Connector APIs

Connector [connectors.alauda.

ConnectorClass [connectors.alauda.io/v1alph

Connector [connectors.alauda.io/v1alpha1]

`connectors.alauda.io` group

Connector is the Schema for the connectors API

`v1alpha1` version

▼ spec `object`

ConnectorSpec defines the desired state of Connector

▼ address `string`

Address is connector address

▼ auth `object`

Auth represents authenticate data of current connector

▼ name `string`

Name represent auth name that configured in

ConnectorClass.spec.auth.types[].name

▼ params `[]object`

Param declares an ParamValues to use for the parameter called name.

▼ name `string` required

▼ **value** required

▼ **secretRef** `object`

SecretRef secret reference when doing authentication

▼ **apiVersion** `string`

API version of the referent.

▼ **fieldPath** `string`

If referring to a piece of an object instead of an entire object, this string should contain a valid JSON/Go field access statement, such as desiredState.manifest.containers[2]. For example, if the object reference is to a container within a pod, this would take on a value like: "spec.containers{name}" (where "name" refers to the name of the container that triggered the event) or if no container name is specified "spec.containers[2]" (container with index 2 in this pod). This syntax is chosen only to have some well-defined way of referencing a part of an object.

▼ **kind** `string`

Kind of the referent. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds↗

▼ **name** `string`

Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗

▼ **namespace** `string`

Namespace of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/namespaces/ ↗

▼ **resourceVersion** `string`

Specific resourceVersion to which this reference is made, if any. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#concurrency-control-and-consistency ↗

▼ **uid** `string`

UID of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#uids ↗

▼ **connectorClassName** `string`

ConnectorClassName of current connector

▼ **params** `[]object`

Param declares an ParamValues to use for the parameter called name.

▼ **name** `string` required

▼ **value** required

▼ **status** `object`

ConnectorStatus defines the observed state of Connector

▼ **annotations** `object`

Annotations is additional Status fields for the Resource to save some additional State as well as convey more information to the user. This is roughly akin to Annotations on any k8s resource, just the reconciler conveying richer information outwards.

▼ **api** `object`

API contains the status information for the connector's api

▼ **path** `string`

Path provides the path for the connector API. it is the path of the connector API. it is used to construct the api path for the connector

▼ **conditions** `[]object`

Condition defines a readiness condition for a Knative resource. See: https://github.com/kubernetes/community/blob/master/contributors/devel/sig-architecture/api-conventions.md#typical-status-properties ↗

▼ **lastTransitionTime** `string`

LastTransitionTime is the last time the condition transitioned from one status to another. We use VolatileTime in place of metav1.Time to exclude this from creating equality.Semantic differences (all other things held constant).

▼ **message** `string`

A human readable message indicating details about the transition.

▼ **reason** `string`

The reason for the condition's last transition.

▼ **severity** `string`

Severity with which to treat failures of this type of condition. When this is not specified, it defaults to Error.

▼ **status** `string` required

Status of the condition, one of True, False, Unknown.

▼ **type** `string` required

Type of condition.

▼ **observedGeneration** `integer`

ObservedGeneration is the 'Generation' of the Service that was last processed by the controller.

▼ **proxy** `object`

Proxy contains the status information for the connector's proxy

▼ **httpAddress** `object`

HTTPAddress provides the addressable HTTP endpoint for the connector proxy.

▼ **CACerts** `string`

CACerts is the Certification Authority (CA) certificates in PEM format according to https://www.rfc-editor.org/rfc/rfc7468↗.

▼ **audience** `string`

Audience is the OIDC audience for this address.

▼ **name** `string`

Name is the name of the address.

▼ **url** `string`

ConnectorClass [connectors.alauda.io/v1alpha1]

`connectors.alauda.io` group

ConnectorClass is the Schema for the connectorclasses API

`v1alpha1` version

▼ **spec** `object`

Spec defines the desired state of ConnectorClass

▼ **address** `object`

Address indicates address param constraints for this ConnectorClass of connectors we only support string param type

▼ **default**

Default is the value a parameter takes if no input value is supplied.

▼ **description** `string`

Description is a user-facing description of the parameter that may be used to populate a UI.

▼ **enum** `[]string`

Enum declares a set of allowed param input values. If Enum is not set, no input validation is performed for the param.

▼ **name** `string` required

Name declares the name by which a parameter is referenced.

▼ **properties** `object`

Properties is the JSON Schema properties to support key-value pairs parameter.

▼ **type** `string`

Type is the user-specified type of the parameter. The possible types are currently "string", "array" and "object", and "string" is the default.

▼ **api** `object`

API defines connectorclass plugin api address and openapi specification `api.ref` can be address of plugin api, it should be a kubernetes svc `api.uri` can be an absolute URL(non-empty scheme and non-empty host) pointing to the target or a relative URI. Relative URIs will be resolved using the base URI retrieved from Ref. `api.CACerts` and `api.audience` is not implemented now

▼ **CACerts** `string`

CACerts are Certification Authority (CA) certificates in PEM format according to https://www.rfc-editor.org/rfc/rfc7468↗. If set, these CAs are appended to the set of CAs provided by the Addressable target, if any.

▼ **audience** `string`

Audience is the OIDC audience. This need only be set, if the target is not an Addressable and thus the Audience can't be received from the Addressable itself. In case the Addressable specifies an Audience too, the Destinations Audience takes preference.

▼ openapi

OpenAPI defines the openapi specification for the connectorclass api This OpenAPI specification can describe an API customized for the ConnectorClass, or it can describe the original tool's API exposed via the ConnectorClass's proxy. if client request path is not in the OpenAPI specification, it will just forward the request to proxy of the connectorclass.

▼ permissions `object`

Permissions defines the request matching rules for different operations on the connectorclass api

▼ rego `string`

Rego contains the Rego policy script that maps API requests to permission verbs. The script evaluates incoming requests and determines the required permission level.

Requirements:

1. Must be defined under the 'permissions' package
2. Must produce a 'result' object matching the PermissionMappingResult structure

Input Variables:

- request.path (string): Request path
- request.method (string): HTTP method (GET, POST, PUT, DELETE, etc.)
- request.headers (map<string, list>): Request headers
- request.query (map<string, list>): Request query parameters

- request.body (dynamic): Parsed JSON body (available when Content-Type is application/json)

Output:

- result.verb (string): Permission verb - "read", "write", "delete", or "" (default to http verb mapping)

Example: package permissions import future.keywords.if

result = { "verb": "read" } if { startswith(input.request.path, "/search") input.request.method == "POST" } else = { "verb": "" }

▼ ref `object`

Ref points to an Addressable.

▼ address `string`

Address points to a specific Address Name.

▼ apiVersion `string`

API version of the referent.

▼ group `string`

Group of the API, without the version of the group. This can be used as an alternative to the APIVersion, and then resolved using ResolveGroup. Note: This API is EXPERIMENTAL and might break anytime. For more details: https://github.com/knative/eventing/issues/5086 ↗

▼ kind `string` required

Kind of the referent. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-

conventions.md#types-kinds↗

▼ **name** `string` required

Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names↗

▼ **namespace** `string`

Namespace of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/namespaces/↗ This is optional field, it gets defaulted to the object holding it if left out.

▼ **uri** `string`

URI can be an absolute URL(non-empty scheme and non-empty host) pointing to the target or a relative URI. Relative URIs will be resolved using the base URI retrieved from Ref.

▼ **auth** `object`

Auth indicates authentication constraints for this ConnectorClass of connectors

▼ **types** `[]object`

ConnectorClassAuthType represent the authentication types supported by connectors of the current connectorclass type

▼ **description** `string`

Description is the description of the AuthType

▼ displayName `string`

DisplayName is the human readable name of the AuthType

▼ generator `object`

Generator specifies how to generate authentication data dynamically. Can be used to implement custom authentication logic.

▼ rego `string`

Rego contains the Rego policy script that will be evaluated to generate authentication data. The script must define an 'auth' object that matches the following rules:

1. Define its rules under the 'proxy' package
2. Produce an 'auth' object containing AuthInjection structure.

▼ name `string` required

Name of the AuthType Must be unique within the ConnectorClass.

▼ optional `boolean`

Optional indicates whether the authentication information is optional for this ConnectorClass of connectors the default value is false

▼ params `[]object`

ParamSpec defines arbitrary parameters needed beyond typed inputs (such as resources).

▼ default

Default is the value a parameter takes if no input value is supplied.

▼ description `string`

Description is a user-facing description of the parameter that may be used to populate a UI.

▼ enum `[]string`

Enum declares a set of allowed param input values. If Enum is not set, no input validation is performed for the param.

▼ name `string` required

Name declares the name by which a parameter is referenced.

▼ properties `object`

Properties is the JSON Schema properties to support key-value pairs parameter.

▼ type `string`

Type is the user-specified type of the parameter. The possible types are currently "string", "array" and "object", and "string" is the default.

▼ secretType `string`

SecretType represents the secret type of the current authentication information follow k8s secret type definition. eg.kubernetes.io/basic-auth,

kubernetes.io/ssh-auth, kubernetes.io/opaque

▼ authProbes `[]object`

ConnectorClassAuthProbe represents network the detection configuration

▼ authName `string` required

AuthName corresponds to `spec.auth.types[].name`, indicating the way to check for the corresponding authentication type

▼ params `[]object`

ParamSpec defines arbitrary parameters needed beyond typed inputs (such as resources).

▼ default

Default is the value a parameter takes if no input value is supplied.

▼ description `string`

Description is a user-facing description of the parameter that may be used to populate a UI.

▼ enum `[]string`

Enum declares a set of allowed param input values. If Enum is not set, no input validation is performed for the param.

▼ name `string` required

Name declares the name by which a parameter is referenced.

▼ properties `object`

Properties is the JSON Schema properties to support key-value pairs parameter.

▼ type `string`

Type is the user-specified type of the parameter. The possible types are currently "string", "array" and "object", and "string" is the default.

▼ probe `object`

Probe represents current detection configuration

▼ api `object`

API defines a network detection probe that uses the ConnectorClass API. The API endpoint for the probe is resolved from the ConnectorClass's `spec.api` configuration. If `spec.api` is not configured or is invalid, no API detection will be performed. The `Path` field within `APIProbeAction` specifies a relative path that is appended to the host of URI resolved from `spec.api` (e.g., `spec.api.uri`).

▼ path `string`

Path represents the API address accessed during the current detection it is relative path, it will be appended to the host of URI resolved from `spec.api` (e.g., `spec.api.uri`) when execute the probe

▼ http `object`

Http represents the network detection using the http get method

▼ disableRedirect `boolean`

DisableRedirect indicates whether the probe should follow redirects, default is false

▼ host `string`

Host represents the tool address for the current detection. usually, it would be empty, it will use the `spec.address` of connector

▼ httpHeaders `[]object`

HTTPHeader describes a custom header to be used in HTTP probes

▼ name `string` required

The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.

▼ value `string` required

The header field value

▼ method `string`

Method represents the HTTP method to use for the probe, support method: GET, POST. default is GET

▼ path `string` required

Path represents the API address accessed during the current detection it is relative path, it will be appended to the host of URI resolved from `spec.address` of connector when execute the probe

▼ response `object`

Response defines the expected response validation rules. If not specified, the probe is considered successful if the request completes without error and returns a 2xx status code.

▼ cel `string`

CEL contains a CEL (Common Expression Language) expression for response validation. The expression must evaluate to a boolean value. Available variables:

- response.statusCode (int): HTTP status code
- response.headers (map<string, list>): Response headers
- response.body (dynamic): Parsed JSON body (if Content-Type is application/json and body is valid JSON)
- response.bodyString (string): Response body as string

Example expressions:

- response.statusCode == 200 && response.body.status == 'healthy'
- response.body.uptime > 60 && response.body.errors.size() == 0

- response.body.contains('success') && !response.body.contains('error')
- response.headers['content-type'][0].startsWith('application/json')

CEL is recommended for simple to medium complexity validations:

- Intuitive syntax similar to JavaScript/C/Python
- Better performance for simple expressions
- Native support in Kubernetes ecosystem
- Lower learning curve

When both CEL and built-in rules (Contains, Regex, JSONPath) are specified, all rules must pass (AND logic).

▼ scheme `string`

Scheme to use for connecting to the host. If empty:

- When Host is empty or matches the connector's address, the scheme from the connector's address is used.
- Otherwise, defaults to http. If specified, this value will be used regardless of Host or connector's address.

▼ configurations `[]object`

ConnectorClassConfiguration defines connectorclass configuration

▼ annotations `object`

Annotations is an unstructured key value map stored with a resource that may be set by external tools to store and retrieve arbitrary metadata. They are not

queryable and should be preserved when modifying objects. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/annotations ↗

▼ **data** `object`

Data contains the configuration data. Each key must consist of alphanumeric characters, '-', '_' or '.'.

▼ **name** `string`

Name of the configuration

▼ **livenessProbe** `object`

LivenessProbe defines liveness probe for this ConnectorClass of connectors

▼ **api** `object`

API defines a network detection probe that uses the ConnectorClass API. The API endpoint for the probe is resolved from the ConnectorClass's `spec.api` configuration. If `spec.api` is not configured or is invalid, no API detection will be performed. The `Path` field within `APIProbeAction` specifies a relative path that is appended to the host of URI resolved from `spec.api` (e.g., `spec.api.uri`).

▼ **path** `string`

Path represents the API address accessed during the current detection it is relative path, it will be appended to the host of URI resolved from `spec.api` (e.g., `spec.api.uri`) when execute the probe

▼ **http** `object`

Http represents the network detection using the http get method

▼ disableRedirect `boolean`

DisableRedirect indicates whether the probe should follow redirects, default is false

▼ host `string`

Host represents the tool address for the current detection. usually, it would be empty, it will use the `spec.address` of connector

▼ httpHeaders `[]object`

HTTPHeader describes a custom header to be used in HTTP probes

▼ name `string` required

The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.

▼ value `string` required

The header field value

▼ method `string`

Method represents the HTTP method to use for the probe, support method: GET, POST. default is GET

▼ path `string` required

Path represents the API address accessed during the current detection it is relative path, it will be appended to the host of URI resolved from

`spec.address` of connector when execute the probe

▼ response `object`

Response defines the expected response validation rules. If not specified, the probe is considered successful if the request completes without error and returns a 2xx status code.

▼ cel `string`

CEL contains a CEL (Common Expression Language) expression for response validation. The expression must evaluate to a boolean value. Available variables:

- response.statusCode (int): HTTP status code
- response.headers (map<string, list>): Response headers
- response.body (dynamic): Parsed JSON body (if Content-Type is application/json and body is valid JSON)
- response.bodyString (string): Response body as string

Example expressions:

- response.statusCode == 200 && response.body.status == 'healthy'
- response.body.uptime > 60 && response.body.errors.size() == 0
- response.body.contains('success') && !response.body.contains('error')
- response.headers['content-type'][0].startsWith('application/json')

CEL is recommended for simple to medium complexity validations:

- Intuitive syntax similar to JavaScript/C/Python
- Better performance for simple expressions
- Native support in Kubernetes ecosystem
- Lower learning curve

When both CEL and built-in rules (Contains, Regex, JSONPath) are specified, all rules must pass (AND logic).

▼ scheme `string`

Scheme to use for connecting to the host. If empty:

- When Host is empty or matches the connector's address, the scheme from the connector's address is used.
- Otherwise, defaults to http. If specified, this value will be used regardless of Host or connector's address.

▼ params `[]object`

ParamSpec defines arbitrary parameters needed beyond typed inputs (such as resources).

▼ default

Default is the value a parameter takes if no input value is supplied.

▼ description `string`

Description is a user-facing description of the parameter that may be used to populate a UI.

▼ enum `[]string`

Enum declares a set of allowed param input values. If Enum is not set, no input validation is performed for the param.

▼ name `string` required

Name declares the name by which a parameter is referenced.

▼ properties `object`

Properties is the JSON Schema properties to support key-value pairs parameter.

▼ type `string`

Type is the user-specified type of the parameter. The possible types are currently "string", "array" and "object", and "string" is the default.

▼ proxy `object`

Proxy defines the proxy configuration for this ConnectorClass. Specifies how network traffic should be routed through a proxy server.

▼ CACerts `string`

CACerts are Certification Authority (CA) certificates in PEM format according to https://www.rfc-editor.org/rfc/rfc7468↗. If set, these CAs are appended to the set of CAs provided by the Addressable target, if any.

▼ audience `string`

Audience is the OIDC audience. This need only be set, if the target is not an Addressable and thus the Audience can't be received from the Addressable itself. In case the Addressable specifies an Audience too, the Destinations Audience takes preference.

▼ authExtractor `object`

AuthExtractor specifies the method used to extract authentication data from incoming requests. The AuthExtractor is responsible for extracting the token

required to perform proxy permission validation.

▼ **rego** `string`

Rego contains the Rego policy script that will be evaluated to extract proxy authentication data from the request. The script must define an 'auth' object that matches the following rules:

1. Define its rules under the 'proxy' package
2. Produce an 'auth' object has structure same as ProxyAuthExtractorResult. Example: package proxy auth = { "token": input.request.headers["Private-Token"][0] }

▼ **ref** `object`

Ref points to an Addressable.

▼ **address** `string`

Address points to a specific Address Name.

▼ **apiVersion** `string`

API version of the referent.

▼ **group** `string`

Group of the API, without the version of the group. This can be used as an alternative to the APIVersion, and then resolved using ResolveGroup. Note: This API is EXPERIMENTAL and might break anytime. For more details: https://github.com/knative/eventing/issues/5086 ↗

▼ **kind** `string` required

Kind of the referent. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds↗

▼ **name** `string` required

Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names↗

▼ **namespace** `string`

Namespace of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/namespaces/↗ This is optional field, it gets defaulted to the object holding it if left out.

▼ **uri** `string`

URI can be an absolute URL(non-empty scheme and non-empty host) pointing to the target or a relative URI. Relative URIs will be resolved using the base URI retrieved from Ref.

▼ **status** `object`

Status defines the actual state of ConnectorClass

▼ **annotations** `object`

Annotations is additional Status fields for the Resource to save some additional State as well as convey more information to the user. This is roughly akin to Annotations on any k8s resource, just the reconciler conveying richer information outwards.

▼ api `object`

API represents status of connectorclass api it will resolved based on `spec.api` if `spec.api` is empty or invalid, it will not be set if current field is empty, the connectorclass cannot provides any api service.

▼ address `object`

Address is a single Addressable address.

▼ CACerts `string`

CACerts is the Certification Authority (CA) certificates in PEM format according to https://www.rfc-editor.org/rfc/rfc7468↗.

▼ audience `string`

Audience is the OIDC audience for this address.

▼ name `string`

Name is the name of the address.

▼ url `string`

▼ conditions `[]object`

Condition defines a readiness condition for a Knative resource. See: https://github.com/kubernetes/community/blob/master/contributors/devel/sig-architecture/api-conventions.md#typical-status-properties↗

▼ lastTransitionTime `string`

LastTransitionTime is the last time the condition transitioned from one status to another. We use VolatileTime in place of metav1.Time to exclude this from creating equality.Semantic differences (all other things held constant).

▼ message `string`

A human readable message indicating details about the transition.

▼ reason `string`

The reason for the condition's last transition.

▼ severity `string`

Severity with which to treat failures of this type of condition. When this is not specified, it defaults to Error.

▼ status `string` required

Status of the condition, one of True, False, Unknown.

▼ type `string` required

Type of condition.

▼ observedGeneration `integer`

ObservedGeneration is the 'Generation' of the Service that was last processed by the controller.

▼ proxy `object`

Proxy represents status of connectorclass proxy it will resolved based on `spec.proxy` if `spec.proxy` is empty or invalid, it will not be set if current field is empty, the connectorclass cannot provides any proxy service.

▼ httpAddress `object`

HttpAddress is a single Addressable address.

▼ CACerts `string`

CACerts is the Certification Authority (CA) certificates in PEM format according to https://www.rfc-editor.org/rfc/rfc7468↗.

▼ audience `string`

Audience is the OIDC audience for this address.

▼ name `string`

Name is the name of the address.

▼ url `string`