



# Custom Rules for API Standardization

Available in SwaggerHub SaaS for organizations on the Enterprise plan and in SwaggerHub On-Premise 1.23 and later.

Organizations can add custom rules for [API standardization\[api-standardization.html\]](#) that will be used to validate OpenAPI definitions for compliance with API design guidelines. These rules can help you verify that:

- a specific node exists in YAML, regardless of the value,
- the node name or value matches (or does not match) a regular expression.

Each rule consists of a node path (such as `$.info.version` ) and, optionally, a regular expression that defines valid values for this node. You can customize the rule severity and applicable OpenAPI versions.


## Considerations

Custom standardization rules have the same limitations as the built-in rules:

- They apply only to APIs, not to domains.
- They check the YAML structure as is and do not resolve `$refs` .

## Create a custom rule

To create rules, you must be an organization owner.

- 1 Go to **My Hub** and click  next to the organization name in the sidebar.
- 2 Switch to the **Standardization** tab.
- 3 Scroll to the bottom and click **Add Custom Rule**.



- 4 Configure the rule (see the parameters below) and click **Save**.



### Tip

You can also [test the rule\[custom-rules-for-api-standardization.html#test-a-rule\]](#) directly in the rule editor.



### Important

After you create or edit a rule, click **Save** at the top of the Standardization page to re-scan your organization's APIs.



Add Custom Rule

Details

Name \*

Specification

OpenAPI (all)

Rule Severity ⓘ

Error

Error Message ⓘ

optional

Description ⓘ

Path Required \* ⓘ

\$.

Test For

pattern

Field ⓘ

Pattern \* ⓘ

Must Match

Try It Out ⓘ

IMPORT API

1

Import or Paste YAML  
to test your Custom Rule  
Like [Help](#) creating a Custom Rule?

CANCEL

SAVE

Name

Required. A unique name for this rule, up to 35 characters. The name is also used as a prefix for the rule's error message.

Specification

Select whether the rule applies to All API definitions, OpenAPI (all), OpenAPI 3.0, OpenAPI 2.0 or AsyncAPI. OpenAPI 2.0 and 3.0 definitions use a slightly different structure and keywords, so some validations may need separate rules for different OpenAPI versions.

Rule Severity

Error or Warning. Errors will prevent the API from being published [manage-apis/publishing-an-api.html] if the Require API to pass Standardization to be publishable option is selected in the Standardization settings.

Error Message

An optional additional error message displayed in the SwaggerHub editor [edit-apis/code-editor-for-openapi.html], up to 130 chars. If omitted, the rule name is used as the error message. If both the name and the error message are specified, the name will prefix the error message. For example, given:

- Rule name: Version
- Error message: `info.version` must be major.minor.patch

the full error message in the editor will be:

Version -> `info.version` must be major.minor.patch

Description

Optional. It is a short description of the rule.



Support[/] [\[https://support.smartbear.com\]](https://support.smartbear.com)

### Path Required

A JSONPath expression that specifies the target node or nodes to check. For example, `$.info.version`.

### Case-sensitive.

We use the *jsonpath-plus* library that supports some extra selectors in addition to the standard JSONPath syntax. See the [jsonpath-plus documentation](https://github.com/s3u/JSONPath#syntax-through-examples) for the supported syntax.

### Test For

Specify **pattern** to search for the regular expression specified in the **Pattern** field (below).

Alternately, you can choose **defined** to to check that the node is present, regardless of the value. For example, check that tag objects have a `description` specified. See also [How "defined" rules work](#) below.

### Field

Field to be tested. Set this to `@key` if you want to test against a field name rather than a value.

### Pattern

A [regular expression](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Regular_Expressions#Writing_a_regular_expression_pattern) to test against the result of the specified JSONPath. For example, `^\d+\.\d+\.\d+$`.

Notes:

- This option is available if the defined option is not selected in the **Test For** field.
- If the path expression returns an array, each array item is tested against this regular expression.
- Patterns look for a partial match unless the pattern contains the `^` (beginning of value) and `$` (end of value) anchors. For example, `a` matches *any string* that contains this character, but `^a$` matches the exact string "a".
- Patterns are **case-sensitive** by default. For a case-insensitive match, use the format `/pattern/i`, for example, `/https/i`.

### Must Match

Regular expression match type: "must match" (check box is selected) or "must not match" (unselected).

### Try It Out

Use this section to [test the rule](#) against an existing API. You can either paste the API definition into the **Try It Out** field or use the **IMPORT API** button to import an API into the **Try It Out** field for testing purposes.

### Save

Save the rule.

## Examples

### Version must be major.minor.patch (digits only)

**Path Required:** `$.info.version`

**Test For:** pattern

**Pattern:** `^\d+\.\d+\.\d+$`

**Must Match:** selected

### Paths should not end with a slash

**Path Required:** `$.paths.*~`

**Test For:** pattern

**Pattern:** `^\/.+\/$`

**Must Match:** unselected

Path names like `/pets` and `/pets/{id}` are considered valid, but `/pets/` and `/pets/{id}/` will be flagged.

### Response content type must match application/json

**Path Required:** `$.paths.*.responses.*.content`

Test for: pattern

Field: @key

Pattern: application/json

Must Match: selected

The **content** field within the **responses** node can only contain a field that is **application/json**. Any other field within the **content** field will cause the API definition to fail validation.

## POST responses must contain 200 response

Path Required: \$.paths.\*.post.responses

Test For: defined

Field: 200

All POST responses in the document must contain the 200 response field.

## How "defined" rules work

If you specify "defined" in the **Test For** field, then your custom rule verifies the presence of a specific node or nodes in YAML. The specified JSONPath expression is first split into two components: the last component (the target key name) and everything that precedes it (let's call this "base path"). SwaggerHub then finds all the nodes with the "base path" and checks whether each of them contains the target key. The nodes that do not contain this key will fail the validation.

Sample rule: "201 responses to POST requests must include the Location header"

```
$.paths.*.post.responses.201.headers.Location
|_____|/ |_____|/
|find all the nodes at this path|and check whether
|                               |they have this key
```

Keep in mind that the **presence of intermediate nodes is not enforced**. For example, the rule above will not flag 201 responses without headers – because these responses will not be found by the "base path" expression `...201.headers`.

```
# Rule will flag this response
responses:
  201:
    description: Created
    headers:
      X-RateLimit-Remaining:
        type: string

# but not this response
responses:
  201:
    description: Created
```

In this example, the "Location header exists" rule needs to be implemented as two "defined" rules in order to verify that: a) 201 responses have headers ; and b) headers include the Location header:

Defined: \$.paths.\*.post.responses.201.headers

Defined: \$.paths.\*.post.responses.201.headers.Location

## Test a rule

You can quickly test custom rules against existing APIs directly in the rule editor. To do that:

- 1 Edit a rule or start creating a new one.
- 2 Paste the YAML to test against or click **Import API** and load an existing API.



### Note

Pasting works only if the path has not been specified yet.


- 3 Enter the JSONPath and, optionally, a regular expression to test the value.
- 4 The validation results will appear in the **Try It Out** section.

The **Try It Out** section displays YAML nodes found by the specified JSONPath. Leading dashes ( – ) indicate individual matches. Note that a match can be a standalone value, a simple `key: value` node, an array, or a multi-property object (including nested objects). The error and warning icons in the gutter indicate the nodes that failed the validation.

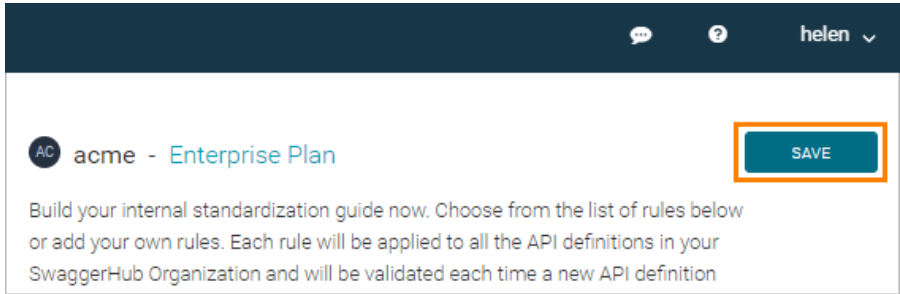
The results may look differently depending on whether you test a "defined" rule or a pattern-based rule. For example, when testing the existence of tag descriptions ( `$.tags[*].description` ), the results would show all the tag objects ( `"base"` nodes) and indicate those without the `description` property. However, when testing the *values* of tag descriptions against a regular expression, the results will show the descriptions themselves.

## Delete a custom rule

**Note:** In SwaggerHub On-Premise, this is supported since v. 1.27.

1 In the **Custom Rules** list, click  next to the rule you want to delete and confirm the deletion.

2  **Important**  
Click **Save** at the top of the Standardization page to save the changes.



## Enable or disable a rule


- 1 Select or clear the check box next to the rule.
- 2 Click **Save** at the top of the page to save the changes.

## Public style guides

Style guides enforce industry best practices and standards.





To add a public style guide:



-  **Note**
- Preview** takes you to the style guide documentation pages.  
[About Us](https://smartbear.com/company/about-us/) | [Careers](https://smartbear.com/company/careers/) | [Solutions](https://smartbear.com/solutions/) | [Partners](https://smartbear.com/partners/) | [Responsibility](https://smartbear.com/company/responsibility/)
1. Go to the **Standardization** page.  
[Contact Us](https://smartbear.com/contact-us/) | +1 617-684-2600 USA | +353 91 398300 EUR | +61 391929960 AUS
  2. Click **Add Public Style Guides**.
  3. Select an existing public style guide from the list in the **Manage Public Style Guides** modal window.
  4. Add it to your set of rules on the **Standardization** page.



© 2023 SmartBear Software. All Rights Reserved.  
Here's the list of supported style guides:  
[Privacy](https://smartbear.com/privacy/) | [Terms of Use](https://smartbear.com/terms-of-use/) | [Site Map](https://smartbear.com/site-map/) | [OWASP Top 10](https://smartbear.com/custom-rules-for-api-standardization/owasp-top-10/) | [Terms of Use](https://smartbear.com/terms-of-use/) | [Security](https://smartbear.com/security/)

See Also  [\[http://twitter.com/smartbear\]](http://twitter.com/smartbear)  [\[http://www.facebook.com/smartbear\]](http://www.facebook.com/smartbear)  [\[https://www.linkedin.com/company/smartbear/\]](https://www.linkedin.com/company/smartbear/)  [\[http://www.youtube.com/user/SmartBearSoftware\]](http://www.youtube.com/user/SmartBearSoftware)

- [API Standardization](#) [api-standardization.html](#)