API Validation Report

Automated API Specification Analysis

Sample Pet Store API

Version: 1.0.0

Owner: test-organization

API Quality Score

' VALIDATION FAILED

Generated: 2/20/2026, 9:03:25 AM
By: API Governance Team

Executive Summary

20	1	12	7
Total Issues	Errors	Warnings	Info

The API specification "Sample Pet Store API" (version 1.0.0) was analyzed against OpenAPI compliance rules and API design best practices. A total of 20 issue(s) were identified: 1 error(s), 12 warning(s), and 7 informational finding(s). The API specification FAILED validation due to critical errors that must be resolved before the API can be approved for production use.

Issues by Category

Category	Issues	Errors	Warnings
Security	1	1	0
Documentation	10	0	6
Spec Compliance	4	0	1
Structure	2	0	2
Naming Conventions	1	0	1
Response Design	1	0	1
Best Practice	1	0	1

Changes Since Last Scan

Compared against: version 0.9.0 (scanned 2/13/2026)

30	!’	47	%² +17
Previous Score		Current Score	

Issue Count Changes

Total Issues: **-5**
Errors: **-1**
Warnings: **0**
Informational: **-1**

Resolved Issues (5)

'[Error] Schema object "Pet" has an invalid "type" value
Path: components.schemas.Pet.type
'[Error] Description contains script tags
Path: info.description
'[Warning] Path should use kebab-case. Avoid camelCase or snake_case in URLs.
Path: paths./userAccounts
'[Warning] Operation "description" must be present and non-empty string.
Path: paths./users.get
'[Warning] Operation "description" must be present and non-empty string.
Path: paths./users.post

New Issues Introduced (10)

[Warning] Operation must have non-empty "tags" array.
Path: paths./petCategories.get
[Warning] Operation must define at least one 2xx or 3xx response.
Path: paths./petCategories.get.responses
[Warning] POST operation should have a requestBody defined.
Path: paths./pets.post
[Information] Parameter should have a description.
Path: paths./pets/{petId}.get.parameters.0
[Information] Response should have a meaningful description.
Path: paths./pets.post.responses.201
[Information] Tag should have a description.
Path: tags.0
[Information] Tag should have a description.
Path: tags.1
[Information] Property "id" should have a description.
Path: components.schemas.Pet.properties.id
[Information] Property "name" should have a description.
Path: components.schemas.Pet.properties.name
[Information] Property "tag" should have a description.
Path: components.schemas.Pet.properties.tag

10 issue(s) remain unchanged from the previous scan.

Overall improvement: score increased by 17 point(s).

Detailed Findings

#1 ERROR Rule: no-eval-in-markdown

Markdown descriptions should not contain "eval(" expressions.

Path: info.description

#2 WARNING Rule: info-contact

Info object must have "contact" object.

Path: info

#3 WARNING Rule: info-description

Info "description" must be present and non-empty string.

Path: info

#4 WARNING Rule: info-license

Info object must have "license" object.

Path: info

#5 WARNING Rule: oas3-api-servers

OpenAPI "servers" must be present and non-empty array.

Path: line 1

#6 WARNING Rule: operation-description

Operation "description" must be present and non-empty string.

Path: paths./pets.get

#7 WARNING Rule: operation-operationId

Operation must have "operationId".

Path: paths./pets.get

#8 WARNING Rule: operation-description

Operation "description" must be present and non-empty string.

Path: paths./pets.post

#9 WARNING Rule: operation-description

Operation "description" must be present and non-empty string.

Path: paths./pets/{petId}.get

#10 WARNING Rule: bp-path-casing

Path should use kebab-case. Avoid camelCase or snake_case in URLs.

Path: paths./petCategories

#11 WARNING Rule: operation-tags

Operation must have non-empty "tags" array.

Path: paths./petCategories.get

#12 WARNING Rule: operation-success-response

Operation must define at least one 2xx or 3xx response.

Path: paths./petCategories.get.responses

#13 WARNING Rule: bp-request-body-required

POST operation should have a requestBody defined.

Path: paths./pets.post

#14 INFORMATION Rule: bp-parameter-descriptions

Parameter should have a description.

Path: paths./pets/{petId}.get.parameters.0

#15 INFORMATION Rule: bp-response-descriptions

Response should have a meaningful description.

Path: paths./pets.post.responses.201

#16 INFORMATION Rule: bp-tags-description

Tag should have a description.

Path: tags.0

#17 INFORMATION Rule: bp-tags-description

Tag should have a description.

Path: tags.1

#18 INFORMATION Rule: oas3-schema

Property "id" should have a description.

Path: components.schemas.Pet.properties.id

#19 INFORMATION Rule: oas3-schema

Property "name" should have a description.

Path: components.schemas.Pet.properties.name

#20 INFORMATION Rule: oas3-schema

Property "tag" should have a description.

Path: components.schemas.Pet.properties.tag

Category Analysis

Security

1 issues

- Markdown descriptions should not contain "eval(" expressions.

Documentation

10 issues

- Info object must have "contact" object.
- Info "description" must be present and non-empty string.
- Info object must have "license" object.
- Operation "description" must be present and non-empty string.
- Operation "description" must be present and non-empty string.

... and 5 more

Spec Compliance

4 issues

- OpenAPI "servers" must be present and non-empty array.
- Property "id" should have a description.
- Property "name" should have a description.
- Property "tag" should have a description.

Structure

2 issues

- Operation must have "operationId".
- Operation must have non-empty "tags" array.

Naming Conventions

1 issues

- Path should use kebab-case. Avoid camelCase or snake_case in URLs.

Response Design

1 issues

- Operation must define at least one 2xx or 3xx response.

Best Practice

1 issues

- POST operation should have a requestBody defined.

Recommendations

1. Improve API Documentation

Add missing descriptions, contact information, and licensing details. Well-documented APIs are easier for consumers to understand and integrate with.

Priority: Medium

2. Standardize Naming Conventions

Ensure all URL paths use kebab-case and operation IDs follow a consistent pattern. Consistent naming improves developer experience and API discoverability.

Priority: Medium

3. Define Complete Response Models

Ensure all operations define success and error responses with appropriate schemas. Include standard error response models (400, 401, 404, 500) for consistency.

Priority: Medium

4. Address Security Findings

Review and remediate security-related findings. Ensure proper authentication schemes are defined and no sensitive data is exposed in the specification.

Priority: High

5. Adopt API Design Best Practices

Review the best practice findings and align your API design with organizational standards. This includes proper error handling, consistent patterns, and comprehensive schemas.

Priority: Low