

---

[Alauda Container Platform](#) > [API References](#) > [Kubernetes APIs](#) > Marketplace APIs

---

# Marketplace APIs

---

[Artifact \[app.alauda.io/v1alpha1\]](#) [ArtifactVersion \[app.alauda.io/v1alpha1\]](#) [Chart \[app.alauda.io/v1alpha1\]](#)

[ChartRepo \[app.alauda.io/v1alpha1\]](#) [HelmRequest \[app.alauda.io/v1alpha1\]](#) [ImageWhite](#)

[Library \[app.alauda.io/v1alpha1\]](#)

# Artifact [app.alauda.io/v1alpha1]

## Description

Artifact is the Schema for the artifacts API

## Type

object

## Specification

Property	Type	Description
<code>apiVersion</code>	<code>string</code>	<p>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: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources</a></p>

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: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds</a>
metadata	ObjectMeta	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
spec	object	ArtifactSpec defines the desired state of Artifact
status	object	ArtifactStatus defines the observed state of Artifact

## .spec

### Description

ArtifactSpec defines the desired state of Artifact

### Type

object

### Required

artifactVersionSelector

present

registry

repository

type

Property	Type	Description
<code>artifactVersionSelector</code>	<code>object</code>	A label selector is a label query over a set of resources. The result of <code>matchLabels</code> and <code>matchExpressions</code> are ANDed. An empty label selector matches all objects. A null label selector matches no objects.
<code>description</code>	<code>string</code>	
<code>displayName</code>	<code>string</code>	
<code>imagePullSecrets</code>	<code>array</code>	
<code>present</code>	<code>boolean</code>	
<code>registry</code>	<code>string</code>	
<code>repository</code>	<code>string</code>	
<code>tagPatterns</code>	<code>array</code>	
<code>type</code>	<code>string</code>	

## `.spec.artifactVersionSelector`

### Description

A label selector is a label query over a set of resources. The result of `matchLabels` and `matchExpressions` are ANDed. An empty label selector matches all objects. A null label selector matches no objects.

### Type

`object`

Property	Type	Description
<code>matchExpressions</code>	<code>array</code>	<code>matchExpressions</code> is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	<code>object</code>	<code>matchLabels</code> is a map of {key,value} pairs. A single {key,value} in the <code>matchLabels</code> map is equivalent to an element of <code>matchExpressions</code> , whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

## `.spec.artifactVersionSelector.matchExpressions`

### Description

`matchExpressions` is a list of label selector requirements. The requirements are ANDed.

### Type

`array`

## `.spec.artifactVersionSelector.matchExpressions[]`

### Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

### Type

`object`

### Required

`key`

`operator`

Property	Type	Description
<code>key</code>	<code>string</code>	key is the label key that the selector applies to.
<code>operator</code>	<code>string</code>	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
<code>values</code>	<code>array</code>	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

## `.spec.artifactVersionSelector.matchExpressions[].values`

### Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

### Type

`array`

## `.spec.artifactVersionSelector.matchExpressions[].values[]`

### Type

`string`

## `.spec.artifactVersionSelector.matchLabels`

### Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

**Type**

object

**.spec.imagePullSecrets****Type**

array

**.spec.imagePullSecrets[]****Type**

string

**.spec.tagPatterns****Type**

array

**.spec.tagPatterns[]****Description**

Pattern is image version pattern

**Type**

object

**Required**

maxQuantity

name

regex

Property	Type	Description
maxQuantity		
name	string	
regex	string	

## .status

### Description

ArtifactStatus defines the observed state of Artifact

### Type

object

Property	Type	Description
availableArtifactVersion	integer	INSERT ADDITIONAL STATUS FIELD - define observed state of cluster Important: Run "make" to regenerate code after modifying this file
lastUpdateTime	string	
synced	boolean	

## API Endpoints

The following API endpoints are available:

- `/apis/app.alauda.io/v1alpha1/namespaces/{namespace}/artifacts`
  - **DELETE** : delete collection of Artifact
  - **GET** : list objects of kind Artifact

- **POST** : create a new Artifact
- `/apis/app.alauda.io/v1alpha1/namespaces/{namespace}/artifacts/{name}`
- **DELETE** : delete the specified Artifact
- **GET** : read the specified Artifact
- **PATCH** : partially update the specified Artifact
- **PUT** : replace the specified Artifact
- `/apis/app.alauda.io/v1alpha1/namespaces/{namespace}/artifacts/{name}/status`
- **GET** : read status of the specified Artifact
- **PATCH** : partially update status of the specified Artifact
- **PUT** : replace status of the specified Artifact

## `/apis/app.alauda.io/v1alpha1/namespaces/{namespace}/artifacts`

### HTTP method

**DELETE**

### Description

delete collection of Artifact

### HTTP responses

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

### HTTP method

**GET**

### Description

list objects of kind Artifact

## HTTP responses

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

## HTTP method

POST

## Description

create a new Artifact

## 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
body	Artifact schema	application/json formatted

## HTTP responses

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

# /apis/app.alauda.io/v1alpha1/namespaces/{namespace}/artifacts/{name}

## HTTP method

DELETE

## Description

delete the specified Artifact

## 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	<code>Status</code> schema
202 - Accepted	<code>Status</code> schema
401 - Unauthorized	Empty

## HTTP method

GET

## Description

read the specified Artifact

## HTTP responses

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

## HTTP method

PATCH

## Description

partially update the specified Artifact

## 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:

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.

## HTTP responses

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

## HTTP method

PUT

## Description

replace the specified Artifact

## 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>	<p><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:</p> <ul style="list-style-type: none"> <li>- 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.</li> <li>- 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+.</li> <li>- 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.</li> </ul>

### Body parameters

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

### HTTP responses

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

**`/apis/app.alauda.io/v1alpha1/namespaces/{namespace}/artifacts/{name}/status`**

## HTTP method

GET

## Description

read status of the specified Artifact

## HTTP responses

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

## HTTP method

PATCH

## Description

partially update status of the specified Artifact

## 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

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

## HTTP method

PUT

## Description

replace status of the specified Artifact

## 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

Parameter	Type	Description
		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	Artifact schema	application/json formatted

## HTTP responses

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

# ArtifactVersion [app.alauda.io/v1alpha1]

## Description

ArtifactVersion is the Schema for the artifactversions API

## Type

object

## Specification

Property	Type	Description
<code>apiVersion</code>	<code>string</code>	<p>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: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources</a></p>

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: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds</a>
<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>	ArtifactVersionSpec defines the desired state of ArtifactVersion
<code>status</code>	<code>object</code>	ArtifactVersionStatus defines the observed state of ArtifactVersion

## .spec

### Description

ArtifactVersionSpec defines the desired state of ArtifactVersion

### Type

`object`

### Required

`present`

`tag`

Property	Type	Description
present	boolean	
tag	string	

## .status

### Description

ArtifactVersionStatus defines the observed state of ArtifactVersion

### Type

object

Property	Type	Description
message	string	
name	string	
phase	string	INSERT ADDITIONAL STATUS FIELD - define observed state of cluster Important: Run "make" to regenerate code after modifying this file
reason	string	
type	string	
version	string	
chartInfo	object	ChartInfo defines the struct of chart info

## .status.chartInfo

## Description

ChartInfo defines the struct of chart info

## Type

object

Property	Type	Description
files	array	Files are miscellaneous files in a chart archive, e.g. README, LICENSE, etc.
metadata	ObjectMeta	Metadata is the contents of the Chartfile.
values	object	Values are default config for this chart.

## .status.chartInfo.files

### Description

Files are miscellaneous files in a chart archive, e.g. README, LICENSE, etc.

### Type

array

## .status.chartInfo.files[]

### Description

File represents a file as a name/value pair. By convention, name is a relative path within the scope of the chart's base directory.

### Type

object

### Required

data name

Property	Type	Description
data	string	Data is the template as byte data.
name	string	Name is the path-like name of the template.

## .status.chartInfo.values

### Description

Values are default config for this chart.

### Type

object

## API Endpoints

The following API endpoints are available:

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

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

## `/apis/app.alauda.io/v1alpha1/namespaces/{namespace}/artifactversions`

### HTTP method

`DELETE`

### Description

delete collection of ArtifactVersion

### HTTP responses

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

### HTTP method

`GET`

### Description

list objects of kind ArtifactVersion

### HTTP responses

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

### HTTP method

POST

## Description

create a new ArtifactVersion

## 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>ArtifactVersion</code> schema	<code>application/json</code> formatted

## HTTP responses

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

## /apis/app.alauda.io/v1alpha1/namespaces/{namespace}/artifactversions/{name}

### HTTP method

DELETE

### Description

delete the specified ArtifactVersion

### 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 ArtifactVersion

## HTTP responses

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

## HTTP method

PATCH

## Description

partially update the specified ArtifactVersion

## 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

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

## HTTP method

PUT

## Description

replace the specified ArtifactVersion

## 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

Parameter	Type	Description
		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	ArtifactVersion schema	application/json formatted

### HTTP responses

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

## /apis/app.alauda.io/v1alpha1/namespaces/{namespace}/artifactversions/{name}/status

### HTTP method

GET

### Description

read status of the specified ArtifactVersion

### HTTP responses

HTTP code	Response body
200 - OK	ArtifactVersion schema

HTTP code	Response body
401 - Unauthorized	Empty

## HTTP method

PATCH

## Description

partially update status of the specified ArtifactVersion

## 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>ArtifactVersion</code> schema
401 - Unauthorized	Empty

## HTTP method

PUT

## Description

replace status of the specified ArtifactVersion

## 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
body	ArtifactVersion schema	application/json formatted

## HTTP responses

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

# Chart [app.alauda.io/v1alpha1]

## Type

`object`

## Required

`spec`

## Specification

Property	Type	Description
<code>apiVersion</code>	<code>string</code>	<p>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: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources</a></p>

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: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds</a>
metadata	ObjectMeta	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
spec	object	

## .spec

### Type

object

Property	Type	Description
versions	array	

## .spec.versions

### Type

array

## .spec.versions[]

### Type

object

## API Endpoints

The following API endpoints are available:

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

## `/apis/app.alauda.io/v1beta1/namespaces/{namespace}/charts`

### HTTP method

**DELETE**

### Description

delete collection of Chart

### HTTP responses

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

## HTTP method

GET

## Description

list objects of kind Chart

## HTTP responses

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

## HTTP method

POST

## Description

create a new Chart

## 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

Parameter	Type	Description
		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	Chart schema	application/json formatted

## HTTP responses

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

# /apis/app.alauda.io/v1beta1/namespaces/{namespace}/charts/{name}

## HTTP method

DELETE

## Description

delete the specified Chart

## 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

## 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 Chart

## HTTP responses

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

## HTTP method

`PATCH`

## Description

partially update the specified Chart

## 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>Chart</code> schema
401 - Unauthorized	Empty

## HTTP method

`PUT`

## Description

replace the specified Chart

## 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>Chart</code> schema	<code>application/json</code> formatted

## HTTP responses

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



# ChartRepo [app.alauda.io/v1alpha1]

## Type

`object`

## Required

`spec`

## Specification

Property	Type	Description
<code>apiVersion</code>	<code>string</code>	<p>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: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources</a></p>

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: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds</a>
metadata	ObjectMeta	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
spec	object	
status	object	

## .spec

### Type

object

### Required

type url

Property	Type	Description
secret	object	Secret contains information about how to auth to this repo
source	object	new in v1beta1.if type is Chart, this is optional and it will provide some compatible with v1alpha1

Property	Type	Description
<code>type</code>	<code>string</code>	new in v1beta1
<code>url</code>	<code>string</code>	URL is the repo's url

## **.spec.secret**

### **Description**

Secret contains information about how to auth to this repo

### **Type**

`object`

Property	Type	Description
<code>name</code>	<code>string</code>	Name is unique within a namespace to reference a secret resource.
<code>namespace</code>	<code>string</code>	Namespace defines the space within which the secret name must be unique.

## **.spec.source**

### **Description**

new in v1beta1.if type is Chart, this is optional and it will provide some compatible with v1alpha1

### **Type**

`object`

## Required

`url`

Property	Type	Description
<code>path</code>	<code>string</code>	may be root, may be a subdir
<code>url</code>	<code>string</code>	vcs url

## .status

### Type

`object`

Property	Type	Description
<code>phase</code>	<code>string</code>	Phase ... After create, this phase will be updated to indicate it's sync status If receive update event, and some field in spec changed, sync agagin.
<code>reason</code>	<code>string</code>	Reason is the failed reason

## API Endpoints

The following API endpoints are available:

- `/apis/app.alauda.io/v1beta1/namespaces/{namespace}/chartrepos`
- `DELETE`: delete collection of ChartRepo

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

## `/apis/app.alauda.io/v1beta1/namespaces/{namespace}/chartrepos`

### HTTP method

**DELETE**

### Description

delete collection of ChartRepo

### HTTP responses

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

### HTTP method

**GET**

### Description

list objects of kind ChartRepo

## HTTP responses

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

## HTTP method

POST

## Description

create a new ChartRepo

## 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

Parameter	Type	Description
		present. The error returned from the server will contain all unknown and duplicate fields encountered.

### Body parameters

Parameter	Type	Description
body	ChartRepo schema	application/json formatted

### HTTP responses

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

## /apis/app.alauda.io/v1beta1/namespaces/{namespace}/chartrepos/{name}

### HTTP method

DELETE

### Description

delete the specified ChartRepo

### Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result

Parameter	Type	Description
		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 ChartRepo

## HTTP responses

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

## HTTP method

PATCH

## Description

partially update the specified ChartRepo

## 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

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

## HTTP method

`PUT`

## Description

replace the specified ChartRepo

## 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>ChartRepo</code> schema	<code>application/json</code> formatted

## HTTP responses

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

# /apis/app.alauda.io/v1beta1/namespaces/{namespace}/chartrepos/{name}/status

## HTTP method

GET

## Description

read status of the specified ChartRepo

## HTTP responses

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

## HTTP method

PATCH

## Description

partially update status of the specified ChartRepo

## 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

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>ChartRepo</code> schema
401 - Unauthorized	Empty

## HTTP method

`PUT`

## Description

replace status of the specified ChartRepo

## 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	ChartRepo schema	application/json formatted

## HTTP responses

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

# HelmRequest [app.alauda.io/v1]

## 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: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources</a>

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: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds</a>
<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`

Property	Type	Description
<code>chart</code>	<code>string</code>	
<code>clusterName</code>	<code>string</code>	ClusterName is the cluster where the chart will be installed. If InstallToAllClusters=true, this field will be ignored
<code>dependencies</code>	<code>array</code>	Dependencies is the dependencies of this HelmRequest, it's a list of there names THE dependencies must lives in the same

Property	Type	Description
		namespace, and each of them must be in Synced status before we sync this HelmRequest
<code>installToAllClusters</code>	<code>boolean</code>	InstallToAllClusters will install this chart to all available clusters, even the cluster was created after this chart. If this field is true, ClusterName will be ignored(useless)
<code>namespace</code>	<code>string</code>	Namespace is the namespace where the Release object will be lived in. Notes this should be used with the values defined in the chart , otherwise the install will failed
<code>releaseName</code>	<code>string</code>	ReleaseName is the Release name to be generated, default to HelmRequest.Name. If we want to manually install this chart to multi clusters, we may have different HelmRequest name(with cluster prefix or suffix) and same release name
<code>source</code>	<code>object</code>	Source defines the source of chart, If this field is set, Chart and Version field will be ignored(useless)
<code>targetClusters</code>	<code>object</code>	TargetClusters defines the target clusters which the chart will be installed

Property	Type	Description
<code>values</code>	<code>object</code>	Values represents a collection of chart values.
<code>valuesFrom</code>	<code>array</code>	ValuesFrom represents values from ConfigMap/Secret...
<code>version</code>	<code>string</code>	

## `.spec.dependencies`

### Description

Dependencies is the dependencies of this HelmRequest, it's a list of their names. The dependencies must live in the same namespace, and each of them must be in Synced status before we sync this HelmRequest.

### Type

`array`

## `.spec.dependencies[]`

### Type

`string`

## `.spec.source`

### Description

Source defines the source of chart. If this field is set, Chart and Version fields will be ignored (useless).

### Type

`object`

Property	Type	Description
<code>http</code>	<code>object</code>	
<code>oci</code>	<code>object</code>	

## `.spec.source.http`

### Type

`object`

### Required

`url`

Property	Type	Description
<code>secretRef</code>	<code>string</code>	SecretRef A Secret reference, the secret should contain accessKeyId (user name) base64 encoded, and secretKey (password) also base64 encoded
<code>url</code>	<code>string</code>	URL is the URL of the http(s) endpoint

## `.spec.source.oci`

### Type

`object`

### Required

`repo`

Property	Type	Description
<code>repo</code>	<code>string</code>	Repo is the repo of the oci artifact
<code>secretRef</code>	<code>string</code>	SecretRef A Secret reference, the secret should contain accessKeyId (user name) base64 encoded, and secretKey (password) also base64 encoded

## **.spec.targetClusters**

### **Description**

TargetClusters defines the target clusters which the chart will be installed

### **Type**

`object`

Property	Type	Description
<code>clusterLabels</code>	<code>object</code>	ClusterLabels store the key value of labels to match clusters
<code>clusterNames</code>	<code>object</code>	ClusterNames store a list of cluster names

## **.spec.targetClusters.clusterLabels**

### **Description**

ClusterLabels store the key value of labels to match clusters

### **Type**

`object`

## Required

labelSelector

Property	Type	Description
labelSelector	object	

## `.spec.targetClusters.clusterLabels.labelSelector`

### Type

object

## `.spec.targetClusters.clusterNames`

### Description

ClusterNames store a list of cluster names

### Type

object

## Required

names

Property	Type	Description
names	array	

## `.spec.targetClusters.clusterNames.names`

### Type

array

## `.spec.targetClusters.clusterNames.names[]`

### Type

`string`

## `.spec.values`

### Description

Values represents a collection of chart values.

### Type

`object`

## `.spec.valuesFrom`

### Description

ValuesFrom represents values from ConfigMap/Secret...

### Type

`array`

## `.spec.valuesFrom[]`

### Description

ValuesFromSource represents a source of values, only one of it's fields may be set

### Type

`object`

Property	Type	Description
<code>configMapKeyRef</code>	<code>object</code>	ConfigMapKeyRef selects a key of a ConfigMap
<code>secretKeyRef</code>	<code>object</code>	SecretKeyRef selects a key of a Secret

## .spec.valuesFrom[].configMapKeyRef

### Description

ConfigMapKeyRef selects a key of a ConfigMap

### Type

object

### Required

key

Property	Type	Description
key	string	The key to select.
name	string	Name of the referent. More info: <a href="https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names">https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names</a> <sup>↗</sup> TODO: Add other useful fields. apiVersion, kind, uid?
optional	boolean	Specify whether the ConfigMap or its key must be defined

## .spec.valuesFrom[].secretKeyRef

### Description

SecretKeyRef selects a key of a Secret

### Type

object

### Required

key

Property	Type	Description
<code>key</code>	<code>string</code>	The key of the secret to select from. Must be a valid secret key.
<code>name</code>	<code>string</code>	Name of the referent. More info: <a href="https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names">https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names</a> <sup>↗</sup> TODO: Add other useful fields. apiVersion, kind, uid?
<code>optional</code>	<code>boolean</code>	Specify whether the Secret or its key must be defined

## .status

### Type

`object`

Property	Type	Description
<code>conditions</code>	<code>array</code>	
<code>lastSpecHash</code>	<code>string</code>	LastSpecHash store the has value of the synced spec, if this value not equal to the current one, means we need to do a update for the chart
<code>notes</code>	<code>string</code>	Notes is the contents from helm (after helm install successfully it will be printed to the console

Property	Type	Description
<code>phase</code>	<code>string</code>	HelmRequestPhase is a label for the condition of a HelmRequest at the current time.
<code>reason</code>	<code>string</code>	Reason will store the reason why the HelmRequest deploy failed
<code>syncedClusters</code>	<code>array</code>	SyncedClusters will store the synced clusters if InstallToAllClusters is true
<code>targetClusterSyncResults</code>	<code>object</code>	TargetClusterSyncResults will store the chart sync result of every target cluster
<code>version</code>	<code>string</code>	Verions is the real version that installed

## **.status.conditions**

### Type

`array`

## **.status.conditions[]**

### Type

`object`

Property	Type	Description
<code>lastProbeTime</code>	<code>string</code>	Last time we probed the condition.
<code>lastTransitionTime</code>	<code>string</code>	Last time the condition transitioned from one status to another.
<code>message</code>	<code>string</code>	Human-readable message indicating details about last transition.
<code>reason</code>	<code>string</code>	Unique, one-word, CamelCase reason for the condition's last transition.
<code>status</code>	<code>string</code>	Status is the status of the condition. Can be True, False, Unknown. More info: <a href="https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#pod-conditions">https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#pod-conditions</a> ↗
<code>type</code>	<code>string</code>	Type is the type of the condition. More info: <a href="https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#pod-conditions">https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#pod-conditions</a> ↗

## .status.syncedClusters

### Description

SyncedClusters will store the synced clusters if InstallToAllClusters is true

### Type

`array`

## `.status.syncedClusters[]`

### Type

`string`

## `.status.targetClusterSyncResults`

### Description

TargetClusterSyncResults will store the chart sync result of every target cluster

### Type

`object`

## API Endpoints

The following API endpoints are available:

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

- **PATCH** : partially update status of the specified HelmRequest
- **PUT** : replace status of the specified HelmRequest

## /apis/app.alauda.io/v1/namespaces/{namespace}/helmrequests

### HTTP method

**DELETE**

### Description

delete collection of HelmRequest

### HTTP responses

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

### HTTP method

**GET**

### Description

list objects of kind HelmRequest

### HTTP responses

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

### HTTP method

**POST**

### Description

create a new HelmRequest

## 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>HelmRequest</code> schema	<code>application/json</code> formatted

## HTTP responses

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

HTTP code	Response body
202 - Accepted	<code>HelmRequest</code> schema
401 - Unauthorized	Empty

## /apis/app.alauda.io/v1/namespaces/{namespace}/helmrequests/{name}

### HTTP method

DELETE

### Description

delete the specified HelmRequest

### 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 HelmRequest

## HTTP responses

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

## HTTP method

PATCH

## Description

partially update the specified HelmRequest

## 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

Parameter	Type	Description
		present. The error returned from the server will contain all unknown and duplicate fields encountered.

## HTTP responses

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

## HTTP method

PUT

## Description

replace the specified HelmRequest

## 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

Parameter	Type	Description
		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	HelmRequest schema	application/json formatted

### HTTP responses

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

## /apis/app.alauda.io/v1/namespaces/{namespace}/helmrequests/{name}/status

### HTTP method

GET

### Description

read status of the specified HelmRequest

### HTTP responses

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

## HTTP method

PATCH

## Description

partially update status of the specified HelmRequest

## 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>HelmRequest</code> schema
401 - Unauthorized	Empty

## HTTP method

PUT

## Description

replace status of the specified HelmRequest

## 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>HelmRequest</code> schema	<code>application/json</code> formatted

## HTTP responses

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

# ImageWhiteList [app.alauda.io/v1alpha1]

## Description

ImageWhiteList is the Schema for the imagewhitelists API

## Type

object

## Specification

Property	Type	Description
<code>apiVersion</code>	<code>string</code>	<p>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: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources</a></p>

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: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds</a>
metadata	ObjectMeta	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
spec	object	ImageWhiteListSpec defines the desired state of ImageWhiteList
status	object	ImageWhiteListStatus defines the observed state of ImageWhiteList

## .spec

### Description

ImageWhiteListSpec defines the desired state of ImageWhiteList

### Type

object

Property	Type	Description
repoList	array	
rewriteRules	array	

## **.spec.repoList**

### Type

array

## **.spec.repoList[]**

### Type

string

## **.spec.rewriteRules**

### Type

array

## **.spec.rewriteRules[]**

### Description

RewriteRule defines the rewrite rule of ImageWhiteList

### Type

object

Property	Type	Description
regexp	string	
replacement	string	

## **.status**

### Description

ImageWhiteListStatus defines the observed state of ImageWhiteList

### Type

`object`

## API Endpoints

The following API endpoints are available:

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

## `/apis/app.alauda.io/v1alpha1/namespaces/{namespace}/imagewhitelists`

### HTTP method

`DELETE`

### Description

delete collection of ImageWhiteList

### HTTP responses

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

## HTTP method

GET

## Description

list objects of kind ImageWhiteList

## HTTP responses

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

## HTTP method

POST

## Description

create a new ImageWhiteList

## 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	ImageWhiteList schema	application/json formatted

### HTTP responses

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

**/apis/app.alauda.io/v1alpha1/namespaces/{namespace}/imagewhitelists/{name}**

### HTTP method

DELETE

## Description

delete the specified ImageWhiteList

## 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 ImageWhiteList

## HTTP responses

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

## HTTP method

`PATCH`

## Description

partially update the specified ImageWhiteList

## 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>ImageWhiteList</code> schema
401 - Unauthorized	Empty

## HTTP method

`PUT`

## Description

replace the specified ImageWhiteList

## 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>ImageWhiteList</code> schema	<code>application/json</code> formatted

## HTTP responses

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

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

## /apis/app.alauda.io/v1alpha1/namespaces/{namespace}/imagewhitelists/{name}/status

### HTTP method

`GET`

### Description

read status of the specified ImageWhiteList

### HTTP responses

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

### HTTP method

`PATCH`

### Description

partially update status of the specified ImageWhiteList

### 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>ImageWhiteList</code> schema
401 - Unauthorized	Empty

## HTTP method

`PUT`

## Description

replace status of the specified ImageWhiteList

## 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

Parameter	Type	Description
		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>ImageWhiteList</code> schema	<code>application/json</code> formatted

## HTTP responses

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

# Library [app.alauda.io/v1alpha1]

## Description

Library is the Schema for the libraries API

## Type

object

## Specification

Property	Type	Description
<code>apiVersion</code>	<code>string</code>	<p>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: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources</a></p>

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: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds</a>
<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>	LibrarySpec defines the desired state of Library
<code>status</code>	<code>object</code>	LibraryStatus defines the observed state of Library

## .spec

### Description

LibrarySpec defines the desired state of Library

### Type

`object`

### Required

`artifactSelector`

`imagePullSecrets`

Property	Type	Description
<code>artifactSelector</code>	<code>object</code>	A label selector is a label query over a set of resources. The result of <code>matchLabels</code> and <code>matchExpressions</code> are ANDed. An empty label selector matches all objects. A null label selector matches no objects.
<code>imagePullSecrets</code>	<code>array</code>	

## `.spec.artifactSelector`

### Description

A label selector is a label query over a set of resources. The result of `matchLabels` and `matchExpressions` are ANDed. An empty label selector matches all objects. A null label selector matches no objects.

### Type

`object`

Property	Type	Description
<code>matchExpressions</code>	<code>array</code>	<code>matchExpressions</code> is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	<code>object</code>	<code>matchLabels</code> is a map of {key,value} pairs. A single {key,value} in the <code>matchLabels</code> map is equivalent to an element of <code>matchExpressions</code> , whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

## `.spec.artifactSelector.matchExpressions`

## Description

matchExpressions is a list of label selector requirements. The requirements are ANDed.

## Type

array

## .spec.artifactSelector.matchExpressions[]

## Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

## Type

object

## Required

key

operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

## .spec.artifactSelector.matchExpressions[].values

## Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

## Type

array

## **.spec.artifactSelector.matchExpressions[].values[]**

## Type

string

## **.spec.artifactSelector.matchLabels**

## Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

## Type

object

## **.spec.imagePullSecrets**

## Type

array

## **.spec.imagePullSecrets[]**

## Type

string

## **.status**

## Description

LibraryStatus defines the observed state of Library

## Type

object

## Required

lastUpdateTime

phase

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

## API Endpoints

The following API endpoints are available:

- `/apis/app.alauda.io/v1.alpha1/namespaces/{namespace}/libraries`
  - `DELETE` : delete collection of Library
  - `GET` : list objects of kind Library
  - `POST` : create a new Library
- `/apis/app.alauda.io/v1.alpha1/namespaces/{namespace}/libraries/{name}`
  - `DELETE` : delete the specified Library
  - `GET` : read the specified Library
  - `PATCH` : partially update the specified Library
  - `PUT` : replace the specified Library
- `/apis/app.alauda.io/v1.alpha1/namespaces/{namespace}/libraries/{name}/status`

- **GET** : read status of the specified Library
- **PATCH** : partially update status of the specified Library
- **PUT** : replace status of the specified Library

## /apis/app.alauda.io/v1alpha1/namespaces/{namespace}/libraries

### HTTP method

**DELETE**

### Description

delete collection of Library

### HTTP responses

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

### HTTP method

**GET**

### Description

list objects of kind Library

### HTTP responses

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

### HTTP method

**POST**

### Description

## create a new Library

## 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>Library</code> schema	<code>application/json</code> formatted

## HTTP responses

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

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

## /apis/app.alauda.io/v1.alpha1/namespaces/{namespace}/libraries/{name}

### HTTP method

DELETE

### Description

delete the specified Library

### 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 Library

## HTTP responses

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

## HTTP method

`PATCH`

## Description

partially update the specified Library

## 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

Parameter	Type	Description
		present. The error returned from the server will contain all unknown and duplicate fields encountered.

## HTTP responses

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

## HTTP method

`PUT`

## Description

replace the specified Library

## 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

Parameter	Type	Description
		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	Library schema	application/json formatted

### HTTP responses

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

## /apis/app.alauda.io/v1.alpha1/namespaces/{namespace}/libraries/{name}/status

### HTTP method

GET

### Description

read status of the specified Library

### HTTP responses

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

## HTTP method

PATCH

## Description

partially update status of the specified Library

## 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>Library</code> schema
401 - Unauthorized	Empty

## HTTP method

PUT

## Description

replace status of the specified Library

## 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>Library</code> schema	<code>application/json</code> formatted

## HTTP responses

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