OpenShift Container Platform 4.18

Schedule and quota APIs

Reference guide for schedule and quota APIs

Last Updated: 2025-05-15

OpenShift Container Platform 4.18 Schedule and quota APIs

Reference guide for schedule and quota APIs

Legal Notice

Abstract

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

Table of Contents

CHAPTER 1. SCHEDULE AND QUOTA APIS

1.1. APPLIEDCLUSTERRESOURCEQUOTA [QUOTA.OPENSHIFT.IO/V1]

Description

AppliedClusterResourceQuota mirrors ClusterResourceQuota at a project scope, for projection into a project. It allows a project-admin to know which ClusterResourceQuotas are applied to his project and their associated usage.
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. CLUSTERRESOURCEQUOTA [QUOTA.OPENSHIFT.IO/V1]

Description

ClusterResourceQuota mirrors ResourceQuota at a cluster scope. This object is easily convertible to synthetic ResourceQuota object to allow quota evaluation re-use.
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. FLOWSCHEMA [FLOWCONTROL.APISERVER.K8S.IO/V1]

Description

FlowSchema defines the schema of a group of flows. Note that a flow is made up of a set of inbound API requests with similar attributes and is identified by a pair of strings: the name of the FlowSchema and a "flow distinguisher".

Type

object

1.4. LIMITRANGE [V1]

Description

LimitRange sets resource usage limits for each kind of resource in a Namespace.

Type

object

1.5. PRIORITYCLASS [SCHEDULING.K8S.IO/V1]

Description

PriorityClass defines mapping from a priority class name to the priority integer value. The value can be any valid integer.

Type

object

1.6. PRIORITYLEVELCONFIGURATION [FLOWCONTROL.APISERVER.K8S.IO/V1]

Description

PriorityLevelConfiguration represents the configuration of a priority level.

Type

object

1.7. RESOURCEQUOTA [V1]

Description

ResourceQuota sets aggregate quota restrictions enforced per namespace

Type

object

CHAPTER 2. APPLIEDCLUSTERRESOURCEQUOTA [QUOTA.OPENSHIFT.IO/V1]

Description

AppliedClusterResourceQuota mirrors ClusterResourceQuota at a project scope, for projection into a project. It allows a project-admin to know which ClusterResourceQuotas are applied to his project and their associated usage.
Compatibility level 1: Stable within a major release for a minimum of 12 months or 3 minor releases (whichever is longer).

Type

object

Required

- **metadata**
- **spec**

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

Property	Type	Description
metadata	**ObjectMeta_v2**	metadata is the standard object's metadata. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata
spec	**object**	ClusterResourceQuotaSpec defines the desired quota restrictions
status	**object**	ClusterResourceQuotaStatus defines the actual enforced quota and its current usage

2.1.1. .spec

Description

ClusterResourceQuotaSpec defines the desired quota restrictions

Type

object

Required

- **selector**
- **quota**

Property	Type	Description
quota	**ResourceQuotaSpec_v2**	Quota defines the desired quota
selector	**object**	ClusterResourceQuotaSelector is used to select projects. At least one of LabelSelector or AnnotationSelector must present. If only one is present, it is the only selection criteria. If both are specified, the project must match both restrictions.

2.1.2. .spec.selector

Description

ClusterResourceQuotaSelector is used to select projects. At least one of LabelSelector or AnnotationSelector must present. If only one is present, it is the only selection criteria. If both are specified, the project must match both restrictions.

Type

object

Property	Type	Description
annotations	**object (string)**	AnnotationSelector is used to select projects by annotation.
labels	**LabelSelector_v4**	LabelSelector is used to select projects by label.

2.1.3. .status

Description

ClusterResourceQuotaStatus defines the actual enforced quota and its current usage

Type

object

Required

- **total**

Property	Type	Description
namespaces	**array**	Namespaces slices the usage by project. This division allows for quick resolution of deletion reconciliation inside of a single project without requiring a recalculation across all projects. This can be used to pull the deltas for a given project.
namespaces[]	**object**	ResourceQuotaStatusByNamespace gives status for a particular project
total	**ResourceQuotaStatus**	Total defines the actual enforced quota and its current usage across all projects

2.1.4. .status.namespaces

Description

Namespaces slices the usage by project. This division allows for quick resolution of deletion reconciliation inside of a single project without requiring a recalculation across all projects. This can be used to pull the deltas for a given project.

Type

array

2.1.5. .status.namespaces[]

Description

ResourceQuotaStatusByNamespace gives status for a particular project

Type

object

Required

- **namespace**
- **status**

Property	Type	Description
namespace	**string**	Namespace the project this status applies to
status	**ResourceQuotaStatus**	Status indicates how many resources have been consumed by this project

2.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/quota.openshift.io/v1/appliedclusterresourcequotas**
  - **GET**: list objects of kind AppliedClusterResourceQuota
- **/apis/quota.openshift.io/v1/namespaces/{namespace}/appliedclusterresourcequotas**
  - **GET**: list objects of kind AppliedClusterResourceQuota
- **/apis/quota.openshift.io/v1/namespaces/{namespace}/appliedclusterresourcequotas/{name}**
  - **GET**: read the specified AppliedClusterResourceQuota

2.2.1. /apis/quota.openshift.io/v1/appliedclusterresourcequotas

HTTP method

GET

Description

list objects of kind AppliedClusterResourceQuota

Table 2.1. HTTP responses

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

2.2.2. /apis/quota.openshift.io/v1/namespaces/{namespace}/appliedclusterresourcec

HTTP method

GET

Description

list objects of kind AppliedClusterResourceQuota

Table 2.2. HTTP responses

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

2.2.3. /apis/quota.openshift.io/v1/namespaces/{namespace}/appliedclusterresourcec

Table 2.3. Global path parameters

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

HTTP method

GET

Description

read the specified AppliedClusterResourceQuota

Table 2.4. HTTP responses

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

CHAPTER 3. CLUSTERRESOURCEQUOTA [QUOTA.OPENSHIFT.IO/V1]

Description

ClusterResourceQuota mirrors ResourceQuota at a cluster scope. This object is easily convertible to synthetic ResourceQuota object to allow quota evaluation re-use.
Compatibility level 1: Stable within a major release for a minimum of 12 months or 3 minor releases (whichever is longer).

Type

object

Required

- **metadata**
- **spec**

3.1. SPECIFICATION

Property	Type	Description
apiVersion	**string**	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
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. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata

Property	Type	Description
spec	**object**	Spec defines the desired quota
status	**object**	Status defines the actual enforced quota and its current usage

3.1.1. .spec

Description

Spec defines the desired quota

Type

object

Required

- **quota**
- **selector**

Property	Type	Description
quota	**object**	Quota defines the desired quota
selector	**object**	Selector is the selector used to match projects. It should only select active projects on the scale of dozens (though it can select many more less active projects). These projects will contend on object creation through this resource.

3.1.2. .spec.quota

Description

Quota defines the desired quota

Type

object

Property	Type	Description

Property	Type	Description
hard	**integer-or-string**	hard is the set of desired hard limits for each named resource. More info: https://kubernetes.io/docs/concepts/policy/resource-quotas/
scopeSelector	**object**	scopeSelector is also a collection of filters like scopes that must match each object tracked by a quota but expressed using ScopeSelectorOperator in combination with possible values. For a resource to match, both scopes AND scopeSelector (if specified in spec), must be matched.
scopes	**array (string)**	A collection of filters that must match each object tracked by a quota. If not specified, the quota matches all objects.

3.1.3. .spec.quota.scopeSelector

Description

scopeSelector is also a collection of filters like scopes that must match each object tracked by a quota but expressed using ScopeSelectorOperator in combination with possible values. For a resource to match, both scopes AND scopeSelector (if specified in spec), must be matched.

Type

object

Property	Type	Description
matchExpressions	**array**	A list of scope selector requirements by scope of the resources.
matchExpressions[]	**object**	A scoped-resource selector requirement is a selector that contains values, a scope name, and an operator that relates the scope name and values.

3.1.4. .spec.quota.scopeSelector.matchExpressions

Description

A list of scope selector requirements by scope of the resources.

Type

array

3.1.5. .spec.quota.scopeSelector.matchExpressions[]

Description

A scoped-resource selector requirement is a selector that contains values, a scope name, and an operator that relates the scope name and values.

Type

object

Required

- **operator**
- **scopeName**

Property	Type	Description
operator	**string**	Represents a scope's relationship to a set of values. Valid operators are In, NotIn, Exists, DoesNotExist.
scopeName	**string**	The name of the scope that the selector applies to.
values	**array (string)**	An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

3.1.6. .spec.selector

Description

Selector is the selector used to match projects. It should only select active projects on the scale of dozens (though it can select many more less active projects). These projects will contend on object creation through this resource.

Type

object

Property	Type	Description
annotations	**undefined (string)**	AnnotationSelector is used to select projects by annotation.

Property	Type	Description
labels	``	LabelSelector is used to select projects by label.

3.1.7. .status

Description

Status defines the actual enforced quota and its current usage

Type

object

Required

- **total**

Property	Type	Description
namespaces	``	Namespaces slices the usage by project. This division allows for quick resolution of deletion reconciliation inside of a single project without requiring a recalculation across all projects. This can be used to pull the deltas for a given project.
total	**object**	Total defines the actual enforced quota and its current usage across all projects

3.1.8. .status.total

Description

Total defines the actual enforced quota and its current usage across all projects

Type

object

Property	Type	Description
hard	**integer-or-string**	Hard is the set of enforced hard limits for each named resource. More info: https://kubernetes.io/docs/concepts/policy/resource-quotas/

Property	Type	Description
used	**integer-or-string**	Used is the current observed total usage of the resource in the namespace.

3.2. API ENDPOINTS

The following API endpoints are available:

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

3.2.1. /apis/quota.openshift.io/v1/clusterresourcequotas

HTTP method

DELETE

Description

delete collection of ClusterResourceQuota

Table 3.1. HTTP responses

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

HTTP method

GET

Description

list objects of kind ClusterResourceQuota

Table 3.2. HTTP responses

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

HTTP method

POST

Description

create a ClusterResourceQuota

Table 3.3. 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

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 3.4. Body parameters

Parameter	Type	Description
body	**ClusterResourceQuota** schema	

Table 3.5. HTTP responses

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

3.2.2. /apis/quota.openshift.io/v1/watch/clusterresourcequotas

HTTP method

GET

Description

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

Table 3.6. HTTP responses

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

3.2.3. /apis/quota.openshift.io/v1/clusterresourcequotas/{name}

Table 3.7. Global path parameters

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

HTTP method

DELETE

Description

delete a ClusterResourceQuota

Table 3.8. Query parameters

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

Table 3.9. HTTP responses

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

HTTP method

GET

Description

read the specified ClusterResourceQuota

Table 3.10. HTTP responses

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

HTTP method

PATCH

Description

partially update the specified ClusterResourceQuota

Table 3.11. 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
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 3.12. HTTP responses

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

HTTP method

PUT

Description

replace the specified ClusterResourceQuota

Table 3.13. 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 3.14. Body parameters

Parameter	Type	Description
body	**ClusterResourceQuota** schema	

Table 3.15. HTTP responses

HTTP code	Reponse body
200 - OK	**ClusterResourceQuota** schema
201 - Created	**ClusterResourceQuota** schema

HTTP code	Reponse body
401 - Unauthorized	Empty

3.2.4. /apis/quota.openshift.io/v1/watch/clusterresourcequotas/{name}

Table 3.16. Global path parameters

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

HTTP method

GET

Description

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

Table 3.17. HTTP responses

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

3.2.5. /apis/quota.openshift.io/v1/clusterresourcequotas/{name}/status

Table 3.18. Global path parameters

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

HTTP method

GET

Description

read status of the specified ClusterResourceQuota

Table 3.19. HTTP responses

HTTP code	Reponse body
200 - OK	**ClusterResourceQuota** schema

HTTP code	Reponse body
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update status of the specified ClusterResourceQuota

Table 3.20. 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 3.21. HTTP responses

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

HTTP method

PUT

Description

replace status of the specified ClusterResourceQuota

Table 3.22. 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 3.23. Body parameters

Parameter	Type	Description
body	**ClusterResourceQuota** schema	

Table 3.24. HTTP responses

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

CHAPTER 4. FLOWSCHEMA [FLOWCONTROL.APISERVER.K8S.IO/V1]

Description

FlowSchema defines the schema of a group of flows. Note that a flow is made up of a set of inbound API requests with similar attributes and is identified by a pair of strings: the name of the FlowSchema and a "flow distinguisher".

Type

object

4.1. SPECIFICATION

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

Property	Type	Description
status	**object**	FlowSchemaStatus represents the current state of a FlowSchema.

4.1.1. .spec

Description

FlowSchemaSpec describes how the FlowSchema's specification looks like.

Type

object

Required

- **priorityLevelConfiguration**

Property	Type	Description
distinguisherMethod	**object**	FlowDistinguisherMethod specifies the method of a flow distinguisher.
matchingPrecedence	**integer**	**matchingPrecedence** is used to choose among the FlowSchemas that match a given request. The chosen FlowSchema is among those with the numerically lowest (which we take to be logically highest) MatchingPrecedence. Each MatchingPrecedence value must be ranged in [1,10000]. Note that if the precedence is not specified, it will be set to 1000 as default.
priorityLevelConfiguration	**object**	PriorityLevelConfigurationReference contains information that points to the "request-priority" being used.
rules	**array**	**rules** describes which requests will match this flow schema. This FlowSchema matches a request if and only if at least one member of rules matches the request. if it is an empty slice, there will be no requests matching the FlowSchema.

Property	Type	Description
rules[]	**object**	PolicyRulesWithSubjects prescribes a test that applies to a request to an apiserver. The test considers the subject making the request, the verb being requested, and the resource to be acted upon. This PolicyRulesWithSubjects matches a request if and only if both (a) at least one member of subjects matches the request and (b) at least one member of resourceRules or nonResourceRules matches the request.

4.1.2. .spec.distinguisherMethod

Description

FlowDistinguisherMethod specifies the method of a flow distinguisher.

Type

object

Required

- **type**

Property	Type	Description
type	**string**	**type** is the type of flow distinguisher method The supported types are "ByUser" and "ByNamespace". Required.

4.1.3. .spec.priorityLevelConfiguration

Description

PriorityLevelConfigurationReference contains information that points to the "request-priority" being used.

Type

object

Required

- **name**

Property	Type	Description
name	**string**	**name** is the name of the priority level configuration being referenced Required.

4.1.4. .spec.rules

Description

rules describes which requests will match this flow schema. This FlowSchema matches a request if and only if at least one member of rules matches the request. if it is an empty slice, there will be no requests matching the FlowSchema.

Type

array

4.1.5. .spec.rules[]

Description

PolicyRulesWithSubjects prescribes a test that applies to a request to an apiserver. The test considers the subject making the request, the verb being requested, and the resource to be acted upon. This PolicyRulesWithSubjects matches a request if and only if both (a) at least one member of subjects matches the request and (b) at least one member of resourceRules or nonResourceRules matches the request.

Type

object

Required

- **subjects**

Property	Type	Description
nonResourceRules	**array**	**nonResourceRules** is a list of NonResourcePolicyRules that identify matching requests according to their verb and the target non-resource URL.
nonResourceRules[]	**object**	NonResourcePolicyRule is a predicate that matches non-resource requests according to their verb and the target non-resource URL. A NonResourcePolicyRule matches a request if and only if both (a) at least one member of verbs matches the request and (b) at least one member of nonResourceURLs matches the request.

Property	Type	Description
resourceRules	**array**	**resourceRules** is a slice of ResourcePolicyRules that identify matching requests according to their verb and the target resource. At least one of **resourceRules** and **nonResourceRules** has to be non-empty.
resourceRules[]	**object**	ResourcePolicyRule is a predicate that matches some resource requests, testing the request's verb and the target resource. A ResourcePolicyRule matches a resource request if and only if: (a) at least one member of verbs matches the request, (b) at least one member of apiGroups matches the request, (c) at least one member of resources matches the request, and (d) either (d1) the request does not specify a namespace (i.e., **Namespace==""**) and clusterScope is true or (d2) the request specifies a namespace and least one member of namespaces matches the request's namespace.

Property	Type	Description
subjects	**array**	subjects is the list of normal user, serviceaccount, or group that this rule cares about. There must be at least one member in this slice. A slice that includes both the system:authenticated and system:unauthenticated user groups matches every request. Required.
subjects[]	**object**	Subject matches the originator of a request, as identified by the request authentication system. There are three ways of matching an originator; by user, group, or service account.

4.1.6. .spec.rules[].nonResourceRules

Description

nonResourceRules is a list of NonResourcePolicyRules that identify matching requests according to their verb and the target non-resource URL.

Type

array

4.1.7. .spec.rules[].nonResourceRules[]

Description

NonResourcePolicyRule is a predicate that matches non-resource requests according to their verb and the target non-resource URL. A NonResourcePolicyRule matches a request if and only if both (a) at least one member of verbs matches the request and (b) at least one member of nonResourceURLs matches the request.

Type

object

Required

- **verbs**
- **nonResourceURLs**

Property	Type	Description

Property	Type	Description
nonResourceURLs	**array (string)**	**nonResourceURLs** is a set of url prefixes that a user should have access to and may not be empty. For example: - "/healthz" is legal - "/hea*" is illegal - "/hea" is legal but matches nothing - "/hea/**" also matches nothing - "/healthz/**" matches all per-component health checks. "*" matches all non-resource urls. if it is present, it must be the only entry. Required.
verbs	**array (string)**	**verbs** is a list of matching verbs and may not be empty. "*" matches all verbs. If it is present, it must be the only entry. Required.

4.1.8. .spec.rules[].resourceRules

Description

resourceRules is a slice of ResourcePolicyRules that identify matching requests according to their verb and the target resource. At least one of **resourceRules** and **nonResourceRules** has to be non-empty.

Type

array

4.1.9. .spec.rules[].resourceRules[]

Description

ResourcePolicyRule is a predicate that matches some resource requests, testing the request's verb and the target resource. A ResourcePolicyRule matches a resource request if and only if: (a) at least one member of verbs matches the request, (b) at least one member of apiGroups matches the request, (c) at least one member of resources matches the request, and (d) either (d1) the request does not specify a namespace (i.e., **Namespace==""**) and clusterScope is true or (d2) the request specifies a namespace and least one member of namespaces matches the request's namespace.

Type

object

Required

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

Property	Type	Description
apiGroups	**array (string)**	**apiGroups** is a list of matching API groups and may not be empty. "*" matches all API groups and, if present, must be the only entry. Required.
clusterScope	**boolean**	**clusterScope** indicates whether to match requests that do not specify a namespace (which happens either because the resource is not namespaced or the request targets all namespaces). If this field is omitted or false then the **namespaces** field must contain a non-empty list.
namespaces	**array (string)**	**namespaces** is a list of target namespaces that restricts matches. A request that specifies a target namespace matches only if either (a) this list contains that target namespace or (b) this list contains "**". Note that "**" matches any specified namespace but does not match a request that *does not specify* a namespace (see the **clusterScope** field for that). This list may be empty, but only if **clusterScope** is true.
resources	**array (string)**	**resources** is a list of matching resources (i.e., lowercase and plural) with, if desired, subresource. For example, ["services", "nodes/status"]. This list may not be empty. "*" matches all resources and, if present, must be the only entry. Required.
verbs	**array (string)**	**verbs** is a list of matching verbs and may not be empty. "*" matches all verbs and, if present, must be the only entry. Required.

4.1.10. .spec.rules[].subjects

Description

subjects is the list of normal user, serviceaccount, or group that this rule cares about. There must be at least one member in this slice. A slice that includes both the system:authenticated and system:unauthenticated user groups matches every request. Required.

Type

array

4.1.11. .spec.rules[].subjects[]

Description

Subject matches the originator of a request, as identified by the request authentication system. There are three ways of matching an originator; by user, group, or service account.

Type

object

Required

- **kind**

Property	Type	Description
group	**object**	GroupSubject holds detailed information for group-kind subject.
kind	**string**	**kind** indicates which one of the other fields is non-empty. Required
serviceAccount	**object**	ServiceAccountSubject holds detailed information for service-account-kind subject.
user	**object**	UserSubject holds detailed information for user-kind subject.

4.1.12. .spec.rules[].subjects[].group

Description

GroupSubject holds detailed information for group-kind subject.

Type

object

Required

- **name**

Property	Type	Description
name	**string**	name is the user group that matches, or "*" to match all user groups. See https://github.com/kubernetes/apiserver/blob/master/pkg/authentication/user/user.go for some well-known group names. Required.

4.1.13. .spec.rules[].subjects[].serviceAccount

Description

ServiceAccountSubject holds detailed information for service-account-kind subject.

Type

object

Required

- **namespace**
- **name**

Property	Type	Description
name	**string**	**name** is the name of matching ServiceAccount objects, or "*" to match regardless of name. Required.
namespace	**string**	**namespace** is the namespace of matching ServiceAccount objects. Required.

4.1.14. .spec.rules[].subjects[].user

Description

UserSubject holds detailed information for user-kind subject.

Type

object

Required

- **name**

Property	Type	Description
name	**string**	**name** is the username that matches, or "*" to match all usernames. Required.

4.1.15. .status

Description

FlowSchemaStatus represents the current state of a FlowSchema.

Type

object

Property	Type	Description
conditions	**array**	**conditions** is a list of the current states of FlowSchema.
conditions[]	**object**	FlowSchemaCondition describes conditions for a FlowSchema.

4.1.16. .status.conditions

Description

conditions is a list of the current states of FlowSchema.

Type

array

4.1.17. .status.conditions[]

Description

FlowSchemaCondition describes conditions for a FlowSchema.

Type

object

Property	Type	Description
lastTransitionTime	Time	**lastTransitionTime** is the last time the condition transitioned from one status to another.
message	**string**	**message** is a human-readable message indicating details about last transition.

Property	Type	Description
reason	**string**	**reason** is a unique, one-word, CamelCase reason for the condition's last transition.
status	**string**	**status** is the status of the condition. Can be True, False, Unknown. Required.
type	**string**	**type** is the type of the condition. Required.

4.2. API ENDPOINTS

The following API endpoints are available:

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

4.2.1. /apis/flowcontrol.apiserver.k8s.io/v1/flowschemas

HTTP method

DELETE

Description

delete collection of FlowSchema

Table 4.1. 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 4.2. HTTP responses

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

HTTP method

GET

Description

list or watch objects of kind FlowSchema

Table 4.3. HTTP responses

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

HTTP method

POST

Description

create a FlowSchema

Table 4.4. 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 4.5. Body parameters

Parameter	Type	Description
body	**FlowSchema** schema	

Table 4.6. HTTP responses

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

4.2.2. /apis/flowcontrol.apiserver.k8s.io/v1/watch/flowschemas

HTTP method

GET

Description

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

Table 4.7. HTTP responses

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

4.2.3. /apis/flowcontrol.apiserver.k8s.io/v1/flowschemas/{name}

Table 4.8. Global path parameters

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

HTTP method

DELETE

Description

delete a FlowSchema

Table 4.9. 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 4.10. HTTP responses

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

HTTP method

GET

Description

read the specified FlowSchema

Table 4.11. HTTP responses

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

HTTP method

PATCH

Description

partially update the specified FlowSchema

Table 4.12. Query parameters

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

Table 4.13. HTTP responses

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

HTTP method

PUT

Description

replace the specified FlowSchema

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
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. Body parameters

Parameter	Type	Description
body	**FlowSchema** schema	

Table 4.16. HTTP responses

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

4.2.4. /apis/flowcontrol.apiserver.k8s.io/v1/watch/flowschemas/{name}

Table 4.17. Global path parameters

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

HTTP method

GET

Description

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

Table 4.18. HTTP responses

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

4.2.5. /apis/flowcontrol.apiserver.k8s.io/v1/flowschemas/{name}/status

Table 4.19. Global path parameters

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

HTTP method

GET

Description

read status of the specified FlowSchema

Table 4.20. HTTP responses

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

HTTP method

PATCH

Description

partially update status of the specified FlowSchema

Table 4.21. Query parameters

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

HTTP code	Reponse body
200 - OK	**FlowSchema** schema
201 - Created	**FlowSchema** schema

HTTP code	Reponse body
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace status of the specified FlowSchema

Table 4.23. 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 4.24. Body parameters

Parameter	Type	Description
body	**FlowSchema** schema	

Table 4.25. HTTP responses

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

CHAPTER 5. LIMITRANGE [V1]

Description

LimitRange sets resource usage limits for each kind of resource in a Namespace.

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. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata
spec	**object**	LimitRangeSpec defines a min/max usage limit for resources that match on kind.

5.1.1. .spec

Description

LimitRangeSpec defines a min/max usage limit for resources that match on kind.

Type

object

Required

- **limits**

Property	Type	Description
limits	**array**	Limits is the list of LimitRangeItem objects that are enforced.
limits[]	**object**	LimitRangeItem defines a min/max usage limit for any resource that matches on kind.

5.1.2. .spec.limits

Description

Limits is the list of LimitRangeItem objects that are enforced.

Type

array

5.1.3. .spec.limits[]

Description

LimitRangeItem defines a min/max usage limit for any resource that matches on kind.

Type

object

Required

- **type**

Property	Type	Description
default	**object (Quantity)**	Default resource requirement limit value by resource name if resource limit is omitted.
defaultRequest	**object (Quantity)**	DefaultRequest is the default resource requirement request value by resource name if resource request is omitted.
max	**object (Quantity)**	Max usage constraints on this kind by resource name.

Property	Type	Description
maxLimitRequestRatio	**object (Quantity)**	MaxLimitRequestRatio if specified, the named resource must have a request and limit that are both non-zero where limit divided by request is less than or equal to the enumerated value; this represents the max burst for the named resource.
min	**object (Quantity)**	Min usage constraints on this kind by resource name.
type	**string**	Type of resource that this limit applies to.

5.2. API ENDPOINTS

The following API endpoints are available:

- **/api/v1/limitranges**
  - **GET**: list or watch objects of kind LimitRange
- **/api/v1/watch/limitranges**
  - **GET**: watch individual changes to a list of LimitRange. deprecated: use the 'watch' parameter with a list operation instead.
- **/api/v1/namespaces/{namespace}/limitranges**
  - **DELETE**: delete collection of LimitRange
  - **GET**: list or watch objects of kind LimitRange
  - **POST**: create a LimitRange
- **/api/v1/watch/namespaces/{namespace}/limitranges**
  - **GET**: watch individual changes to a list of LimitRange. deprecated: use the 'watch' parameter with a list operation instead.
- **/api/v1/namespaces/{namespace}/limitranges/{name}**
  - **DELETE**: delete a LimitRange
  - **GET**: read the specified LimitRange
  - **PATCH**: partially update the specified LimitRange
  - **PUT**: replace the specified LimitRange
- **/api/v1/watch/namespaces/{namespace}/limitranges/{name}**

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

5.2.1. /api/v1/limitranges

HTTP method

GET

Description

list or watch objects of kind LimitRange

Table 5.1. HTTP responses

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

5.2.2. /api/v1/watch/limitranges

HTTP method

GET

Description

watch individual changes to a list of LimitRange. 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. /api/v1/namespaces/{namespace}/limitranges

HTTP method

DELETE

Description

delete collection of LimitRange

Table 5.3. 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 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 LimitRange

Table 5.5. HTTP responses

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

HTTP method

POST

Description

create a LimitRange

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	**LimitRange** schema	

Table 5.8. HTTP responses

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

5.2.4. /api/v1/watch/namespaces/{namespace}/limitranges

HTTP method

GET

Description

watch individual changes to a list of LimitRange. 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. /api/v1/namespaces/{namespace}/limitranges/{name}

Table 5.10. Global path parameters

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

HTTP method

DELETE

Description

delete a LimitRange

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 LimitRange

Table 5.13. HTTP responses

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

HTTP method

PATCH

Description

partially update the specified LimitRange

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	Reponse body
200 - OK	**LimitRange** schema
201 - Created	**LimitRange** schema

HTTP code	Reponse body
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace the specified LimitRange

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	**LimitRange** schema	

Table 5.18. HTTP responses

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

5.2.6. /api/v1/watch/namespaces/{namespace}/limitranges/{name}

Table 5.19. Global path parameters

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

HTTP method

GET

Description

watch changes to an object of kind LimitRange. 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

CHAPTER 6. PRIORITYCLASS [SCHEDULING.K8S.IO/V1]

Description

PriorityClass defines mapping from a priority class name to the priority integer value. The value can be any valid integer.

Type

object

Required

- **value**

6.1. SPECIFICATION

Property	Type	Description
apiVersion	**string**	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
description	**string**	description is an arbitrary string that usually provides guidelines on when this priority class should be used.
globalDefault	**boolean**	globalDefault specifies whether this PriorityClass should be considered as the default priority for pods that do not have any priority class. Only one PriorityClass can be marked as **globalDefault**. However, if more than one PriorityClasses exists with their **globalDefault** field set to true, the smallest value of such global default PriorityClasses will be used as the default priority.

Property	Type	Description
kind	**string**	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	**ObjectMeta**	Standard object's metadata. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata
preemptionPolicy	**string**	preemptionPolicy is the Policy for preempting pods with lower priority. One of Never, PreemptLowerPriority. Defaults to PreemptLowerPriority if unset. Possible enum values: - **"Never"** means that pod never preempts other pods with lower priority. - **"PreemptLowerPriority"** means that pod can preempt other pods with lower priority.
value	**integer**	value represents the integer value of this priority class. This is the actual priority that pods receive when they have the name of this class in their pod spec.

6.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/scheduling.k8s.io/v1/priorityclasses**
  - **DELETE**: delete collection of PriorityClass
  - **GET**: list or watch objects of kind PriorityClass
  - **POST**: create a PriorityClass

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

6.2.1. /apis/scheduling.k8s.io/v1/priorityclasses

HTTP method

DELETE

Description

delete collection of PriorityClass

Table 6.1. 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 6.2. HTTP responses

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

HTTP method

GET

Description

list or watch objects of kind PriorityClass

Table 6.3. HTTP responses

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

HTTP method

POST

Description

create a PriorityClass

Table 6.4. 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 6.5. Body parameters

Parameter	Type	Description
body	**PriorityClass** schema	

Table 6.6. HTTP responses

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

6.2.2. /apis/scheduling.k8s.io/v1/watch/priorityclasses

HTTP method

GET

Description

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

Table 6.7. HTTP responses

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

6.2.3. /apis/scheduling.k8s.io/v1/priorityclasses/{name}

Table 6.8. Global path parameters

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

HTTP method

DELETE

Description

delete a PriorityClass

Table 6.9. 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 6.10. HTTP responses

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

HTTP method

GET

Description

read the specified PriorityClass

Table 6.11. HTTP responses

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

HTTP method

PATCH

Description

partially update the specified PriorityClass

Table 6.12. 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 6.13. HTTP responses

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

HTTP method

PUT

Description

replace the specified PriorityClass

Table 6.14. 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 6.15. Body parameters

Parameter	Type	Description
body	**PriorityClass** schema	

Table 6.16. HTTP responses

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

6.2.4. /apis/scheduling.k8s.io/v1/watch/priorityclasses/{name}

Table 6.17. Global path parameters

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

HTTP method

GET

Description

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

Table 6.18. HTTP responses

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

CHAPTER 7. PRIORITYLEVELCONFIGURATION [FLOWCONTROL.APISERVER.K8S.IO/V1]

Description

PriorityLevelConfiguration represents the configuration of a priority level.

Type

object

7.1. SPECIFICATION

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

7.1.1. .spec

Description

PriorityLevelConfigurationSpec specifies the configuration of a priority level.

Type

object

Required

- **type**

Property	Type	Description
exempt	**object**	ExemptPriorityLevelConfiguration describes the configurable aspects of the handling of exempt requests. In the mandatory exempt configuration object the values in the fields here can be modified by authorized users, unlike the rest of the **spec**.
limited	**object**	LimitedPriorityLevelConfiguration specifies how to handle requests that are subject to limits. It addresses two issues: - How are requests for this priority level limited? - What should be done with requests that exceed the limit?
type	**string**	**type** indicates whether this priority level is subject to limitation on request execution. A value of **"Exempt"** means that requests of this priority level are not subject to a limit (and thus are never queued) and do not detract from the capacity made available to other priority levels. A value of **"Limited"** means that (a) requests of this priority level *are* subject to limits and (b) some of the server's limited capacity is made available exclusively to this priority level. Required.

7.1.2. .spec.exempt

Description

ExemptPriorityLevelConfiguration describes the configurable aspects of the handling of exempt requests. In the mandatory exempt configuration object the values in the fields here can be modified by authorized users, unlike the rest of the **spec**.

Type

object

Property	Type	Description
lendablePercent	**integer**	**lendablePercent** prescribes the fraction of the level's NominalCL that can be borrowed by other priority levels. This value of this field must be between 0 and 100, inclusive, and it defaults to 0. The number of seats that other levels can borrow from this level, known as this level's LendableConcurrencyLimit (LendableCL), is defined as follows. LendableCL(i) = round(NominalCL(i) * lendablePercent(i)/100.0)
nominalConcurrencyShares	**integer**	**nominalConcurrencyShares** (NCS) contributes to the computation of the NominalConcurrencyLimit (NominalCL) of this level. This is the number of execution seats nominally reserved for this priority level. This DOES NOT limit the dispatching from this priority level but affects the other priority levels through the borrowing mechanism. The server's concurrency limit (ServerCL) is divided among all the priority levels in proportion to their NCS values: NominalCL(i) = ceil(ServerCL * NCS(i) / sum_ncs) sum_ncs = sum[priority level k] NCS(k) Bigger numbers mean a larger nominal concurrency limit, at the expense of every other priority level. This field has a default value of zero.

7.1.3. .spec.limited

Description

LimitedPriorityLevelConfiguration specifies how to handle requests that are subject to limits. It addresses two issues: - How are requests for this priority level limited? - What should be done with requests that exceed the limit?

Type

object

Property	Type	Description
borrowingLimitPercent	**integer**	**borrowingLimitPercent**, if present, configures a limit on how many seats this priority level can borrow from other priority levels. The limit is known as this level's BorrowingConcurrencyLimit (BorrowingCL) and is a limit on the total number of seats that this level may borrow at any one time. This field holds the ratio of that limit to the level's nominal concurrency limit. When this field is non-nil, it must hold a non-negative integer and the limit is calculated as follows. BorrowingCL(i) = round(NominalCL(i) * borrowingLimitPercent(i)/100.0) The value of this field can be more than 100, implying that this priority level can borrow a number of seats that is greater than its own nominal concurrency limit (NominalCL). When this field is left **nil**, the limit is effectively infinite.
lendablePercent	**integer**	**lendablePercent** prescribes the fraction of the level's NominalCL that can be borrowed by other priority levels. The value of this field must be between 0 and 100, inclusive, and it defaults to 0. The number of seats that other levels can borrow from this level, known as this level's LendableConcurrencyLimit (LendableCL), is defined as follows. LendableCL(i) = round(NominalCL(i) * lendablePercent(i)/100.0)

Property	Type	Description
limitResponse	**object**	LimitResponse defines how to handle requests that can not be executed right now.

Property	Type	Description
nominalConcurrencyShares	**integer**	**nominalConcurrencyShares** (NCS) contributes to the computation of the NominalConcurrencyLimit (NominalCL) of this level. This is the number of execution seats available at this priority level. This is used both for requests dispatched from this priority level as well as requests dispatched from other priority levels borrowing seats from this level. The server's concurrency limit (ServerCL) is divided among the Limited priority levels in proportion to their NCS values: NominalCL(i) = ceil(ServerCL * NCS(i) / sum_ncs) sum_ncs = sum[priority level k] NCS(k) Bigger numbers mean a larger nominal concurrency limit, at the expense of every other priority level. If not specified, this field defaults to a value of 30. Setting this field to zero supports the construction of a "jail" for this priority level that is used to hold some request(s)

7.1.4. .spec.limited.limitResponse

Description

LimitResponse defines how to handle requests that can not be executed right now.

Type

object

Required

- **type**

Property	Type	Description
queuing	**object**	QueuingConfiguration holds the configuration parameters for queuing

Property	Type	Description
type	**string**	**type** is "Queue" or "Reject". "Queue" means that requests that can not be executed upon arrival are held in a queue until they can be executed or a queuing limit is reached. "Reject" means that requests that can not be executed upon arrival are rejected. Required.

7.1.5. .spec.limited.limitResponse.queuing

Description

QueuingConfiguration holds the configuration parameters for queuing

Type

object

Property	Type	Description
handSize	**integer**	**handSize** is a small positive number that configures the shuffle sharding of requests into queues. When enqueuing a request at this priority level the request's flow identifier (a string pair) is hashed and the hash value is used to shuffle the list of queues and deal a hand of the size specified here. The request is put into one of the shortest queues in that hand. **handSize** must be no larger than **queues**, and should be significantly smaller (so that a few heavy flows do not saturate most of the queues). See the user-facing documentation for more extensive guidance on setting this field. This field has a default value of 8.
queueLengthLimit	**integer**	**queueLengthLimit** is the maximum number of requests allowed to be waiting in a given queue of this priority level at a time; excess requests are rejected. This value must be positive. If not specified, it will be defaulted to 50.

Property	Type	Description
queues	**integer**	**queues** is the number of queues for this priority level. The queues exist independently at each apiserver. The value must be positive. Setting it to 1 effectively precludes shufflesharding and thus makes the distinguisher method of associated flow schemas irrelevant. This field has a default value of 64.

7.1.6. .status

Description

PriorityLevelConfigurationStatus represents the current state of a "request-priority".

Type

object

Property	Type	Description
conditions	**array**	**conditions** is the current state of "request-priority".
conditions[]	**object**	PriorityLevelConfigurationCondition defines the condition of priority level.

7.1.7. .status.conditions

Description

conditions is the current state of "request-priority".

Type

array

7.1.8. .status.conditions[]

Description

PriorityLevelConfigurationCondition defines the condition of priority level.

Type

object

Property	Type	Description

Property	Type	Description
lastTransitionTime	Time	**lastTransitionTime** is the last time the condition transitioned from one status to another.
message	**string**	**message** is a human-readable message indicating details about last transition.
reason	**string**	**reason** is a unique, one-word, CamelCase reason for the condition's last transition.
status	**string**	**status** is the status of the condition. Can be True, False, Unknown. Required.
type	**string**	**type** is the type of the condition. Required.

7.2. API ENDPOINTS

The following API endpoints are available:

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

'fieldSelector' parameter.

- **/apis/flowcontrol.apiserver.k8s.io/v1/prioritylevelconfigurations/{name}/status**
  - **GET**: read status of the specified PriorityLevelConfiguration
  - **PATCH**: partially update status of the specified PriorityLevelConfiguration
  - **PUT**: replace status of the specified PriorityLevelConfiguration

7.2.1. /apis/flowcontrol.apiserver.k8s.io/v1/prioritylevelconfigurations

HTTP method

DELETE

Description

delete collection of PriorityLevelConfiguration

Table 7.1. 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 7.2. HTTP responses

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

HTTP method

GET

Description

list or watch objects of kind PriorityLevelConfiguration

Table 7.3. HTTP responses

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

HTTP method

POST

Description

create a PriorityLevelConfiguration

Table 7.4. 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 7.5. Body parameters

Parameter	Type	Description
body	**PriorityLevelConfiguration** schema	

Table 7.6. HTTP responses

HTTP code	Reponse body
200 - OK	**PriorityLevelConfiguration** schema
201 - Created	**PriorityLevelConfiguration** schema
202 - Accepted	**PriorityLevelConfiguration** schema

HTTP code	Reponse body
401 - Unauthorized	Empty

7.2.2. /apis/flowcontrol.apiserver.k8s.io/v1/watch/prioritylevelconfigurations

HTTP method

GET

Description

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

Table 7.7. HTTP responses

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

7.2.3. /apis/flowcontrol.apiserver.k8s.io/v1/prioritylevelconfigurations/{name}

Table 7.8. Global path parameters

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

HTTP method

DELETE

Description

delete a PriorityLevelConfiguration

Table 7.9. 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 7.10. HTTP responses

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

HTTP method

GET

Description

read the specified PriorityLevelConfiguration

Table 7.11. HTTP responses

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

HTTP method

PATCH

Description

partially update the specified PriorityLevelConfiguration

Table 7.12. 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 7.13. HTTP responses

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

HTTP method

PUT

Description

replace the specified PriorityLevelConfiguration

Table 7.14. 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 7.15. Body parameters

Parameter	Type	Description
body	**PriorityLevelConfiguration** schema	

Table 7.16. HTTP responses

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

7.2.4. /apis/flowcontrol.apiserver.k8s.io/v1/watch/prioritylevelconfigurations/{name}

Table 7.17. Global path parameters

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

HTTP method

GET

Description

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

Table 7.18. HTTP responses

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

7.2.5. /apis/flowcontrol.apiserver.k8s.io/v1/prioritylevelconfigurations/{name}/status

Table 7.19. Global path parameters

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

HTTP method

GET

Description

read status of the specified PriorityLevelConfiguration

Table 7.20. HTTP responses

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

HTTP method

PATCH

Description

partially update status of the specified PriorityLevelConfiguration

Table 7.21. 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 7.22. HTTP responses

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

HTTP method

PUT

Description

replace status of the specified PriorityLevelConfiguration

Table 7.23. 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 7.24. Body parameters

Parameter	Type	Description
body	**PriorityLevelConfiguration** schema	

Table 7.25. HTTP responses

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

CHAPTER 8. RESOURCEQUOTA [V1]

Description

ResourceQuota sets aggregate quota restrictions enforced per namespace

Type

object

8.1. 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. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata
spec	**object**	ResourceQuotaSpec defines the desired hard limits to enforce for Quota.
status	**object**	ResourceQuotaStatus defines the enforced hard limits and observed use.

8.1.1. .spec

Description

ResourceQuotaSpec defines the desired hard limits to enforce for Quota.

Type

object

Property	Type	Description
hard	**object (Quantity)**	hard is the set of desired hard limits for each named resource. More info: https://kubernetes.io/docs/concepts/policy/resource-quotas/
scopeSelector	**object**	A scope selector represents the AND of the selectors represented by the scoped-resource selector requirements.
scopes	**array (string)**	A collection of filters that must match each object tracked by a quota. If not specified, the quota matches all objects.

8.1.2. .spec.scopeSelector

Description

A scope selector represents the AND of the selectors represented by the scoped-resource selector requirements.

Type

object

Property	Type	Description
matchExpressions	**array**	A list of scope selector requirements by scope of the resources.
matchExpressions[]	**object**	A scoped-resource selector requirement is a selector that contains values, a scope name, and an operator that relates the scope name and values.

8.1.3. .spec.scopeSelector.matchExpressions

Description

A list of scope selector requirements by scope of the resources.

Type

array

8.1.4. .spec.scopeSelector.matchExpressions[]

Description

A scoped-resource selector requirement is a selector that contains values, a scope name, and an operator that relates the scope name and values.

Type

object

Required

- **scopeName**
- **operator**

Property	Type	Description
operator	**string**	Represents a scope's relationship to a set of values. Valid operators are In, NotIn, Exists, DoesNotExist. Possible enum values: - **"DoesNotExist"** - **"Exists"** - **"In"** - **"NotIn"**
scopeName	**string**	The name of the scope that the selector applies to. Possible enum values: - **"BestEffort"** Match all pod objects that have best effort quality of service - **"CrossNamespacePodAffinity"** Match all pod objects that have cross-namespace pod (anti)affinity mentioned. - **"NotBestEffort"** Match all pod objects that do not have best effort quality of service - **"NotTerminating"** Match all pod objects where spec.activeDeadlineSeconds is nil - **"PriorityClass"** Match all pod objects that have priority class mentioned - **"Terminating"** Match all pod objects where spec.activeDeadlineSeconds >=0

Property	Type	Description
values	**array (string)**	An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

8.1.5. .status

Description

ResourceQuotaStatus defines the enforced hard limits and observed use.

Type

object

Property	Type	Description
hard	**object (Quantity)**	Hard is the set of enforced hard limits for each named resource. More info: https://kubernetes.io/docs/concepts/policy/resource-quotas/
used	**object (Quantity)**	Used is the current observed total usage of the resource in the namespace.

8.2. API ENDPOINTS

The following API endpoints are available:

- **/api/v1/resourcequotas**
  - **GET**: list or watch objects of kind ResourceQuota
- **/api/v1/watch/resourcequotas**
  - **GET**: watch individual changes to a list of ResourceQuota. deprecated: use the 'watch' parameter with a list operation instead.
- **/api/v1/namespaces/{namespace}/resourcequotas**
  - **DELETE**: delete collection of ResourceQuota
  - **GET**: list or watch objects of kind ResourceQuota
  - **POST**: create a ResourceQuota

- **/api/v1/watch/namespaces/{namespace}/resourcequotas**
  - **GET**: watch individual changes to a list of ResourceQuota. deprecated: use the 'watch' parameter with a list operation instead.
- **/api/v1/namespaces/{namespace}/resourcequotas/{name}**
  - **DELETE**: delete a ResourceQuota
  - **GET**: read the specified ResourceQuota
  - **PATCH**: partially update the specified ResourceQuota
  - **PUT**: replace the specified ResourceQuota
- **/api/v1/watch/namespaces/{namespace}/resourcequotas/{name}**
  - **GET**: watch changes to an object of kind ResourceQuota. deprecated: use the 'watch' parameter with a list operation instead, filtered to a single item with the 'fieldSelector' parameter.
- **/api/v1/namespaces/{namespace}/resourcequotas/{name}/status**
  - **GET**: read status of the specified ResourceQuota
  - **PATCH**: partially update status of the specified ResourceQuota
  - **PUT**: replace status of the specified ResourceQuota

8.2.1. /api/v1/resourcequotas

HTTP method

GET

Description

list or watch objects of kind ResourceQuota

Table 8.1. HTTP responses

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

8.2.2. /api/v1/watch/resourcequotas

HTTP method

GET

Description

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

Table 8.2. HTTP responses

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

8.2.3. /api/v1/namespaces/{namespace}/resourcequotas

HTTP method

DELETE

Description

delete collection of ResourceQuota

Table 8.3. 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 8.4. HTTP responses

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

HTTP method

GET

Description

list or watch objects of kind ResourceQuota

Table 8.5. HTTP responses

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

HTTP method

POST

Description

create a ResourceQuota

Table 8.6. 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 8.7. Body parameters

Parameter	Type	Description
body	**ResourceQuota** schema	

Table 8.8. HTTP responses

HTTP code	Reponse body
200 - OK	**ResourceQuota** schema
201 - Created	**ResourceQuota** schema
202 - Accepted	**ResourceQuota** schema

HTTP code	Reponse body
401 - Unauthorized	Empty

8.2.4. /api/v1/watch/namespaces/{namespace}/resourcequotas

HTTP method

GET

Description

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

Table 8.9. HTTP responses

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

8.2.5. /api/v1/namespaces/{namespace}/resourcequotas/{name}

Table 8.10. Global path parameters

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

HTTP method

DELETE

Description

delete a ResourceQuota

Table 8.11. 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 8.12. HTTP responses

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

HTTP method

GET

Description

read the specified ResourceQuota

Table 8.13. HTTP responses

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

HTTP method

PATCH

Description

partially update the specified ResourceQuota

Table 8.14. 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 8.15. HTTP responses

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

HTTP method

PUT

Description

replace the specified ResourceQuota

Table 8.16. 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 8.17. Body parameters

Parameter	Type	Description
body	**ResourceQuota** schema	

Table 8.18. HTTP responses

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

8.2.6. /api/v1/watch/namespaces/{namespace}/resourcequotas/{name}

Table 8.19. Global path parameters

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

HTTP method

GET

Description

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

Table 8.20. HTTP responses

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

8.2.7. /api/v1/namespaces/{namespace}/resourcequotas/{name}/status

Table 8.21. Global path parameters

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

HTTP method

GET

Description

read status of the specified ResourceQuota

Table 8.22. HTTP responses

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

HTTP method

PATCH

Description

partially update status of the specified ResourceQuota

Table 8.23. 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 8.24. HTTP responses

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

HTTP method

PUT

Description

replace status of the specified ResourceQuota

Table 8.25. 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 8.26. Body parameters

Parameter	Type	Description
body	**ResourceQuota** schema	

Table 8.27. HTTP responses

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