

Platform Cluster APIs

AlaudaFeatureGate [alauda.io/v1]

Description

AlaudaFeatureGate describes a feature gate of the alauda product.

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 ↗

Property	Type	Description
<code>kind</code>	<code>string</code>	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 ↗
<code>metadata</code>	<code>ObjectMeta</code>	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
<code>spec</code>	<code>object</code>	AlaudaFeatureGateSpec defines the desired state of AlaudaFeatureGate.
<code>status</code>	<code>object</code>	AlaudaFeatureGateStatus defines the observed state of AlaudaFeatureGate

.spec

Description

AlaudaFeatureGateSpec defines the desired state of AlaudaFeatureGate.

Type

`object`

Required

`description`

`enabled`

`stage`

Property	Type	Description
<code>dependency</code>	<code>object</code>	Dependency is the dependencies of the feature gate.
<code>description</code>	<code>string</code>	Description is a human readable text of the feature gate.
<code>descriptionEn</code>	<code>string</code>	DescriptionEn is an English human readable text of the feature gate.
<code>enabled</code>	<code>boolean</code>	Enabled indicates if the feature gate is enabled manually.
<code>stage</code>	<code>string</code>	Stage is the stage of the feature gate, should be one of "Alpha", "Beta", "GA" or "EOF".

.spec.dependency

Description

Dependency is the dependencies of the feature gate.

Type

`object`

Property	Type	Description
<code>featureGates</code>	<code>array</code>	FeatureGates is the names of feature gates that depends on.

Property	Type	Description
<code>type</code>	<code>string</code>	Type is the type of dependency, should be one of "all" or "any".

`.spec.dependency.featureGates`

Description

FeatureGates is the names of feature gates that depends on.

Type

`array`

`.spec.dependency.featureGates[]`

Type

`string`

`.status`

Description

AlaudaFeatureGateStatus defines the observed state of AlaudaFeatureGate

Type

`object`

Required

`enabled`

Property	Type	Description
<code>enabled</code>	<code>boolean</code>	Enabled indicates if the feature gate is enabled at runtime.

API Endpoints

The following API endpoints are available:

- `/apis/alauda.io/v1/namespaces/{namespace}/alaudafeaturegates`
 - `DELETE` : delete collection of AlaudaFeatureGate
 - `GET` : list objects of kind AlaudaFeatureGate
 - `POST` : create a new AlaudaFeatureGate
- `/apis/alauda.io/v1/namespaces/{namespace}/alaudafeaturegates/{name}`
 - `DELETE` : delete the specified AlaudaFeatureGate
 - `GET` : read the specified AlaudaFeatureGate
 - `PATCH` : partially update the specified AlaudaFeatureGate
 - `PUT` : replace the specified AlaudaFeatureGate
- `/apis/alauda.io/v1/namespaces/{namespace}/alaudafeaturegates/{name}/status`
 - `GET` : read status of the specified AlaudaFeatureGate
 - `PATCH` : partially update status of the specified AlaudaFeatureGate
 - `PUT` : replace status of the specified AlaudaFeatureGate

/apis/alauda.io/v1/namespaces/{namespace}/alaudafeatur egates

HTTP method

`DELETE`

Description

delete collection of AlaudaFeatureGate

HTTP responses

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

HTTP method

GET

Description

list objects of kind AlaudaFeatureGate

HTTP responses

HTTP code	Response body
200 - OK	<code>AlaudaFeatureGateList</code> schema
401 - Unauthorized	Empty

HTTP method

POST

Description

create a new AlaudaFeatureGate

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	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
<code>fieldValidation</code>	<code>string</code>	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

Parameter	Type	Description
		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	AlaudaFeatureGate schema	application/json formatted

HTTP responses

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

**/apis/alauda.io/v1/namespaces/{namespace}/alaudafeatur
egates/{name}**

HTTP method

DELETE

Description

delete the specified AlaudaFeatureGate

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	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	<code>Status</code> schema
202 - Accepted	<code>Status</code> schema
401 - Unauthorized	Empty

HTTP method

`GET`

Description

read the specified AlaudaFeatureGate

HTTP responses

HTTP code	Response body
200 - OK	<code>AlaudaFeatureGate</code> schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update the specified AlaudaFeatureGate

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	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
<code>fieldValidation</code>	<code>string</code>	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	<code>AlaudaFeatureGate</code> schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace the specified AlaudaFeatureGate

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	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
<code>fieldValidation</code>	<code>string</code>	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
<code>body</code>	<code>AlaudaFeatureGate</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>AlaudaFeatureGate</code> schema
201 - Created	<code>AlaudaFeatureGate</code> schema
401 - Unauthorized	Empty

/apis/alauda.io/v1/namespaces/{namespace}/alaudafeaturegates/{name}/status

HTTP method

`GET`

Description

read status of the specified AlaudaFeatureGate

HTTP responses

HTTP code	Response body
200 - OK	<code>AlaudaFeatureGate</code> schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update status of the specified AlaudaFeatureGate

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further

Parameter	Type	Description
		processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	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	<code>AlaudaFeatureGate</code> schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace status of the specified AlaudaFeatureGate

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> 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
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> 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 <code>BadRequest</code> 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
<code>body</code>	<code>AlaudaFeatureGate</code> schema	<code>application/json</code> formatted

HTTP responses

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

HTTP code	Response body
401 - Unauthorized	Empty

AlertRule [ait.alauda.io/v1]

Description

AlertRule defines alerting rules for a Prometheus instance AlertRule is a copy of PrometheusRule, just different kind/group/version.

Type

`object`

Required

`spec`

Specification

Property	Type	Description
<code>apiVersion</code>	<code>string</code>	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 ↗
<code>kind</code>	<code>string</code>	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint

Property	Type	Description
		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 ↗
<code>metadata</code>	<code>ObjectMeta</code>	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
<code>spec</code>	<code>object</code>	Specification of desired alerting rule definitions for Prometheus.

.spec

Description

Specification of desired alerting rule definitions for Prometheus.

Type

`object`

Property	Type	Description
<code>groups</code>	<code>array</code>	Content of Prometheus rule file

.spec.groups

Description

Content of Prometheus rule file

Type

`array`

`.spec.groups[]`

Description

RuleGroup is a list of sequentially evaluated recording and alerting rules. Note: PartialResponseStrategy is only used by ThanosRuler and will be ignored by Prometheus instances. Valid values for this field are 'warn' or 'abort'. More info: <https://github.com/thanos-io/thanos/blob/master/docs/components/rule.md#partial-response>

Type

`object`

Required

`name``rules`

Property	Type	Description
<code>interval</code>	<code>string</code>	
<code>name</code>	<code>string</code>	
<code>partial_response_strategy</code>	<code>string</code>	
<code>rules</code>	<code>array</code>	

`.spec.groups[].rules`

Type

`array`

`.spec.groups[].rules[]`

Description

Rule describes an alerting or recording rule.

Type

`object`

Required

`expr`

Property	Type	Description
<code>alert</code>	<code>string</code>	
<code>annotations</code>	<code>object</code>	
<code>expr</code>		
<code>for</code>	<code>string</code>	
<code>labels</code>	<code>object</code>	
<code>record</code>	<code>string</code>	

`.spec.groups[].rules[].annotations`

Type

`object`

`.spec.groups[].rules[].labels`

Type

`object`

API Endpoints

The following API endpoints are available:

- `/apis/ait.alauda.io/v1/namespaces/{namespace}/alertrules`
 - `DELETE` : delete collection of AlertRule

- **GET** : list objects of kind AlertRule
- **POST** : create a new AlertRule
- `/apis/ait.alauda.io/v1/namespaces/{namespace}/alertrules/{name}`
 - **DELETE** : delete the specified AlertRule
 - **GET** : read the specified AlertRule
 - **PATCH** : partially update the specified AlertRule
 - **PUT** : replace the specified AlertRule

/apis/ait.alauda.io/v1/namespaces/{namespace}/alertrules

HTTP method

DELETE

Description

delete collection of AlertRule

HTTP responses

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

HTTP method

GET

Description

list objects of kind AlertRule

HTTP responses

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

HTTP method

POST

Description

create a new AlertRule

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	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
<code>fieldValidation</code>	<code>string</code>	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
<code>body</code>	<code>AlertRule</code> schema	<code>application/json</code> formatted

HTTP responses

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

/apis/ait.alauda.io/v1/namespaces/{namespace}/alertrules/{name}

HTTP method

`DELETE`

Description

delete the specified AlertRule

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	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	<code>Status</code> schema
202 - Accepted	<code>Status</code> schema

HTTP code	Response body
401 - Unauthorized	Empty

HTTP method

GET

Description

read the specified AlertRule

HTTP responses

HTTP code	Response body
200 - OK	<code>AlertRule</code> schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update the specified AlertRule

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	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
<code>fieldValidation</code>	<code>string</code>	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

Parameter	Type	Description
		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	<code>AlertRule</code> schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace the specified AlertRule

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	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
<code>fieldValidation</code>	<code>string</code>	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

Parameter	Type	Description
		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
<code>body</code>	<code>AlertRule</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>AlertRule</code> schema
201 - Created	<code>AlertRule</code> schema
401 - Unauthorized	Empty

ClusterAlaudaFeatureGate [alauda.io/v1]

Description

ClusterAlaudaFeatureGate describes a cluster level feature gate of the alauda product

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 ↗

Property	Type	Description
<code>kind</code>	<code>string</code>	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 ↗
<code>metadata</code>	<code>ObjectMeta</code>	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
<code>spec</code>	<code>object</code>	AlaudaFeatureGateSpec defines the desired state of AlaudaFeatureGate.
<code>status</code>	<code>object</code>	AlaudaFeatureGateStatus defines the observed state of AlaudaFeatureGate

.spec

Description

AlaudaFeatureGateSpec defines the desired state of AlaudaFeatureGate.

Type

`object`

Required

`description`

`enabled`

`stage`

Property	Type	Description
<code>dependency</code>	<code>object</code>	Dependency is the dependencies of the feature gate.
<code>description</code>	<code>string</code>	Description is a human readable text of the feature gate.
<code>descriptionEn</code>	<code>string</code>	DescriptionEn is an English human readable text of the feature gate.
<code>enabled</code>	<code>boolean</code>	Enabled indicates if the feature gate is enabled manually.
<code>stage</code>	<code>string</code>	Stage is the stage of the feature gate, should be one of "Alpha", "Beta", "GA" or "EOF".

.spec.dependency

Description

Dependency is the dependencies of the feature gate.

Type

`object`

Property	Type	Description
<code>featureGates</code>	<code>array</code>	FeatureGates is the names of feature gates that depends on.

Property	Type	Description
<code>type</code>	<code>string</code>	Type is the type of dependency, should be one of "all" or "any".

`.spec.dependency.featureGates`

Description

FeatureGates is the names of feature gates that depends on.

Type

`array`

`.spec.dependency.featureGates[]`

Type

`string`

`.status`

Description

AlaudaFeatureGateStatus defines the observed state of AlaudaFeatureGate

Type

`object`

Required

`enabled`

Property	Type	Description
<code>enabled</code>	<code>boolean</code>	Enabled indicates if the feature gate is enabled at runtime.

API Endpoints

The following API endpoints are available:

- `/apis/alauda.io/v1/clusteralaudafeaturegates`
 - **DELETE** : delete collection of ClusterAlaudaFeatureGate
 - **GET** : list objects of kind ClusterAlaudaFeatureGate
 - **POST** : create a new ClusterAlaudaFeatureGate
- `/apis/alauda.io/v1/clusteralaudafeaturegates/{name}`
 - **DELETE** : delete the specified ClusterAlaudaFeatureGate
 - **GET** : read the specified ClusterAlaudaFeatureGate
 - **PATCH** : partially update the specified ClusterAlaudaFeatureGate
 - **PUT** : replace the specified ClusterAlaudaFeatureGate
- `/apis/alauda.io/v1/clusteralaudafeaturegates/{name}/status`
 - **GET** : read status of the specified ClusterAlaudaFeatureGate
 - **PATCH** : partially update status of the specified ClusterAlaudaFeatureGate
 - **PUT** : replace status of the specified ClusterAlaudaFeatureGate

/apis/alauda.io/v1/clusteralaudafeaturegates

HTTP method

DELETE

Description

delete collection of ClusterAlaudaFeatureGate

HTTP responses

HTTP code	Response body
200 - OK	Status schema

HTTP code	Response body
401 - Unauthorized	Empty

HTTP method

GET

Description

list objects of kind ClusterAlaudaFeatureGate

HTTP responses

HTTP code	Response body
200 - OK	<code>ClusterAlaudaFeatureGateList</code> schema
401 - Unauthorized	Empty

HTTP method

POST

Description

create a new ClusterAlaudaFeatureGate

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	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
<code>fieldValidation</code>	<code>string</code>	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

Parameter	Type	Description
		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	ClusterAlaudaFeatureGate schema	application/json formatted

HTTP responses

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

/apis/alauda.io/v1/clusteralaudafeaturegates/{name}

HTTP method

DELETE

Description

delete the specified ClusterAlaudaFeatureGate

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	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	<code>Status</code> schema
202 - Accepted	<code>Status</code> schema
401 - Unauthorized	Empty

HTTP method

`GET`

Description

read the specified ClusterAlaudaFeatureGate

HTTP responses

HTTP code	Response body
200 - OK	<code>ClusterAlaudaFeatureGate</code> schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update the specified ClusterAlaudaFeatureGate

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> 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
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> 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 <code>BadRequest</code> 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	<code>ClusterAlaudaFeatureGate</code> schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace the specified `ClusterAlaudaFeatureGate`

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> 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
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> 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 <code>BadRequest</code> 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
<code>body</code>	<code>ClusterAlaudaFeatureGate</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>ClusterAlaudaFeatureGate</code> schema

HTTP code	Response body
201 - Created	<code>ClusterAlaudaFeatureGate</code> schema
401 - Unauthorized	Empty

/apis/alauda.io/v1/clusteralaudafeaturegates/{name}/status

HTTP method

`GET`

Description

read status of the specified ClusterAlaudaFeatureGate

HTTP responses

HTTP code	Response body
200 - OK	<code>ClusterAlaudaFeatureGate</code> schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update status of the specified ClusterAlaudaFeatureGate

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	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
<code>fieldValidation</code>	<code>string</code>	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	<code>ClusterAlaudaFeatureGate</code> schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace status of the specified ClusterAlaudaFeatureGate

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun

Parameter	Type	Description
		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
<code>fieldValidation</code>	<code>string</code>	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
<code>body</code>	<code>ClusterAlaudaFeatureGate</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>ClusterAlaudaFeatureGate</code> schema
201 - Created	<code>ClusterAlaudaFeatureGate</code> schema
401 - Unauthorized	Empty

Dashboard [ait.alauda.io/v1alpha1]

Description

Dashboard is the Schema for the dashboards 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 ↗

Property	Type	Description
<code>kind</code>	<code>string</code>	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 ↗
<code>metadata</code>	<code>ObjectMeta</code>	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
<code>spec</code>	<code>object</code>	DashboardSpec defines the desired state of Dashboard

.spec

Description

DashboardSpec defines the desired state of Dashboard

Type

`object`

Property	Type	Description
<code>panels</code>	<code>array</code>	Dashboard contains panels to show monitor indicator.

.spec.panels

Description

Dashboard contains panels to show monitor indicator.

Type

array

.spec.panels[]

Description

DashboardPanel defines the info of panel, including title, targets, position.

Type

object

Required

description

title

Property	Type	Description
description	string	Description is the description of a panel
targets	array	Targets define an array of indicator
title	string	Title is the title of a panel

.spec.panels[].targets

Description

Targets define an array of indicator

Type

array

.spec.panels[].targets[]

Type

`object`

Required

`displayName`

Property	Type	Description
<code>displayName</code>	<code>string</code>	DisplayName is the displayName of indicator
<code>expr</code>	<code>string</code>	Expr is the prometheus expression to query for custom indicator; no need for builtin indicator
<code>indicator</code>	<code>string</code>	Indicator is the name of indicator for builtin indicator, md5 of expr for custom indicator
<code>legend</code>	<code>string</code>	Legend is format to show the query result
<code>resources</code>	<code>array</code>	Resources defines the resources to monitor, empty array for all resources in a application; no need for custom indicator
<code>type</code>	<code>string</code>	Type is the type of a indicator, builtin or custom
<code>unit</code>	<code>string</code>	Unit is the unit of indicator

.spec.panels[].targets[].resources

Description

Resources defines the resources to monitor, empty array for all resources in a application;
no need for custom indicator

Type

array

.spec.panels[].targets[].resources[]

Type

object

Required

kind

name

Property	Type	Description
kind	string	Kind defines the kind of a resource
name	string	Name defines the name of a resource
namespace	string	Namespace defines the namespace of a resource

API Endpoints

The following API endpoints are available:

- `/apis/ait.alauda.io/v1alpha1/namespaces/{namespace}/dashboards`
 - `DELETE` : delete collection of Dashboard
 - `GET` : list objects of kind Dashboard

- **POST** : create a new Dashboard
- `/apis/ait.alauda.io/v1alpha1/namespaces/{namespace}/dashboards/{name}`
- **DELETE** : delete the specified Dashboard
- **GET** : read the specified Dashboard
- **PATCH** : partially update the specified Dashboard
- **PUT** : replace the specified Dashboard

/apis/ait.alauda.io/v1alpha1/namespaces/{namespace}/dashboards

HTTP method

DELETE

Description

delete collection of Dashboard

HTTP responses

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

HTTP method

GET

Description

list objects of kind Dashboard

HTTP responses

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

HTTP method

POST

Description

create a new Dashboard

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	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
<code>fieldValidation</code>	<code>string</code>	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
<code>body</code>	<code>Dashboard</code> schema	<code>application/json</code> formatted

HTTP responses

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

/apis/ait.alauda.io/v1alpha1/namespaces/{namespace}/dashboards/{name}

HTTP method

`DELETE`

Description

delete the specified Dashboard

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	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	<code>Status</code> schema
202 - Accepted	<code>Status</code> schema

HTTP code	Response body
401 - Unauthorized	Empty

HTTP method

GET

Description

read the specified Dashboard

HTTP responses

HTTP code	Response body
200 - OK	<code>Dashboard</code> schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update the specified Dashboard

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	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
<code>fieldValidation</code>	<code>string</code>	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

Parameter	Type	Description
		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	<code>Dashboard</code> schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace the specified Dashboard

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	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
<code>fieldValidation</code>	<code>string</code>	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

Parameter	Type	Description
		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	Dashboard schema	application/json formatted

HTTP responses

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

DisableToken [auth.alauda.io/v1]

This API is installed only when the chart is deployed for the global cluster. It is not available on workload clusters.

Description

DisableToken is the Schema for the disabletokens API

Type

object

Specification

Property	Type	Description
<code>apiVersion</code>	<code>string</code>	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
<code>kind</code>	<code>string</code>	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

Property	Type	Description
		CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗
metadata	ObjectMeta	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
spec	object	DisableTokenSpec defines the desired state of DisableToken
status	object	DisableTokenStatus defines the observed state of DisableToken

.spec

Description

DisableTokenSpec defines the desired state of DisableToken

Type

object

Required

Expiration

TokenMd5

Property	Type	Description
Expiration	string	
TokenMd5	string	

.status

Description

DisableTokenStatus defines the observed state of DisableToken

Type

object

API Endpoints

The following API endpoints are available:

- `/apis/auth.alauda.io/v1/disabletokens`
 - `DELETE` : delete collection of DisableToken
 - `GET` : list objects of kind DisableToken
 - `POST` : create a new DisableToken
- `/apis/auth.alauda.io/v1/disabletokens/{name}`
 - `DELETE` : delete the specified DisableToken
 - `GET` : read the specified DisableToken
 - `PATCH` : partially update the specified DisableToken
 - `PUT` : replace the specified DisableToken
- `/apis/auth.alauda.io/v1/disabletokens/{name}/status`
 - `GET` : read status of the specified DisableToken
 - `PATCH` : partially update status of the specified DisableToken
 - `PUT` : replace status of the specified DisableToken

/apis/auth.alauda.io/v1/disabletokens

HTTP method

DELETE

Description

delete collection of DisableToken

HTTP responses

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

HTTP method

GET

Description

list objects of kind DisableToken

HTTP responses

HTTP code	Response body
200 - OK	<code>DisableTokenList</code> schema
401 - Unauthorized	Empty

HTTP method

POST

Description

create a new DisableToken

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	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
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> 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 <code>BadRequest</code> 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
<code>body</code>	<code>DisableToken</code> schema	<code>application/json</code> formatted

HTTP responses

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

`/apis/auth.alauda.io/v1/disabletokens/{name}`

HTTP method

DELETE

Description

delete the specified DisableToken

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	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	<code>Status</code> schema
202 - Accepted	<code>Status</code> schema
401 - Unauthorized	Empty

HTTP method

GET

Description

read the specified DisableToken

HTTP responses

HTTP code	Response body
200 - OK	<code>DisableToken</code> schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update the specified DisableToken

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	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
<code>fieldValidation</code>	<code>string</code>	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	<code>DisableToken</code> schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace the specified DisableToken

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	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
<code>fieldValidation</code>	<code>string</code>	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
<code>body</code>	<code>DisableToken</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>DisableToken</code> schema
201 - Created	<code>DisableToken</code> schema
401 - Unauthorized	Empty

/apis/auth.alauda.io/v1/disabletokens/{name}/status

HTTP method

`GET`

Description

read status of the specified DisableToken

HTTP responses

HTTP code	Response body
200 - OK	<code>DisableToken</code> schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update status of the specified DisableToken

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further

Parameter	Type	Description
		processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	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	<code>DisableToken</code> schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace status of the specified DisableToken

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	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
<code>fieldValidation</code>	<code>string</code>	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
<code>body</code>	<code>DisableToken</code> schema	<code>application/json</code> formatted

HTTP responses

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

HTTP code	Response body
401 - Unauthorized	Empty

DisableUser [auth.alauda.io/v1]

This API is installed only when the chart is deployed for the global cluster. It is not available on workload clusters.

Description

DisableUser is the Schema for the disableusers 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

Property	Type	Description
		CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗
metadata	ObjectMeta	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
spec	object	DisableUserSpec defines the desired state of DisableUser
status	object	DisableUserStatus defines the observed state of DisableUser

.spec

Description

DisableUserSpec defines the desired state of DisableUser

Type

object

Required

Expiration User

Property	Type	Description
Expiration	string	
User	string	

.status

Description

DisableUserStatus defines the observed state of DisableUser

Type

object

API Endpoints

The following API endpoints are available:

- `/apis/auth.alauda.io/v1/disableusers`
 - `DELETE` : delete collection of DisableUser
 - `GET` : list objects of kind DisableUser
 - `POST` : create a new DisableUser
- `/apis/auth.alauda.io/v1/disableusers/{name}`
 - `DELETE` : delete the specified DisableUser
 - `GET` : read the specified DisableUser
 - `PATCH` : partially update the specified DisableUser
 - `PUT` : replace the specified DisableUser
- `/apis/auth.alauda.io/v1/disableusers/{name}/status`
 - `GET` : read status of the specified DisableUser
 - `PATCH` : partially update status of the specified DisableUser
 - `PUT` : replace status of the specified DisableUser

/apis/auth.alauda.io/v1/disableusers

HTTP method

DELETE

Description

delete collection of DisableUser

HTTP responses

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

HTTP method

GET

Description

list objects of kind DisableUser

HTTP responses

HTTP code	Response body
200 - OK	<code>DisableUserList</code> schema
401 - Unauthorized	Empty

HTTP method

POST

Description

create a new DisableUser

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	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
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> 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 <code>BadRequest</code> 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
<code>body</code>	<code>DisableUser</code> schema	<code>application/json</code> formatted

HTTP responses

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

`/apis/auth.alauda.io/v1/disableusers/{name}`

HTTP method

DELETE

Description

delete the specified DisableUser

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	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	<code>Status</code> schema
202 - Accepted	<code>Status</code> schema
401 - Unauthorized	Empty

HTTP method

GET

Description

read the specified DisableUser

HTTP responses

HTTP code	Response body
200 - OK	<code>DisableUser</code> schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update the specified DisableUser

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	<code>DisableUser</code> schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace the specified DisableUser

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	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
<code>fieldValidation</code>	<code>string</code>	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
<code>body</code>	<code>DisableUser</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>DisableUser</code> schema
201 - Created	<code>DisableUser</code> schema
401 - Unauthorized	Empty

/apis/auth.alauda.io/v1/disableusers/{name}/status

HTTP method

`GET`

Description

read status of the specified DisableUser

HTTP responses

HTTP code	Response body
200 - OK	<code>DisableUser</code> schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update status of the specified DisableUser

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further

Parameter	Type	Description
		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	DisableUser schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace status of the specified DisableUser

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> 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
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> 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 <code>BadRequest</code> 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
<code>body</code>	<code>DisableUser</code> schema	<code>application/json</code> formatted

HTTP responses

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

HTTP code	Response body
401 - Unauthorized	Empty

Feature [infrastructure.alauda.io/v1alpha1]

Type

`object`

Specification

Property	Type	Description
<code>apiVersion</code>	<code>string</code>	
<code>kind</code>	<code>string</code>	
<code>metadata</code>	<code>ObjectMeta</code>	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
<code>spec</code>	<code>object</code>	
<code>status</code>	<code>object</code>	

.spec

Type

`object`

Required

version

type

Property	Type	Description
accessInfo	object	accessInfo defines the server info,if the addon support some interface to access. eg promtheus, grafana
deployInfo	object	
instanceType	string	instanceType defines the tool to support the type. eg: prometheus, fluentd
resourceInfo	array	resourceInfo defines all the resource the addons contains
type	string	type defines the type of the addon. eg: metric, log, application etc...
version	string	version defines the version of the addon. it is used to compare with the version of the instance to descide upgrade or not

.spec.accessInfo

Description

accessInfo defines the server info,if the addon support some interface to access. eg promtheus, grafana

Type

object

.spec.deployInfo

Type

object

.spec.resourceInfo

Description

resourceInfo defines all the resource the addons contains

Type

array

.spec.resourceInfo[]

Type

object

Property	Type	Description
kind	string	
name	string	
namespace	string	

.status

Type

object

Property	Type	Description
status	object	

.status.status

Type

object

API Endpoints

The following API endpoints are available:

- `/apis/infrastructure.alauda.io/v1alpha1/features`
 - `DELETE` : delete collection of Feature
 - `GET` : list objects of kind Feature
 - `POST` : create a new Feature
- `/apis/infrastructure.alauda.io/v1alpha1/features/{name}`
 - `DELETE` : delete the specified Feature
 - `GET` : read the specified Feature
 - `PATCH` : partially update the specified Feature
 - `PUT` : replace the specified Feature
- `/apis/infrastructure.alauda.io/v1alpha1/features/{name}/status`
 - `GET` : read status of the specified Feature
 - `PATCH` : partially update status of the specified Feature
 - `PUT` : replace status of the specified Feature

/apis/infrastructure.alauda.io/v1alpha1/features

HTTP method

DELETE

Description

delete collection of Feature

HTTP responses

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

HTTP method

`GET`

Description

list objects of kind Feature

HTTP responses

HTTP code	Response body
200 - OK	<code>FeatureList</code> schema
401 - Unauthorized	Empty

HTTP method

`POST`

Description

create a new Feature

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	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
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore:

Parameter	Type	Description
		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
<code>body</code>	<code>Feature</code> schema	<code>application/json</code> formatted

HTTP responses

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

/apis/infrastructure.alauda.io/v1alpha1/features/{name}

HTTP method

DELETE

Description

delete the specified Feature

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	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	<code>Status</code> schema
202 - Accepted	<code>Status</code> schema
401 - Unauthorized	Empty

HTTP method

`GET`

Description

read the specified Feature

HTTP responses

HTTP code	Response body
200 - OK	<code>Feature</code> schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update the specified Feature

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	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
<code>fieldValidation</code>	<code>string</code>	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	<code>Feature</code> schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace the specified Feature

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	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
<code>fieldValidation</code>	<code>string</code>	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
<code>body</code>	<code>Feature</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>Feature</code> schema
201 - Created	<code>Feature</code> schema
401 - Unauthorized	Empty

/apis/infrastructure.alauda.io/v1alpha1/features/{name}/status

HTTP method

`GET`

Description

read status of the specified Feature

HTTP responses

HTTP code	Response body
200 - OK	<code>Feature</code> schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update status of the specified Feature

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further

Parameter	Type	Description
		processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	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	<code>Feature</code> schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace status of the specified Feature

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> 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
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> 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 <code>BadRequest</code> 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
<code>body</code>	<code>Feature</code> schema	<code>application/json</code> formatted

HTTP responses

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

HTTP code	Response body
401 - Unauthorized	Empty

GrafanaDashboard [ait.alauda.io/v1beta1]

Description

GrafanaDashboard is the Schema for the grafanadashboards 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 ↗

Property	Type	Description
kind	string	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗
metadata	ObjectMeta	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
spec	object	GrafanaDashboardSpec defines the desired state of GrafanaDashboard
status	object	GrafanaDashboardStatus defines the observed state of GrafanaDashboard

.spec

Description

GrafanaDashboardSpec defines the desired state of GrafanaDashboard

Type

object

Required

folder

json

Property	Type	Description
<code>folder</code>	<code>string</code>	Folder is an folder title for GrafanaDashboard. Edit GrafanaDashboard_types.go to remove/update
<code>json</code>	<code>string</code>	Json is an json content for GrafanaDashboard. Edit GrafanaDashboard_types.go to remove/update

.status

Description

GrafanaDashboardStatus defines the observed state of GrafanaDashboard

Type

`object`

Property	Type	Description
<code>conditions</code>	<code>array</code>	INSERT ADDITIONAL STATUS FIELD - define observed state of cluster Important: Run "make" to regenerate code after modifying this file

.status.conditions

Description

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

Type

`array`

.status.conditions[]

Description

GrafanaDashboardCondition contains condition information for a grafana dashboard object.

Type

object

Required

status

type

Property	Type	Description
destMD5	string	DestMD5 is md5 for content in grafana
lastTransitionTime	string	Last time the condition transit from one status to another.
message	string	Human readable message indicating details about last transition.
reason	string	Brief reason for the condition's last transition.
specMD5	string	SpecMD5 is md5 for spec field in grafana dashboard instance
status	string	Status of the condition, one of True, False, Unknown.

Property	Type	Description
<code>type</code>	<code>string</code>	Type of grafana dashboard condition.

API Endpoints

The following API endpoints are available:

- `/apis/ait.alauda.io/v1beta1/namespaces/{namespace}/grafanadashboards`
 - `DELETE` : delete collection of GrafanaDashboard
 - `GET` : list objects of kind GrafanaDashboard
 - `POST` : create a new GrafanaDashboard
- `/apis/ait.alauda.io/v1beta1/namespaces/{namespace}/grafanadashboards/{name}`
 - `DELETE` : delete the specified GrafanaDashboard
 - `GET` : read the specified GrafanaDashboard
 - `PATCH` : partially update the specified GrafanaDashboard
 - `PUT` : replace the specified GrafanaDashboard
- `/apis/ait.alauda.io/v1beta1/namespaces/{namespace}/grafanadashboards/{name}/status`
 - `GET` : read status of the specified GrafanaDashboard
 - `PATCH` : partially update status of the specified GrafanaDashboard
 - `PUT` : replace status of the specified GrafanaDashboard

`/apis/ait.alauda.io/v1beta1/namespaces/{namespace}/grafanadashboards`

HTTP method

DELETE

Description

delete collection of GrafanaDashboard

HTTP responses

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

HTTP method

GET

Description

list objects of kind GrafanaDashboard

HTTP responses

HTTP code	Response body
200 - OK	<code>GrafanaDashboardList</code> schema
401 - Unauthorized	Empty

HTTP method

POST

Description

create a new GrafanaDashboard

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further

Parameter	Type	Description
		processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	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
<code>body</code>	<code>GrafanaDashboard</code> schema	<code>application/json</code> formatted

HTTP responses

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

/apis/ait.alauda.io/v1beta1/namespaces/{namespace}/grafanadashboards/{name}

HTTP method

DELETE

Description

delete the specified GrafanaDashboard

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 GrafanaDashboard

HTTP responses

HTTP code	Response body
200 - OK	<code>GrafanaDashboard</code> schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update the specified GrafanaDashboard

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	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
<code>fieldValidation</code>	<code>string</code>	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	<code>GrafanaDashboard</code> schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace the specified GrafanaDashboard

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	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
<code>fieldValidation</code>	<code>string</code>	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
<code>body</code>	<code>GrafanaDashboard</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>GrafanaDashboard</code> schema
201 - Created	<code>GrafanaDashboard</code> schema
401 - Unauthorized	Empty

`/apis/ait.alauda.io/v1beta1/namespaces/{namespace}/grafanadashboards/{name}/status`

HTTP method

GET

Description

read status of the specified GrafanaDashboard

HTTP responses

HTTP code	Response body
200 - OK	<code>GrafanaDashboard</code> schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update status of the specified GrafanaDashboard

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	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
<code>fieldValidation</code>	<code>string</code>	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	<code>GrafanaDashboard</code> schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace status of the specified GrafanaDashboard

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	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
<code>fieldValidation</code>	<code>string</code>	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	GrafanaDashboard schema	application/json formatted

HTTP responses

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

ImageVersion [middleware.alauda.io/v1]

Description

ImageVersion is the Schema for the imageversions 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 ↗

Property	Type	Description
<code>kind</code>	<code>string</code>	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 ↗
<code>metadata</code>	<code>ObjectMeta</code>	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
<code>spec</code>	<code>object</code>	ImageVersionSpec defines the desired state of ImageVersion
<code>status</code>	<code>object</code>	ImageVersionStatus defines the observed state of ImageVersion

.spec

Description

ImageVersionSpec defines the desired state of ImageVersion

Type

`object`

Property	Type	Description
<code>components</code>	<code>object</code>	

Property	Type	Description
<code>crVersion</code>	<code>string</code>	INSERT ADDITIONAL SPEC FIELDS - desired state of cluster Important: Run "make" to regenerate code after modifying this file

.spec.components

Type

`object`

.status

Description

ImageVersionStatus defines the observed state of ImageVersion

Type

`object`

API Endpoints

The following API endpoints are available:

- `/apis/middleware.alauda.io/v1/imageversions`
 - `DELETE` : delete collection of ImageVersion
 - `GET` : list objects of kind ImageVersion
 - `POST` : create a new ImageVersion
- `/apis/middleware.alauda.io/v1/imageversions/{name}`
 - `DELETE` : delete the specified ImageVersion
 - `GET` : read the specified ImageVersion

- **PATCH** : partially update the specified ImageVersion
- **PUT** : replace the specified ImageVersion
- `/apis/middleware.alauda.io/v1/imageversions/{name}/status`
 - **GET** : read status of the specified ImageVersion
 - **PATCH** : partially update status of the specified ImageVersion
 - **PUT** : replace status of the specified ImageVersion

/apis/middleware.alauda.io/v1/imageversions

HTTP method

DELETE

Description

delete collection of ImageVersion

HTTP responses

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

HTTP method

GET

Description

list objects of kind ImageVersion

HTTP responses

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

HTTP method

POST

Description

create a new ImageVersion

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	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
<code>fieldValidation</code>	<code>string</code>	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
<code>body</code>	<code>ImageVersion</code> schema	<code>application/json</code> formatted

HTTP responses

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

/apis/middleware.alauda.io/v1/imageversions/{name}

HTTP method

`DELETE`

Description

delete the specified ImageVersion

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	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	<code>Status</code> schema
202 - Accepted	<code>Status</code> schema
401 - Unauthorized	Empty

HTTP method

GET

Description

read the specified ImageVersion

HTTP responses

HTTP code	Response body
200 - OK	<code>ImageVersion</code> schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update the specified ImageVersion

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> 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
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> 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	<code>ImageVersion</code> schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace the specified ImageVersion

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	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
<code>fieldValidation</code>	<code>string</code>	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	ImageVersion schema	application/json formatted

HTTP responses

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

/apis/middleware.alauda.io/v1/imageversions/{name}/status

S

HTTP method

GET

Description

read status of the specified ImageVersion

HTTP responses

HTTP code	Response body
200 - OK	<code>ImageVersion</code> schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update status of the specified ImageVersion

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	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
<code>fieldValidation</code>	<code>string</code>	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	<code>ImageVersion</code> schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace status of the specified ImageVersion

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	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
<code>fieldValidation</code>	<code>string</code>	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
<code>body</code>	<code>ImageVersion</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>ImageVersion</code> schema
201 - Created	<code>ImageVersion</code> schema
401 - Unauthorized	Empty

MonitorDashboard [ait.alauda.io/v1alpha1]

The installer creates this CRD only when it is not already present in the cluster.

Description

MonitorDashboard is the Schema for the templates API

Type

object

Required

spec

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 ↗

Property	Type	Description
<code>kind</code>	<code>string</code>	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 ↗
<code>metadata</code>	<code>ObjectMeta</code>	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
<code>spec</code>	<code>object</code>	

.spec

Type

`object`

Property	Type	Description
<code>body</code>	<code>object</code>	Body is a json content for MonitorDashboard. Same as GrafanaDashboard's Json

.spec.body

Description

Body is a json content for MonitorDashboard. Same as GrafanaDashboard's Json

Type

`object`

API Endpoints

The following API endpoints are available:

- `/apis/ait.alauda.io/v1alpha2/namespaces/{namespace}/monitordashboards`
 - **DELETE** : delete collection of MonitorDashboard
 - **GET** : list objects of kind MonitorDashboard
 - **POST** : create a new MonitorDashboard
- `/apis/ait.alauda.io/v1alpha2/namespaces/{namespace}/monitordashboards/{name}`
 - **DELETE** : delete the specified MonitorDashboard
 - **GET** : read the specified MonitorDashboard
 - **PATCH** : partially update the specified MonitorDashboard
 - **PUT** : replace the specified MonitorDashboard

/apis/ait.alauda.io/v1alpha2/namespaces/{namespace}/monitordashboards

HTTP method

DELETE

Description

delete collection of MonitorDashboard

HTTP responses

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

HTTP method

GET

Description

list objects of kind MonitorDashboard

HTTP responses

HTTP code	Response body
200 - OK	<code>MonitorDashboardList</code> schema
401 - Unauthorized	Empty

HTTP method

POST

Description

create a new MonitorDashboard

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	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
<code>fieldValidation</code>	<code>string</code>	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

Parameter	Type	Description
		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	MonitorDashboard schema	application/json formatted

HTTP responses

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

/apis/ait.alauda.io/v1alpha2/namespaces/{namespace}/monitordashboards/{name}

HTTP method

DELETE

Description

delete the specified MonitorDashboard

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	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	<code>Status</code> schema
202 - Accepted	<code>Status</code> schema
401 - Unauthorized	Empty

HTTP method

`GET`

Description

read the specified MonitorDashboard

HTTP responses

HTTP code	Response body
200 - OK	<code>MonitorDashboard</code> schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update the specified MonitorDashboard

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	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
<code>fieldValidation</code>	<code>string</code>	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	<code>MonitorDashboard</code> schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace the specified MonitorDashboard

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	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
<code>fieldValidation</code>	<code>string</code>	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
<code>body</code>	<code>MonitorDashboard</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>MonitorDashboard</code> schema

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

ProductEntry [alauda.io/v1alpha1]

This API is installed only when the chart is deployed for the global cluster. It is not available on workload clusters.

Description

ProductEntry is the Schema for the productentries API

Type

object

Specification

Property	Type	Description
<code>apiVersion</code>	<code>string</code>	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
<code>kind</code>	<code>string</code>	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

Property	Type	Description
		CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗
metadata	ObjectMeta	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
spec	object	ProductEntrySpec defines the desired state of ProductEntry
status	object	ProductEntryStatus defines the observed state of ProductEntry

.spec

Description

ProductEntrySpec defines the desired state of ProductEntry

Type

object

Required

description

displayName

entrypoint

logo

packType

version

Property	Type	Description
catalog	object	Catalog stands for the product's catalog

Property	Type	Description
<code>description</code>	<code>object</code>	Description stands for the product's description
<code>displayName</code>	<code>object</code>	DisplayName stands for the product's display name
<code>entrypoint</code>	<code>string</code>	Entrypoint stands for the entrypoint of sub app
<code>logo</code>	<code>string</code>	Logo stands for the product logo with base64 encoded data uri
<code>packType</code>	<code>string</code>	PackType is the product's package type
<code>ssoEnabled</code>	<code>boolean</code>	SSOEnabled stands for if SSO enabled
<code>version</code>	<code>string</code>	Version stands for product version
<code>entryTarget</code>	<code>string</code>	

.spec.catalog

Description

Catalog stands for the product's catalog

Type

`object`

Property	Type	Description
en	string	EN stands for text with English
zh	string	ZH stands for text with Simple Chinese

.spec.description

Description

Description stands for the product's description

Type

object

Property	Type	Description
en	string	EN stands for text with English
zh	string	ZH stands for text with Simple Chinese

.spec.displayName

Description

DisplayName stands for the product's display name

Type

object

Property	Type	Description
<code>en</code>	<code>string</code>	EN stands for text with English
<code>zh</code>	<code>string</code>	ZH stands for text with Simple Chinese

.status

Description

ProductEntryStatus defines the observed state of ProductEntry

Type

`object`

Property	Type	Description
<code>hiddenOnMenu</code>	<code>boolean</code>	HiddenOnPortal stands for if this product is hidden on menu
<code>hiddenOnPortal</code>	<code>boolean</code>	HiddenOnPortal stands for if this product is hidden on portal
<code>orderOnMenu</code>	<code>integer</code>	OrderOnMenu stands for the product's order on menu
<code>orderOnPortal</code>	<code>integer</code>	OrderOnPortal stands for the product's order on portal

API Endpoints

The following API endpoints are available:

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

/apis/alauda.io/v1alpha1/productentries

HTTP method

`DELETE`

Description

delete collection of ProductEntry

HTTP responses

HTTP code	Response body
200 - OK	<code>Status</code> schema

HTTP code	Response body
401 - Unauthorized	Empty

HTTP method

GET

Description

list objects of kind ProductEntry

HTTP responses

HTTP code	Response body
200 - OK	<code>ProductEntryList</code> schema
401 - Unauthorized	Empty

HTTP method

POST

Description

create a new ProductEntry

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	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
<code>fieldValidation</code>	<code>string</code>	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

Parameter	Type	Description
		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	ProductEntry schema	application/json formatted

HTTP responses

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

/apis/alauda.io/v1alpha1/productentries/{name}

HTTP method

DELETE

Description

delete the specified ProductEntry

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	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	<code>Status</code> schema
202 - Accepted	<code>Status</code> schema
401 - Unauthorized	Empty

HTTP method

`GET`

Description

read the specified ProductEntry

HTTP responses

HTTP code	Response body
200 - OK	<code>ProductEntry</code> schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update the specified ProductEntry

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	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
<code>fieldValidation</code>	<code>string</code>	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	<code>ProductEntry</code> schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace the specified ProductEntry

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> 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
<code>fieldValidation</code>	<code>string</code>	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
<code>body</code>	<code>ProductEntry</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>ProductEntry</code> schema

HTTP code	Response body
201 - Created	<code>ProductEntry</code> schema
401 - Unauthorized	Empty

/apis/alauda.io/v1.alpha1/productentries/{name}/status

HTTP method

`GET`

Description

read status of the specified ProductEntry

HTTP responses

HTTP code	Response body
200 - OK	<code>ProductEntry</code> schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update status of the specified ProductEntry

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	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
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing

Parameter	Type	Description
		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	<code>ProductEntry</code> schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace status of the specified ProductEntry

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> 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
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> 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 <code>BadRequest</code> 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
<code>body</code>	<code>ProductEntry</code> schema	<code>application/json</code> formatted

HTTP responses

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

HTTP code	Response body
401 - Unauthorized	Empty

Project [auth.alauda.io/v1]

This API is installed only when the chart is deployed for the global cluster. It is not available on workload clusters.

Description

Project is the Schema for the projects API

Type

object

Specification

Property	Type	Description
<code>apiVersion</code>	<code>string</code>	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
<code>kind</code>	<code>string</code>	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

Property	Type	Description
		CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗
metadata	ObjectMeta	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
spec	object	ProjectSpec defines the desired state of Project
status	object	ProjectStatus defines the observed state of Project

.spec

Description

ProjectSpec defines the desired state of Project

Type

object

Property	Type	Description
clusterDeletePolicy	string	
clusters	array	Clusters contains the clusters associated with this Project

.spec.clusters

Description

Clusters contains the clusters associated with this Project

Type

array

.spec.clusters[]

Description

ProjectClusters is the attribute for ProjectSpec

Type

object

Required

name

quota

Property	Type	Description
name	string	Name of the cluster
quota	object	Quota store the quota info for Project
type	string	Type of the cluster

.spec.clusters[].quota

Description

Quota store the quota info for Project

Type

object

.status

Description

ProjectStatus defines the observed state of Project

Type

object

Property	Type	Description
conditions	array	
phase	string	Phase record the state of project
reason	string	
version	string	

.status.conditions

Type

array

.status.conditions[]

Description

ProjectCondition contains condition information for a cluster.

Type

object

Required

status

type

Property	Type	Description
<code>lastTransitionTime</code>	<code>string</code>	LastTransitionTime is the last time the condition changed from one status to another.
<code>message</code>	<code>string</code>	Message is a human-readable message indicating details about the last status change.
<code>projectNameSpaceDetail</code>	<code>object</code>	用来记录 project 同名命名空间详细信息
<code>reason</code>	<code>string</code>	Reason is a (brief) reason for the condition's last status change.
<code>status</code>	<code>string</code>	Status is the status of the condition. One of True, False, Unknown.
<code>type</code>	<code>string</code>	Type is the type of the cluster condition.

`.status.conditions[].projectNameSpaceDetail`

Description

用来记录 project 同名命名空间详细信息

Type

`object`

Property	Type	Description
<code>failedClusters</code>	<code>object</code>	
<code>name</code>	<code>string</code>	原始的项目名称，即要创建 ns 的名称
<code>successClusters</code>	<code>object</code>	

`.status.conditions[].projectNameSpaceDetail.failedClusterS`

Type

`object`

`.status.conditions[].projectNameSpaceDetail.successClusters`

Type

`object`

API Endpoints

The following API endpoints are available:

- `/apis/auth.alauda.io/v1/projects`
 - `DELETE` : delete collection of Project
 - `GET` : list objects of kind Project
 - `POST` : create a new Project
- `/apis/auth.alauda.io/v1/projects/{name}`
 - `DELETE` : delete the specified Project

- **GET** : read the specified Project
- **PATCH** : partially update the specified Project
- **PUT** : replace the specified Project
- **/apis/auth.alauda.io/v1/projects/{name}/status**
 - **GET** : read status of the specified Project
 - **PATCH** : partially update status of the specified Project
 - **PUT** : replace status of the specified Project

/apis/auth.alauda.io/v1/projects

HTTP method

DELETE

Description

delete collection of Project

HTTP responses

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

HTTP method

GET

Description

list objects of kind Project

HTTP responses

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

HTTP method

POST

Description

create a new Project

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	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
<code>fieldValidation</code>	<code>string</code>	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
<code>body</code>	<code>Project</code> schema	<code>application/json</code> formatted

HTTP responses

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

/apis/auth.alauda.io/v1/projects/{name}

HTTP method

`DELETE`

Description

delete the specified Project

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	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	<code>Status</code> schema
202 - Accepted	<code>Status</code> schema
401 - Unauthorized	Empty

HTTP method

GET

Description

read the specified Project

HTTP responses

HTTP code	Response body
200 - OK	<code>Project</code> schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update the specified Project

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> 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
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> 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	<code>Project</code> schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace the specified Project

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	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
<code>fieldValidation</code>	<code>string</code>	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	Project schema	application/json formatted

HTTP responses

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

/apis/auth.alauda.io/v1/projects/{name}/status

HTTP method

GET

Description

read status of the specified Project

HTTP responses

HTTP code	Response body
200 - OK	<code>Project</code> schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update status of the specified Project

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	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
<code>fieldValidation</code>	<code>string</code>	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	<code>Project</code> schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace status of the specified Project

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	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
<code>fieldValidation</code>	<code>string</code>	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
<code>body</code>	<code>Project</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>Project</code> schema
201 - Created	<code>Project</code> schema
401 - Unauthorized	Empty

User [auth.alauda.io/v1]

This API is installed only when the chart is deployed for the global cluster. It is not available on workload clusters.

Description

User is the Schema for the users API

Type

object

Specification

Property	Type	Description
<code>apiVersion</code>	<code>string</code>	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
<code>kind</code>	<code>string</code>	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

Property	Type	Description
		CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗
<code>metadata</code>	<code>ObjectMeta</code>	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
<code>spec</code>	<code>object</code>	UserSpec defines the desired state of User
<code>status</code>	<code>object</code>	UserStatus defines the observed state of User

.spec

Description

UserSpec defines the desired state of User

Type

`object`

Required

`connector_name`

`connector_type`

`email`

`is_admin`

`username`

Property	Type	Description
<code>account</code>	<code>string</code>	
<code>connector_id</code>	<code>string</code>	
<code>connector_name</code>	<code>string</code>	

Property	Type	Description
connector_type	string	INSERT ADDITIONAL SPEC FIELDS - desired state of cluster Important: Run "make" to regenerate code after modifying this file
continuity_errors	integer	
email	string	
expired	object	Expired ...
groups	array	
ids	array	
is_admin	boolean	
is_disabled	boolean	
last_login_time	string	
mail	string	
mobile	string	
old_password	string	
password	string	
state	string	State is User's State
username	string	
valid	boolean	

Property	Type	Description
<code>webhookType</code>	<code>string</code>	
<code>webhookUrl</code>	<code>string</code>	
<code>extra</code>	<code>object</code>	Extra contains additional arbitrary metadata for the user from third-party systems

`.spec.expired`

Description

Expired ...

Type

`object`

Required

`begin`

`end`

Property	Type	Description
<code>begin</code>	<code>string</code>	
<code>end</code>	<code>string</code>	

`.spec.groups`

Type

`array`

`.spec.groups[]`

Type

`string`

.spec.ids

Type

`array`

.spec.ids[]

Type

`object`

Required

`id``type`

Property	Type	Description
<code>id</code>	<code>string</code>	
<code>type</code>	<code>string</code>	

.spec.extra

Description

Extra contains additional arbitrary metadata for the user from third-party systems

Type

`object`

.status

Description

UserStatus defines the observed state of User

Type

object

API Endpoints

The following API endpoints are available:

- `/apis/auth.alauda.io/v1/users`
 - `DELETE` : delete collection of User
 - `GET` : list objects of kind User
 - `POST` : create a new User
- `/apis/auth.alauda.io/v1/users/{name}`
 - `DELETE` : delete the specified User
 - `GET` : read the specified User
 - `PATCH` : partially update the specified User
 - `PUT` : replace the specified User
- `/apis/auth.alauda.io/v1/users/{name}/status`
 - `GET` : read status of the specified User
 - `PATCH` : partially update status of the specified User
 - `PUT` : replace status of the specified User

/apis/auth.alauda.io/v1/users

HTTP method

`DELETE`

Description

delete collection of User

HTTP responses

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

HTTP method

`GET`

Description

list objects of kind User

HTTP responses

HTTP code	Response body
200 - OK	<code>UserList</code> schema
401 - Unauthorized	Empty

HTTP method

`POST`

Description

create a new User

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	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
<code>fieldValidation</code>	<code>string</code>	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

Parameter	Type	Description
		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	User schema	application/json formatted

HTTP responses

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

/apis/auth.alauda.io/v1/users/{name}

HTTP method

DELETE

Description

delete the specified User

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	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	<code>Status</code> schema
202 - Accepted	<code>Status</code> schema
401 - Unauthorized	Empty

HTTP method

`GET`

Description

read the specified User

HTTP responses

HTTP code	Response body
200 - OK	<code>User</code> schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update the specified User

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	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
<code>fieldValidation</code>	<code>string</code>	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	<code>User</code> schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace the specified User

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	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
<code>fieldValidation</code>	<code>string</code>	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
<code>body</code>	<code>User</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>User</code> schema
201 - Created	<code>User</code> schema
401 - Unauthorized	Empty

/apis/auth.alauda.io/v1/users/{name}/status

HTTP method

`GET`

Description

read status of the specified User

HTTP responses

HTTP code	Response body
200 - OK	<code>User</code> schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update status of the specified User

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	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
<code>fieldValidation</code>	<code>string</code>	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	<code>User</code> schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace status of the specified User

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun

Parameter	Type	Description
		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
<code>fieldValidation</code>	<code>string</code>	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
<code>body</code>	<code>User</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>User</code> schema
201 - Created	<code>User</code> schema
401 - Unauthorized	Empty

UserBinding [auth.alauda.io/v1]

Description

UserBinding is the Schema for the userbindings 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 ↗

Property	Type	Description
<code>kind</code>	<code>string</code>	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 ↗
<code>metadata</code>	<code>ObjectMeta</code>	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
<code>spec</code>	<code>object</code>	UserBindingSpec defines the desired state of UserBinding
<code>status</code>	<code>object</code>	UserBindingStatus defines the observed state of UserBinding

.spec

Description

UserBindingSpec defines the desired state of UserBinding

Type

`object`

Property	Type	Description
<code>constraint</code>	<code>array</code>	
<code>roleRef</code>	<code>string</code>	

Property	Type	Description
<code>scope</code>	<code>string</code>	
<code>subjects</code>	<code>array</code>	

`.spec.constraint`

Type

`array`

`.spec.constraint[]`

Type

`object`

Property	Type	Description
<code>cluster</code>	<code>string</code>	
<code>namespace</code>	<code>string</code>	
<code>project</code>	<code>string</code>	

`.spec.subjects`

Type

`array`

`.spec.subjects[]`

Type

`object`

Required

kind

name

Property	Type	Description
kind	string	
name	string	

.status

Description

UserBindingStatus defines the observed state of UserBinding

Type

object

Property	Type	Description
bindingsCreated	object	BindingsCreated records the count of created bindings
conditions	array	
failedClusters	array	FailedClusters lists clusters that failed to sync (global cluster only)
failureDetails	object	FailureDetails contains detailed error information for troubleshooting

Property	Type	Description
<code>lastReconcileTime</code>	<code>string</code>	LastReconcileTime records the last successful reconcile time Used to detect if reconciliation is stuck
<code>lastSpecHash</code>	<code>string</code>	
<code>nextRetryTime</code>	<code>string</code>	NextRetryTime indicates when the next retry will be attempted
<code>observedGeneration</code>	<code>integer</code>	ObservedGeneration records the last processed Generation Used to determine if status is stale
<code>phase</code>	<code>string</code>	
<code>retryCount</code>	<code>integer</code>	RetryCount tracks the number of retry attempts
<code>syncedClusters</code>	<code>array</code>	SyncedClusters lists successfully synced clusters (global cluster only)

`.status.bindingsCreated`

Description

BindingsCreated records the count of created bindings

Type

`object`

Property	Type	Description
clusterRoleBindings	integer	ClusterRoleBindings count of created ClusterRoleBindings
roleBindings	integer	RoleBindings count of created RoleBindings

.status.conditions

Type

array

.status.conditions[]

Type

object

Property	Type	Description
lastTransitionTime	string	
message	string	
reason	string	
status	string	
type	string	

.status.failedClusters

Description

FailedClusters lists clusters that failed to sync (global cluster only)

Type

array

.status.failedClusters[]

Type

string

.status.failureDetails

Description

FailureDetails contains detailed error information for troubleshooting

Type

object

Property	Type	Description
affectedItems	array	AffectedItems lists the specific resources that failed
failureMessage	string	FailureMessage is a human-readable detailed error message
failureReason	string	FailureReason is a short, machine-readable reason for the failure
lastFailureTime	string	LastFailureTime records when the last failure occurred

`.status.failureDetails.affectedItems`

Description

AffectedItems lists the specific resources that failed

Type

array

`.status.failureDetails.affectedItems[]`

Description

AffectedItem represents a specific resource that failed during reconciliation

Type

object

Property	Type	Description
<code>error</code>	<code>string</code>	Error is the specific error message for this item
<code>name</code>	<code>string</code>	Name is the name of the affected resource
<code>namespace</code>	<code>string</code>	Namespace is the namespace of the resource (for RoleBinding)
<code>type</code>	<code>string</code>	Type is the resource type (ClusterRoleBinding, RoleBinding, etc.)

`.status.syncedClusters`

Description

SyncedClusters lists successfully synced clusters (global cluster only)

Type

array

.status.syncedClusters[]

Type

string

API Endpoints

The following API endpoints are available:

- `/apis/auth.alauda.io/v1/userbindings`
 - `DELETE` : delete collection of UserBinding
 - `GET` : list objects of kind UserBinding
 - `POST` : create a new UserBinding
- `/apis/auth.alauda.io/v1/userbindings/{name}`
 - `DELETE` : delete the specified UserBinding
 - `GET` : read the specified UserBinding
 - `PATCH` : partially update the specified UserBinding
 - `PUT` : replace the specified UserBinding
- `/apis/auth.alauda.io/v1/userbindings/{name}/status`
 - `GET` : read status of the specified UserBinding
 - `PATCH` : partially update status of the specified UserBinding
 - `PUT` : replace status of the specified UserBinding

/apis/auth.alauda.io/v1/userbindings

HTTP method

DELETE

Description

delete collection of UserBinding

HTTP responses

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

HTTP method

GET

Description

list objects of kind UserBinding

HTTP responses

HTTP code	Response body
200 - OK	<code>UserBindingList</code> schema
401 - Unauthorized	Empty

HTTP method

POST

Description

create a new UserBinding

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further

Parameter	Type	Description
		processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	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
<code>body</code>	<code>UserBinding</code> schema	<code>application/json</code> formatted

HTTP responses

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

/apis/auth.alauda.io/v1/userbindings/{name}

HTTP method

DELETE

Description

delete the specified UserBinding

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	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	<code>Status</code> schema
202 - Accepted	<code>Status</code> schema
401 - Unauthorized	Empty

HTTP method

GET

Description

read the specified UserBinding

HTTP responses

HTTP code	Response body
200 - OK	<code>UserBinding</code> schema

HTTP code	Response body
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update the specified UserBinding

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	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
<code>fieldValidation</code>	<code>string</code>	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	<code>UserBinding</code> schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace the specified UserBinding

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	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
<code>fieldValidation</code>	<code>string</code>	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
<code>body</code>	<code>UserBinding</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>UserBinding</code> schema
201 - Created	<code>UserBinding</code> schema
401 - Unauthorized	Empty

/apis/auth.alauda.io/v1/userbindings/{name}/status

HTTP method

`GET`

Description

read status of the specified UserBinding

HTTP responses

HTTP code	Response body
200 - OK	<code>UserBinding</code> schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update status of the specified UserBinding

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> 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
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> 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 <code>BadRequest</code> 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	<code>UserBinding</code> schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace status of the specified `UserBinding`

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> 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
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> 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 <code>BadRequest</code> 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
<code>body</code>	<code>UserBinding</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>UserBinding</code> schema

HTTP code	Response body
201 - Created	<code>UserBinding</code> schema
401 - Unauthorized	Empty