

jsonpickle Documentation

`jsonpickle` is a Python library for serialization and deserialization of complex Python objects to and from JSON. The standard Python libraries for encoding Python into JSON, such as the stdlib's `json` and `simplejson` can only handle Python primitives that have a direct JSON equivalent (e.g. dicts, lists, strings, ints, etc.). `jsonpickle` builds on top of these libraries and allows more complex data structures to be serialized to JSON. `jsonpickle` is highly configurable and extendable—allowing the user to choose the JSON backend and add additional backends.

jsonpickle Usage

Python library for serializing any arbitrary object graph into JSON.

⚠ Warning

The `jsonpickle` module **is not secure**. Only unpickle data you trust.

It is possible to construct malicious pickle data which will **execute arbitrary code during unpickling**. Never unpickle data that could have come from an untrusted source, or that could have been tampered with.

Consider signing data with an HMAC if you need to ensure that it has not been tampered with.

Safer deserialization approaches, such as reading the raw JSON directly, may be more appropriate if you are processing untrusted data.

`jsonpickle` can take almost any Python object and turn the object into JSON. Additionally, it can reconstitute the object back into Python.

The object must be accessible globally via a module and must inherit from `object` (AKA new-style classes).

Create an object:

```
class Thing(object):
    def __init__(self, name):
        self.name = name

obj = Thing('Awesome')
```

Use `jsonpickle` to transform the object into a JSON string:

Skip to content

jsonpickle

```
frozen = jsonpickle.encode(obj)
```

Use jsonpickle to recreate a Python object from a JSON string:

```
thawed = jsonpickle.decode(frozen)
```

The new object has the same type and data, but essentially is now a copy of the original.

```
assert obj.name == thawed.name
```

If you will never need to load (regenerate the Python class from JSON), you can pass in the keyword `unpicklable=False` to prevent extra information from being added to JSON:

```
oneway = jsonpickle.encode(obj, unpicklable=False)
result = jsonpickle.decode(oneway)
assert obj.name == result['name'] == 'Awesome'
```

Note

Please see the note in the [jsonpickle – High Level API](#) when serializing dictionaries that contain non-string dictionary keys.

Download & Install

The easiest way to get jsonpickle is via [PyPi](#) with [pip](#):

```
$ pip install -U jsonpickle
```

jsonpickle has no required dependencies (it uses the standard library's `json` module by default).

You can also download or [checkout](#) the latest code and install from source:

```
$ python setup.py install
```

[Skip to content](#)

API Reference

- [jsonpickle API](#)
 - [jsonpickle – High Level API](#)
 - [encode\(\)](#)
 - [decode\(\)](#)
 - [Choosing and Loading Backends](#)
 - [set_preferred_backend\(\)](#)
 - [load_backend\(\)](#)
 - [remove_backend\(\)](#)
 - [set_encoder_options\(\)](#)
 - [set_decoder_options\(\)](#)
 - [Customizing JSON output](#)
 - [object.__getstate__\(\)](#)
 - [object.__setstate__\(\)](#)

[Skip to content](#)

- [jsonpickle.handlers – Custom Serialization Handlers](#)
 - [ArrayHandler](#)
 - [ArrayHandler.flatten\(\)](#)
 - [ArrayHandler.restore\(\)](#)
 - [BaseHandler](#)
 - [BaseHandler.flatten\(\)](#)
 - [BaseHandler.handles\(\)](#)
 - [BaseHandler.restore\(\)](#)
 - [CloneFactory](#)
 - [DatetimeHandler](#)
 - [DatetimeHandler.flatten\(\)](#)
 - [DatetimeHandler.restore\(\)](#)
 - [LockHandler](#)
 - [LockHandler.flatten\(\)](#)
 - [LockHandler.restore\(\)](#)
 - [QueueHandler](#)
 - [QueueHandler.flatten\(\)](#)
 - [QueueHandler.restore\(\)](#)
 - [RegexHandler](#)
 - [RegexHandler.flatten\(\)](#)
 - [RegexHandler.restore\(\)](#)
 - [Registry](#)
 - [Registry.get\(\)](#)
 - [Registry.register\(\)](#)
 - [Registry.unregister\(\)](#)
 - [TextIOHandler](#)
 - [TextIOHandler.flatten\(\)](#)
 - [TextIOHandler.restore\(\)](#)
 - [UUIDHandler](#)
 - [UUIDHandler.flatten\(\)](#)
 - [UUIDHandler.restore\(\)](#)
 - [get\(\)](#)
 - [register\(\)](#)
 - [unregister\(\)](#)

[Skip to content](#)

- o [Low Level API](#)

- [jsonpickle.pickler – Python to JSON-compatible dict](#)
 - [Pickler](#)
 - [Pickler.flatten\(\)](#)
 - [Pickler.reset\(\)](#)
 - [encode\(\)](#)
- [jsonpickle.unpickler – JSON-compatible dict to Python](#)
 - [Unpickler](#)
 - [Unpickler.register_classes\(\)](#)
 - [Unpickler.reset\(\)](#)
 - [Unpickler.restore\(\)](#)
 - [decode\(\)](#)
 - [getargs\(\)](#)
 - [has_tag\(\)](#)
 - [has_tag_dict\(\)](#)
 - [loadclass\(\)](#)
 - [loadrepr\(\)](#)
 - [make_blank_classic\(\)](#)
- [jsonpickle.backend – JSON Backend Management](#)
 - [JSONBackend](#)
 - [JSONBackend.decode\(\)](#)
 - [JSONBackend.dumps\(\)](#)
 - [JSONBackend.enable_fallthrough\(\)](#)
 - [JSONBackend.encode\(\)](#)
 - [JSONBackend.load_backend\(\)](#)
 - [JSONBackend.loads\(\)](#)
 - [JSONBackend.remove_backend\(\)](#)
 - [JSONBackend.set_decoder_options\(\)](#)
 - [JSONBackend.set_encoder_options\(\)](#)
 - [JSONBackend.set_preferred_backend\(\)](#)

[Skip to content](#)

- [jsonpickle.util – Helper functions](#)

- [b64decode\(\)](#)
- [b64encode\(\)](#)
- [b85decode\(\)](#)
- [b85encode\(\)](#)
- [has_method\(\)](#)
- [has_reduce\(\)](#)
- [importable_name\(\)](#)
- [in_dict\(\)](#)
- [in_slots\(\)](#)
- [is_bytes\(\)](#)
- [is_collections\(\)](#)
- [is_cython_function\(\)](#)
- [is_dictionary\(\)](#)
- [is_dictionary_subclass\(\)](#)
- [is_enum\(\)](#)
- [is_function\(\)](#)
- [is_installed\(\)](#)
- [is_iterator\(\)](#)
- [is_list\(\)](#)
- [is_list_like\(\)](#)
- [is_module\(\)](#)
- [is_module_function\(\)](#)
- [is_noncomplex\(\)](#)
- [is_not_class\(\)](#)
- [is_object\(\)](#)
- [is_picklable\(\)](#)
- [is_primitive\(\)](#)
- [is_readonly\(\)](#)
- [is_reducible\(\)](#)
- [is_reducible_sequence_subclass\(\)](#)
- [is_sequence\(\)](#)
- [is_sequence_subclass\(\)](#)
- [is_set\(\)](#)
- [is_tuple\(\)](#)
- [is_type\(\)](#)
- [is_unicode\(\)](#)

Skip to content

- [tter\(\)](#)
- [translate_module_name\(\)](#)

- `untranslate_module_name()`

Extensions

- [jsonpickle extensions](#)
 - [NumPy](#)
 - [Ecdsa](#)

Contributing

- [Contributing to jsonpickle](#)
 - [Get the Code](#)
 - [Run the Test Suite](#)
 - [Generate Documentation](#)

Contact

Please join our [mailing list](#). You can send email to jsonpickle@googlegroups.com.

Check <https://github.com/jsonpickle/jsonpickle> for project updates.

Authors

- John Paulett - john -at- paulett.org - <https://github.com/johnpaulett>
- David Aguilar - davvid -at- gmail.com - <https://github.com/davvid>
- Theelx - <https://github.com/Theelx>
- Dan Buch - <https://github.com/meatballhat>
- Ian Schenck - <https://github.com/ianschenck>
- David K. Hess - <https://github.com/davidkhess>
- Alec Thomas - <https://github.com/alecthomas>
- jaraco - <https://github.com/jaraco>
- Marcin Tustin - <https://github.com/marcintustin>

[Skip to content](#)

Change Log

- [History](#)
 - [Upcoming](#)
 - [v4.1.1](#)
 - [v4.1.0](#)
 - [v4.0.5](#)
 - [v4.0.4](#)
 - [v4.0.3](#)
 - [v4.0.2](#)
 - [v4.0.1](#)
 - [v4.0.0](#)
 - [v3.4.2](#)
 - [v3.4.1](#)
 - [v3.4.0](#)
 - [v3.3.0](#)
 - [v3.2.2](#)
 - [v3.2.1](#)
 - [v3.2.0](#)
 - [v3.1.0](#)
 - [v3.0.4](#)
 - [v3.0.3](#)
 - [v3.0.2](#)
 - [v3.0.1](#)
 - [v3.0.0](#)
 - [v2.2.0](#)
 - [v2.1.0](#)
 - [v2.0.0](#)
 - [v1.5.2](#)
 - [v1.5.1](#)
 - [v1.5.0](#)
 - [v1.4.2](#)
 - [v1.4.1](#)
 - [v1.4](#)
 - [v1.3](#)
 - [v1.2](#)
 - [v1.1](#)

[Skip to content](#)

- [v0.9.5](#)

- [v0.9.4](#)
- [v0.9.3](#)
- [v0.9.2](#)
- [v0.9.1](#)
- [v0.9.0](#)
- [v0.8.0](#)
- [v0.7.2](#)
- [v0.7.1](#)
- [v0.7.0](#)
- [v0.6.1](#)
- [v0.6.0](#)
- [v0.5.0](#)
- [0.4.0](#)
- [0.3.1](#)
- [0.3.0](#)
- [0.2.0](#)
- [0.1.0](#)
- [0.0.5](#)
- [0.0.3](#)
- [0.0.2](#)
- [0.0.1](#)

License

jsonpickle is provided under a [New BSD license](#),

Copyright (C) 2008-2011 John Paulett (john -at- paulett.org) Copyright (C) 2009-2021 David Aguilar (davvid -at- gmail.com)

Copyright ©

Made with [Sphinx](#) and @pradyunsg's [Furo](#)