Search projects

Help Sponsors Log in Register

jsonschema 4.22.0

✓ Latest version

```
pip install jsonschema
```

Released: Apr 30, 2024

An implementation of JSON Schema validation for Python

Navigation

Project description

Release history

Download files

Verified details

These details have been verified by PyPI

Owner

Project description

pypi v4.22.0 | python 3.8 | 3.9 | 3.10 | 3.11 | 3.12 | CI passing | docs passing | pre-commit.ci passed | DOI 10.5281/zenodo.11094907

`jsonschema` is an implementation of the JSON Schema specification for Python.

```
>>> from jsonschema import validate

>>> # A sample schema, like what we'd get from json.loa
>>> schema = {
...     "type" : "object",
...     "properties" : {
...         "price" : {"type" : "number"},
...         "name" : {"type" : "string"},
...     },
... }

>>> # If no exception is raised by validate(), the inst
>>> validate(instance={"name" : "Eggs", "price" : 34.99
```

JSON
Schema

Unverified details

*These details have **not** been verified by PyPI*

Project links

- Changelog
- Documentation
- Funding
- Homepage
- Issues
- Source
- Tidelift

GitHub Statistics

- **Stars:** 4498
- **Forks:** 570
- **Open issues:** 35
- **Open PRs:** 3

View statistics for this project via Libraries.io, or by using our public dataset on Google BigQuery

Meta

License: MIT License (MIT)

Author: Julian Berman

```
>>> validate(
...     instance={"name" : "Eggs", "price" : "Invalid"}
... )                                   # doctest: +IGN
Traceback (most recent call last):
    ...
ValidationError: 'Invalid' is not of type 'number'
```

It can also be used from the command line by installing check-jsonschema.

Features

- Full support for Draft 2020-12, Draft 2019-09, Draft 7, Draft 6, Draft 4 and Draft 3
- Lazy validation that can iteratively report *all* validation errors.
- Programmatic querying of which properties or items failed validation.

Installation

`jsonschema` is available on PyPI. You can install using pip:

```
$ pip install jsonschema
```

Extras

Two extras are available when installing the package, both currently related to `format` validation:

> - *`format`*
> - *`format-nongpl`*

They can be used when installing in order to include additional dependencies, e.g.:

```
$ pip install jsonschema'[format]'
```

Be aware that the mere presence of these dependencies – or even the specification of `format` checks in a schema – do *not* activate format checks (as per the specification). Please read the format validation documentation for further details.

About

I'm Julian Berman.

`jsonschema` is on GitHub.

Get in touch, via GitHub or otherwise, if you've got something to contribute, it'd be most welcome!

You can also generally find me on Libera (nick: `Julian`) in various channels, including `#python`.

If you feel overwhelmingly grateful, you can also sponsor me.

And for companies who appreciate `jsonschema` and its continued support and growth, `jsonschema` is also now supportable via TideLift.

Release Information

v4.22.0

- Improve `best_match` (and thereby error messages from `jsonschema.validate`) in cases where there are multiple *sibling* errors from applying `anyOf` / `allOf` – i.e. when multiple elements of a JSON array have errors, we now do prefer showing errors from earlier elements rather than simply showing an error for the full array (#1250).
- (Micro-)optimize equality checks when comparing for JSON Schema equality by first checking for object identity, as `==` would.

data validation, json, json schema, jsonschema, validation

Requires: Python >=3.8

Classifiers

- **Development Status**
  - 5 - Production/Stable
- **Intended Audience**
  - Developers
- **License**
  - OSI Approved :: MIT License
- **Operating System**
  - OS Independent
- **Programming Language**
  - Python
  - Python :: 3.8
  - Python :: 3.9
  - Python :: 3.10
  - Python :: 3.11
  - Python :: 3.12
  - Python :: Implementation :: CPython
  - Python :: Implementation :: PyPy
- **Topic**
  - File Formats ::

JSON
- File Formats :: JSON :: JSON Schema

indeed

Indeed is a Contributing sponsor of the Python Software Foundation.

PSF Sponsor · Served ethically

Help

Installing packages
Uploading packages
User guide
Project name retention
FAQs

About PyPI

PyPI Blog
Infrastructure dashboard
Statistics
Logos & trademarks
Our sponsors

Contributing to PyPI

Bugs and feedback
Contribute on GitHub
Translate PyPI
Sponsor PyPI
Development credits

Using PyPI

Code of conduct
Report security issue
Privacy policy
Terms of use
Acceptable Use Policy

Status: All Systems Operational

Developed and maintained by the Python community, for the Python community.
Donate today!

Site map

Switch to desktop version

English español français 日本語 português (Brasil) українська Ελληνικά Deutsch 中文 (简体) 中文 (繁體) русский עברית Esperanto

AWS Cloud computing and Security Sponsor	**Datadog** Monitoring	**Fastly** CDN	**Google** Download Analytics	**Microsoft** PSF Sponsor
	Pingdom Monitoring	**Sentry** Error logging	**StatusPage** Status page	