operator.alauda.io

operatorstatuses

Description

OperatorStatus is the Schema for the operatorstatuses API

Type

`object`

Specification

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

Property	Type	Description
`metadata`	`ObjectMeta`	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
`spec`	`object`	OperatorStatusSpec defines the desired state of OperatorStatus
`status`	`object`	OperatorStatusStatus defines the observed state of OperatorStatus

.spec

Description

OperatorStatusSpec defines the desired state of OperatorStatus

Type

`object`

Required

`cr` `operator`

Property	Type	Description
`cr`	`object`	
`installedNamespace`	`array`	
`operator`	`object`	TypedLocalObjectReference contains enough information to let you locate the typed referenced object inside the same namespace.

.spec.cr

Type

`object`

Required

`kind` `name`

Property	Type	Description
`apiGroup`	`string`	APIGroup is the group for the resource being referenced. If APIGroup is not specified, the specified Kind must be in the core API group. For any other third-party types, APIGroup is required.
`kind`	`string`	Kind is the type of resource being referenced
`name`	`string`	Name is the name of resource being referenced
`namespace`	`string`	

.spec.installedNamespace

Type

`array`

.spec.installedNamespace[]

Type

`string`

.spec.operator

Description

TypedLocalObjectReference contains enough information to let you locate the typed referenced object inside the same namespace.

Type

`object`

Required

`kind` `name`

Property	Type	Description
`apiGroup`	`string`	APIGroup is the group for the resource being referenced. If APIGroup is not specified, the specified Kind must be in the core API group. For any other third-party types, APIGroup is required.
`kind`	`string`	Kind is the type of resource being referenced
`name`	`string`	Name is the name of resource being referenced

.status

Description

OperatorStatusStatus defines the observed state of OperatorStatus

Type

`object`

Property	Type	Description
`componentStatus`	`object`	INSERT ADDITIONAL STATUS FIELD - define observed state of cluster Important: Run "make" to regenerate code after modifying this file

.status.componentStatus

Description

INSERT ADDITIONAL STATUS FIELD - define observed state of cluster Important: Run "make" to regenerate code after modifying this file

Type

`object`

API Endpoints

The following API endpoints are available:

- `/apis/operator.alauda.io/v1alpha1/namespaces/{namespace}/operatorstatuses`
  - `DELETE`: delete collection of OperatorStatus
  - `GET`: list objects of kind OperatorStatus
  - `POST`: create a new OperatorStatus
- `/apis/operator.alauda.io/v1alpha1/namespaces/{namespace}/operatorstatuses/{name}`
  - `DELETE`: delete the specified OperatorStatus
  - `GET`: read the specified OperatorStatus
  - `PATCH`: partially update the specified OperatorStatus
  - `PUT`: replace the specified OperatorStatus

- `/apis/operator.alauda.io/v1alpha1/namespaces/{namespace}/operatorstatuses/{name}/status`
  - `GET`: read status of the specified OperatorStatus
  - `PATCH`: partially update status of the specified OperatorStatus
  - `PUT`: replace status of the specified OperatorStatus

/apis/operator.alauda.io/v1alpha1/namespaces/{namespace}/operatorstatuses

HTTP method

`DELETE`

Description

delete collection of OperatorStatus

HTTP responses

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

HTTP method

`GET`

Description

list objects of kind OperatorStatus

HTTP responses

HTTP code	Response body
200 - OK	`OperatorStatusList` schema
401 - Unauthorized	Empty

HTTP method

`POST`

Description

create a new OperatorStatus

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.

Body parameters

Parameter	Type	Description
`body`	`OperatorStatus` schema	`application/json` formatted

HTTP responses

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

/apis/operator.alauda.io/v1alpha1/namespaces/{namespace}/operatorstatuses/{name}

HTTP method

`DELETE`

Description

delete the specified OperatorStatus

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

HTTP responses

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

HTTP method

`GET`

Description

read the specified OperatorStatus

HTTP responses

HTTP code	Response body
200 - OK	`OperatorStatus` schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update the specified OperatorStatus

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.

Parameter	Type	Description
		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 responses

HTTP code	Response body
200 - OK	`OperatorStatus` schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace the specified OperatorStatus

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,

Parameter	Type	Description
		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.

Body parameters

Parameter	Type	Description
`body`	`OperatorStatus` schema	`application/json` formatted

HTTP responses

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

/apis/operator.alauda.io/v1alpha1/namespaces/{namespace}/operatorstatuses/{name}/status

HTTP method

`GET`

Description

read status of the specified OperatorStatus

HTTP responses

HTTP code	Response body
200 - OK	`OperatorStatus` schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update status of the specified OperatorStatus

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 responses

HTTP code	Response body
200 - OK	`OperatorStatus` schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace status of the specified OperatorStatus

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.

Body parameters

Parameter	Type	Description
`body`	`OperatorStatus` schema	`application/json` formatted

HTTP responses

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