
STAC API Validator

Phil Varner

Apr 29, 2024

CONTENTS

1	Introduction	3
2	Documentation	5
3	Installation	7
4	Contributing	9
5	Usage	11
6	Features	13
7	Command-line Reference	15
8	Validating OGC API Features - Part 1 compliance	17
9	Common Mistakes	19
10	License	21
11	Issues	23
12	Credits	25
	Python Module Index	37
	Index	39

INTRODUCTION

The STAC API Validator is the official validation suite for the [STAC API](#) family of specifications.

DOCUMENTATION

See the [stable](#) or [latest](#) documentation pages.

INSTALLATION

STAC API Validator requires Python 3.10.

You can install *STAC API Validator* via [pip](#) from [PyPI](#):

```
pip install stac-api-validator
```

and then run it:

```
stac-api-validator \
  --root-url https://planetarycomputer.microsoft.com/api/stac/v1/ \
  --conformance core \
  --conformance features \
  --conformance item-search \
  --collection sentinel-2-l2a \
  --geometry '{"type": "Polygon", "coordinates": [[[100.0, 0.0], [101.0, 0.0], [101.0, ↵
↵1.0], [100.0, 1.0], [100.0, 0.0]]]}'
```


CONTRIBUTING

Contributions are very welcome. To learn more, see the *Contributor Guide*.

USAGE

Please see the *Command-line Reference* for details.

The conformance class validations to run are selected with the `--conformance` parameters. This parameter can be used more than once to specify multiple conformance classes to validate. The STAC API - Core conformance class will always be validated, even if not specified.

If `item-search`, `collections`, and/or `features` are specified, the `--collection` and `--geometry` parameters must also be specified. The `--collection` parameter specifies the name of a collection to use for some of the validations. The `--geometry` should specify an AOI over which there are between 100 and 20,000 results for the collection (more results means longer time to run).

FEATURES

Work in Progress – this currently only validates a subset of behavior

This validation suite focuses on validating STAC API interactions. Tools such as [pystac](#) and [stac4s](#) do a good job of validating STAC objects (Catalog, Collection, Item). This suite focuses on the STAC API behavior validation.

The three key concepts within a STAC API are:

1. *Conformance classes* advertising the capabilities of the API
2. *Link relations* between resources within the web API (hypermedia)
3. *Parameters* that filter search results

The conformance classes, as defined in the `conformsTo` field of the Landing Page (root, /), advertise to clients which capabilities are available in the API. Without this field, a client would not even be able to tell that a root URI was a STAC API.

The link relations define how to navigate a STAC catalog through parent-child links and find resources such as the OpenAPI specification. While many OGC API and STAC API endpoint have a fixed value (e.g., `/collections`), it is preferable for clients discover the paths via hypermedia.

The parameters that filter results apply to the Items resource and Item Search endpoints.

The current validity status of several popular STAC API implementations can be found [here](#).

COMMAND-LINE REFERENCE

Usage:

```
Usage: stac-api-validator [OPTIONS]
```

STAC API Validator.

Options:

<code>--version</code>	Show the version and exit.
<code>--log-level TEXT</code>	Logging level, one of DEBUG, INFO, WARN, ERROR, CRITICAL
<code>--root-url TEXT</code>	STAC API Root / Landing Page URL [required]
<code>--collection TEXT</code>	The name of the collection to use for item-search, collections, and features tests.
<code>--geometry TEXT</code>	The GeoJSON geometry to use for intersection tests.
<code>--conformance [core browseable item-search features collections children filter]</code>	The conformance classes to validate. [required]
<code>--auth-bearer-token TEXT</code>	Authorization Bearer token value to append to all requests.
<code>--auth-query-parameter TEXT</code>	Query parameter key and value to pass for authorization, e.g., <code>'key=xyz'</code> .
<code>--help</code>	Show this message and exit.

Conformance classes item-search, features, and collections require the `--collection` parameter with the id of a collection to run some tests on.

Conformance class item-search requires `--geometry` with a GeoJSON geometry that returns some items for the specified collection.

Example:

```
stac-api-validator \  
  --root-url https://planetarycomputer.microsoft.com/api/stac/v1/ \  
  --conformance core \  
  --conformance item-search \  
  --conformance features \  
  --collection sentinel-2-l2a \  
  --geometry '{"type": "Polygon", "coordinates": [[[100.0, 0.0], [101.0, 0.0], [101.0, ↵  
↪1.0], [100.0, 1.0], [100.0, 0.0]]]}'
```

Example output:

```
Validating https://cmr.earthdata.nasa.gov/stac/LARC_ASDC ...
STAC API - Core conformance class found.
STAC API - Item Search conformance class found.
warnings: none
errors:
- service-desc (https://api.stacspec.org/v1.0.0-beta.1/openapi.yaml): should have
  ↳ content-type header 'application/vnd.oai.openapi+json;version=3.0', actually 'text/
  ↳ yaml'
- service-desc (https://api.stacspec.org/v1.0.0-beta.1/openapi.yaml): should return JSON,
  ↳ instead got non-JSON text
- GET Search with bbox=100.0, 0.0, 105.0, 1.0 returned status code 400
- POST Search with bbox:[100.0, 0.0, 105.0, 1.0] returned status code 502
- GET Search with bbox=100.0,0.0,0.0,105.0,1.0,1.0 returned status code 400
- POST Search with bbox:[100.0, 0.0, 0.0, 105.0, 1.0, 1.0] returned status code 400
```

Example with authorization using parameters:

```
stac-api-validator --root-url https://api.radiant.earth/mlhub/v1 --conformance core --
  ↳ auth-query-parameter 'key=xxx'
```

VALIDATING OGC API FEATURES - PART 1 COMPLIANCE

A STAC API that conforms to the “STAC API - Features” conformance class will also be a valid implementation of OGC API Features - Part 1. In general, this validator focuses on those aspects of API behavior that are different between STAC and OGC. It is recommended that implementers also use the [OGC API Features - Part 1 validation test suite](#) to validate conformance.

Full instructions are available at the link above, but the simplest way to run this is with:

```
docker run -p 8081:8080 ogccite/ets-ogcapi-features10
```

Then, open <http://localhost:8081/teamengine/> and login with the username and password ogctest, Create a new session, with Organization OGC, Specification OGC API - Features, Start a new test session, input the root URL for the service, and Start.

COMMON MISTAKES

- incorrect `conformsTo` in the Landing Page. This was added between STAC API 0.9 and 1.0. It should be the same as the value in the `conformsTo` in the `OAFeat /conformance` endpoint.
- OGC API Features uses `data` relation link relation at the root to point to the Collections endpoint (`/collections`), not `collections` relation
- media type for link relation `service-desc` and endpoint is `application/vnd.oai.openapi+json; version=3.0` (not `application/json`) and link relation `search` and endpoint is `application/geo+json` (not `application/json`)
- Use of OGC API “`req`” urls instead of “`conf`” urls, e.g. `http://www.opengis.net/spec/ogcapi-features-1/1.0/conf/core` should be used, not `http://www.opengis.net/spec/ogcapi-features-1/1.0/req/core`

LICENSE

Distributed under the terms of the [Apache 2.0 license](#), *STAC API Validator* is free and open source software.

ISSUES

If you encounter any problems, please [file an issue](#) along with a detailed description.

This project was generated from [@cjolowicz's Hypermodern Python Cookiecutter](#) template.

12.1 Usage

12.1.1 stac-api-validator

STAC API Validator.

```
stac-api-validator [OPTIONS]
```

Options

--version

Show the version and exit.

--log-level <log_level>

Logging level, one of DEBUG, INFO, WARN, ERROR, CRITICAL

--root-url <root_url>

Required STAC API Root / Landing Page URL

--collection <collection>

The name of the collection to use for item-search, collections, and features tests.

--geometry <geometry>

The GeoJSON geometry to use for intersection tests.

--conformance <conformance_classes>

Required The conformance classes to validate.

Options

core | browseable | item-search | features | collections | children | filter | item-search#sort | item-search#fields | item-search#query | features#sort | features#fields | features#query | transaction

--auth-bearer-token <auth_bearer_token>

Authorization Bearer token value to append to all requests.

--auth-query-parameter <auth_query_parameter>

Query parameter key and value to pass for authorization, e.g., 'key=xyz'.

--fields-nested-property <fields_nested_property>
Fields Extension: name of a field in Item Properties, e.g. 'properties.eo:cloud_cover'

--validate-pagination, --no-validate-pagination
Validate pagination behavior (can take a long time to run)

--query-comparison-field <query_comparison_field>
Query Extension: name of field to use for comparison operators tests (eq, neq, lt, lte, gt, gte)

--query-eq-value <query_eq_value>
Query Extension: value of field to use for eq

--query-neq-value <query_neq_value>
Query Extension: value of field to use for neq

--query-lt-value <query_lt_value>
Query Extension: value of field to use for lt

--query-lte-value <query_lte_value>
Query Extension: value of field to use for lte

--query-gt-value <query_gt_value>
Query Extension: value of field to use for gt

--query-gte-value <query_gte_value>
Query Extension: value of field to use for gte

--query-substring-field <query_substring_field>
Query Extension: name of field to use for substring operators tests (startsWith, endsWith, contains)

--query-starts-with-value <query_starts_with_value>
Query Extension: value of field to use for startsWith

--query-ends-with-value <query_ends_with_value>
Query Extension: value of field to use for endsWith

--query-contains-value <query_contains_value>
Query Extension: value of field to use for contains

--query-in-field <query_in_field>
Query Extension: name of field to use for 'in' operator tests

--query-in-values <query_in_values>
Query Extension: comma-separated values of field to use for 'in' operator tests

--transaction-collection <transaction_collection>
The name of the collection to use for Transaction Extension tests.

12.2 Reference

12.2.1 stac_api_validator

STAC API Validator.

12.3 Contributor Guide

Thank you for your interest in improving this project. This project is open-source under the [Apache 2.0 license](#) and welcomes contributions in the form of bug reports, feature requests, and pull requests.

Here is a list of important resources for contributors:

- [Source Code](#)
- [Documentation](#)
- [Issue Tracker](#)
- *[Code of Conduct](#)*

12.3.1 How to report a bug

Report bugs on the [Issue Tracker](#).

When filing an issue, make sure to answer these questions:

- Which operating system and Python version are you using?
- Which version of this project are you using?
- What did you do?
- What did you expect to see?
- What did you see instead?

The best way to get your bug fixed is to provide a test case, and/or steps to reproduce the issue.

12.3.2 How to request a feature

Request features on the [Issue Tracker](#).

12.3.3 How to set up your development environment

You need Python 3.10 and the following tools:

- [Poetry](#) version 1.3.x
- [Nox](#)
- [nox-poetry](#)

Install the package with development requirements:

```
$ poetry install
```

You can now run an interactive Python session, or the command-line interface:

```
$ poetry run python
$ poetry run stac-api-validator
```

Note: if `poetry update` or `poetry install` hang, try running `poetry cache clear --all pypi` to fix the issue.

12.3.4 How to test the project

Run the full test suite:

```
$ nox
```

List the available Nox sessions:

```
$ nox --list-sessions
```

You can also run a specific Nox session. For example, invoke the unit test suite like this:

```
$ nox --session=tests
```

Unit tests are located in the `tests` directory, and are written using the [pytest](#) testing framework.

12.3.5 How to submit changes

Open a [pull request](#) to submit changes to this project.

Your pull request needs to meet the following guidelines for acceptance:

- The Nox test suite must pass without errors and warnings.
- Include unit tests. This project maintains 100% code coverage.
- If your changes add functionality, update the documentation accordingly.

Feel free to submit early, though—we can always iterate on this.

To run linting and code formatting checks before committing your change, you can install pre-commit as a Git hook by running the following command:

```
$ nox --session=pre-commit -- install
```

It is recommended to open an issue before starting work on anything. This will allow a chance to talk it over with the owners and validate your approach.

12.4 Contributor Covenant Code of Conduct

12.4.1 Our Pledge

We as members, contributors, and leaders pledge to make participation in our community a harassment-free experience for everyone, regardless of age, body size, visible or invisible disability, ethnicity, sex characteristics, gender identity and expression, level of experience, education, socio-economic status, nationality, personal appearance, race, caste, color, religion, or sexual identity and orientation.

We pledge to act and interact in ways that contribute to an open, welcoming, diverse, inclusive, and healthy community.

12.4.2 Our Standards

Examples of behavior that contributes to a positive environment for our community include:

- Demonstrating empathy and kindness toward other people
- Being respectful of differing opinions, viewpoints, and experiences
- Giving and gracefully accepting constructive feedback
- Accepting responsibility and apologizing to those affected by our mistakes, and learning from the experience
- Focusing on what is best not just for us as individuals, but for the overall community

Examples of unacceptable behavior include:

- The use of sexualized language or imagery, and sexual attention or advances of any kind
- Trolling, insulting or derogatory comments, and personal or political attacks
- Public or private harassment
- Publishing others' private information, such as a physical or email address, without their explicit permission
- Other conduct which could reasonably be considered inappropriate in a professional setting

12.4.3 Enforcement Responsibilities

Community leaders are responsible for clarifying and enforcing our standards of acceptable behavior and will take appropriate and fair corrective action in response to any behavior that they deem inappropriate, threatening, offensive, or harmful.

Community leaders have the right and responsibility to remove, edit, or reject comments, commits, code, wiki edits, issues, and other contributions that are not aligned to this Code of Conduct, and will communicate reasons for moderation decisions when appropriate.

12.4.4 Scope

This Code of Conduct applies within all community spaces, and also applies when an individual is officially representing the community in public spaces. Examples of representing our community include using an official e-mail address, posting via an official social media account, or acting as an appointed representative at an online or offline event.

12.4.5 Enforcement

Instances of abusive, harassing, or otherwise unacceptable behavior may be reported to the community leaders responsible for enforcement at phil@philvarner.com. All complaints will be reviewed and investigated promptly and fairly.

All community leaders are obligated to respect the privacy and security of the reporter of any incident.

12.4.6 Enforcement Guidelines

Community leaders will follow these Community Impact Guidelines in determining the consequences for any action they deem in violation of this Code of Conduct:

1. Correction

Community Impact: Use of inappropriate language or other behavior deemed unprofessional or unwelcome in the community.

Consequence: A private, written warning from community leaders, providing clarity around the nature of the violation and an explanation of why the behavior was inappropriate. A public apology may be requested.

2. Warning

Community Impact: A violation through a single incident or series of actions.

Consequence: A warning with consequences for continued behavior. No interaction with the people involved, including unsolicited interaction with those enforcing the Code of Conduct, for a specified period of time. This includes avoiding interactions in community spaces as well as external channels like social media. Violating these terms may lead to a temporary or permanent ban.

3. Temporary Ban

Community Impact: A serious violation of community standards, including sustained inappropriate behavior.

Consequence: A temporary ban from any sort of interaction or public communication with the community for a specified period of time. No public or private interaction with the people involved, including unsolicited interaction with those enforcing the Code of Conduct, is allowed during this period. Violating these terms may lead to a permanent ban.

4. Permanent Ban

Community Impact: Demonstrating a pattern of violation of community standards, including sustained inappropriate behavior, harassment of an individual, or aggression toward or disparagement of classes of individuals.

Consequence: A permanent ban from any sort of public interaction within the community.

12.4.7 Attribution

This Code of Conduct is adapted from the Contributor Covenant, version 2.1, available at https://www.contributor-covenant.org/version/2/1/code_of_conduct.html.

Community Impact Guidelines were inspired by Mozilla's code of conduct enforcement ladder.

For answers to common questions about this code of conduct, see the FAQ at <https://www.contributor-covenant.org/faq>. Translations are available at <https://www.contributor-covenant.org/translations>.

12.5 License

Copyright 2022 Radiant Earth Foundation

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

(continues on next page)

(continued from previous page)

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate

(continues on next page)

(continued from previous page)

as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
 - (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.
5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. Trademarks. This License does not grant permission to use the trade

(continues on next page)

(continued from previous page)

- names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

(continues on next page)

(continued from previous page)

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

PYTHON MODULE INDEX

S

stac_api_validator, [26](#)

Symbols

- auth-bearer-token
 - stac-api-validator command line option, 25
- auth-query-parameter
 - stac-api-validator command line option, 25
- collection
 - stac-api-validator command line option, 25
- conformance
 - stac-api-validator command line option, 25
- fields-nested-property
 - stac-api-validator command line option, 25
- geometry
 - stac-api-validator command line option, 25
- log-level
 - stac-api-validator command line option, 25
- no-validate-pagination
 - stac-api-validator command line option, 26
- query-comparison-field
 - stac-api-validator command line option, 26
- query-contains-value
 - stac-api-validator command line option, 26
- query-ends-with-value
 - stac-api-validator command line option, 26
- query-eq-value
 - stac-api-validator command line option, 26
- query-gt-value
 - stac-api-validator command line option, 26
- query-gte-value
 - stac-api-validator command line option,

- 26
- query-in-field
 - stac-api-validator command line option, 26
- query-in-values
 - stac-api-validator command line option, 26
- query-lt-value
 - stac-api-validator command line option, 26
- query-lte-value
 - stac-api-validator command line option, 26
- query-neq-value
 - stac-api-validator command line option, 26
- query-starts-with-value
 - stac-api-validator command line option, 26
- query-substring-field
 - stac-api-validator command line option, 26
- root-url
 - stac-api-validator command line option, 25
- transaction-collection
 - stac-api-validator command line option, 26
- validate-pagination
 - stac-api-validator command line option, 26
- version
 - stac-api-validator command line option, 25

M

- module
 - stac_api_validator, 26

S

- stac_api_validator
 - module, 26

stac-api-validator command line option

- auth-bearer-token, [25](#)
- auth-query-parameter, [25](#)
- collection, [25](#)
- conformance, [25](#)
- fields-nested-property, [25](#)
- geometry, [25](#)
- log-level, [25](#)
- no-validate-pagination, [26](#)
- query-comparison-field, [26](#)
- query-contains-value, [26](#)
- query-ends-with-value, [26](#)
- query-eq-value, [26](#)
- query-gt-value, [26](#)
- query-gte-value, [26](#)
- query-in-field, [26](#)
- query-in-values, [26](#)
- query-lt-value, [26](#)
- query-lte-value, [26](#)
- query-neq-value, [26](#)
- query-starts-with-value, [26](#)
- query-substring-field, [26](#)
- root-url, [25](#)
- transaction-collection, [26](#)
- validate-pagination, [26](#)
- version, [25](#)