|  |
| --- |
| Intel |
| ISC API |
| Arvind Nadendla |

Contents

[API Documentation 3](#_Toc436044086)

[API Versioning 3](#_Toc436044087)

[Optional field was added to an API call/ Optional field was removed 3](#_Toc436044088)

[ISC Developer Impact 3](#_Toc436044089)

[User Impact 3](#_Toc436044090)

[Required field was added to an API call 3](#_Toc436044091)

[ISC Developer Impact 3](#_Toc436044092)

[User Impact 3](#_Toc436044093)

[Required field is removed 4](#_Toc436044094)

[ISC Developer Impact 4](#_Toc436044095)

[User Impact 4](#_Toc436044096)

[Required field is made optional 4](#_Toc436044097)

[ISC Developer Impact 4](#_Toc436044098)

[User Impact 4](#_Toc436044099)

[API payload changes completely 4](#_Toc436044100)

[Response Changes 5](#_Toc436044101)

[ISC Developer Impact 5](#_Toc436044102)

[User Impact 5](#_Toc436044103)

[API Error Codes 5](#_Toc436044104)

[Validation/Locking errors 5](#_Toc436044105)

[Remote system errors 5](#_Toc436044106)

[General ISC Errors 5](#_Toc436044107)

[Internal server error 5](#_Toc436044108)

# API Documentation

Swagger is used to document ISC API’s

# API Versioning

Versioning of API’s in ISC is handled by using the content-type header in each of the HTML requests.

Because we started with versions in the URL, we will continue to have v1 in the URL.

Cases:

## Element/Field name change

## Optional field was added/remove

Taking the example of a Virtualization Connector and a new optional field is added/removed to the DTO.

### ISC Developer Impact

As the field is optional, the API itself does not change. If the new field is provided, it is persisted in ISC if not it is left blank.

The change does not need the API to be versioned

### User Impact

As the field is optional, the API itself does not change. If the new field is provided, it is persisted in ISC if not it is left blank.

The User can continue to use the API either with older payload or the newer payload and ISC is able to accept either.

## Required field was added

Taking the example of a Virtualization Connector a new required field is added to the DTO.

### ISC Developer Impact

As the field is a required field, for older API we need to allow omitting the field. For the Older API calls which come into ISC we can choose a sensible default for the field value and persist it in ISC.

For Newer API calls we require the field to have a valid value, we have to validate the value is not null.

This change will required the API to be versioned. Based on the content type, ISC will be able to determine whether the new API or the old API is being invoked and behaves appropriately.

Alternatively, we could handle all API calls as older API calls and for API coming in with the required field value set to something, we use that value.

This strategy would not require the API to be versioned.

### User Impact

For Users using the old API and wanting to continue using the old API, nothing changes.

For user trying to use the new API calls with the required field value set to something would need to use a content type like below for example

*Content-Type: application/vnd.isc.v2.0+json*

This would indicate that the user is trying to invoke the new API and ISC will validate it appropriately.

## Required field is removed

Taking the example of a Virtualization Connector a required field is removed from the DTO.

### ISC Developer Impact

As field was required in the older version, any API calls coming in will still require the field to be present and ISC will validate it as such

For newer API version, since this field has been removed we will validate that the field has an empty value.

This change will require API to be versioned.

### User Impact

For Users using the old API and wanting to continue using the old API, nothing changes.

For user trying to use the new API calls without the required field value they would need to use a content type like below for example

*Content-Type: application/vnd.isc.v2.0+json*

This would indicate that the user is trying to invoke the new API and ISC will validate it appropriately.

## Required field is made optional

Taking the example of a Virtualization Connector a required field is made optional from the DTO.

### ISC Developer Impact

As field was required in the older version, any API calls coming in will still require the field to be present and ISC will validate it as such

For newer API version, since this field has been made optional we will accept the value if provided.

This change will require API to be versioned.

### User Impact

For Users using the old API and wanting to continue using the old API, nothing changes.

For user trying to use the new API calls without the required field value they would need to use a content type like below for example

*Content-Type: application/vnd.isc.v2.0+json*

This would indicate that the user is trying to invoke the new API and ISC will validate it appropriately.

## API payload changes completely

If there are a lot of changes in the API with respect to adding/removing many required fields, we might need to consider creating a new API which accepts the new payload.

## Response Changes

ISC will add the appropriate response headers to its responses. Based on the response headers, the client will need to deserialize into the response objects as indicated by the headers.

For example, if a newer ISC API is called by using a specific content type header, ISC will return the same content-type to indicate the newer response.

If the Old ISC API is called we will return the old response.

### ISC Developer Impact

If response needs to be changed, you need to introduce a new versioned content-type header.

### User Impact

The response object changes will be managed the same way. The client is expected to check the response media type string to accommodate changes if needed.

# API Error Codes

In addition to the standard HTTP error codes, ISC provides more context sensistive error code so API users can react appropriately

## Validation/Locking errors

For the following exceptions we return a HTTP 400 error with 4000 as the ISC error code

VmidcBrokerInvalidEntryException, VmidcBrokerInvalidRequestException, VmidcBrokerValidationException

## Remote system errors

For exceptions reported to us by other systems we interact with via API we return HTTP 400 error code with 6000 as the ISC error code

RestClientException, RemoteException

## General ISC Errors

For other ISC error we return HTTP error code 500 with ISC error code 5000

## Internal server error

We return HTTP error code 500 with NO corresponding ISC error code

{

errorCode: "4000"

errorMessage: "Intel Security Controller: VirtualizationConnector entry with ID 600 is not found."

}