



# OpenShift Container Platform 4.18

## Console APIs

Reference guide for console APIs



# OpenShift Container Platform 4.18 Console APIs

---

Reference guide for console APIs

## Legal Notice

Copyright © 2025 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux<sup>®</sup> is the registered trademark of Linus Torvalds in the United States and other countries.

Java<sup>®</sup> is a registered trademark of Oracle and/or its affiliates.

XFS<sup>®</sup> is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL<sup>®</sup> is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js<sup>®</sup> is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack<sup>®</sup> Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

## Abstract

This document describes the OpenShift Container Platform console API objects and their detailed specifications.

# Table of Contents

<b>CHAPTER 1. CONSOLE APIS</b>	<b>4</b>
1.1. CONSOLECLIDOWNLOAD [CONSOLE.OPENSIFT.IO/V1]	4
1.2. CONSOLEEXTERNALLOGLINK [CONSOLE.OPENSIFT.IO/V1]	4
1.3. CONSOLELINK [CONSOLE.OPENSIFT.IO/V1]	4
1.4. CONSOLENOTIFICATION [CONSOLE.OPENSIFT.IO/V1]	4
1.5. CONSOLEPLUGIN [CONSOLE.OPENSIFT.IO/V1]	4
1.6. CONSOLEQUICKSTART [CONSOLE.OPENSIFT.IO/V1]	5
1.7. CONSOLESAMPLE [CONSOLE.OPENSIFT.IO/V1]	5
1.8. CONSOLEYAMLSAMPLE [CONSOLE.OPENSIFT.IO/V1]	5
<b>CHAPTER 2. CONSOLECLIDOWNLOAD [CONSOLE.OPENSIFT.IO/V1]</b>	<b>6</b>
2.1. SPECIFICATION	6
2.1.1. .spec	7
2.1.2. .spec.links	7
2.1.3. .spec.links[]	7
2.2. API ENDPOINTS	8
2.2.1. /apis/console.openshift.io/v1/consoleclidownloads	8
2.2.2. /apis/console.openshift.io/v1/consoleclidownloads/{name}	10
2.2.3. /apis/console.openshift.io/v1/consoleclidownloads/{name}/status	13
<b>CHAPTER 3. CONSOLEEXTERNALLOGLINK [CONSOLE.OPENSIFT.IO/V1]</b>	<b>17</b>
3.1. SPECIFICATION	17
3.1.1. .spec	18
3.2. API ENDPOINTS	20
3.2.1. /apis/console.openshift.io/v1/consoleexternalloglinks	20
3.2.2. /apis/console.openshift.io/v1/consoleexternalloglinks/{name}	22
3.2.3. /apis/console.openshift.io/v1/consoleexternalloglinks/{name}/status	25
<b>CHAPTER 4. CONSOLELINK [CONSOLE.OPENSIFT.IO/V1]</b>	<b>28</b>
4.1. SPECIFICATION	28
4.1.1. .spec	29
4.1.2. .spec.applicationMenu	29
4.1.3. .spec.namespaceDashboard	30
4.1.4. .spec.namespaceDashboard.namespaceSelector	30
4.1.5. .spec.namespaceDashboard.namespaceSelector.matchExpressions	31
4.1.6. .spec.namespaceDashboard.namespaceSelector.matchExpressions[]	31
4.2. API ENDPOINTS	32
4.2.1. /apis/console.openshift.io/v1/consolelinks	32
4.2.2. /apis/console.openshift.io/v1/consolelinks/{name}	34
4.2.3. /apis/console.openshift.io/v1/consolelinks/{name}/status	37
<b>CHAPTER 5. CONSOLENOTIFICATION [CONSOLE.OPENSIFT.IO/V1]</b>	<b>41</b>
5.1. SPECIFICATION	41
5.1.1. .spec	42
5.1.2. .spec.link	42
5.2. API ENDPOINTS	43
5.2.1. /apis/console.openshift.io/v1/consolenotifications	43
5.2.2. /apis/console.openshift.io/v1/consolenotifications/{name}	45
5.2.3. /apis/console.openshift.io/v1/consolenotifications/{name}/status	48
<b>CHAPTER 6. CONSOLEPLUGIN [CONSOLE.OPENSIFT.IO/V1]</b>	<b>52</b>
6.1. SPECIFICATION	52

6.1.1. .spec	53
6.1.2. .spec.backend	53
6.1.3. .spec.backend.service	54
6.1.4. .spec.i18n	55
6.1.5. .spec.proxy	55
6.1.6. .spec.proxy[]	55
6.1.7. .spec.proxy[].endpoint	56
6.1.8. .spec.proxy[].endpoint.service	57
6.2. API ENDPOINTS	57
6.2.1. /apis/console.openshift.io/v1/consoleplugins	58
6.2.2. /apis/console.openshift.io/v1/consoleplugins/{name}	59
<b>CHAPTER 7. CONSOLEQUICKSTART [CONSOLE.OPENSIFT.IO/V1]</b>	<b>63</b>
7.1. SPECIFICATION	63
7.1.1. .spec	64
7.1.2. .spec.accessReviewResources	65
7.1.3. .spec.accessReviewResources[]	65
7.1.4. .spec.accessReviewResources[].fieldSelector	67
7.1.5. .spec.accessReviewResources[].fieldSelector.requirements	68
7.1.6. .spec.accessReviewResources[].fieldSelector.requirements[]	68
7.1.7. .spec.accessReviewResources[].labelSelector	69
7.1.8. .spec.accessReviewResources[].labelSelector.requirements	70
7.1.9. .spec.accessReviewResources[].labelSelector.requirements[]	70
7.1.10. .spec.tasks	71
7.1.11. .spec.tasks[]	71
7.1.12. .spec.tasks[].review	72
7.1.13. .spec.tasks[].summary	72
7.2. API ENDPOINTS	73
7.2.1. /apis/console.openshift.io/v1/consolequickstarts	73
7.2.2. /apis/console.openshift.io/v1/consolequickstarts/{name}	75
<b>CHAPTER 8. CONSOLESAMPLE [CONSOLE.OPENSIFT.IO/V1]</b>	<b>79</b>
8.1. SPECIFICATION	79
8.1.1. .spec	80
8.1.2. .spec.source	82
8.1.3. .spec.source.containerImport	83
8.1.4. .spec.source.containerImport.service	83
8.1.5. .spec.source.gitImport	84
8.1.6. .spec.source.gitImport.repository	84
8.1.7. .spec.source.gitImport.service	85
8.2. API ENDPOINTS	86
8.2.1. /apis/console.openshift.io/v1/consolesamples	86
8.2.2. /apis/console.openshift.io/v1/consolesamples/{name}	88
<b>CHAPTER 9. CONSOLEYAMLSAMPLE [CONSOLE.OPENSIFT.IO/V1]</b>	<b>92</b>
9.1. SPECIFICATION	92
9.1.1. .spec	93
9.1.2. .spec.targetResource	94
9.2. API ENDPOINTS	94
9.2.1. /apis/console.openshift.io/v1/consoleyaml samples	95
9.2.2. /apis/console.openshift.io/v1/consoleyaml samples/{name}	96



## CHAPTER 1. CONSOLE APIS

### 1.1. CONSOLECLIDOWNLOAD [CONSOLE.OPENSIFT.IO/V1]

#### Description

ConsoleCLIDownload is an extension for configuring openshift web console command line interface (CLI) downloads.

Compatibility level 2: Stable within a major release for a minimum of 9 months or 3 minor releases (whichever is longer).

#### Type

**object**

### 1.2. CONSOLEEXTERNALLOGLINK [CONSOLE.OPENSIFT.IO/V1]

#### Description

ConsoleExternalLogLink is an extension for customizing OpenShift web console log links.

Compatibility level 2: Stable within a major release for a minimum of 9 months or 3 minor releases (whichever is longer).

#### Type

**object**

### 1.3. CONSOLELINK [CONSOLE.OPENSIFT.IO/V1]

#### Description

ConsoleLink is an extension for customizing OpenShift web console links.

Compatibility level 2: Stable within a major release for a minimum of 9 months or 3 minor releases (whichever is longer).

#### Type

**object**

### 1.4. CONSOLENOTIFICATION [CONSOLE.OPENSIFT.IO/V1]

#### Description

ConsoleNotification is the extension for configuring openshift web console notifications.

Compatibility level 2: Stable within a major release for a minimum of 9 months or 3 minor releases (whichever is longer).

#### Type

**object**

### 1.5. CONSOLEPLUGIN [CONSOLE.OPENSIFT.IO/V1]

#### Description

ConsolePlugin is an extension for customizing OpenShift web console by dynamically loading code from another service running on the cluster.



Compatibility level 1: Stable within a major release for a minimum of 12 months or 3 minor releases (whichever is longer).

**Type**

**object**

## 1.6. CONSOLEQUICKSTART [CONSOLE.OPENSIFT.IO/V1]

**Description**

ConsoleQuickStart is an extension for guiding user through various workflows in the OpenShift web console.

Compatibility level 2: Stable within a major release for a minimum of 9 months or 3 minor releases (whichever is longer).

**Type**

**object**

## 1.7. CONSOLESAMPLE [CONSOLE.OPENSIFT.IO/V1]

**Description**

ConsoleSample is an extension to customizing OpenShift web console by adding samples.

Compatibility level 1: Stable within a major release for a minimum of 12 months or 3 minor releases (whichever is longer).

**Type**

**object**

## 1.8. CONSOLEYAMLSAMPLE [CONSOLE.OPENSIFT.IO/V1]

**Description**

ConsoleYAMLSample is an extension for customizing OpenShift web console YAML samples.

Compatibility level 2: Stable within a major release for a minimum of 9 months or 3 minor releases (whichever is longer).

**Type**

**object**

## CHAPTER 2. CONSOLECLIDOWNLOAD [CONSOLE.OPENSIFT.IO/V1]

### Description

ConsoleCLIDownload is an extension for configuring openshift web console command line interface (CLI) downloads.

Compatibility level 2: Stable within a major release for a minimum of 9 months or 3 minor releases (whichever is longer).

### Type

**object**

### Required

- **spec**

## 2.1. SPECIFICATION

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

Property	Type	Description
<b>spec</b>	<b>object</b>	ConsoleCLIDownloadSpec is the desired cli download configuration.

### 2.1.1. .spec

#### Description

ConsoleCLIDownloadSpec is the desired cli download configuration.

#### Type

**object**

#### Required

- **description**
- **displayName**
- **links**

Property	Type	Description
<b>description</b>	<b>string</b>	description is the description of the CLI download (can include markdown).
<b>displayName</b>	<b>string</b>	displayName is the display name of the CLI download.
<b>links</b>	<b>array</b>	links is a list of objects that provide CLI download link details.
<b>links[]</b>	<b>object</b>	

### 2.1.2. .spec.links

#### Description

links is a list of objects that provide CLI download link details.

#### Type

**array**

### 2.1.3. .spec.links[]

#### Description

Type

**object**

Required

- **href**

Property	Type	Description
<b>href</b>	<b>string</b>	href is the absolute secure URL for the link (must use https)
<b>text</b>	<b>string</b>	text is the display text for the link

## 2.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/console.openshift.io/v1/consoleclidownloads**
  - **DELETE:** delete collection of ConsoleCLIDownload
  - **GET:** list objects of kind ConsoleCLIDownload
  - **POST:** create a ConsoleCLIDownload
- **/apis/console.openshift.io/v1/consoleclidownloads/{name}**
  - **DELETE:** delete a ConsoleCLIDownload
  - **GET:** read the specified ConsoleCLIDownload
  - **PATCH:** partially update the specified ConsoleCLIDownload
  - **PUT:** replace the specified ConsoleCLIDownload
- **/apis/console.openshift.io/v1/consoleclidownloads/{name}/status**
  - **GET:** read status of the specified ConsoleCLIDownload
  - **PATCH:** partially update status of the specified ConsoleCLIDownload
  - **PUT:** replace status of the specified ConsoleCLIDownload

### 2.2.1. /apis/console.openshift.io/v1/consoleclidownloads

HTTP method

**DELETE**

Description

delete collection of ConsoleCLIDownload

Table 2.1. HTTP responses

HTTP code	Reponse body
200 - OK	<a href="#">Status</a> schema
401 - Unauthorized	Empty

**HTTP method****GET****Description**

list objects of kind ConsoleCLIDownload

**Table 2.2. HTTP responses**

HTTP code	Reponse body
200 - OK	<a href="#">ConsoleCLIDownloadList</a> schema
401 - Unauthorized	Empty

**HTTP method****POST****Description**

create a ConsoleCLIDownload

**Table 2.3. Query parameters**

Parameter	Type	Description
<b>dryRun</b>	<b>string</b>	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
<b>fieldValidation</b>	<b>string</b>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: <ul style="list-style-type: none"> <li>- Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23.</li> <li>- Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+</li> <li>- Strict: This will fail the request with a 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.</li> </ul>

Table 2.4. Body parameters

Parameter	Type	Description
<b>body</b>	<a href="#">ConsoleCLIDownload</a> schema	

Table 2.5. HTTP responses

HTTP code	Response body
200 - OK	<a href="#">ConsoleCLIDownload</a> schema
201 - Created	<a href="#">ConsoleCLIDownload</a> schema
202 - Accepted	<a href="#">ConsoleCLIDownload</a> schema
401 - Unauthorized	Empty

### 2.2.2. /apis/console.openshift.io/v1/consoleclidownloads/{name}

Table 2.6. Global path parameters

Parameter	Type	Description
<b>name</b>	<b>string</b>	name of the ConsoleCLIDownload

**HTTP method****DELETE****Description**

delete a ConsoleCLIDownload

**Table 2.7. Query parameters**

Parameter	Type	Description
<b>dryRun</b>	<b>string</b>	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

**Table 2.8. HTTP responses**

HTTP code	Reponse body
200 - OK	<b>Status</b> schema
202 - Accepted	<b>Status</b> schema
401 - Unauthorized	Empty

**HTTP method****GET****Description**

read the specified ConsoleCLIDownload

**Table 2.9. HTTP responses**

HTTP code	Reponse body
200 - OK	<b>ConsoleCLIDownload</b> schema
401 - Unauthorized	Empty

**HTTP method****PATCH****Description**

partially update the specified ConsoleCLIDownload

**Table 2.10. Query parameters**

Parameter	Type	Description
<b>dryRun</b>	<b>string</b>	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
<b>fieldValidation</b>	<b>string</b>	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.

Table 2.11. HTTP responses

HTTP code	Response body
200 - OK	<a href="#">ConsoleCLIDownload</a> schema
401 - Unauthorized	Empty

**HTTP method****PUT****Description**

replace the specified ConsoleCLIDownload

Table 2.12. Query parameters

Parameter	Type	Description
-----------	------	-------------



Parameter	Type	Description
<b>dryRun</b>	<b>string</b>	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
<b>fieldValidation</b>	<b>string</b>	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.

Table 2.13. Body parameters

Parameter	Type	Description
<b>body</b>	<b>ConsoleCLIDownload schema</b>	

Table 2.14. HTTP responses

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

### 2.2.3. /apis/console.openshift.io/v1/consoleclidownloads/{name}/status

Table 2.15. Global path parameters

Parameter	Type	Description
<b>name</b>	<b>string</b>	name of the ConsoleCLIDownload

**HTTP method****GET****Description**

read status of the specified ConsoleCLIDownload

**Table 2.16. HTTP responses**

HTTP code	Reponse body
200 - OK	<a href="#">ConsoleCLIDownload</a> schema
401 - Unauthorized	Empty

**HTTP method****PATCH****Description**

partially update status of the specified ConsoleCLIDownload

**Table 2.17. Query parameters**

Parameter	Type	Description
<b>dryRun</b>	<b>string</b>	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
<b>fieldValidation</b>	<b>string</b>	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.

Table 2.18. HTTP responses

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

**HTTP method****PUT****Description**

replace status of the specified ConsoleCLIDownload

Table 2.19. Query parameters

Parameter	Type	Description
<b>dryRun</b>	<b>string</b>	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
<b>fieldValidation</b>	<b>string</b>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: <ul style="list-style-type: none"> <li>- Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23.</li> <li>- Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+</li> <li>- Strict: This will fail the request with a 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.</li> </ul>

Table 2.20. Body parameters

Parameter	Type	Description
<b>body</b>	<b>ConsoleCLIDownload</b> schema	

Table 2.21. HTTP responses

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

## CHAPTER 3. CONSOLEEXTERNALLOGLINK [CONSOLE.OPENSIFT.IO/V1]

### Description

ConsoleExternalLogLink is an extension for customizing OpenShift web console log links.  
Compatibility level 2: Stable within a major release for a minimum of 9 months or 3 minor releases (whichever is longer).

### Type

**object**

### Required

- **spec**

## 3.1. SPECIFICATION

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

Property	Type	Description
<b>spec</b>	<b>object</b>	ConsoleExternalLogLinkSpec is the desired log link configuration. The log link will appear on the logs tab of the pod details page.

### 3.1.1. .spec

#### Description

ConsoleExternalLogLinkSpec is the desired log link configuration. The log link will appear on the logs tab of the pod details page.

#### Type

**object**

#### Required

- **hrefTemplate**
- **text**

Property	Type	Description
----------	------	-------------

Property	Type	Description
<b>hrefTemplate</b>	<b>string</b>	<p>hrefTemplate is an absolute secure URL (must use https) for the log link including variables to be replaced. Variables are specified in the URL with the format <code>\${variableName}</code>, for instance, <code>\${containerName}</code> and will be replaced with the corresponding values from the resource. Resource is a pod. Supported variables are: - <code>\${resourceName}</code> - name of the resource which contains the logs - <code>\${resourceUID}</code> - UID of the resource which contains the logs - e.g. <b>11111111-2222-3333-4444-555555555555</b> - <code>\${containerName}</code> - name of the resource's container that contains the logs - <code>\${resourceNamespace}</code> - namespace of the resource that contains the logs - <code>\${resourceNamespaceUID}</code> - namespace UID of the resource that contains the logs - <code>\${podLabels}</code> - JSON representation of labels matching the pod with the logs - e.g. <b>{"key1":"value1","key2":"value2"}</b></p> <p>e.g., <a href="https://example.com/logs?resourceName=\${resourceName}&amp;containerName=\${containerName}&amp;resourceNamespace=\${resourceNamespace}&amp;podLabels=\${podLabels}">https://example.com/logs?resourceName=\${resourceName}&amp;containerName=\${containerName}&amp;resourceNamespace=\${resourceNamespace}&amp;podLabels=\${podLabels}</a></p>
<b>namespaceFilter</b>	<b>string</b>	<p>namespaceFilter is a regular expression used to restrict a log link to a matching set of namespaces (e.g., <b>^openshift-</b>). The string is converted into a regular expression using the JavaScript RegExp constructor. If not specified, links will be displayed for all the namespaces.</p>
<b>text</b>	<b>string</b>	text is the display text for the link

### 3.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/console.openshift.io/v1/consoleexternalloglinks**
  - **DELETE:** delete collection of ConsoleExternalLogLink
  - **GET:** list objects of kind ConsoleExternalLogLink
  - **POST:** create a ConsoleExternalLogLink
- **/apis/console.openshift.io/v1/consoleexternalloglinks/{name}**
  - **DELETE:** delete a ConsoleExternalLogLink
  - **GET:** read the specified ConsoleExternalLogLink
  - **PATCH:** partially update the specified ConsoleExternalLogLink
  - **PUT:** replace the specified ConsoleExternalLogLink
- **/apis/console.openshift.io/v1/consoleexternalloglinks/{name}/status**
  - **GET:** read status of the specified ConsoleExternalLogLink
  - **PATCH:** partially update status of the specified ConsoleExternalLogLink
  - **PUT:** replace status of the specified ConsoleExternalLogLink

#### 3.2.1. /apis/console.openshift.io/v1/consoleexternalloglinks

HTTP method

**DELETE**

Description

delete collection of ConsoleExternalLogLink

Table 3.1. HTTP responses

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

HTTP method

**GET**

Description

list objects of kind ConsoleExternalLogLink

Table 3.2. HTTP responses



HTTP code	Reponse body
200 - OK	<a href="#">ConsoleExternalLogLinkList</a> schema
401 - Unauthorized	Empty

**HTTP method****POST****Description**

create a ConsoleExternalLogLink

**Table 3.3. Query parameters**

Parameter	Type	Description
<b>dryRun</b>	<b>string</b>	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
<b>fieldValidation</b>	<b>string</b>	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.

**Table 3.4. Body parameters**

Parameter	Type	Description
<b>body</b>	<a href="#">ConsoleExternalLogLink</a> schema	

**Table 3.5. HTTP responses**

HTTP code	Reponse body
200 - OK	<a href="#">ConsoleExternalLogLink</a> schema
201 - Created	<a href="#">ConsoleExternalLogLink</a> schema
202 - Accepted	<a href="#">ConsoleExternalLogLink</a> schema
401 - Unauthorized	Empty

### 3.2.2. /apis/console.openshift.io/v1/consoleexternalloglinks/{name}

Table 3.6. Global path parameters

Parameter	Type	Description
<b>name</b>	<b>string</b>	name of the ConsoleExternalLogLink

#### HTTP method

#### DELETE

#### Description

delete a ConsoleExternalLogLink

Table 3.7. Query parameters

Parameter	Type	Description
<b>dryRun</b>	<b>string</b>	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

Table 3.8. HTTP responses

HTTP code	Reponse body
200 - OK	<a href="#">Status</a> schema
202 - Accepted	<a href="#">Status</a> schema
401 - Unauthorized	Empty

#### HTTP method

#### GET

#### Description

read the specified ConsoleExternalLogLink

**Table 3.9. HTTP responses**

HTTP code	Response body
200 - OK	<a href="#">ConsoleExternalLogLink</a> schema
401 - Unauthorized	Empty

## HTTP method

### PATCH

## Description

partially update the specified ConsoleExternalLogLink

**Table 3.10. Query parameters**

Parameter	Type	Description
<b>dryRun</b>	<b>string</b>	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
<b>fieldValidation</b>	<b>string</b>	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.

**Table 3.11. HTTP responses**

HTTP code	Response body
200 - OK	<a href="#">ConsoleExternalLogLink</a> schema

HTTP code	Response body
401 - Unauthorized	Empty

**HTTP method****PUT****Description**

replace the specified ConsoleExternalLogLink

**Table 3.12. Query parameters**

Parameter	Type	Description
<b>dryRun</b>	<b>string</b>	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
<b>fieldValidation</b>	<b>string</b>	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.

**Table 3.13. Body parameters**

Parameter	Type	Description
<b>body</b>	<a href="#">ConsoleExternalLog Link</a> schema	

**Table 3.14. HTTP responses**

HTTP code	Reponse body
200 - OK	<a href="#">ConsoleExternalLogLink</a> schema
201 - Created	<a href="#">ConsoleExternalLogLink</a> schema
401 - Unauthorized	Empty

### 3.2.3. /apis/console.openshift.io/v1/consoleexternalloglinks/{name}/status

Table 3.15. Global path parameters

Parameter	Type	Description
<b>name</b>	<b>string</b>	name of the ConsoleExternalLogLink

#### HTTP method

#### GET

#### Description

read status of the specified ConsoleExternalLogLink

Table 3.16. HTTP responses

HTTP code	Reponse body
200 - OK	<a href="#">ConsoleExternalLogLink</a> schema
401 - Unauthorized	Empty

#### HTTP method

#### PATCH

#### Description

partially update status of the specified ConsoleExternalLogLink

Table 3.17. Query parameters

Parameter	Type	Description
<b>dryRun</b>	<b>string</b>	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
<b>fieldValidation</b>	<b>string</b>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: <ul style="list-style-type: none"> <li>- Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23.</li> <li>- Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+</li> <li>- Strict: This will fail the request with a 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.</li> </ul>

Table 3.18. HTTP responses

HTTP code	Response body
200 - OK	<a href="#">ConsoleExternalLogLink</a> schema
401 - Unauthorized	Empty

**HTTP method****PUT****Description**

replace status of the specified ConsoleExternalLogLink

Table 3.19. Query parameters

Parameter	Type	Description
<b>dryRun</b>	<b>string</b>	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: <ul style="list-style-type: none"> <li>- All: all dry run stages will be processed</li> </ul>

Parameter	Type	Description
<b>fieldValidation</b>	<b>string</b>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: <ul style="list-style-type: none"> <li>- Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23.</li> <li>- Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+</li> <li>- Strict: This will fail the request with a 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.</li> </ul>

Table 3.20. Body parameters

Parameter	Type	Description
<b>body</b>	<a href="#">ConsoleExternalLogLink</a> schema	

Table 3.21. HTTP responses

HTTP code	Response body
200 - OK	<a href="#">ConsoleExternalLogLink</a> schema
201 - Created	<a href="#">ConsoleExternalLogLink</a> schema
401 - Unauthorized	Empty

## CHAPTER 4. CONSOLELINK [CONSOLE.OPENSIFT.IO/V1]

### Description

ConsoleLink is an extension for customizing OpenShift web console links.

Compatibility level 2: Stable within a major release for a minimum of 9 months or 3 minor releases (whichever is longer).

### Type

**object**

### Required

- **spec**

## 4.1. SPECIFICATION

Property	Type	Description
<b>apiVersion</b>	<b>string</b>	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources</a>
<b>kind</b>	<b>string</b>	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds</a>
<b>metadata</b>	<b>ObjectMeta</b>	Standard object's metadata. More info: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata</a>
<b>spec</b>	<b>object</b>	ConsoleLinkSpec is the desired console link configuration.



### 4.1.1. .spec

#### Description

ConsoleLinkSpec is the desired console link configuration.

#### Type

**object**

#### Required

- **href**
- **location**
- **text**

Property	Type	Description
<b>applicationMenu</b>	<b>object</b>	applicationMenu holds information about section and icon used for the link in the application menu, and it is applicable only when location is set to ApplicationMenu.
<b>href</b>	<b>string</b>	href is the absolute secure URL for the link (must use https)
<b>location</b>	<b>string</b>	location determines which location in the console the link will be appended to (ApplicationMenu, HelpMenu, UserMenu, NamespaceDashboard).
<b>namespaceDashboard</b>	<b>object</b>	namespaceDashboard holds information about namespaces in which the dashboard link should appear, and it is applicable only when location is set to NamespaceDashboard. If not specified, the link will appear in all namespaces.
<b>text</b>	<b>string</b>	text is the display text for the link

### 4.1.2. .spec.applicationMenu

#### Description

applicationMenu holds information about section and icon used for the link in the application menu, and it is applicable only when location is set to ApplicationMenu.

#### Type

**object****Required**

- **section**

Property	Type	Description
<b>imageUrl</b>	<b>string</b>	imageUrl is the URL for the icon used in front of the link in the application menu. The URL must be an HTTPS URL or a Data URI. The image should be square and will be shown at 24x24 pixels.
<b>section</b>	<b>string</b>	section is the section of the application menu in which the link should appear. This can be any text that will appear as a subheading in the application menu dropdown. A new section will be created if the text does not match text of an existing section.

### 4.1.3. .spec.namespaceDashboard

**Description**

namespaceDashboard holds information about namespaces in which the dashboard link should appear, and it is applicable only when location is set to NamespaceDashboard. If not specified, the link will appear in all namespaces.

**Type****object**

Property	Type	Description
<b>namespaceSelector</b>	<b>object</b>	namespaceSelector is used to select the Namespaces that should contain dashboard link by label. If the namespace labels match, dashboard link will be shown for the namespaces.
<b>namespaces</b>	<b>array (string)</b>	namespaces is an array of namespace names in which the dashboard link should appear.

### 4.1.4. .spec.namespaceDashboard.namespaceSelector

**Description**

namespaceSelector is used to select the Namespaces that should contain dashboard link by label. If the namespace labels match, dashboard link will be shown for the namespaces.

#### Type

**object**

Property	Type	Description
<b>matchExpressions</b>	<b>array</b>	matchExpressions is a list of label selector requirements. The requirements are ANDed.
<b>matchExpressions[]</b>	<b>object</b>	A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.
<b>matchLabels</b>	<b>object (string)</b>	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.

#### 4.1.5. .spec.namespaceDashboard.namespaceSelector.matchExpressions

##### Description

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

##### Type

**array**

#### 4.1.6. .spec.namespaceDashboard.namespaceSelector.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
<b>key</b>	<b>string</b>	key is the label key that the selector applies to.
<b>operator</b>	<b>string</b>	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
<b>values</b>	<b>array (string)</b>	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.

## 4.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/console.openshift.io/v1/consolelinks**
  - **DELETE**: delete collection of ConsoleLink
  - **GET**: list objects of kind ConsoleLink
  - **POST**: create a ConsoleLink
- **/apis/console.openshift.io/v1/consolelinks/{name}**
  - **DELETE**: delete a ConsoleLink
  - **GET**: read the specified ConsoleLink
  - **PATCH**: partially update the specified ConsoleLink
  - **PUT**: replace the specified ConsoleLink
- **/apis/console.openshift.io/v1/consolelinks/{name}/status**
  - **GET**: read status of the specified ConsoleLink
  - **PATCH**: partially update status of the specified ConsoleLink
  - **PUT**: replace status of the specified ConsoleLink

### 4.2.1. /apis/console.openshift.io/v1/consolelinks

HTTP method

**DELETE****Description**

delete collection of ConsoleLink

**Table 4.1. HTTP responses**

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

**HTTP method****GET****Description**

list objects of kind ConsoleLink

**Table 4.2. HTTP responses**

HTTP code	Reponse body
200 - OK	<b>ConsoleLinkList</b> schema
401 - Unauthorized	Empty

**HTTP method****POST****Description**

create a ConsoleLink

**Table 4.3. Query parameters**

Parameter	Type	Description
<b>dryRun</b>	<b>string</b>	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
<b>fieldValidation</b>	<b>string</b>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: <ul style="list-style-type: none"> <li>- Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23.</li> <li>- Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+</li> <li>- Strict: This will fail the request with a 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.</li> </ul>

Table 4.4. Body parameters

Parameter	Type	Description
<b>body</b>	<a href="#">ConsoleLink</a> schema	

Table 4.5. HTTP responses

HTTP code	Response body
200 - OK	<a href="#">ConsoleLink</a> schema
201 - Created	<a href="#">ConsoleLink</a> schema
202 - Accepted	<a href="#">ConsoleLink</a> schema
401 - Unauthorized	Empty

#### 4.2.2. /apis/console.openshift.io/v1/consolelinks/{name}

Table 4.6. Global path parameters

Parameter	Type	Description
<b>name</b>	<b>string</b>	name of the ConsoleLink

**HTTP method****DELETE****Description**

delete a ConsoleLink

**Table 4.7. Query parameters**

Parameter	Type	Description
<b>dryRun</b>	<b>string</b>	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

**Table 4.8. HTTP responses**

HTTP code	Reponse body
200 - OK	<b>Status</b> schema
202 - Accepted	<b>Status</b> schema
401 - Unauthorized	Empty

**HTTP method****GET****Description**

read the specified ConsoleLink

**Table 4.9. HTTP responses**

HTTP code	Reponse body
200 - OK	<b>ConsoleLink</b> schema
401 - Unauthorized	Empty

**HTTP method****PATCH****Description**

partially update the specified ConsoleLink

**Table 4.10. Query parameters**

Parameter	Type	Description
<b>dryRun</b>	<b>string</b>	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
<b>fieldValidation</b>	<b>string</b>	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.

Table 4.11. HTTP responses

HTTP code	Response body
200 - OK	<a href="#">ConsoleLink</a> schema
401 - Unauthorized	Empty

**HTTP method****PUT****Description**

replace the specified ConsoleLink

Table 4.12. Query parameters

Parameter	Type	Description
-----------	------	-------------



Parameter	Type	Description
<b>dryRun</b>	<b>string</b>	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
<b>fieldValidation</b>	<b>string</b>	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.

Table 4.13. Body parameters

Parameter	Type	Description
<b>body</b>	<b>ConsoleLink</b> schema	

Table 4.14. HTTP responses

HTTP code	Reponse body
200 - OK	<b>ConsoleLink</b> schema
201 - Created	<b>ConsoleLink</b> schema
401 - Unauthorized	Empty

### 4.2.3. /apis/console.openshift.io/v1/consolelinks/{name}/status

Table 4.15. Global path parameters

Parameter	Type	Description
<b>name</b>	<b>string</b>	name of the ConsoleLink

**HTTP method****GET****Description**

read status of the specified ConsoleLink

**Table 4.16. HTTP responses**

HTTP code	Reponse body
200 - OK	<a href="#">ConsoleLink</a> schema
401 - Unauthorized	Empty

**HTTP method****PATCH****Description**

partially update status of the specified ConsoleLink

**Table 4.17. Query parameters**

Parameter	Type	Description
<b>dryRun</b>	<b>string</b>	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
<b>fieldValidation</b>	<b>string</b>	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.

Table 4.18. HTTP responses

HTTP code	Response body
200 - OK	<a href="#">ConsoleLink</a> schema
401 - Unauthorized	Empty

**HTTP method****PUT****Description**

replace status of the specified ConsoleLink

Table 4.19. Query parameters

Parameter	Type	Description
<b>dryRun</b>	<b>string</b>	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
<b>fieldValidation</b>	<b>string</b>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: <ul style="list-style-type: none"> <li>- Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23.</li> <li>- Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+</li> <li>- Strict: This will fail the request with a 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.</li> </ul>

Table 4.20. Body parameters

Parameter	Type	Description
<b>body</b>	<b>ConsoleLink</b> schema	

Table 4.21. HTTP responses

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

## CHAPTER 5. CONSOLENOTIFICATION [CONSOLE.OPENSIFT.IO/V1]

### Description

ConsoleNotification is the extension for configuring openshift web console notifications.  
Compatibility level 2: Stable within a major release for a minimum of 9 months or 3 minor releases (whichever is longer).

### Type

**object**

### Required

- **spec**

## 5.1. SPECIFICATION

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

Property	Type	Description
<b>spec</b>	<b>object</b>	ConsoleNotificationSpec is the desired console notification configuration.

### 5.1.1. .spec

#### Description

ConsoleNotificationSpec is the desired console notification configuration.

#### Type

**object**

#### Required

- **text**

Property	Type	Description
<b>backgroundColor</b>	<b>string</b>	backgroundColor is the color of the background for the notification as CSS data type color.
<b>color</b>	<b>string</b>	color is the color of the text for the notification as CSS data type color.
<b>link</b>	<b>object</b>	link is an object that holds notification link details.
<b>location</b>	<b>string</b>	location is the location of the notification in the console. Valid values are: "BannerTop", "BannerBottom", "BannerTopBottom".
<b>text</b>	<b>string</b>	text is the visible text of the notification.

### 5.1.2. .spec.link

#### Description

link is an object that holds notification link details.

#### Type

**object**

#### Required

- **href**

- **text**

Property	Type	Description
<b>href</b>	<b>string</b>	href is the absolute secure URL for the link (must use https)
<b>text</b>	<b>string</b>	text is the display text for the link

## 5.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/console.openshift.io/v1/consolenotifications**
  - **DELETE**: delete collection of ConsoleNotification
  - **GET**: list objects of kind ConsoleNotification
  - **POST**: create a ConsoleNotification
- **/apis/console.openshift.io/v1/consolenotifications/{name}**
  - **DELETE**: delete a ConsoleNotification
  - **GET**: read the specified ConsoleNotification
  - **PATCH**: partially update the specified ConsoleNotification
  - **PUT**: replace the specified ConsoleNotification
- **/apis/console.openshift.io/v1/consolenotifications/{name}/status**
  - **GET**: read status of the specified ConsoleNotification
  - **PATCH**: partially update status of the specified ConsoleNotification
  - **PUT**: replace status of the specified ConsoleNotification

### 5.2.1. /apis/console.openshift.io/v1/consolenotifications

HTTP method

**DELETE**

Description

delete collection of ConsoleNotification

Table 5.1. HTTP responses

HTTP code	Reponse body
200 - OK	<a href="#">Status</a> schema
401 - Unauthorized	Empty

**HTTP method****GET****Description**

list objects of kind ConsoleNotification

**Table 5.2. HTTP responses**

HTTP code	Reponse body
200 - OK	<a href="#">ConsoleNotificationList</a> schema
401 - Unauthorized	Empty

**HTTP method****POST****Description**

create a ConsoleNotification

**Table 5.3. Query parameters**

Parameter	Type	Description
<b>dryRun</b>	<b>string</b>	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
<b>fieldValidation</b>	<b>string</b>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: <ul style="list-style-type: none"> <li>- Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23.</li> <li>- Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+</li> <li>- Strict: This will fail the request with a 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.</li> </ul>

Table 5.4. Body parameters

Parameter	Type	Description
<b>body</b>	<b>ConsoleNotification</b> schema	

Table 5.5. HTTP responses

HTTP code	Reponse body
200 - OK	<b>ConsoleNotification</b> schema
201 - Created	<b>ConsoleNotification</b> schema
202 - Accepted	<b>ConsoleNotification</b> schema
401 - Unauthorized	Empty

### 5.2.2. /apis/console.openshift.io/v1/consolenotifications/{name}

Table 5.6. Global path parameters

Parameter	Type	Description
<b>name</b>	<b>string</b>	name of the ConsoleNotification

**HTTP method****DELETE****Description**

delete a ConsoleNotification

**Table 5.7. Query parameters**

Parameter	Type	Description
<b>dryRun</b>	<b>string</b>	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

**Table 5.8. HTTP responses**

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

**HTTP method****GET****Description**

read the specified ConsoleNotification

**Table 5.9. HTTP responses**

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

**HTTP method****PATCH****Description**

partially update the specified ConsoleNotification

Table 5.10. Query parameters

Parameter	Type	Description
<b>dryRun</b>	<b>string</b>	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
<b>fieldValidation</b>	<b>string</b>	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.

Table 5.11. HTTP responses

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

**HTTP method****PUT****Description**

replace the specified ConsoleNotification

Table 5.12. Query parameters

Parameter	Type	Description
-----------	------	-------------

Parameter	Type	Description
<b>dryRun</b>	<b>string</b>	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
<b>fieldValidation</b>	<b>string</b>	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.

Table 5.13. Body parameters

Parameter	Type	Description
<b>body</b>	<b>ConsoleNotification</b> schema	

Table 5.14. HTTP responses

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

### 5.2.3. /apis/console.openshift.io/v1/consolenotifications/{name}/status

Table 5.15. Global path parameters

Parameter	Type	Description
<b>name</b>	<b>string</b>	name of the ConsoleNotification

**HTTP method****GET****Description**

read status of the specified ConsoleNotification

**Table 5.16. HTTP responses**

HTTP code	Reponse body
200 - OK	<a href="#">ConsoleNotification</a> schema
401 - Unauthorized	Empty

**HTTP method****PATCH****Description**

partially update status of the specified ConsoleNotification

**Table 5.17. Query parameters**

Parameter	Type	Description
<b>dryRun</b>	<b>string</b>	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
<b>fieldValidation</b>	<b>string</b>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: <ul style="list-style-type: none"> <li>- Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23.</li> <li>- Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+</li> <li>- Strict: This will fail the request with a 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.</li> </ul>

Table 5.18. HTTP responses

HTTP code	Response body
200 - OK	<a href="#">ConsoleNotification</a> schema
401 - Unauthorized	Empty

**HTTP method****PUT****Description**

replace status of the specified ConsoleNotification

Table 5.19. Query parameters

Parameter	Type	Description
<b>dryRun</b>	<b>string</b>	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: <ul style="list-style-type: none"> <li>- All: all dry run stages will be processed</li> </ul>

Parameter	Type	Description
<b>fieldValidation</b>	<b>string</b>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: <ul style="list-style-type: none"> <li>- Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23.</li> <li>- Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+</li> <li>- Strict: This will fail the request with a 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.</li> </ul>

Table 5.20. Body parameters

Parameter	Type	Description
<b>body</b>	<b>ConsoleNotification</b> schema	

Table 5.21. HTTP responses

HTTP code	Reponse body
200 - OK	<b>ConsoleNotification</b> schema
201 - Created	<b>ConsoleNotification</b> schema
401 - Unauthorized	Empty

## CHAPTER 6. CONSOLEPLUGIN [CONSOLE.OPENSIFT.IO/V1]

### Description

ConsolePlugin is an extension for customizing OpenShift web console by dynamically loading code from another service running on the cluster.

Compatibility level 1: Stable within a major release for a minimum of 12 months or 3 minor releases (whichever is longer).

### Type

**object**

### Required

- **metadata**
- **spec**

## 6.1. SPECIFICATION

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



Property	Type	Description
<b>spec</b>	<b>object</b>	ConsolePluginSpec is the desired plugin configuration.

### 6.1.1. .spec

#### Description

ConsolePluginSpec is the desired plugin configuration.

#### Type

**object**

#### Required

- **backend**
- **displayName**

Property	Type	Description
<b>backend</b>	<b>object</b>	backend holds the configuration of backend which is serving console's plugin .
<b>displayName</b>	<b>string</b>	displayName is the display name of the plugin. The display name should be between 1 and 128 characters.
<b>i18n</b>	<b>object</b>	i18n is the configuration of plugin's localization resources.
<b>proxy</b>	<b>array</b>	proxy is a list of proxies that describe various service type to which the plugin needs to connect to.
<b>proxy[]</b>	<b>object</b>	ConsolePluginProxy holds information on various service types to which console's backend will proxy the plugin's requests.

### 6.1.2. .spec.backend

#### Description

backend holds the configuration of backend which is serving console's plugin .

#### Type

**object**

**Required**

- **type**

Property	Type	Description
<b>service</b>	<b>object</b>	service is a Kubernetes Service that exposes the plugin using a deployment with an HTTP server. The Service must use HTTPS and Service serving certificate. The console backend will proxy the plugins assets from the Service using the service CA bundle.
<b>type</b>	<b>string</b>	type is the backend type which servers the console's plugin. Currently only "Service" is supported.

**6.1.3. .spec.backend.service****Description**

service is a Kubernetes Service that exposes the plugin using a deployment with an HTTP server. The Service must use HTTPS and Service serving certificate. The console backend will proxy the plugins assets from the Service using the service CA bundle.

**Type****object****Required**

- **name**
- **namespace**
- **port**

Property	Type	Description
<b>basePath</b>	<b>string</b>	basePath is the path to the plugin's assets. The primary asset is the manifest file called <b>plugin-manifest.json</b> , which is a JSON document that contains metadata about the plugin and the extensions.
<b>name</b>	<b>string</b>	name of Service that is serving the plugin assets.

Property	Type	Description
<b>namespace</b>	<b>string</b>	namespace of Service that is serving the plugin assets.
<b>port</b>	<b>integer</b>	port on which the Service that is serving the plugin is listening to.

#### 6.1.4. .spec.i18n

##### Description

i18n is the configuration of plugin's localization resources.

##### Type

**object**

##### Required

- **loadType**

Property	Type	Description
<b>loadType</b>	<b>string</b>	loadType indicates how the plugin's localization resource should be loaded. Valid values are Preload, Lazy and the empty string. When set to Preload, all localization resources are fetched when the plugin is loaded. When set to Lazy, localization resources are lazily loaded as and when they are required by the console. When omitted or set to the empty string, the behaviour is equivalent to Lazy type.

#### 6.1.5. .spec.proxy

##### Description

proxy is a list of proxies that describe various service type to which the plugin needs to connect to.

##### Type

**array**

#### 6.1.6. .spec.proxy[]

##### Description

ConsolePluginProxy holds information on various service types to which console's backend will proxy the plugin's requests.

##### Type

**object****Required**

- **alias**
- **endpoint**

Property	Type	Description
<b>alias</b>	<b>string</b>	<p>alias is a proxy name that identifies the plugin's proxy. An alias name should be unique per plugin. The console backend exposes following proxy endpoint:</p> <pre>/api/proxy/plugin/&lt;plugin-name&gt;/&lt;proxy-alias&gt;/&lt;request-path&gt;?&lt;optional-query-parameters&gt;</pre> <p>Request example path:</p> <pre>/api/proxy/plugin/acm/search/pods?namespace=openshift-apiserver</pre>
<b>authorization</b>	<b>string</b>	authorization provides information about authorization type, which the proxied request should contain
<b>caCertificate</b>	<b>string</b>	caCertificate provides the cert authority certificate contents, in case the proxied Service is using custom service CA. By default, the service CA bundle provided by the service-ca operator is used.
<b>endpoint</b>	<b>object</b>	endpoint provides information about endpoint to which the request is proxied to.

**6.1.7. .spec.proxy[].endpoint****Description**

endpoint provides information about endpoint to which the request is proxied to.

**Type****object****Required**

- **type**

Property	Type	Description
<b>service</b>	<b>object</b>	service is an in-cluster Service that the plugin will connect to. The Service must use HTTPS. The console backend exposes an endpoint in order to proxy communication between the plugin and the Service. Note: service field is required for now, since currently only "Service" type is supported.
<b>type</b>	<b>string</b>	type is the type of the console plugin's proxy. Currently only "Service" is supported.

### 6.1.8. .spec.proxy[].endpoint.service

#### Description

service is an in-cluster Service that the plugin will connect to. The Service must use HTTPS. The console backend exposes an endpoint in order to proxy communication between the plugin and the Service. Note: service field is required for now, since currently only "Service" type is supported.

#### Type

**object**

#### Required

- **name**
- **namespace**
- **port**

Property	Type	Description
<b>name</b>	<b>string</b>	name of Service that the plugin needs to connect to.
<b>namespace</b>	<b>string</b>	namespace of Service that the plugin needs to connect to
<b>port</b>	<b>integer</b>	port on which the Service that the plugin needs to connect to is listening on.

## 6.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/console.openshift.io/v1/consoleplugins**
  - **DELETE**: delete collection of ConsolePlugin
  - **GET**: list objects of kind ConsolePlugin
  - **POST**: create a ConsolePlugin
- **/apis/console.openshift.io/v1/consoleplugins/{name}**
  - **DELETE**: delete a ConsolePlugin
  - **GET**: read the specified ConsolePlugin
  - **PATCH**: partially update the specified ConsolePlugin
  - **PUT**: replace the specified ConsolePlugin

### 6.2.1. /apis/console.openshift.io/v1/consoleplugins

#### HTTP method

#### **DELETE**

#### Description

delete collection of ConsolePlugin

Table 6.1. HTTP responses

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

#### HTTP method

#### **GET**

#### Description

list objects of kind ConsolePlugin

Table 6.2. HTTP responses

HTTP code	Reponse body
200 - OK	<b>ConsolePluginList</b> schema
401 - Unauthorized	Empty

#### HTTP method

#### **POST**

#### Description

create a ConsolePlugin

Table 6.3. Query parameters

Parameter	Type	Description
<b>dryRun</b>	<b>string</b>	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
<b>fieldValidation</b>	<b>string</b>	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.

Table 6.4. Body parameters

Parameter	Type	Description
<b>body</b>	<b>ConsolePlugin</b> schema	

Table 6.5. HTTP responses

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

### 6.2.2. /apis/console.openshift.io/v1/consoleplugins/{name}

Table 6.6. Global path parameters

Parameter	Type	Description
<b>name</b>	<b>string</b>	name of the ConsolePlugin

## HTTP method

**DELETE**

## Description

delete a ConsolePlugin

Table 6.7. Query parameters

Parameter	Type	Description
<b>dryRun</b>	<b>string</b>	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

Table 6.8. HTTP responses

HTTP code	Reponse body
200 - OK	<a href="#">Status</a> schema
202 - Accepted	<a href="#">Status</a> schema
401 - Unauthorized	Empty

## HTTP method

**GET**

## Description

read the specified ConsolePlugin

Table 6.9. HTTP responses

HTTP code	Reponse body
200 - OK	<a href="#">ConsolePlugin</a> schema
401 - Unauthorized	Empty

## HTTP method

**PATCH**



**Description**

partially update the specified ConsolePlugin

**Table 6.10. Query parameters**

Parameter	Type	Description
<b>dryRun</b>	<b>string</b>	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
<b>fieldValidation</b>	<b>string</b>	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.

**Table 6.11. HTTP responses**

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

**HTTP method****PUT****Description**

replace the specified ConsolePlugin

**Table 6.12. Query parameters**

Parameter	Type	Description
<b>dryRun</b>	<b>string</b>	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
<b>fieldValidation</b>	<b>string</b>	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.

Table 6.13. Body parameters

Parameter	Type	Description
<b>body</b>	<b>ConsolePlugin</b> schema	

Table 6.14. HTTP responses

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

## CHAPTER 7. CONSOLEQUICKSTART [CONSOLE.OPENSIFT.IO/V1]

### Description

ConsoleQuickStart is an extension for guiding user through various workflows in the OpenShift web console.

Compatibility level 2: Stable within a major release for a minimum of 9 months or 3 minor releases (whichever is longer).

### Type

**object**

### Required

- **spec**

## 7.1. SPECIFICATION

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

Property	Type	Description
<b>spec</b>	<b>object</b>	ConsoleQuickStartSpec is the desired quick start configuration.

### 7.1.1. .spec

#### Description

ConsoleQuickStartSpec is the desired quick start configuration.

#### Type

**object**

#### Required

- **description**
- **displayName**
- **durationMinutes**
- **introduction**
- **tasks**

Property	Type	Description
<b>accessReviewResources</b>	<b>array</b>	accessReviewResources contains a list of resources that the user's access will be reviewed against in order for the user to complete the Quick Start. The Quick Start will be hidden if any of the access reviews fail.
<b>accessReviewResources[]</b>	<b>object</b>	ResourceAttributes includes the authorization attributes available for resource requests to the Authorizer interface
<b>conclusion</b>	<b>string</b>	conclusion sums up the Quick Start and suggests the possible next steps. (includes markdown)
<b>description</b>	<b>string</b>	description is the description of the Quick Start. (includes markdown)
<b>displayName</b>	<b>string</b>	displayName is the display name of the Quick Start.

Property	Type	Description
<b>durationMinutes</b>	<b>integer</b>	durationMinutes describes approximately how many minutes it will take to complete the Quick Start.
<b>icon</b>	<b>string</b>	icon is a base64 encoded image that will be displayed beside the Quick Start display name. The icon should be an vector image for easy scaling. The size of the icon should be 40x40.
<b>introduction</b>	<b>string</b>	introduction describes the purpose of the Quick Start. (includes markdown)
<b>nextQuickStart</b>	<b>array (string)</b>	nextQuickStart is a list of the following Quick Starts, suggested for the user to try.
<b>prerequisites</b>	<b>array (string)</b>	prerequisites contains all prerequisites that need to be met before taking a Quick Start. (includes markdown)
<b>tags</b>	<b>array (string)</b>	tags is a list of strings that describe the Quick Start.
<b>tasks</b>	<b>array</b>	tasks is the list of steps the user has to perform to complete the Quick Start.
<b>tasks[]</b>	<b>object</b>	ConsoleQuickStartTask is a single step in a Quick Start.

### 7.1.2. .spec.accessReviewResources

#### Description

accessReviewResources contains a list of resources that the user's access will be reviewed against in order for the user to complete the Quick Start. The Quick Start will be hidden if any of the access reviews fail.

#### Type

**array**

### 7.1.3. .spec.accessReviewResources[]

#### Description

ResourceAttributes includes the authorization attributes available for resource requests to the Authorizer interface

## Type

**object**

Property	Type	Description
<b>fieldSelector</b>	<b>object</b>	<p>fieldSelector describes the limitation on access based on field. It can only limit access, not broaden it.</p> <p>This field is alpha-level. To use this field, you must enable the <b>AuthorizeWithSelectors</b> feature gate (disabled by default).</p>
<b>group</b>	<b>string</b>	Group is the API Group of the Resource. "*" means all.
<b>labelSelector</b>	<b>object</b>	<p>labelSelector describes the limitation on access based on labels. It can only limit access, not broaden it.</p> <p>This field is alpha-level. To use this field, you must enable the <b>AuthorizeWithSelectors</b> feature gate (disabled by default).</p>
<b>name</b>	<b>string</b>	Name is the name of the resource being requested for a "get" or deleted for a "delete". "" (empty) means all.
<b>namespace</b>	<b>string</b>	<p>Namespace is the namespace of the action being requested. Currently, there is no distinction between no namespace and all namespaces "" (empty) is defaulted for LocalSubjectAccessReviews "" (empty) is empty for cluster-scoped resources "" (empty) means "all" for namespace scoped resources from a SubjectAccessReview or SelfSubjectAccessReview</p>

Property	Type	Description
<b>resource</b>	<b>string</b>	Resource is one of the existing resource types. "*" means all.
<b>subresource</b>	<b>string</b>	Subresource is one of the existing resource types. "" means none.
<b>verb</b>	<b>string</b>	Verb is a kubernetes resource API verb, like: get, list, watch, create, update, delete, proxy. "*" means all.
<b>version</b>	<b>string</b>	Version is the API Version of the Resource. "*" means all.

#### 7.1.4. .spec.accessReviewResources[].fieldSelector

##### Description

fieldSelector describes the limitation on access based on field. It can only limit access, not broaden it. This field is alpha-level. To use this field, you must enable the **AuthorizeWithSelectors** feature gate (disabled by default).

##### Type

**object**

Property	Type	Description
<b>rawSelector</b>	<b>string</b>	rawSelector is the serialization of a field selector that would be included in a query parameter. Webhook implementations are encouraged to ignore rawSelector. The kube-apiserver's *SubjectAccessReview will parse the rawSelector as long as the requirements are not present.

Property	Type	Description
<b>requirements</b>	<b>array</b>	requirements is the parsed interpretation of a field selector. All requirements must be met for a resource instance to match the selector. Webhook implementations should handle requirements, but how to handle them is up to the webhook. Since requirements can only limit the request, it is safe to authorize as unlimited request if the requirements are not understood.
<b>requirements[]</b>	<b>object</b>	FieldSelectorRequirement is a selector that contains values, a key, and an operator that relates the key and values.

### 7.1.5. .spec.accessReviewResources[].fieldSelector.requirements

#### Description

requirements is the parsed interpretation of a field selector. All requirements must be met for a resource instance to match the selector. Webhook implementations should handle requirements, but how to handle them is up to the webhook. Since requirements can only limit the request, it is safe to authorize as unlimited request if the requirements are not understood.

#### Type

**array**

### 7.1.6. .spec.accessReviewResources[].fieldSelector.requirements[]

#### Description

FieldSelectorRequirement 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
<b>key</b>	<b>string</b>	key is the field selector key that the requirement applies to.



Property	Type	Description
<b>operator</b>	<b>string</b>	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists, DoesNotExist. The list of operators may grow in the future.
<b>values</b>	<b>array (string)</b>	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.

### 7.1.7. .spec.accessReviewResources[].labelSelector

#### Description

labelSelector describes the limitation on access based on labels. It can only limit access, not broaden it.

This field is alpha-level. To use this field, you must enable the **AuthorizeWithSelectors** feature gate (disabled by default).

#### Type

**object**

Property	Type	Description
<b>rawSelector</b>	<b>string</b>	rawSelector is the serialization of a field selector that would be included in a query parameter. Webhook implementations are encouraged to ignore rawSelector. The kube-apiserver's *SubjectAccessReview will parse the rawSelector as long as the requirements are not present.

Property	Type	Description
<b>requirements</b>	<b>array</b>	requirements is the parsed interpretation of a label selector. All requirements must be met for a resource instance to match the selector. Webhook implementations should handle requirements, but how to handle them is up to the webhook. Since requirements can only limit the request, it is safe to authorize as unlimited request if the requirements are not understood.
<b>requirements[]</b>	<b>object</b>	A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

### 7.1.8. .spec.accessReviewResources[].labelSelector.requirements

#### Description

requirements is the parsed interpretation of a label selector. All requirements must be met for a resource instance to match the selector. Webhook implementations should handle requirements, but how to handle them is up to the webhook. Since requirements can only limit the request, it is safe to authorize as unlimited request if the requirements are not understood.

#### Type

**array**

### 7.1.9. .spec.accessReviewResources[].labelSelector.requirements[]

#### 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
----------	------	-------------

Property	Type	Description
<b>key</b>	<b>string</b>	key is the label key that the selector applies to.
<b>operator</b>	<b>string</b>	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
<b>values</b>	<b>array (string)</b>	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.

### 7.1.10. .spec.tasks

#### Description

tasks is the list of steps the user has to perform to complete the Quick Start.

#### Type

**array**

### 7.1.11. .spec.tasks[]

#### Description

ConsoleQuickStartTask is a single step in a Quick Start.

#### Type

**object**

#### Required

- **description**
- **title**

Property	Type	Description
<b>description</b>	<b>string</b>	description describes the steps needed to complete the task. (includes markdown)

Property	Type	Description
<b>review</b>	<b>object</b>	review contains instructions to validate the task is complete. The user will select 'Yes' or 'No'. using a radio button, which indicates whether the step was completed successfully.
<b>summary</b>	<b>object</b>	summary contains information about the passed step.
<b>title</b>	<b>string</b>	title describes the task and is displayed as a step heading.

### 7.1.12. `.spec.tasks[].review`

#### Description

review contains instructions to validate the task is complete. The user will select 'Yes' or 'No'. using a radio button, which indicates whether the step was completed successfully.

#### Type

**object**

#### Required

- **failedTaskHelp**
- **instructions**

Property	Type	Description
<b>failedTaskHelp</b>	<b>string</b>	failedTaskHelp contains suggestions for a failed task review and is shown at the end of task. (includes markdown)
<b>instructions</b>	<b>string</b>	instructions contains steps that user needs to take in order to validate his work after going through a task. (includes markdown)

### 7.1.13. `.spec.tasks[].summary`

#### Description

summary contains information about the passed step.

#### Type

**object****Required**

- **failed**
- **success**

Property	Type	Description
<b>failed</b>	<b>string</b>	failed briefly describes the unsuccessfully passed task. (includes markdown)
<b>success</b>	<b>string</b>	success describes the succesfully passed task.

## 7.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/console.openshift.io/v1/consolequickstarts**
  - **DELETE**: delete collection of ConsoleQuickStart
  - **GET**: list objects of kind ConsoleQuickStart
  - **POST**: create a ConsoleQuickStart
- **/apis/console.openshift.io/v1/consolequickstarts/{name}**
  - **DELETE**: delete a ConsoleQuickStart
  - **GET**: read the specified ConsoleQuickStart
  - **PATCH**: partially update the specified ConsoleQuickStart
  - **PUT**: replace the specified ConsoleQuickStart

### 7.2.1. /apis/console.openshift.io/v1/consolequickstarts

**HTTP method****DELETE****Description**

delete collection of ConsoleQuickStart

Table 7.1. HTTP responses

HTTP code	Reponse body
200 - OK	<b>Status</b> schema

HTTP code	Reponse body
401 - Unauthorized	Empty

**HTTP method****GET****Description**

list objects of kind ConsoleQuickStart

**Table 7.2. HTTP responses**

HTTP code	Reponse body
200 - OK	<a href="#">ConsoleQuickStartList</a> schema
401 - Unauthorized	Empty

**HTTP method****POST****Description**

create a ConsoleQuickStart

**Table 7.3. Query parameters**

Parameter	Type	Description
<b>dryRun</b>	<b>string</b>	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
<b>fieldValidation</b>	<b>string</b>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: <ul style="list-style-type: none"> <li>- Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23.</li> <li>- Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+</li> <li>- Strict: This will fail the request with a 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.</li> </ul>

Table 7.4. Body parameters

Parameter	Type	Description
<b>body</b>	<b>ConsoleQuickStart</b> schema	

Table 7.5. HTTP responses

HTTP code	Reponse body
200 - OK	<b>ConsoleQuickStart</b> schema
201 - Created	<b>ConsoleQuickStart</b> schema
202 - Accepted	<b>ConsoleQuickStart</b> schema
401 - Unauthorized	Empty

### 7.2.2. /apis/console.openshift.io/v1/consolequickstarts/{name}

Table 7.6. Global path parameters

Parameter	Type	Description
<b>name</b>	<b>string</b>	name of the ConsoleQuickStart

**HTTP method****DELETE****Description**

delete a ConsoleQuickStart

**Table 7.7. Query parameters**

Parameter	Type	Description
<b>dryRun</b>	<b>string</b>	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

**Table 7.8. HTTP responses**

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

**HTTP method****GET****Description**

read the specified ConsoleQuickStart

**Table 7.9. HTTP responses**

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

**HTTP method****PATCH****Description**

partially update the specified ConsoleQuickStart



Table 7.10. Query parameters

Parameter	Type	Description
<b>dryRun</b>	<b>string</b>	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
<b>fieldValidation</b>	<b>string</b>	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.

Table 7.11. HTTP responses

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

**HTTP method****PUT****Description**

replace the specified ConsoleQuickStart

Table 7.12. Query parameters

Parameter	Type	Description
-----------	------	-------------

Parameter	Type	Description
<b>dryRun</b>	<b>string</b>	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
<b>fieldValidation</b>	<b>string</b>	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.

Table 7.13. Body parameters

Parameter	Type	Description
<b>body</b>	<b>ConsoleQuickStart</b> schema	

Table 7.14. HTTP responses

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

## CHAPTER 8. CONSOLESAMPLE [CONSOLE.OPENSIFT.IO/V1]

### Description

ConsoleSample is an extension to customizing OpenShift web console by adding samples.  
Compatibility level 1: Stable within a major release for a minimum of 12 months or 3 minor releases (whichever is longer).

### Type

**object**

### Required

- **metadata**
- **spec**

## 8.1. SPECIFICATION

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

Property	Type	Description
<b>spec</b>	<b>object</b>	spec contains configuration for a console sample.

### 8.1.1. .spec

#### Description

spec contains configuration for a console sample.

#### Type

**object**

#### Required

- **abstract**
- **description**
- **source**
- **title**

Property	Type	Description
<b>abstract</b>	<b>string</b>	<p>abstract is a short introduction to the sample.</p> <p>It is required and must be no more than 100 characters in length.</p> <p>The abstract is shown on the sample card tile below the title and provider and is limited to three lines of content.</p>
<b>description</b>	<b>string</b>	<p>description is a long form explanation of the sample.</p> <p>It is required and can have a maximum length of <b>4096</b> characters.</p> <p>It is a README.md-like content for additional information, links, pre-conditions, and other instructions. It will be rendered as Markdown so that it can contain line breaks, links, and other simple formatting.</p>

Property	Type	Description
<b>icon</b>	<b>string</b>	<p>icon is an optional base64 encoded image and shown beside the sample title.</p> <p>The format must follow the data: URL format and can have a maximum size of <b>10 KB</b>.</p> <p>data:[&lt;mediatype&gt;][;base64], &lt;base64 encoded image&gt;</p> <p>For example:</p> <p>data:image;base64, plus the base64 encoded image.</p> <p>Vector images can also be used. SVG icons must start with:</p> <p>data:image/svg+xml;base64, plus the base64 encoded SVG image.</p> <p>All sample catalog icons will be shown on a white background (also when the dark theme is used). The web console ensures that different aspect ratios work correctly. Currently, the surface of the icon is at most 40x100px.</p> <p>For more information on the data URL format, please visit <a href="https://developer.mozilla.org/en-US/docs/Web/HTTP/Basics_of_HTTP/Data_URLs">https://developer.mozilla.org/en-US/docs/Web/HTTP/Basics_of_HTTP/Data_URLs</a>.</p>
<b>provider</b>	<b>string</b>	<p>provider is an optional label to honor who provides the sample.</p> <p>It is optional and must be no more than 50 characters in length.</p> <p>A provider can be a company like "Red Hat" or an organization like "CNCF" or "Knative".</p> <p>Currently, the provider is only shown on the sample card tile below the title with the prefix "Provided by "</p>

Property	Type	Description
<b>source</b>	<b>object</b>	source defines where to deploy the sample service from. The sample may be sourced from an external git repository or container image.
<b>tags</b>	<b>array (string)</b>	<p>tags are optional string values that can be used to find samples in the samples catalog.</p> <p>Examples of common tags may be "Java", "Quarkus", etc.</p> <p>They will be displayed on the samples details page.</p>
<b>title</b>	<b>string</b>	<p>title is the display name of the sample.</p> <p>It is required and must be no more than 50 characters in length.</p>
<b>type</b>	<b>string</b>	<p>type is an optional label to group multiple samples.</p> <p>It is optional and must be no more than 20 characters in length.</p> <p>Recommendation is a singular term like "Builder Image", "Devfile" or "Serverless Function".</p> <p>Currently, the type is shown a badge on the sample card tile in the top right corner.</p>

### 8.1.2. .spec.source

#### Description

source defines where to deploy the sample service from. The sample may be sourced from an external git repository or container image.

#### Type

**object**

#### Required

- **type**

Property	Type	Description
<b>containerImport</b>	<b>object</b>	containerImport allows the user import a container image.
<b>gitImport</b>	<b>object</b>	gitImport allows the user to import code from a git repository.
<b>type</b>	<b>string</b>	type of the sample, currently supported: "GitImport";"ContainerImport"

### 8.1.3. .spec.source.containerImport

#### Description

containerImport allows the user import a container image.

#### Type

**object**

#### Required

- **image**

Property	Type	Description
<b>image</b>	<b>string</b>	reference to a container image that provides a HTTP service. The service must be exposed on the default port (8080) unless otherwise configured with the port field.  Supported formats: - <repository-name>/<image-name> - docker.io/<repository-name>/<image-name> - quay.io/<repository-name>/<image-name> - quay.io/<repository-name>/<image-name>@sha256:<image hash> - quay.io/<repository-name>/<image-name>:<tag>
<b>service</b>	<b>object</b>	service contains configuration for the Service resource created for this sample.

### 8.1.4. .spec.source.containerImport.service

**Description**

service contains configuration for the Service resource created for this sample.

**Type**

**object**

Property	Type	Description
<b>targetPort</b>	<b>integer</b>	targetPort is the port that the service listens on for HTTP requests. This port will be used for Service and Route created for this sample. Port must be in the range 1 to 65535. Default port is 8080.

### 8.1.5. .spec.source.gitImport

**Description**

gitImport allows the user to import code from a git repository.

**Type**

**object**

**Required**

- **repository**

Property	Type	Description
<b>repository</b>	<b>object</b>	repository contains the reference to the actual Git repository.
<b>service</b>	<b>object</b>	service contains configuration for the Service resource created for this sample.

### 8.1.6. .spec.source.gitImport.repository

**Description**

repository contains the reference to the actual Git repository.

**Type**

**object**

**Required**

- **url**



Property	Type	Description
<b>contextDir</b>	<b>string</b>	contextDir is used to specify a directory within the repository to build the component. Must start with / and have a maximum length of 256 characters. When omitted, the default value is to build from the root of the repository.
<b>revision</b>	<b>string</b>	revision is the git revision at which to clone the git repository Can be used to clone a specific branch, tag or commit SHA. Must be at most 256 characters in length. When omitted the repository's default branch is used.
<b>url</b>	<b>string</b>	<p>url of the Git repository that contains a HTTP service. The HTTP service must be exposed on the default port (8080) unless otherwise configured with the port field.</p> <p>Only public repositories on GitHub, GitLab and Bitbucket are currently supported:</p> <ul style="list-style-type: none"> <li>- <a href="https://github.com/&lt;org&gt;/&lt;repository&gt;">https://github.com/&lt;org&gt;/&lt;repository&gt;</a> -</li> <li><a href="https://gitlab.com/&lt;org&gt;/&lt;repository&gt;">https://gitlab.com/&lt;org&gt;/&lt;repository&gt;</a> -</li> <li><a href="https://bitbucket.org/&lt;org&gt;/&lt;repository&gt;">https://bitbucket.org/&lt;org&gt;/&lt;repository&gt;</a></li> </ul> <p>The url must have a maximum length of 256 characters.</p>

### 8.1.7. .spec.source.gitImport.service

#### Description

service contains configuration for the Service resource created for this sample.

#### Type

**object**

Property	Type	Description
----------	------	-------------

Property	Type	Description
<b>targetPort</b>	<b>integer</b>	targetPort is the port that the service listens on for HTTP requests. This port will be used for Service created for this sample. Port must be in the range 1 to 65535. Default port is 8080.

## 8.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/console.openshift.io/v1/consolesamples**
  - **DELETE**: delete collection of ConsoleSample
  - **GET**: list objects of kind ConsoleSample
  - **POST**: create a ConsoleSample
- **/apis/console.openshift.io/v1/consolesamples/{name}**
  - **DELETE**: delete a ConsoleSample
  - **GET**: read the specified ConsoleSample
  - **PATCH**: partially update the specified ConsoleSample
  - **PUT**: replace the specified ConsoleSample

### 8.2.1. /apis/console.openshift.io/v1/consolesamples

HTTP method

**DELETE**

Description

delete collection of ConsoleSample

Table 8.1. HTTP responses

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

HTTP method

**GET**

Description

list objects of kind ConsoleSample

Table 8.2. HTTP responses

HTTP code	Response body
200 - OK	<a href="#">ConsoleSampleList</a> schema
401 - Unauthorized	Empty

## HTTP method

**POST**

## Description

create a ConsoleSample

Table 8.3. Query parameters

Parameter	Type	Description
<b>dryRun</b>	<b>string</b>	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
<b>fieldValidation</b>	<b>string</b>	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.

Table 8.4. Body parameters

Parameter	Type	Description
<b>body</b>	<a href="#">ConsoleSample</a> schema	

Table 8.5. HTTP responses

HTTP code	Reponse body
200 - OK	<a href="#">ConsoleSample</a> schema
201 - Created	<a href="#">ConsoleSample</a> schema
202 - Accepted	<a href="#">ConsoleSample</a> schema
401 - Unauthorized	Empty

### 8.2.2. /apis/console.openshift.io/v1/consolesamples/{name}

Table 8.6. Global path parameters

Parameter	Type	Description
<b>name</b>	<b>string</b>	name of the ConsoleSample

#### HTTP method

##### DELETE

#### Description

delete a ConsoleSample

Table 8.7. Query parameters

Parameter	Type	Description
<b>dryRun</b>	<b>string</b>	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

Table 8.8. HTTP responses

HTTP code	Reponse body
200 - OK	<a href="#">Status</a> schema
202 - Accepted	<a href="#">Status</a> schema
401 - Unauthorized	Empty

#### HTTP method

##### GET

**Description**

read the specified ConsoleSample

**Table 8.9. HTTP responses**

HTTP code	Response body
200 - OK	<a href="#">ConsoleSample</a> schema
401 - Unauthorized	Empty

**HTTP method****PATCH****Description**

partially update the specified ConsoleSample

**Table 8.10. Query parameters**

Parameter	Type	Description
<b>dryRun</b>	<b>string</b>	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
<b>fieldValidation</b>	<b>string</b>	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.

**Table 8.11. HTTP responses**

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

**HTTP method****PUT****Description**

replace the specified ConsoleSample

**Table 8.12. Query parameters**

Parameter	Type	Description
<b>dryRun</b>	<b>string</b>	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
<b>fieldValidation</b>	<b>string</b>	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.

**Table 8.13. Body parameters**

Parameter	Type	Description
<b>body</b>	<b>ConsoleSample</b> schema	

**Table 8.14. HTTP responses**

HTTP code	Reponse body
200 - OK	<b>ConsoleSample</b> schema
201 - Created	<b>ConsoleSample</b> schema
401 - Unauthorized	Empty

## CHAPTER 9. CONSOLEYAMLSAMPLE [CONSOLE.OPENSIFT.IO/V1]

### Description

ConsoleYAMLSample is an extension for customizing OpenShift web console YAML samples. Compatibility level 2: Stable within a major release for a minimum of 9 months or 3 minor releases (whichever is longer).

### Type

**object**

### Required

- **metadata**
- **spec**

## 9.1. SPECIFICATION

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



Property	Type	Description
<b>spec</b>	<b>object</b>	ConsoleYAMLSampleSpec is the desired YAML sample configuration. Samples will appear with their descriptions in a samples sidebar when creating a resources in the web console.

### 9.1.1. .spec

#### Description

ConsoleYAMLSampleSpec is the desired YAML sample configuration. Samples will appear with their descriptions in a samples sidebar when creating a resources in the web console.

#### Type

**object**

#### Required

- **description**
- **targetResource**
- **title**
- **yaml**

Property	Type	Description
<b>description</b>	<b>string</b>	description of the YAML sample.
<b>snippet</b>	<b>boolean</b>	snippet indicates that the YAML sample is not the full YAML resource definition, but a fragment that can be inserted into the existing YAML document at the user's cursor.
<b>targetResource</b>	<b>object</b>	targetResource contains apiVersion and kind of the resource YAML sample is representating.
<b>title</b>	<b>string</b>	title of the YAML sample.
<b>yaml</b>	<b>string</b>	yaml is the YAML sample to display.

### 9.1.2. .spec.targetResource

#### Description

targetResource contains apiVersion and kind of the resource YAML sample is representing.

#### Type

**object**

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

## 9.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/console.openshift.io/v1/consoleyamlsamples**
  - **DELETE**: delete collection of ConsoleYAMLSample
  - **GET**: list objects of kind ConsoleYAMLSample
  - **POST**: create a ConsoleYAMLSample
- **/apis/console.openshift.io/v1/consoleyamlsamples/{name}**
  - **DELETE**: delete a ConsoleYAMLSample
  - **GET**: read the specified ConsoleYAMLSample
  - **PATCH**: partially update the specified ConsoleYAMLSample
  - **PUT**: replace the specified ConsoleYAMLSample

### 9.2.1. /apis/console.openshift.io/v1/consoleyamlsamples

#### HTTP method

#### DELETE

#### Description

delete collection of ConsoleYAMLSample

Table 9.1. HTTP responses

HTTP code	Reponse body
200 - OK	<a href="#">Status</a> schema
401 - Unauthorized	Empty

#### HTTP method

#### GET

#### Description

list objects of kind ConsoleYAMLSample

Table 9.2. HTTP responses

HTTP code	Reponse body
200 - OK	<a href="#">ConsoleYAMLSampleList</a> schema
401 - Unauthorized	Empty

#### HTTP method

#### POST

#### Description

create a ConsoleYAMLSample

Table 9.3. Query parameters

Parameter	Type	Description
<b>dryRun</b>	<b>string</b>	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
<b>fieldValidation</b>	<b>string</b>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: <ul style="list-style-type: none"> <li>- Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23.</li> <li>- Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+</li> <li>- Strict: This will fail the request with a 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.</li> </ul>

Table 9.4. Body parameters

Parameter	Type	Description
<b>body</b>	<a href="#">ConsoleYAMLSample</a> schema	

Table 9.5. HTTP responses

HTTP code	Response body
200 - OK	<a href="#">ConsoleYAMLSample</a> schema
201 - Created	<a href="#">ConsoleYAMLSample</a> schema
202 - Accepted	<a href="#">ConsoleYAMLSample</a> schema
401 - Unauthorized	Empty

### 9.2.2. /apis/console.openshift.io/v1/consoleyamlsamples/{name}

Table 9.6. Global path parameters

Parameter	Type	Description
<b>name</b>	<b>string</b>	name of the ConsoleYAMLSample

**HTTP method****DELETE****Description**

delete a ConsoleYAMLSample

**Table 9.7. Query parameters**

Parameter	Type	Description
<b>dryRun</b>	<b>string</b>	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

**Table 9.8. HTTP responses**

HTTP code	Reponse body
200 - OK	<b>Status</b> schema
202 - Accepted	<b>Status</b> schema
401 - Unauthorized	Empty

**HTTP method****GET****Description**

read the specified ConsoleYAMLSample

**Table 9.9. HTTP responses**

HTTP code	Reponse body
200 - OK	<b>ConsoleYAMLSample</b> schema
401 - Unauthorized	Empty

**HTTP method****PATCH****Description**

partially update the specified ConsoleYAMLSample

**Table 9.10. Query parameters**

Parameter	Type	Description
<b>dryRun</b>	<b>string</b>	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
<b>fieldValidation</b>	<b>string</b>	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.

Table 9.11. HTTP responses

HTTP code	Response body
200 - OK	<a href="#">ConsoleYAMLSample</a> schema
401 - Unauthorized	Empty

**HTTP method****PUT****Description**

replace the specified ConsoleYAMLSample

Table 9.12. Query parameters

Parameter	Type	Description
-----------	------	-------------

Parameter	Type	Description
<b>dryRun</b>	<b>string</b>	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
<b>fieldValidation</b>	<b>string</b>	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.

Table 9.13. Body parameters

Parameter	Type	Description
<b>body</b>	<a href="#">ConsoleYAMLSample</a> schema	

Table 9.14. HTTP responses

HTTP code	Response body
200 - OK	<a href="#">ConsoleYAMLSample</a> schema
201 - Created	<a href="#">ConsoleYAMLSample</a> schema
401 - Unauthorized	Empty