



OpenShift Container Platform 4.20

Metadata APIs

Reference guide for metadata APIs

OpenShift Container Platform 4.20 Metadata APIs

Reference guide for metadata APIs

Legal Notice

Copyright © 2025 Red Hat, Inc.

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

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

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

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

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

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

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

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

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

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

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

All other trademarks are the property of their respective owners.

Abstract

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

Table of Contents

CHAPTER 1. METADATA APIS	5
1.1. APIREQUESTCOUNT [APISERVER.OPENSIFT.IO/V1]	5
1.2. BINDING [V1]	5
1.3. COMPONENTSTATUS [V1]	5
1.4. CONFIGMAP [V1]	5
1.5. CONTROLLERREVISION [APPS/V1]	5
1.6. EVENT [EVENTS.K8S.IO/V1]	6
1.7. EVENT [V1]	6
1.8. LEASE [COORDINATION.K8S.IO/V1]	6
1.9. NAMESPACE [V1]	6
CHAPTER 2. APIREQUESTCOUNT [APISERVER.OPENSIFT.IO/V1]	7
2.1. SPECIFICATION	7
2.1.1. .spec	8
2.1.2. .status	8
2.1.3. .status.conditions	10
2.1.4. .status.conditions[]	10
2.1.5. .status.currentHour	11
2.1.6. .status.currentHour.byNode	12
2.1.7. .status.currentHour.byNode[]	12
2.1.8. .status.currentHour.byNode[].byUser	12
2.1.9. .status.currentHour.byNode[].byUser[]	13
2.1.10. .status.currentHour.byNode[].byUser[].byVerb	13
2.1.11. .status.currentHour.byNode[].byUser[].byVerb[]	13
2.1.12. .status.last24h	14
2.1.13. .status.last24h[]	14
2.1.14. .status.last24h[].byNode	14
2.1.15. .status.last24h[].byNode[]	15
2.1.16. .status.last24h[].byNode[].byUser	15
2.1.17. .status.last24h[].byNode[].byUser[]	15
2.1.18. .status.last24h[].byNode[].byUser[].byVerb	16
2.1.19. .status.last24h[].byNode[].byUser[].byVerb[]	16
2.2. API ENDPOINTS	16
2.2.1. /apis/apiserver.opensift.io/v1/apirequestcounts	17
2.2.2. /apis/apiserver.opensift.io/v1/apirequestcounts/{name}	19
2.2.3. /apis/apiserver.opensift.io/v1/apirequestcounts/{name}/status	22
CHAPTER 3. BINDING [V1]	25
3.1. SPECIFICATION	25
3.1.1. .target	26
3.2. API ENDPOINTS	27
3.2.1. /api/v1/namespaces/{namespace}/bindings	27
3.2.2. /api/v1/namespaces/{namespace}/pods/{name}/binding	28
CHAPTER 4. COMPONENTSTATUS [V1]	31
4.1. SPECIFICATION	31
4.1.1. .conditions	31
4.1.2. .conditions[]	32
4.2. API ENDPOINTS	32
4.2.1. /api/v1/componentstatuses	32
4.2.2. /api/v1/componentstatuses/{name}	33

CHAPTER 5. CONFIGMAP [V1]	34
5.1. SPECIFICATION	34
5.2. API ENDPOINTS	35
5.2.1. /api/v1/configmaps	36
5.2.2. /api/v1/watch/configmaps	36
5.2.3. /api/v1/namespaces/{namespace}/configmaps	37
5.2.4. /api/v1/watch/namespaces/{namespace}/configmaps	38
5.2.5. /api/v1/namespaces/{namespace}/configmaps/{name}	39
5.2.6. /api/v1/watch/namespaces/{namespace}/configmaps/{name}	42
CHAPTER 6. CONTROLLERREVISION [APPS/V1]	43
6.1. SPECIFICATION	43
6.2. API ENDPOINTS	44
6.2.1. /apis/apps/v1/controllerrevisions	45
6.2.2. /apis/apps/v1/watch/controllerrevisions	45
6.2.3. /apis/apps/v1/namespaces/{namespace}/controllerrevisions	45
6.2.4. /apis/apps/v1/watch/namespaces/{namespace}/controllerrevisions	47
6.2.5. /apis/apps/v1/namespaces/{namespace}/controllerrevisions/{name}	48
6.2.6. /apis/apps/v1/watch/namespaces/{namespace}/controllerrevisions/{name}	51
CHAPTER 7. EVENT [EVENTS.K8S.IO/V1]	52
7.1. SPECIFICATION	52
7.1.1. .series	54
7.2. API ENDPOINTS	55
7.2.1. /apis/events.k8s.io/v1/events	56
7.2.2. /apis/events.k8s.io/v1/watch/events	56
7.2.3. /apis/events.k8s.io/v1/namespaces/{namespace}/events	56
7.2.4. /apis/events.k8s.io/v1/watch/namespaces/{namespace}/events	58
7.2.5. /apis/events.k8s.io/v1/namespaces/{namespace}/events/{name}	59
7.2.6. /apis/events.k8s.io/v1/watch/namespaces/{namespace}/events/{name}	62
CHAPTER 8. EVENT [V1]	63
8.1. SPECIFICATION	63
8.1.1. .involvedObject	65
8.1.2. .related	66
8.1.3. .series	68
8.1.4. .source	68
8.2. API ENDPOINTS	68
8.2.1. /api/v1/events	69
8.2.2. /api/v1/watch/events	69
8.2.3. /api/v1/namespaces/{namespace}/events	70
8.2.4. /api/v1/watch/namespaces/{namespace}/events	72
8.2.5. /api/v1/namespaces/{namespace}/events/{name}	72
8.2.6. /api/v1/watch/namespaces/{namespace}/events/{name}	75
CHAPTER 9. LEASE [COORDINATION.K8S.IO/V1]	77
9.1. SPECIFICATION	77
9.1.1. .spec	77
9.2. API ENDPOINTS	78
9.2.1. /apis/coordination.k8s.io/v1/leases	79
9.2.2. /apis/coordination.k8s.io/v1/watch/leases	79
9.2.3. /apis/coordination.k8s.io/v1/namespaces/{namespace}/leases	80
9.2.4. /apis/coordination.k8s.io/v1/watch/namespaces/{namespace}/leases	82
9.2.5. /apis/coordination.k8s.io/v1/namespaces/{namespace}/leases/{name}	82

9.2.6. /apis/coordination.k8s.io/v1/watch/namespaces/{namespace}/leases/{name}	85
CHAPTER 10. NAMESPACE [V1]	87
10.1. SPECIFICATION	87
10.1.1. .spec	87
10.1.2. .status	88
10.1.3. .status.conditions	88
10.1.4. .status.conditions[]	89
10.2. API ENDPOINTS	89
10.2.1. /api/v1/namespaces	90
10.2.2. /api/v1/watch/namespaces	91
10.2.3. /api/v1/namespaces/{name}	92
10.2.4. /api/v1/watch/namespaces/{name}	95
10.2.5. /api/v1/namespaces/{name}/status	95
10.2.6. /api/v1/namespaces/{name}/finalize	98

CHAPTER 1. METADATA APIS

1.1. APIREQUESTCOUNT [APISERVER.OPENSIFT.IO/V1]

Description

APIRequestCount tracks requests made to an API. The instance name must be of the form **resource.version.group**, matching the resource. 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. BINDING [V1]

Description

Binding ties one object to another; for example, a pod is bound to a node by a scheduler.

Type

object

1.3. COMPONENTSTATUS [V1]

Description

ComponentStatus (and ComponentStatusList) holds the cluster validation info. Deprecated: This API is deprecated in v1.19+

Type

object

1.4. CONFIGMAP [V1]

Description

ConfigMap holds configuration data for pods to consume.

Type

object

1.5. CONTROLLERREVISION [APPS/V1]

Description

ControllerRevision implements an immutable snapshot of state data. Clients are responsible for serializing and deserializing the objects that contain their internal state. Once a ControllerRevision has been successfully created, it can not be updated. The API Server will fail validation of all requests that attempt to mutate the Data field. ControllerRevisions may, however, be deleted. Note that, due to its use by both the DaemonSet and StatefulSet controllers for update and rollback, this object is beta. However, it may be subject to name and representation changes in future releases, and clients should not depend on its stability. It is primarily for internal use by controllers.

Type

object

1.6. EVENT [EVENTS.K8S.IO/V1]

Description

Event is a report of an event somewhere in the cluster. It generally denotes some state change in the system. Events have a limited retention time and triggers and messages may evolve with time. Event consumers should not rely on the timing of an event with a given Reason reflecting a consistent underlying trigger, or the continued existence of events with that Reason. Events should be treated as informative, best-effort, supplemental data.

Type

object

1.7. EVENT [V1]

Description

Event is a report of an event somewhere in the cluster. Events have a limited retention time and triggers and messages may evolve with time. Event consumers should not rely on the timing of an event with a given Reason reflecting a consistent underlying trigger, or the continued existence of events with that Reason. Events should be treated as informative, best-effort, supplemental data.

Type

object

1.8. LEASE [COORDINATION.K8S.IO/V1]

Description

Lease defines a lease concept.

Type

object

1.9. NAMESPACE [V1]

Description

Namespace provides a scope for Names. Use of multiple namespaces is optional.

Type

object

CHAPTER 2. APIREQUESTCOUNT [APISERVER.OPENSIFT.IO/V1]

Description

APIRequestCount tracks requests made to an API. The instance name must be of the form **resource.version.group**, matching the resource. Compatibility level 1: Stable within a major release for a minimum of 12 months or 3 minor releases (whichever is longer).

Type

object

Required

- **spec**

2.1. SPECIFICATION

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

Property	Type	Description
status	object	status contains the observed state of the resource.

2.1.1. .spec

Description

spec defines the characteristics of the resource.

Type

object

Property	Type	Description
numberOfUsersToReport	integer	numberOfUsersToReport is the number of users to include in the report. If unspecified or zero, the default is ten. This is default is subject to change.

2.1.2. .status

Description

status contains the observed state of the resource.

Type

object

Property	Type	Description
conditions	array	conditions contains details of the current status of this API Resource.

Property	Type	Description
conditions[]	object	Condition contains details for one aspect of the current state of this API Resource. --- This struct is intended for direct use as an array at the field path .status.conditions. For example, <pre>type FooStatus struct{ // Represents the observations of a foo's current state. // Known .status.conditions.type are: "Available", "Progressing", and "Degraded" // +patchMergeKey=type // +patchStrategy=merge // +listType=map // +listMapKey=type Conditions []metav1.Condition</pre> json:"conditions,omitempty" patchStrategy:"merge" patchMergeKey:"type" protobuf:"bytes,1,rep,name=conditions" // other fields }
currentHour	object	currentHour contains request history for the current hour. This is porcelain to make the API easier to read by humans seeing if they addressed a problem. This field is reset on the hour.
last24h	array	last24h contains request history for the last 24 hours, indexed by the hour, so 12:00AM-12:59 is in index 0, 6am-6:59am is index 6, etc. The index of the current hour is updated live and then duplicated into the requestsLastHour field.
last24h[]	object	PerResourceAPIRequestLog logs request for various nodes.
removedInRelease	string	removedInRelease is when the API will be removed.
requestCount	integer	requestCount is a sum of all requestCounts across all current hours, nodes, and users.

2.1.3. .status.conditions

Description

conditions contains details of the current status of this API Resource.

Type

array

2.1.4. .status.conditions[]

Description

Condition contains details for one aspect of the current state of this API Resource. --- This struct is intended for direct use as an array at the field path .status.conditions. For example, type FooStatus struct{ // Represents the observations of a foo's current state. // Known .status.conditions.type are: "Available", "Progressing", and "Degraded" // +patchMergeKey=type // +patchStrategy=merge // +listType=map // +listMapKey=type Conditions []metav1.Condition **json:"conditions,omitempty"** **patchStrategy:"merge" patchMergeKey:"type" protobuf:"bytes,1,rep,name=conditions"** // other fields }

Type

object

Required

- **lastTransitionTime**
- **message**
- **reason**
- **status**
- **type**

Property	Type	Description
lastTransitionTime	string	lastTransitionTime is the last time the condition transitioned from one status to another. This should be when the underlying condition changed. If that is not known, then using the time when the API field changed is acceptable.
message	string	message is a human readable message indicating details about the transition. This may be an empty string.

Property	Type	Description
observedGeneration	integer	observedGeneration represents the .metadata.generation that the condition was set based upon. For instance, if .metadata.generation is currently 12, but the .status.conditions[x].observedGeneration is 9, the condition is out of date with respect to the current state of the instance.
reason	string	reason contains a programmatic identifier indicating the reason for the condition's last transition. Producers of specific condition types may define expected values and meanings for this field, and whether the values are considered a guaranteed API. The value should be a CamelCase string. This field may not be empty.
status	string	status of the condition, one of True, False, Unknown.
type	string	type of condition in CamelCase or in foo.example.com/CamelCase. -- Many .condition.type values are consistent across resources like Available, but because arbitrary conditions can be useful (see .node.status.conditions), the ability to deconflict is important. The regex it matches is (dns1123SubdomainFmt/)?(qualifiedNameFmt)

2.1.5. .status.currentHour

Description

currentHour contains request history for the current hour. This is porcelain to make the API easier to read by humans seeing if they addressed a problem. This field is reset on the hour.

Type

object

Property	Type	Description
byNode	array	byNode contains logs of requests per node.

Property	Type	Description
byNode[]	object	PerNodeAPIRequestLog contains logs of requests to a certain node.
requestCount	integer	requestCount is a sum of all requestCounts across nodes.

2.1.6. .status.currentHour.byNode

Description

byNode contains logs of requests per node.

Type

array

2.1.7. .status.currentHour.byNode[]

Description

PerNodeAPIRequestLog contains logs of requests to a certain node.

Type

object

Property	Type	Description
byUser	array	byUser contains request details by top .spec.numberOfUsersToReport users. Note that because in the case of an apiserver, restart the list of top users is determined on a best-effort basis, the list might be imprecise. In addition, some system users may be explicitly included in the list.
byUser[]	object	PerUserAPIRequestCount contains logs of a user's requests.
nodeName	string	nodeName where the request are being handled.
requestCount	integer	requestCount is a sum of all requestCounts across all users, even those outside of the top 10 users.

2.1.8. .status.currentHour.byNode[].byUser

Description

byUser contains request details by top .spec.numberOfUsersToReport users. Note that because in the case of an apiserver, restart the list of top users is determined on a best-effort basis, the list might be imprecise. In addition, some system users may be explicitly included in the list.

Type

array

2.1.9. .status.currentHour.byNode[].byUser[]**Description**

PerUserAPIRequestCount contains logs of a user's requests.

Type

object

Property	Type	Description
byVerb	array	byVerb details by verb.
byVerb[]	object	PerVerbAPIRequestCount requestCounts requests by API request verb.
requestCount	integer	requestCount of requests by the user across all verbs.
userAgent	string	userAgent that made the request. The same user often has multiple binaries which connect (pods with many containers). The different binaries will have different userAgents, but the same user. In addition, we have userAgents with version information embedded and the userName isn't likely to change.
username	string	userName that made the request.

2.1.10. .status.currentHour.byNode[].byUser[].byVerb**Description**

byVerb details by verb.

Type

array

2.1.11. .status.currentHour.byNode[].byUser[].byVerb[]**Description**

PerVerbAPIRequestCount requestCounts requests by API request verb.

Type

object

Property	Type	Description
requestCount	integer	requestCount of requests for verb.
verb	string	verb of API request (get, list, create, etc...)

2.1.12. .status.last24h

Description

last24h contains request history for the last 24 hours, indexed by the hour, so 12:00AM-12:59 is in index 0, 6am-6:59am is index 6, etc. The index of the current hour is updated live and then duplicated into the requestsLastHour field.

Type

array

2.1.13. .status.last24h[]

Description

PerResourceAPIRequestLog logs request for various nodes.

Type

object

Property	Type	Description
byNode	array	byNode contains logs of requests per node.
byNode[]	object	PerNodeAPIRequestLog contains logs of requests to a certain node.
requestCount	integer	requestCount is a sum of all requestCounts across nodes.

2.1.14. .status.last24h[].byNode

Description

byNode contains logs of requests per node.

Type

array

2.1.15. .status.last24h[].byNode[]

Description

PerNodeAPIRequestLog contains logs of requests to a certain node.

Type

object

Property	Type	Description
byUser	array	byUser contains request details by top .spec.numberOfUsersToReport users. Note that because in the case of an apiserver, restart the list of top users is determined on a best-effort basis, the list might be imprecise. In addition, some system users may be explicitly included in the list.
byUser[]	object	PerUserAPIRequestCount contains logs of a user's requests.
nodeName	string	nodeName where the request are being handled.
requestCount	integer	requestCount is a sum of all requestCounts across all users, even those outside of the top 10 users.

2.1.16. .status.last24h[].byNode[].byUser

Description

byUser contains request details by top .spec.numberOfUsersToReport users. Note that because in the case of an apiserver, restart the list of top users is determined on a best-effort basis, the list might be imprecise. In addition, some system users may be explicitly included in the list.

Type

array

2.1.17. .status.last24h[].byNode[].byUser[]

Description

PerUserAPIRequestCount contains logs of a user's requests.

Type

object

Property	Type	Description
byVerb	array	byVerb details by verb.
byVerb[]	object	PerVerbAPIRequestCount requestCounts requests by API request verb.
requestCount	integer	requestCount of requests by the user across all verbs.
userAgent	string	userAgent that made the request. The same user often has multiple binaries which connect (pods with many containers). The different binaries will have different userAgents, but the same user. In addition, we have userAgents with version information embedded and the userName isn't likely to change.
username	string	userName that made the request.

2.1.18. .status.last24h[].byNode[].byUser[].byVerb

Description

byVerb details by verb.

Type

array

2.1.19. .status.last24h[].byNode[].byUser[].byVerb[]

Description

PerVerbAPIRequestCount requestCounts requests by API request verb.

Type

object

Property	Type	Description
requestCount	integer	requestCount of requests for verb.
verb	string	verb of API request (get, list, create, etc...)

2.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/apiserver.openshift.io/v1/apirequestcounts**
 - **DELETE**: delete collection of APIRequestCount
 - **GET**: list objects of kind APIRequestCount
 - **POST**: create an APIRequestCount
- **/apis/apiserver.openshift.io/v1/apirequestcounts/{name}**
 - **DELETE**: delete an APIRequestCount
 - **GET**: read the specified APIRequestCount
 - **PATCH**: partially update the specified APIRequestCount
 - **PUT**: replace the specified APIRequestCount
- **/apis/apiserver.openshift.io/v1/apirequestcounts/{name}/status**
 - **GET**: read status of the specified APIRequestCount
 - **PATCH**: partially update status of the specified APIRequestCount
 - **PUT**: replace status of the specified APIRequestCount

2.2.1. /apis/apiserver.openshift.io/v1/apirequestcounts

HTTP method

DELETE

Description

delete collection of APIRequestCount

Table 2.1. HTTP responses

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

HTTP method

GET

Description

list objects of kind APIRequestCount

Table 2.2. HTTP responses

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

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

create an `APIRequestCount`

Table 2.3. 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
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Table 2.4. Body parameters

Parameter	Type	Description
body	APIRequestCount schema	

Table 2.5. HTTP responses

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

2.2.2. /apis/apiserver.openshift.io/v1/apirequestcounts/{name}

Table 2.6. Global path parameters

Parameter	Type	Description
name	string	name of the APIRequestCount

HTTP method

DELETE

Description

delete an APIRequestCount

Table 2.7. Query parameters

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

Table 2.8. HTTP responses

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

HTTP method

GET

Description

read the specified APIRequestCount

Table 2.9. HTTP responses

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

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

partially update the specified APIRequestCount

Table 2.10. Query parameters

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

Table 2.11. HTTP responses

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

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

replace the specified APIRequestCount

Table 2.12. Query parameters

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

Table 2.13. Body parameters

Parameter	Type	Description
body	APIRequestCount schema	

Table 2.14. HTTP responses

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

2.2.3. /apis/apiserver.openshift.io/v1/apirequestcounts/{name}/status

Table 2.15. Global path parameters

Parameter	Type	Description
name	string	name of the APIRequestCount

HTTP method

GET

Description

read status of the specified APIRequestCount

Table 2.16. HTTP responses

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

HTTP method

PATCH

Description

partially update status of the specified APIRequestCount

Table 2.17. Query parameters

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

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

Table 2.18. HTTP responses

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

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

replace status of the specified APIRequestCount

Table 2.19. Query parameters

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

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

Table 2.20. Body parameters

Parameter	Type	Description
body	APIRequestCount schema	

Table 2.21. HTTP responses

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

CHAPTER 3. BINDING [V1]

Description

Binding ties one object to another; for example, a pod is bound to a node by a scheduler.

Type

object

Required

- **target**

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
target	object	ObjectReference contains enough information to let you inspect or modify the referred object.

3.1.1. .target

Description

ObjectReference contains enough information to let you inspect or modify the referred object.

Type

object

Property	Type	Description
apiVersion	string	API version of the referent.
fieldPath	string	If referring to a piece of an object instead of an entire object, this string should contain a valid JSON/Go field access statement, such as <code>desiredState.manifest.containers[2]</code> . For example, if the object reference is to a container within a pod, this would take on a value like: <code>"spec.containers{name}"</code> (where "name" refers to the name of the container that triggered the event) or if no container name is specified <code>"spec.containers[2]"</code> (container with index 2 in this pod). This syntax is chosen only to have some well-defined way of referencing a part of an object.
kind	string	Kind of the referent. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
name	string	Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
namespace	string	Namespace of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/namespaces/

Property	Type	Description
resourceVersion	string	Specific resourceVersion to which this reference is made, if any. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#concurrency-control-and-consistency
uid	string	UID of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#uids

3.2. API ENDPOINTS

The following API endpoints are available:

- **/api/v1/namespaces/{namespace}/bindings**
 - **POST**: create a Binding
- **/api/v1/namespaces/{namespace}/pods/{name}/binding**
 - **POST**: create binding of a Pod

3.2.1. /api/v1/namespaces/{namespace}/bindings

Table 3.1. Global query parameters

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

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.

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

create a Binding

Table 3.2. Body parameters

Parameter	Type	Description
body	Binding schema	

Table 3.3. HTTP responses

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

3.2.2. /api/v1/namespaces/{namespace}/pods/{name}/binding**Table 3.4. Global path parameters**

Parameter	Type	Description
name	string	name of the Binding

Table 3.5. Global query parameters

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

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

create binding of a Pod

Table 3.6. Body parameters

Parameter	Type	Description
body	Binding schema	

Table 3.7. HTTP responses

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

CHAPTER 4. COMPONENTSTATUS [V1]

Description

ComponentStatus (and ComponentStatusList) holds the cluster validation info. Deprecated: This API is deprecated in v1.19+

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
conditions	array	List of component conditions observed
conditions[]	object	Information about the condition of a component.
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

4.1.1. .conditions

Description

List of component conditions observed

Type

array

4.1.2. .conditions[]**Description**

Information about the condition of a component.

Type

object

Required

- **type**
- **status**

Property	Type	Description
error	string	Condition error code for a component. For example, a health check error code.
message	string	Message about the condition for a component. For example, information about a health check.
status	string	Status of the condition for a component. Valid values for "Healthy": "True", "False", or "Unknown".
type	string	Type of condition for a component. Valid value: "Healthy"

4.2. API ENDPOINTS

The following API endpoints are available:

- **/api/v1/componentstatuses**
 - **GET**: list objects of kind ComponentStatus
- **/api/v1/componentstatuses/{name}**
 - **GET**: read the specified ComponentStatus

4.2.1. /api/v1/componentstatuses

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

list objects of kind ComponentStatus

Table 4.1. HTTP responses

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

4.2.2. /api/v1/componentstatuses/{name}**Table 4.2. Global path parameters**

Parameter	Type	Description
name	string	name of the ComponentStatus

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

read the specified ComponentStatus

Table 4.3. HTTP responses

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

CHAPTER 5. CONFIGMAP [V1]

Description

ConfigMap holds configuration data for pods to consume.

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
binaryData	object (string)	BinaryData contains the binary data. Each key must consist of alphanumeric characters, '-', '_' or '.'. BinaryData can contain byte sequences that are not in the UTF-8 range. The keys stored in BinaryData must not overlap with the ones in the Data field, this is enforced during validation process. Using this field will require 1.10+ apiserver and kubelet.
data	object (string)	Data contains the configuration data. Each key must consist of alphanumeric characters, '-', '_' or '.'. Values with non-UTF-8 byte sequences must use the BinaryData field. The keys stored in Data must not overlap with the keys in the BinaryData field, this is enforced during validation process.

Property	Type	Description
immutable	boolean	Immutable, if set to true, ensures that data stored in the ConfigMap cannot be updated (only object metadata can be modified). If not set to true, the field can be modified at any time. Defaulted to nil.
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

5.2. API ENDPOINTS

The following API endpoints are available:

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

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

5.2.1. /api/v1/configmaps

HTTP method

GET

Description

list or watch objects of kind ConfigMap

Table 5.1. HTTP responses

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

5.2.2. /api/v1/watch/configmaps

HTTP method

GET

Description

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

Table 5.2. HTTP responses

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

5.2.3. /api/v1/namespaces/{namespace}/configmaps

HTTP method

DELETE

Description

delete collection of ConfigMap

Table 5.3. Query parameters

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

Table 5.4. HTTP responses

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

HTTP method

GET

Description

list or watch objects of kind ConfigMap

Table 5.5. HTTP responses

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

HTTP method

POST

Description

create a ConfigMap

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
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	ConfigMap schema	

Table 5.8. HTTP responses

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

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

HTTP method

GET**Description**

watch individual changes to a list of ConfigMap. 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}/configmaps/{name}**Table 5.10. Global path parameters**

Parameter	Type	Description
name	string	name of the ConfigMap

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

delete a ConfigMap

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 ConfigMap

Table 5.13. HTTP responses

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

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

partially update the specified ConfigMap

Table 5.14. Query parameters

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

Table 5.15. HTTP responses

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

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

replace the specified ConfigMap

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	ConfigMap schema	

Table 5.18. HTTP responses

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

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

Table 5.19. Global path parameters

Parameter	Type	Description
name	string	name of the ConfigMap

HTTP method

GET

Description

watch changes to an object of kind ConfigMap. 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. CONTROLLERREVISION [APPS/V1]

Description

ControllerRevision implements an immutable snapshot of state data. Clients are responsible for serializing and deserializing the objects that contain their internal state. Once a ControllerRevision has been successfully created, it can not be updated. The API Server will fail validation of all requests that attempt to mutate the Data field. ControllerRevisions may, however, be deleted. Note that, due to its use by both the DaemonSet and StatefulSet controllers for update and rollback, this object is beta. However, it may be subject to name and representation changes in future releases, and clients should not depend on its stability. It is primarily for internal use by controllers.

Type

object

Required

- **revision**

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
data	RawExtension	Data is the serialized representation of the state.
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	Standard object's metadata. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata
revision	integer	Revision indicates the revision of the state represented by Data.

6.2. API ENDPOINTS

The following API endpoints are available:

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

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

6.2.1. /apis/apps/v1/controllerrevisions

HTTP method

GET

Description

list or watch objects of kind ControllerRevision

Table 6.1. HTTP responses

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

6.2.2. /apis/apps/v1/watch/controllerrevisions

HTTP method

GET

Description

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

Table 6.2. HTTP responses

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

6.2.3. /apis/apps/v1/namespaces/{namespace}/controllerrevisions

HTTP method

DELETE

Description

delete collection of ControllerRevision

Table 6.3. 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 6.4. HTTP responses

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

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

list or watch objects of kind ControllerRevision

Table 6.5. HTTP responses

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

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

create a ControllerRevision

Table 6.6. 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 6.7. Body parameters

Parameter	Type	Description
body	ControllerRevision schema	

Table 6.8. HTTP responses

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

6.2.4. /apis/apps/v1/watch/namespaces/{namespace}/controllerrevisions

HTTP method

GET

Description

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

Table 6.9. HTTP responses

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

6.2.5. /apis/apps/v1/namespaces/{namespace}/controllerrevisions/{name}

Table 6.10. Global path parameters

Parameter	Type	Description
name	string	name of the ControllerRevision

HTTP method

DELETE

Description

delete a ControllerRevision

Table 6.11. 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 6.12. HTTP responses

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

HTTP method

GET

Description

read the specified ControllerRevision

Table 6.13. HTTP responses

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

HTTP method

PATCH

Description

partially update the specified ControllerRevision

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
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. HTTP responses

HTTP code	Response body
200 - OK	ControllerRevision schema

HTTP code	Response body
201 - Created	ControllerRevision schema
401 - Unauthorized	Empty

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

replace the specified ControllerRevision

Table 6.16. 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 6.17. Body parameters

Parameter	Type	Description
body	ControllerRevision schema	

Table 6.18. HTTP responses

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

6.2.6. /apis/apps/v1/watch/namespaces/{namespace}/controllerrevisions/{name}

Table 6.19. Global path parameters

Parameter	Type	Description
name	string	name of the ControllerRevision

HTTP method

GET

Description

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

Table 6.20. HTTP responses

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

CHAPTER 7. EVENT [EVENTS.K8S.IO/V1]

Description

Event is a report of an event somewhere in the cluster. It generally denotes some state change in the system. Events have a limited retention time and triggers and messages may evolve with time. Event consumers should not rely on the timing of an event with a given Reason reflecting a consistent underlying trigger, or the continued existence of events with that Reason. Events should be treated as informative, best-effort, supplemental data.

Type

object

Required

- **eventTime**

7.1. SPECIFICATION

Property	Type	Description
action	string	action is what action was taken/failed regarding to the regarding object. It is machine-readable. This field cannot be empty for new Events and it can have at most 128 characters.
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
deprecatedCount	integer	deprecatedCount is the deprecated field assuring backward compatibility with core.v1 Event type.
deprecatedFirstTimestamp	Time	deprecatedFirstTimestamp is the deprecated field assuring backward compatibility with core.v1 Event type.
deprecatedLastTimestamp	Time	deprecatedLastTimestamp is the deprecated field assuring backward compatibility with core.v1 Event type.

Property	Type	Description
deprecatedSource	EventSource	deprecatedSource is the deprecated field assuring backward compatibility with core.v1 Event type.
eventTime	MicroTime	eventTime is the time when this Event was first observed. It is required.
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
note	string	note is a human-readable description of the status of this operation. Maximal length of the note is 1kB, but libraries should be prepared to handle values up to 64kB.
reason	string	reason is why the action was taken. It is human-readable. This field cannot be empty for new Events and it can have at most 128 characters.
regarding	ObjectReference	regarding contains the object this Event is about. In most cases it's an Object reporting controller implements, e.g. ReplicaSetController implements ReplicaSets and this event is emitted because it acts on some changes in a ReplicaSet object.

Property	Type	Description
related	ObjectReference	related is the optional secondary object for more complex actions. E.g. when regarding object triggers a creation or deletion of related object.
reportingController	string	reportingController is the name of the controller that emitted this Event, e.g. kubernetes.io/kubelet . This field cannot be empty for new Events.
reportingInstance	string	reportingInstance is the ID of the controller instance, e.g. kubelet-xyzf . This field cannot be empty for new Events and it can have at most 128 characters.
series	object	EventSeries contain information on series of events, i.e. thing that was/is happening continuously for some time. How often to update the EventSeries is up to the event reporters. The default event reporter in "k8s.io/client-go/tools/events/event_broadcaster.go" shows how this struct is updated on heartbeats and can guide customized reporter implementations.
type	string	type is the type of this event (Normal, Warning), new types could be added in the future. It is machine-readable. This field cannot be empty for new Events.

7.1.1. .series

Description

EventSeries contain information on series of events, i.e. thing that was/is happening continuously for some time. How often to update the EventSeries is up to the event reporters. The default event reporter in "k8s.io/client-go/tools/events/event_broadcaster.go" shows how this struct is updated on heartbeats and can guide customized reporter implementations.

Type

object

Required

- **count**
- **lastObservedTime**

Property	Type	Description
count	integer	count is the number of occurrences in this series up to the last heartbeat time.
lastObservedTime	MicroTime	lastObservedTime is the time when last Event from the series was seen before last heartbeat.

7.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/events.k8s.io/v1/events**
 - **GET**: list or watch objects of kind Event
- **/apis/events.k8s.io/v1/watch/events**
 - **GET**: watch individual changes to a list of Event. deprecated: use the 'watch' parameter with a list operation instead.
- **/apis/events.k8s.io/v1/namespaces/{namespace}/events**
 - **DELETE**: delete collection of Event
 - **GET**: list or watch objects of kind Event
 - **POST**: create an Event
- **/apis/events.k8s.io/v1/watch/namespaces/{namespace}/events**
 - **GET**: watch individual changes to a list of Event. deprecated: use the 'watch' parameter with a list operation instead.
- **/apis/events.k8s.io/v1/namespaces/{namespace}/events/{name}**
 - **DELETE**: delete an Event
 - **GET**: read the specified Event
 - **PATCH**: partially update the specified Event
 - **PUT**: replace the specified Event
- **/apis/events.k8s.io/v1/watch/namespaces/{namespace}/events/{name}**

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

7.2.1. /apis/events.k8s.io/v1/events

HTTP method

GET

Description

list or watch objects of kind Event

Table 7.1. HTTP responses

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

7.2.2. /apis/events.k8s.io/v1/watch/events

HTTP method

GET

Description

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

Table 7.2. HTTP responses

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

7.2.3. /apis/events.k8s.io/v1/namespaces/{namespace}/events

HTTP method

DELETE

Description

delete collection of Event

Table 7.3. 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 7.4. HTTP responses

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

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

list or watch objects of kind Event

Table 7.5. HTTP responses

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

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

create an Event

Table 7.6. 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 7.7. Body parameters

Parameter	Type	Description
body	Event schema	

Table 7.8. HTTP responses

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

7.2.4. /apis/events.k8s.io/v1/watch/namespaces/{namespace}/events

HTTP method

GET

Description

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

Table 7.9. HTTP responses

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

7.2.5. /apis/events.k8s.io/v1/namespaces/{namespace}/events/{name}

Table 7.10. Global path parameters

Parameter	Type	Description
name	string	name of the Event

HTTP method

DELETE

Description

delete an Event

Table 7.11. 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 7.12. HTTP responses

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

HTTP method

GET

Description

read the specified Event

Table 7.13. HTTP responses

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

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

partially update the specified Event

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
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. HTTP responses

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

HTTP code	Response body
401 - Unauthorized	Empty

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

replace the specified Event

Table 7.16. 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 7.17. Body parameters

Parameter	Type	Description
body	Event schema	

Table 7.18. HTTP responses

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

7.2.6. /apis/events.k8s.io/v1/watch/namespaces/{namespace}/events/{name}

Table 7.19. Global path parameters

Parameter	Type	Description
name	string	name of the Event

HTTP method

GET

Description

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

Table 7.20. HTTP responses

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

CHAPTER 8. EVENT [V1]

Description

Event is a report of an event somewhere in the cluster. Events have a limited retention time and triggers and messages may evolve with time. Event consumers should not rely on the timing of an event with a given Reason reflecting a consistent underlying trigger, or the continued existence of events with that Reason. Events should be treated as informative, best-effort, supplemental data.

Type

object

Required

- **metadata**
- **involvedObject**

8.1. SPECIFICATION

Property	Type	Description
action	string	What action was taken/failed regarding to the Regarding object.
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
count	integer	The number of times this event has occurred.
eventTime	MicroTime	Time when this Event was first observed.
firstTimestamp	Time	The time at which the event was first recorded. (Time of server receipt is in TypeMeta.)
involvedObject	object	ObjectReference contains enough information to let you inspect or modify the referred object.

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
lastTimestamp	Time	The time at which the most recent occurrence of this event was recorded.
message	string	A human-readable description of the status of this operation.
metadata	ObjectMeta	Standard object's metadata. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata
reason	string	This should be a short, machine understandable string that gives the reason for the transition into the object's current status.
related	object	ObjectReference contains enough information to let you inspect or modify the referred object.
reportingComponent	string	Name of the controller that emitted this Event, e.g. kubernetes.io/kubelet .
reportingInstance	string	ID of the controller instance, e.g. kubelet-xyzf .
series	object	EventSeries contain information on series of events, i.e. thing that was/is happening continuously for some time.

Property	Type	Description
source	object	EventSource contains information for an event.
type	string	Type of this event (Normal, Warning), new types could be added in the future

8.1.1. .involvedObject

Description

ObjectReference contains enough information to let you inspect or modify the referred object.

Type

object

Property	Type	Description
apiVersion	string	API version of the referent.
fieldPath	string	If referring to a piece of an object instead of an entire object, this string should contain a valid JSON/Go field access statement, such as <code>desiredState.manifest.containers[2]</code> . For example, if the object reference is to a container within a pod, this would take on a value like: <code>"spec.containers{name}"</code> (where "name" refers to the name of the container that triggered the event) or if no container name is specified <code>"spec.containers[2]"</code> (container with index 2 in this pod). This syntax is chosen only to have some well-defined way of referencing a part of an object.
kind	string	Kind of the referent. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
name	string	Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

Property	Type	Description
namespace	string	Namespace of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/namespaces/
resourceVersion	string	Specific resourceVersion to which this reference is made, if any. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#concurrency-control-and-consistency
uid	string	UID of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#uids

8.1.2. .related

Description

ObjectReference contains enough information to let you inspect or modify the referred object.

Type

object

Property	Type	Description
apiVersion	string	API version of the referent.

Property	Type	Description
fieldPath	string	If referring to a piece of an object instead of an entire object, this string should contain a valid JSON/Go field access statement, such as <code>desiredState.manifest.containers[2]</code> . For example, if the object reference is to a container within a pod, this would take on a value like: <code>"spec.containers{name}"</code> (where "name" refers to the name of the container that triggered the event) or if no container name is specified <code>"spec.containers[2]"</code> (container with index 2 in this pod). This syntax is chosen only to have some well-defined way of referencing a part of an object.
kind	string	Kind of the referent. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
name	string	Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
namespace	string	Namespace of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/namespaces/
resourceVersion	string	Specific resourceVersion to which this reference is made, if any. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#concurrency-control-and-consistency

Property	Type	Description
uid	string	UID of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#uids

8.1.3. .series

Description

EventSeries contain information on series of events, i.e. thing that was/is happening continuously for some time.

Type

object

Property	Type	Description
count	integer	Number of occurrences in this series up to the last heartbeat time
lastObservedTime	MicroTime	Time of the last occurrence observed

8.1.4. .source

Description

EventSource contains information for an event.

Type

object

Property	Type	Description
component	string	Component from which the event is generated.
host	string	Node name on which the event is generated.

8.2. API ENDPOINTS

The following API endpoints are available:

- **/api/v1/events**
 - **GET**: list or watch objects of kind Event

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

8.2.1. /api/v1/events

HTTP method

GET

Description

list or watch objects of kind Event

Table 8.1. HTTP responses

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

8.2.2. /api/v1/watch/events

HTTP method

GET

Description

watch individual changes to a list of Event. 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}/events**HTTP method****DELETE****Description**

delete collection of Event

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 Event

Table 8.5. HTTP responses

HTTP code	Reponse body
200 - OK	EventList schema

HTTP code	Response body
401 - Unauthorized	Empty

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

create an Event

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	Event schema	

Table 8.8. HTTP responses

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

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

HTTP method

GET

Description

watch individual changes to a list of Event. 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}/events/{name}

Table 8.10. Global path parameters

Parameter	Type	Description
name	string	name of the Event

HTTP method

DELETE

Description

delete an Event

Table 8.11. 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 8.12. HTTP responses

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

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

read the specified Event

Table 8.13. HTTP responses

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

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

partially update the specified Event

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	Response body
200 - OK	Event schema
201 - Created	Event schema
401 - Unauthorized	Empty

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

replace the specified Event

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	Event schema	

Table 8.18. HTTP responses

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

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

Table 8.19. Global path parameters

Parameter	Type	Description
name	string	name of the Event

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

watch changes to an object of kind Event. 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

CHAPTER 9. LEASE [COORDINATION.K8S.IO/V1]

Description

Lease defines a lease concept.

Type

object

9.1. 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	More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata
spec	object	LeaseSpec is a specification of a Lease.

9.1.1. .spec

Description

LeaseSpec is a specification of a Lease.

Type

object

Property	Type	Description
acquireTime	MicroTime	acquireTime is a time when the current lease was acquired.
holderIdentity	string	holderIdentity contains the identity of the holder of a current lease. If Coordinated Leader Election is used, the holder identity must be equal to the elected LeaseCandidate.metadata.name field.
leaseDurationSeconds	integer	leaseDurationSeconds is a duration that candidates for a lease need to wait to force acquire it. This is measured against the time of last observed renewTime.
leaseTransitions	integer	leaseTransitions is the number of transitions of a lease between holders.
preferredHolder	string	PreferredHolder signals to a lease holder that the lease has a more optimal holder and should be given up. This field can only be set if Strategy is also set.
renewTime	MicroTime	renewTime is a time when the current holder of a lease has last updated the lease.
strategy	string	Strategy indicates the strategy for picking the leader for coordinated leader election. If the field is not specified, there is no active coordination for this lease. (Alpha) Using this field requires the CoordinatedLeaderElection feature gate to be enabled.

9.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/coordination.k8s.io/v1/leases**
 - **GET**: list or watch objects of kind Lease

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

9.2.1. /apis/coordination.k8s.io/v1/leases

HTTP method

GET

Description

list or watch objects of kind Lease

Table 9.1. HTTP responses

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

9.2.2. /apis/coordination.k8s.io/v1/watch/leases

HTTP method

GET

Description

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

Table 9.2. HTTP responses

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

9.2.3. /apis/coordination.k8s.io/v1/namespaces/{namespace}/leases**HTTP method****DELETE****Description**

delete collection of Lease

Table 9.3. 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 9.4. HTTP responses

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

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

list or watch objects of kind Lease

Table 9.5. HTTP responses

HTTP code	Reponse body
200 - OK	LeaseList schema

HTTP code	Response body
401 - Unauthorized	Empty

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

create a Lease

Table 9.6. 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 9.7. Body parameters

Parameter	Type	Description
body	Lease schema	

Table 9.8. HTTP responses

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

9.2.4. /apis/coordination.k8s.io/v1/watch/namespaces/{namespace}/leases

HTTP method

GET

Description

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

Table 9.9. HTTP responses

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

9.2.5. /apis/coordination.k8s.io/v1/namespaces/{namespace}/leases/{name}

Table 9.10. Global path parameters

Parameter	Type	Description
name	string	name of the Lease

HTTP method

DELETE

Description

delete a Lease

Table 9.11. 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 9.12. HTTP responses

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

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

read the specified Lease

Table 9.13. HTTP responses

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

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

partially update the specified Lease

Table 9.14. 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 9.15. HTTP responses

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

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

replace the specified Lease

Table 9.16. 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 9.17. Body parameters

Parameter	Type	Description
body	Lease schema	

Table 9.18. HTTP responses

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

9.2.6. /apis/coordination.k8s.io/v1/watch/namespaces/{namespace}/leases/{name}

Table 9.19. Global path parameters

Parameter	Type	Description
name	string	name of the Lease

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

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

Table 9.20. HTTP responses

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

CHAPTER 10. NAMESPACE [V1]

Description

Namespace provides a scope for Names. Use of multiple namespaces is optional.

Type

object

10.1. 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	NamespaceSpec describes the attributes on a Namespace.
status	object	NamespaceStatus is information about the current status of a Namespace.

10.1.1. .spec

Description

NamespaceSpec describes the attributes on a Namespace.

Type

object

Property	Type	Description
finalizers	array (string)	Finalizers is an opaque list of values that must be empty to permanently remove object from storage. More info: https://kubernetes.io/docs/tasks/administer-cluster/namespaces/

10.1.2. .status

Description

NamespaceStatus is information about the current status of a Namespace.

Type

object

Property	Type	Description
conditions	array	Represents the latest available observations of a namespace's current state.
conditions[]	object	NamespaceCondition contains details about state of namespace.
phase	string	Phase is the current lifecycle phase of the namespace. More info: https://kubernetes.io/docs/tasks/administer-cluster/namespaces/ Possible enum values: - "Active" means the namespace is available for use in the system - "Terminating" means the namespace is undergoing graceful termination

10.1.3. .status.conditions

Description

Represents the latest available observations of a namespace's current state.

Type

array

10.1.4. .status.conditions[]

Description

NamespaceCondition contains details about state of namespace.

Type

object

Required

- **type**
- **status**

Property	Type	Description
lastTransitionTime	Time	Last time the condition transitioned from one status to another.
message	string	Human-readable message indicating details about last transition.
reason	string	Unique, one-word, CamelCase reason for the condition's last transition.
status	string	Status of the condition, one of True, False, Unknown.
type	string	Type of namespace controller condition.

10.2. API ENDPOINTS

The following API endpoints are available:

- **/api/v1/namespaces**
 - **GET**: list or watch objects of kind Namespace
 - **POST**: create a Namespace
- **/api/v1/watch/namespaces**
 - **GET**: watch individual changes to a list of Namespace. deprecated: use the 'watch' parameter with a list operation instead.
- **/api/v1/namespaces/{name}**

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

10.2.1. /api/v1/namespaces

HTTP method

GET

Description

list or watch objects of kind Namespace

Table 10.1. HTTP responses

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

HTTP method

POST

Description

create a Namespace

Table 10.2. 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 10.3. Body parameters

Parameter	Type	Description
body	Namespace schema	

Table 10.4. HTTP responses

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

10.2.2. /api/v1/watch/namespaces

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

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

Table 10.5. HTTP responses

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

10.2.3. /api/v1/namespaces/{name}**Table 10.6. Global path parameters**

Parameter	Type	Description
name	string	name of the Namespace

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

delete a Namespace

Table 10.7. Query parameters

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

Table 10.8. HTTP responses

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

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

read the specified Namespace

Table 10.9. HTTP responses

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

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

partially update the specified Namespace

Table 10.10. Query parameters

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

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

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

replace the specified Namespace

Table 10.12. 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 10.13. Body parameters

Parameter	Type	Description
body	Namespace schema	

Table 10.14. HTTP responses

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

10.2.4. /api/v1/watch/namespaces/{name}

Table 10.15. Global path parameters

Parameter	Type	Description
name	string	name of the Namespace

HTTP method

GET

Description

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

Table 10.16. HTTP responses

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

10.2.5. /api/v1/namespaces/{name}/status

Table 10.17. Global path parameters

Parameter	Type	Description
name	string	name of the Namespace

HTTP method

GET

Description

read status of the specified Namespace

Table 10.18. HTTP responses

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

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

partially update status of the specified Namespace

Table 10.19. 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 10.20. HTTP responses

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

HTTP code	Response body
401 - Unauthorized	Empty

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

replace status of the specified Namespace

Table 10.21. 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 10.22. Body parameters

Parameter	Type	Description
body	Namespace schema	

Table 10.23. HTTP responses

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

10.2.6. /api/v1/namespaces/{name}/finalize

Table 10.24. Global path parameters

Parameter	Type	Description
name	string	name of the Namespace

Table 10.25. Global query parameters

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

HTTP method

PUT

Description

replace finalize of the specified Namespace

Table 10.26. Body parameters

Parameter	Type	Description
body	Namespace schema	

Table 10.27. HTTP responses

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