Marketplace APIs

Artifact [app.alauda.io/v1alpha1] ArtifactVersion [app.alauda.io/v] Chart [app.a

ChartRepo [app.alauda.io/v1alp] HelmRequest [app.alauda.io/v1] 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
`apiVersion`	`string`	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources ↗

Property	Type	Description
`kind`	`string`	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗
`metadata`	`ObjectMeta`	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
`spec`	`object`	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
`artifactVersionSelector`	`object`	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.
`description`	`string`	
`displayName`	`string`	
`imagePullSecrets`	`array`	
`present`	`boolean`	
`registry`	`string`	
`repository`	`string`	
`tagPatterns`	`array`	
`type`	`string`	

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

.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
`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.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`	`integer|string`	
`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	`Status` schema
401 - Unauthorized	Empty

HTTP method

`GET`

Description

list objects of kind Artifact

HTTP responses

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

HTTP method

`POST`

Description

create a new 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

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

Body parameters

Parameter	Type	Description
`body`	`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	`Status` schema
202 - Accepted	`Status` schema
401 - Unauthorized	Empty

HTTP method

`GET`

Description

read the specified Artifact

HTTP responses

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

HTTP method

`PATCH`

Description

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

HTTP responses

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

HTTP method

`PUT`

Description

replace 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

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

Body parameters

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

HTTP responses

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

HTTP method

`PATCH`

Description

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

Parameter	Type	Description
		This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

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

HTTP method

`PUT`

Description

replace status of 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
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object,

Parameter	Type	Description
		and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
`body`	`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
`apiVersion`	`string`	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources ↗

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

HTTP method

`GET`

Description

list objects of kind ArtifactVersion

HTTP responses

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

HTTP method

`POST`

Description

create a new ArtifactVersion

Query parameters

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

Body parameters

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

HTTP responses

HTTP code	Response body
200 - OK	`ArtifactVersion` schema
201 - Created	`ArtifactVersion` schema
202 - Accepted	`ArtifactVersion` 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
`dryRun`	`string`	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

HTTP responses

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

HTTP method

`GET`

Description

read the specified ArtifactVersion

HTTP responses

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

HTTP method

`PATCH`

Description

partially update the specified ArtifactVersion

Query parameters

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

Parameter	Type	Description
		This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

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

HTTP method

`PUT`

Description

replace the specified ArtifactVersion

Query parameters

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

Parameter	Type	Description
		and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
`body`	`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
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update status of the specified ArtifactVersion

Query parameters

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

HTTP responses

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

HTTP method

`PUT`

Description

replace status of the specified ArtifactVersion

Query parameters

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

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

Body parameters

Parameter	Type	Description
`body`	`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
`apiVersion`	`string`	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources ↗

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

.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	`Status` schema
401 - Unauthorized	Empty

HTTP method

`GET`

Description

list objects of kind Chart

HTTP responses

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

HTTP method

`POST`

Description

create a new Chart

Query parameters

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

Parameter	Type	Description
		This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

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

HTTP responses

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

HTTP method

`GET`

Description

read the specified Chart

HTTP responses

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

HTTP method

`PATCH`

Description

partially update the specified Chart

Query parameters

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

HTTP responses

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

HTTP method

`PUT`

Description

replace the specified Chart

Query parameters

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

Body parameters

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

HTTP responses

HTTP code	Response body
200 - OK	`Chart` schema

HTTP code	Response body
201 - Created	`Chart` schema
401 - Unauthorized	Empty

ChartRepo [app.alauda.io/v1alpha1]

Type

`object`

Required

`spec`

Specification

Property	Type	Description
`apiVersion`	`string`	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources ↗

Property	Type	Description
`kind`	`string`	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗
`metadata`	`ObjectMeta`	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
`spec`	`object`	
`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
`type`	`string`	new in v1beta1
`url`	`string`	URL is the repo's url

.spec.secret

Description

Secret contains information about how to auth to this repo

Type

`object`

Property	Type	Description
`name`	`string`	Name is unique within a namespace to reference a secret resource.
`namespace`	`string`	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
`path`	`string`	may be root, may be a subdir
`url`	`string`	vcs url

.status

Type

`object`

Property	Type	Description
`phase`	`string`	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.
`reason`	`string`	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

HTTP method

`DELETE`

Description

delete collection of ChartRepo

HTTP responses

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

HTTP method

`GET`

Description

list objects of kind ChartRepo

HTTP responses

HTTP code	Response body
200 - OK	`ChartRepoList` schema

HTTP code	Response body
401 - Unauthorized	Empty

HTTP method

`POST`

Description

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

Body parameters

Parameter	Type	Description
`body`	`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 in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

HTTP responses

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

HTTP method

`GET`

Description

read the specified ChartRepo

HTTP responses

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

HTTP method

`PATCH`

Description

partially update 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 in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore:

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	`ChartRepo` schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace 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 in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

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

Body parameters

Parameter	Type	Description
`body`	`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
`apiVersion`	`string`	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources ↗

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

.spec

Type

`object`

Property	Type	Description
`chart`	`string`	
`clusterName`	`string`	ClusterName is the cluster where the chart will be installed. If InstallToAllClusters=true, this field will be ignored
`dependencies`	`array`	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
`installToAllClusters`	`boolean`	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)
`namespace`	`string`	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
`releaseName`	`string`	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
`source`	`object`	Source defines the source of chart, If this field is set, Chart and Version field will be ignored(useless)
`targetClusters`	`object`	TargetClusters defines the target clusters which the chart will be installed

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

.spec.dependencies

Description

Dependencies is the dependencies of this HelmRequest, it's a list of there names THe dependencies must lives 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 field will be ignored(useless)

Type

`object`

Property	Type	Description
`http`	`object`	
`oci`	`object`	

.spec.source.http

Type

`object`

Required

`url`

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

.spec.source.oci

Type

`object`

Required

`repo`

Property	Type	Description
`repo`	`string`	Repo is the repo of the oci artifact
`secretRef`	`string`	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
`clusterLabels`	`object`	ClusterLabels store the key value of labels to match clusters
`clusterNames`	`object`	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
`configMapKeyRef`	`object`	ConfigMapKeyRef selects a key of a ConfigMap
`secretKeyRef`	`object`	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: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names↗ 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
`key`	`string`	The key of the secret to select from. Must be a valid secret key.
`name`	`string`	Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗ TODO: Add other useful fields. apiVersion, kind, uid?
`optional`	`boolean`	Specify whether the Secret or its key must be defined

.status

Type

`object`

Property	Type	Description
`conditions`	`array`	
`lastSpecHash`	`string`	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
`notes`	`string`	Notes is the contents from helm (after helm install successfully it will be printed to the console

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

.status.conditions

Type

`array`

.status.conditions[]

Type

`object`

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

.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	`Status` schema
401 - Unauthorized	Empty

HTTP method

`GET`

Description

list objects of kind HelmRequest

HTTP responses

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

HTTP method

`POST`

Description

create a new HelmRequest

Query parameters

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

Body parameters

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

HTTP responses

HTTP code	Response body
200 - OK	`HelmRequest` schema

HTTP code	Response body
201 - Created	`HelmRequest` schema
202 - Accepted	`HelmRequest` 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
`dryRun`	`string`	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

HTTP responses

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

HTTP method

`GET`

Description

read the specified HelmRequest

HTTP responses

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

HTTP method

`PATCH`

Description

partially update the specified HelmRequest

Query parameters

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

Parameter	Type	Description
		request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

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

HTTP method

`PUT`

Description

replace the specified HelmRequest

Query parameters

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

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

HTTP responses

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

HTTP method

`PUT`

Description

replace status of the specified HelmRequest

Query parameters

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

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

Body parameters

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

HTTP responses

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

ImageWhiteList [app.alauda.io/v1alpha1]

Description

ImageWhiteList is the Schema for the imagewhitelists API

Type

`object`

Specification

Property	Type	Description
`apiVersion`	`string`	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources ↗

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

HTTP method

`GET`

Description

list objects of kind ImageWhiteList

HTTP responses

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

HTTP method

`POST`

Description

create a new ImageWhiteList

Query parameters

Parameter	Type	Description
`dryRun`	`string`	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last

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

HTTP responses

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

HTTP method

`GET`

Description

read the specified ImageWhiteList

HTTP responses

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

HTTP method

`PATCH`

Description

partially update the specified ImageWhiteList

Query parameters

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

HTTP responses

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

HTTP method

`PUT`

Description

replace the specified ImageWhiteList

Query parameters

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

Body parameters

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

HTTP responses

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

HTTP method

`PATCH`

Description

partially update status of the specified ImageWhiteList

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

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

HTTP responses

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

HTTP method

`PUT`

Description

replace status of the specified ImageWhiteList

Query parameters

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

Body parameters

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

HTTP responses

HTTP code	Response body
200 - OK	`ImageWhiteList` schema
201 - Created	`ImageWhiteList` schema

HTTP code	Response body
401 - Unauthorized	Empty

Library [app.alauda.io/v1alpha1]

Description

Library is the Schema for the libraries API

Type

`object`

Specification

Property	Type	Description
`apiVersion`	`string`	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources ↗

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

.spec

Description

LibrarySpec defines the desired state of Library

Type

`object`

Required

`artifactSelector` `imagePullSecrets`

Property	Type	Description
`artifactSelector`	`object`	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.
`imagePullSecrets`	`array`	

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

.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/v1alpha1/namespaces/{namespace}/libraries`
  - `DELETE`: delete collection of Library
  - `GET`: list objects of kind Library
  - `POST`: create a new Library
- `/apis/app.alauda.io/v1alpha1/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/v1alpha1/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	`Status` schema
401 - Unauthorized	Empty

HTTP method

`GET`

Description

list objects of kind Library

HTTP responses

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

HTTP method

`POST`

Description

create a new Library

Query parameters

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

Body parameters

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

HTTP responses

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

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

HTTP method

`DELETE`

Description

delete the specified Library

Query parameters

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

HTTP responses

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

HTTP code	Response body
401 - Unauthorized	Empty

HTTP method

`GET`

Description

read the specified Library

HTTP responses

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

HTTP method

`PATCH`

Description

partially update the specified Library

Query parameters

Parameter	Type	Description
`dryRun`	`string`	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a

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	`Library` schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace the specified Library

Query parameters

Parameter	Type	Description
`dryRun`	`string`	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently

Parameter	Type	Description
		dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
`body`	`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/v1alpha1/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
`dryRun`	`string`	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server

Parameter	Type	Description
		will contain all unknown and duplicate fields encountered.

HTTP responses

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

HTTP method

`PUT`

Description

replace status of the specified Library

Query parameters

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

Parameter	Type	Description
		request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

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

HTTP responses

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