Inspection APIs

Inspection [inspections.ait.alauda.io/v1alpha1]

Inspection [inspections.ait.alauda.io/v1alpha1]

Description

Inspection is the Schema for the templates API

Type

`object`

Specification

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

Property	Type	Description
		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.
`status`	`object`	InspectionStatus defines the observed state of Inspection

.status

Description

InspectionStatus defines the observed state of Inspection

Type

`object`

Property	Type	Description
`conditions`	`array`	
`riskInspection`	`array`	
`usageInspection`	`array`	

.status.conditions

Type

`array`

.status.conditions[]

Type

`object`

Required

`inProgress` `status` `type`

Property	Type	Description
`endTime`	`string`	
`inProgress`	`string`	
`lastTransitionTime`	`string`	
`message`	`string`	
`reason`	`string`	
`startTime`	`string`	
`status`	`string`	
`type`	`string`	

.status.riskInspection

Type

`array`

.status.riskInspection[]

Type

`object`

Required

`kind` `stats` `status` `type`

Property	Type	Description
`endTime`	`string`	
`kind`	`string`	
`lastTransitionTime`	`string`	
`message`	`string`	
`reason`	`string`	
`startTime`	`string`	
`stats`	`object`	
`status`	`string`	
`type`	`string`	

.status.riskInspection[].stats

Type

`object`

.status.usageInspection

Type

`array`

.status.usageInspection[]

Type

`object`

Required

`kind` `status` `type`

Property	Type	Description
`endTime`	`string`	
`kind`	`string`	
`lastTransitionTime`	`string`	
`message`	`string`	
`reason`	`string`	
`resources`	`object`	
`startTime`	`string`	
`stats`	`object`	
`status`	`string`	
`type`	`string`	

.status.usageInspection[].resources

Type

`object`

.status.usageInspection[].stats

Type

`object`

Property	Type	Description
`total`	`object`	
`used`	`object`	

.status.usageInspection[].stats.total

Type

`object`

.status.usageInspection[].stats.used

Type

`object`

API Endpoints

The following API endpoints are available:

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

/apis/ait.alauda.io/v1alpha1/inspections

HTTP method

`DELETE`

Description

delete collection of Inspection

HTTP responses

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

HTTP method

`GET`

Description

list objects of kind Inspection

HTTP responses

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

HTTP method

`POST`

Description

create a new Inspection

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.

Body parameters

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

HTTP responses

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

/apis/ait.alauda.io/v1alpha1/inspections/{name}

HTTP method

`DELETE`

Description

delete the specified Inspection

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 Inspection

HTTP responses

HTTP code	Response body
200 - OK	`Inspection` schema

HTTP code	Response body
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update the specified Inspection

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	`Inspection` schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace the specified Inspection

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`	`Inspection` schema	`application/json` formatted

HTTP responses

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

/apis/ait.alauda.io/v1alpha1/inspections/{name}/status

HTTP method

`GET`

Description

read status of the specified Inspection

HTTP responses

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

HTTP method

`PATCH`

Description

partially update status of the specified Inspection

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	`Inspection` schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace status of the specified Inspection

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`	`Inspection` schema	`application/json` formatted

HTTP responses

HTTP code	Response body
200 - OK	`Inspection` schema

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