of kind RoleBinding

Table 4.1. HTTP responses

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

4.2.2. /apis/rbac.authorization.k8s.io/v1/watch/rolebindings

HTTP method

GET

Description

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

Table 4.2. HTTP responses

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

4.2.3. /apis/rbac.authorization.k8s.io/v1/namespaces/{namespace}/rolebindings

HTTP method

DELETE

Description

delete collection of RoleBinding

Table 4.3. Query parameters

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

Table 4.4. HTTP responses

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

HTTP method

GET

Description

list or watch objects of kind RoleBinding

Table 4.5. HTTP responses

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

HTTP method

POST

Description

create a RoleBinding

Table 4.6. Query parameters

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

Table 4.7. Body parameters

Parameter	Type	Description
body	RoleBinding schema	

Table 4.8. HTTP responses

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

4.2.4. /apis/rbac.authorization.k8s.io/v1/watch/namespaces/{namespace}/rolebindings

HTTP method

GET

Description

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

Table 4.9. HTTP responses

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

4.2.5. /apis/rbac.authorization.k8s.io/v1/namespaces/{namespace}/rolebindings/{name}

Table 4.10. Global path parameters

Parameter	Type	Description
name	string	name of the RoleBinding

HTTP method

DELETE

Description

delete a RoleBinding

Table 4.11. Query parameters

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

Table 4.12. HTTP responses

HTTP code	Response body
200 - OK	Status schema
202 - Accepted	Status schema

HTTP code	Reponse body
401 - Unauthorized	Empty

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

read the specified RoleBinding

Table 4.13. HTTP responses

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

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

partially update the specified RoleBinding

Table 4.14. Query parameters

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

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

Table 4.15. HTTP responses

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

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

replace the specified RoleBinding

Table 4.16. Query parameters

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

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

Table 4.17. Body parameters

Parameter	Type	Description
body	RoleBinding schema	

Table 4.18. HTTP responses

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

4.2.6. /apis/rbac.authorization.k8s.io/v1/watch/namespaces/{namespace}/rolebindings

Table 4.19. Global path parameters

Parameter	Type	Description
name	string	name of the RoleBinding

HTTP method

GET

Description

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

Table 4.20. HTTP responses

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

CHAPTER 5. ROLE [RBAC.AUTHORIZATION.K8S.IO/V1]

Description

Role is a namespaced, logical grouping of PolicyRules that can be referenced as a unit by a RoleBinding.

Type

object

5.1. SPECIFICATION

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

5.1.1. .rules

Description

Rules holds all the PolicyRules for this Role

Type**array****5.1.2. .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**

Property	Type	Description
apiGroups	array (string)	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. "" represents the core API group and "*" represents all API groups.
nonResourceURLs	array (string)	NonResourceURLs 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 Since non-resource URLs are not namespaced, this field is only applicable for ClusterRoles referenced from a ClusterRoleBinding. Rules can either apply to API resources (such as "pods" or "secrets") or non-resource URL paths (such as "/api"), but not both.
resourceNames	array (string)	ResourceNames is an optional white list of names that the rule applies to. An empty set means that everything is allowed.
resources	array (string)	Resources is a list of resources this rule applies to. '*' represents all resources.

Property	Type	Description
verbs	array (string)	Verbs is a list of Verbs that apply to ALL the ResourceKinds contained in this rule. '*' represents all verbs.

5.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/rbac.authorization.k8s.io/v1/roles**
 - **GET**: list or watch objects of kind Role
- **/apis/rbac.authorization.k8s.io/v1/watch/roles**
 - **GET**: watch individual changes to a list of Role. deprecated: use the 'watch' parameter with a list operation instead.
- **/apis/rbac.authorization.k8s.io/v1/namespaces/{namespace}/roles**
 - **DELETE**: delete collection of Role
 - **GET**: list or watch objects of kind Role
 - **POST**: create a Role
- **/apis/rbac.authorization.k8s.io/v1/watch/namespaces/{namespace}/roles**
 - **GET**: watch individual changes to a list of Role. deprecated: use the 'watch' parameter with a list operation instead.
- **/apis/rbac.authorization.k8s.io/v1/namespaces/{namespace}/roles/{name}**
 - **DELETE**: delete a Role
 - **GET**: read the specified Role
 - **PATCH**: partially update the specified Role
 - **PUT**: replace the specified Role
- **/apis/rbac.authorization.k8s.io/v1/watch/namespaces/{namespace}/roles/{name}**
 - **GET**: watch changes to an object of kind Role. deprecated: use the 'watch' parameter with a list operation instead, filtered to a single item with the 'fieldSelector' parameter.

5.2.1. /apis/rbac.authorization.k8s.io/v1/roles

HTTP method

GET**Description**

list or watch objects of kind Role

Table 5.1. HTTP responses

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

5.2.2. /apis/rbac.authorization.k8s.io/v1/watch/roles**HTTP method****GET****Description**

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

Table 5.2. HTTP responses

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

5.2.3. /apis/rbac.authorization.k8s.io/v1/namespaces/{namespace}/roles**HTTP method****DELETE****Description**

delete collection of Role

Table 5.3. Query parameters

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

Table 5.4. HTTP responses

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

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

list or watch objects of kind Role

Table 5.5. HTTP responses

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

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

create a Role

Table 5.6. Query parameters

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

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

Table 5.7. Body parameters

Parameter	Type	Description
body	Role schema	

Table 5.8. HTTP responses

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

5.2.4. /apis/rbac.authorization.k8s.io/v1/watch/namespaces/{namespace}/roles

HTTP method

GET

Description

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

Table 5.9. HTTP responses

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

5.2.5. /apis/rbac.authorization.k8s.io/v1/namespaces/{namespace}/roles/{name}

Table 5.10. Global path parameters

Parameter	Type	Description
name	string	name of the Role

HTTP method

DELETE

Description

delete a Role

Table 5.11. Query parameters

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

Table 5.12. HTTP responses

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

HTTP method

GET

Description

read the specified Role

Table 5.13. HTTP responses

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

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

partially update the specified Role

Table 5.14. Query parameters

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

Table 5.15. HTTP responses

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

HTTP code	Response body
401 - Unauthorized	Empty

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

replace the specified Role

Table 5.16. Query parameters

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

Table 5.17. Body parameters

Parameter	Type	Description
body	Role schema	

Table 5.18. HTTP responses

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

5.2.6. /apis/rbac.authorization.k8s.io/v1/watch/namespaces/{namespace}/roles/{name}

Table 5.19. Global path parameters

Parameter	Type	Description
name	string	name of the Role

HTTP method

GET

Description

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

Table 5.20. HTTP responses

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