
Vexy - Generate VEX in CDX

Release 0.3.1

Paul Horton

Mar 03, 2023

CONTENTS:

1	Installation	3
2	Usage	5
3	Configuration	7
3.1	Configuration File Format	7
4	Data Sources	9
4.1	OSS Index	9
4.2	OSV.dev	9
5	Support	11
5.1	Python Version Support	11
6	Changelog	13
6.1	v0.3.1 (2023-03-03)	13
6.2	v0.3.0 (2022-08-02)	13
6.3	v0.2.0 (2022-07-14)	13
6.4	v0.1.7 (2022-07-13)	14
6.5	v0.1.6 (2022-07-13)	14
6.6	v0.1.5 (2022-07-13)	14
6.7	v0.1.4 (2022-07-13)	14
6.8	v0.1.3 (2022-07-13)	14
6.9	v0.1.2 (2022-07-13)	15
6.10	v0.1.1 (2022-07-13)	15
6.11	v0.1.0 (2022-07-13)	15
7	API Reference	17
7.1	vexy	17
	Python Module Index	29
	Index	31

Software Bill of Materials (SBOMs) are gaining traction and are a great way to codify what dependencies your software relies on from the Open Source ecosystems (and internal libraries too!).

The SBOM for a given release of a given piece of software should be static in terms of the components that comprise that release.

[CycloneDX](#), in this authors view - the best Bill of Materials format, also allows for [Vulnerability Exploitability Exchange](#) (or VEX) information to be included in your BOM.

Known vulnerabilities change over time - we always know more about the security posture of Open Source components tomorrow than we did today. So how do we keep our BOMs updated with this information?

[CycloneDX](#) also allows for BOMs to interlink for the above reason. The best way to manage this scenario is to generate a BOM that describes your software release, excluding VEX data, and then have a tool (perhaps *vexy*?) produce you a VEX document (in [CycloneDX](#) format) that links back to your SBOM.

Did I confuse you? If so - read more about [Independent BOM and VEX here](#).

INSTALLATION

Install from pypi.org as you would any other Python module using your preferred package manager:

```
pip install vxy
```


USAGE

vexy is designed to be run as a standalone application.

Once installed, you can call the tool via the following methods:

```
$ python3 -m vexy
$ vexy
```

The full documentation can be issued by running with `--help`:

```
$ vexy --help
usage: vexy [-h] -c VEXY_CONFIG [-q] [-X] -i FILE_PATH [--format {xml,json}] [--schema-
↪version {1.4}] [-o FILE_PATH] [--force]

Vexy VEX Generator

options:
  -h, --help            show this help message and exit
  -c VEXY_CONFIG, --config VEXY_CONFIG
                        Configuration file for Vexy defining data sources to use and
↪their configuration.
  -q                    Quiet - no console output
  -X                    Enable debug output

Input CycloneDX BOM:
  Where Vexy shall obtain its input

  -i FILE_PATH, --in-file FILE_PATH
                        CycloneDX BOM to read input from. Use "-" to read from STDIN.

VEX Output Configuration:
  Choose the output format and schema version

  --format {xml,json}  The output format for your SBOM (default: xml)
  --schema-version {1.4}
                        The CycloneDX schema version for your VEX (default: 1.4)
  -o FILE_PATH, --o FILE_PATH, --output FILE_PATH
                        Output file path for your SBOM (set to '-' to output to STDOUT)
  --force              If outputting to a file and the stated file already exists, it
↪will be overwritten.
```


CONFIGURATION

Vexy will query the data sources you configured to obtain current known vulnerability information that relates to the Open Source components included in your input SBOM.

Data Sources are configured in a YAML formatted file which is supplied to `vexy` using the `-c` or `--config` flag at the command line. A configuration file **MUST** be supplied to run `vexy`.

3.1 Configuration File Format

Currently, the configuration file is used only to describe which data sources you would like `vexy` to utilise and any configuration that datasource requires - e.g. authentication details.

An example configuration file might look as follows:

```
sources:
  ossindex:
    username: <your-username>
    password: <your-password>
  osv:
```

For details of what data sources are available and their specific configuration - see [Data Sources](#).

DATA SOURCES

4.1 OSS Index

- See <https://ossindex.sonatype.org>
- Supports authentication: ✓
- Requires authentication:

4.1.1 Configuration

```
sources:  
  ossindex:  
    username: <your-username>  
    password: <your-password>
```

4.2 OSV.dev

- See <https://osv.dev/>
- Supports authentication:
- Requires authentication:

4.2.1 Configuration

```
sources:  
  osv:
```


SUPPORT

If you run into issues utilising this library, please raise a [GitHub Issue](#). When raising an issue please include as much detail as possible including:

- Version of **vexy** you have installed
- Input(s)
- Expected Output(s)
- Actual Output(s)

5.1 Python Version Support

We endeavour to support all functionality for all [current actively supported Python versions](#). However, some features may not be possible/present in older Python versions due to their lack of support - which are noted below.

CHANGELOG

6.1 v0.3.1 (2023-03-03)

6.1.1 Fix

- Handle parsing of BOM more safely (``39d6f78`` <<https://github.com/madpah/vexy/commit/39d6f78fd517d4cfe53fa07214f69947762d71a0>>`_)
- Update to latest contracts with `cyclonedx-python-lib` (``fda01e0`` <<https://github.com/madpah/vexy/commit/fda01e047717dc1bf952f07cddd5ad2d551e9e35>>`_)
- Bump to latest rc of `cyclonedx-python-lib` (``8e3d05a`` <<https://github.com/madpah/vexy/commit/8e3d05a8cc0d1d50d317f7f1eb5dbceb6fe093f1>>`_)
- Handle parsing of BOM more safely (``a89862a`` <<https://github.com/madpah/vexy/commit/a89862a039ad918f278a57b64197fa9009fa28e0>>`_)

6.2 v0.3.0 (2022-08-02)

6.2.1 Feature

- Added OSV.dev as data source (``402c669`` <<https://github.com/madpah/vexy/commit/402c669ab3a07a7ca485e860635504789107a0f5>>`_)

6.3 v0.2.0 (2022-07-14)

6.3.1 Feature

- Add `vexy` as a Tool to generated VEX documents (``f2378a8`` <<https://github.com/madpah/vexy/commit/f2378a820b88a6ee10036d4f771b5dd0e11925cb>>`_)
- Add `vexy` as a Tool to generated VEX documents (``70ea250`` <<https://github.com/madpah/vexy/commit/70ea250609ed8bf673637483691406d6b56f9dd8>>`_)

6.3.2 Fix

- Disable mypy warn_unused_ignores to get mypy passing on all variants (``e331e72`` <<https://github.com/madpah/vexy/commit/e331e72aac0002543066151841bbbeb661d5be97>>`_`)

6.4 v0.1.7 (2022-07-13)

6.4.1 Fix

- Use a known working release pipeline (``c51e613`` <<https://github.com/madpah/vexy/commit/c51e6132f5a653385486eda5efa54faece7719e7>>`_`)

6.5 v0.1.6 (2022-07-13)

6.5.1 Fix

- Release CI pipeline syntax error (``b5b8529`` <<https://github.com/madpah/vexy/commit/b5b852955810082009a7c308f91d4a1284aa6368>>`_`)

6.6 v0.1.5 (2022-07-13)

6.6.1 Fix

- Release CI pipeline syntax error (``c0c7846`` <<https://github.com/madpah/vexy/commit/c0c78461c2e288825214640300917edfe24cb04f>>`_`)

6.7 v0.1.4 (2022-07-13)

6.7.1 Fix

- Remove date parsing from source BOM (``3d1e0d9`` <<https://github.com/madpah/vexy/commit/3d1e0d94917df6b4b32da06900c846e771720689>>`_`)

6.8 v0.1.3 (2022-07-13)

6.8.1 Fix

- Remove parsing of input BOM timestamp - we do not use it (``8b40f70`` <<https://github.com/madpah/vexy/commit/8b40f70487f20c4e21f72ed329330226082a31f3>>`_`)

6.9 v0.1.2 (2022-07-13)

6.9.1 Fix

- Pin ci to use `python-semantic-release@v7.28.1` as newer breaks CI (``d1a1fe6`` <<https://github.com/madpah/vexy/commit/d1a1fe6f221fc9f557828188613c0e329a19a881>>`_)

6.10 v0.1.1 (2022-07-13)

6.10.1 Fix

- Typing broke some use cases (``5965816`` <<https://github.com/madpah/vexy/commit/59658165a2789b59d93a0e3844b35b5c5fe303dd>>`_)
- Typing broke some use cases (``c766507`` <<https://github.com/madpah/vexy/commit/c766507bcc5a84f61b7371ba8dd1bc51526a0a77>>`_)

6.11 v0.1.0 (2022-07-13)

6.11.1 Feature

- First alpha release supporting OSS Index as the sole data source (``650bf52`` <<https://github.com/madpah/vexy/commit/650bf521675524d7869ebc1b8d0ccc0d2175aab7>>`_)

API REFERENCE

This page contains auto-generated API reference documentation¹.

7.1 vaxy

7.1.1 Subpackages

`vaxy.sources`

Submodules

`vaxy.sources.base`

Module Contents

Classes

<i>BaseSource</i>	Helper class that provides a standard way to create an ABC using
-------------------	--

class `vaxy.sources.base.BaseSource`(***, *config*: *Dict[str, Any] | None = None*)

Bases: `abc.ABC`

Helper class that provides a standard way to create an ABC using inheritance.

property `all_components`: `Set[cyclonedx.model.component.Component]`

property `valid_components`: `Set[cyclonedx.model.component.Component]`

process_components(***, *components*: *Iterable[cyclonedx.model.component.Component]*) \rightarrow `None`

abstract `get_vulnerabilities`() \rightarrow `Set[cyclonedx.model.vulnerability.Vulnerability]`

abstract `_component_complete_for_source`(***, *component*: *cyclonedx.model.component.Component*)
 \rightarrow `bool`

Whether the given Component has enough data (the right fields) for us to query this data source for known vulnerabilities.

¹ Created with sphinx-autoapi

Parameters

component – Component

Returns

bool

abstract _configure_source(*, *config*: Dict[str, Any]) → None

Perform any source specific configuration such as authentication.

Parameters

config – Dict[str, Any]

Returns

None

abstract static source() → cyclonedx.model.vulnerability.VulnerabilitySource

Instance that represents this data source.

Returns

VulnerabilitySource

abstract static source_name() → str

Human-friendly name for this data source.

Returns

str

abstract static source_description() → str

Human-friendly description of this data source.

Returns

str

abstract static source_ecosystems() → Set[vexy.EcoSystem]

Which ecosystems this source has vulnerability data for.

Returns

Set[str]

abstract static source_url() → str

Public URL for this data source

Returns

str

vexy.sources.ossindex

Module Contents

Classes

OssIndexSource

Helper class that provides a standard way to create an ABC using

class vexy.sources.ossindex.OssIndexSource(*, config: Dict[str, Any] | None = None)

Bases: [vexy.sources.base.BaseSource](#)

Helper class that provides a standard way to create an ABC using inheritance.

_component_complete_for_source(component: cyclonedx.model.component.Component) → bool

Whether the given Component has enough data (the right fields) for us to query this data source for known vulnerabilities.

Parameters

component – Component

Returns

bool

_configure_source(config: Dict[str, Any]) → None

Perform any source specific configuration such as authentication.

Parameters

config – Dict[str, Any]

Returns

None

get_vulnerabilities() → Set[cyclonedx.model.vulnerability.Vulnerability]

static source() → cyclonedx.model.vulnerability.VulnerabilitySource

Instance that represents this data source.

Returns

VulnerabilitySource

static source_name() → str

Human-friendly name for this data source.

Returns

str

static source_description() → str

Human-friendly description of this data source.

Returns

str

static source_ecosystems() → Set[vexy.EcoSystem]

Which ecosystems this source has vulnerability data for.

Returns

Set[str]

static source_url() → str

Public URL for this data source

Returns

str

`vexy.sources.osv`

Module Contents

Classes

<i>OsvSource</i>	Helper class that provides a standard way to create an ABC using
------------------	--

class `vexy.sources.osv.OsvSource(*, config: Dict[str, Any] | None = None)`

Bases: `vexy.sources.base.BaseSource`

Helper class that provides a standard way to create an ABC using inheritance.

get_vulnerabilities() → Set[cyclonedx.model.vulnerability.Vulnerability]

_component_complete_for_source(*, component: cyclonedx.model.component.Component) → bool

Whether the given Component has enough data (the right fields) for us to query this data source for known vulnerabilities.

Parameters

component – Component

Returns

bool

_configure_source(*, config: Dict[str, Any]) → None

Perform any source specific configuration such as authentication.

Parameters

config – Dict[str, Any]

Returns

None

static source() → cyclonedx.model.vulnerability.VulnerabilitySource

Instance that represents this data source.

Returns

VulnerabilitySource

static source_name() → str

Human-friendly name for this data source.

Returns

str

static source_description() → str

Human-friendly description of this data source.

Returns

str

static source_ecosystems() → Set[vexy.EcoSystem]

Which ecosystems this source has vulnerability data for.

Returns

Set[str]

static source_url() → str

Public URL for this data source

Returns

str

`vexy.sources.osvdb`

Module Contents

Classes

OsvDbSource

Helper class that provides a standard way to create an ABC using

class `vexy.sources.osvdb.OsvDbSource`(*, *config*: Dict[str, Any] | None = None)

Bases: `vexy.sources.base.BaseSource`

Helper class that provides a standard way to create an ABC using inheritance.

get_vulnerabilities() → Set[cyclonedx.model.vulnerability.Vulnerability]

_component_complete_for_source(*, *component*: cyclonedx.model.component.Component) → bool

Whether the given Component has enough data (the right fields) for us to query this data source for known vulnerabilities.

Parameters

component – Component

Returns

bool

_configure_source(*, *config*: Dict[str, Any]) → None

Perform any source specific configuration such as authentication.

Parameters

config – Dict[str, Any]

Returns

None

static source() → cyclonedx.model.vulnerability.VulnerabilitySource

Instance that represents this data source.

Returns

VulnerabilitySource

static source_name() → str

Human-friendly name for this data source.

Returns

str

static source_description() → str

Human-friendly description of this data source.

Returns

str

static source_ecosystems() → Set[vexy.EcoSystem]

Which ecosystems this source has vulnerability data for.

Returns

Set[str]

static source_url() → str

Public URL for this data source

Returns

str

Package Contents

Classes

<i>BaseSource</i>	Helper class that provides a standard way to create an ABC using
<i>OssIndexSource</i>	Helper class that provides a standard way to create an ABC using
<i>OsvSource</i>	Helper class that provides a standard way to create an ABC using

Attributes

<i>ALL_SOURCES</i>

class vexy.sources.**BaseSource**(**, config: Dict[str, Any] | None = None*)

Bases: abc.ABC

Helper class that provides a standard way to create an ABC using inheritance.

property all_components: Set[cyclonedx.model.component.Component]

property valid_components: Set[cyclonedx.model.component.Component]

process_components(**, components: Iterable[cyclonedx.model.component.Component]*) → None

abstract get_vulnerabilities() → Set[cyclonedx.model.vulnerability.Vulnerability]

abstract _component_complete_for_source(**, component: cyclonedx.model.component.Component*)
→ bool

Whether the given Component has enough data (the right fields) for us to query this data source for known vulnerabilities.

Parameters

component – Component

Returns

bool

abstract **_configure_source**(*, *config: Dict[str, Any]*) → None

Perform any source specific configuration such as authentication.

Parameters**config** – Dict[str, Any]**Returns**

None

abstract static source() → cyclonedx.model.vulnerability.VulnerabilitySource

Instance that represents this data source.

Returns

VulnerabilitySource

abstract static source_name() → str

Human-friendly name for this data source.

Returns

str

abstract static source_description() → str

Human-friendly description of this data source.

Returns

str

abstract static source_ecosystems() → Set[vexy.EcoSystem]

Which ecosystems this source has vulnerability data for.

Returns

Set[str]

abstract static source_url() → str

Public URL for this data source

Returns

str

class vexy.sources.OssIndexSource(*, *config: Dict[str, Any] | None = None*)Bases: [vexy.sources.base.BaseSource](#)

Helper class that provides a standard way to create an ABC using inheritance.

_component_complete_for_source(*component: cyclonedx.model.component.Component*) → bool

Whether the given Component has enough data (the right fields) for us to query this data source for known vulnerabilities.

Parameters**component** – Component**Returns**

bool

_configure_source(*config: Dict[str, Any]*) → None

Perform any source specific configuration such as authentication.

Parameters**config** – Dict[str, Any]

Returns

None

get_vulnerabilities() → Set[cyclonedx.model.vulnerability.Vulnerability]

static source() → cyclonedx.model.vulnerability.VulnerabilitySource

Instance that represents this data source.

Returns

VulnerabilitySource

static source_name() → str

Human-friendly name for this data source.

Returns

str

static source_description() → str

Human-friendly description of this data source.

Returns

str

static source_ecosystems() → Set[vexy.EcoSystem]

Which ecosystems this source has vulnerability data for.

Returns

Set[str]

static source_url() → str

Public URL for this data source

Returns

str

class vexy.sources.OsvSource(*, config: Dict[str, Any] | None = None)

Bases: [vexy.sources.base.BaseSource](#)

Helper class that provides a standard way to create an ABC using inheritance.

get_vulnerabilities() → Set[cyclonedx.model.vulnerability.Vulnerability]

_component_complete_for_source(*, component: cyclonedx.model.component.Component) → bool

Whether the given Component has enough data (the right fields) for us to query this data source for known vulnerabilities.

Parameters

component – Component

Returns

bool

_configure_source(*, config: Dict[str, Any]) → None

Perform any source specific configuration such as authentication.

Parameters

config – Dict[str, Any]

Returns

None

static source() → cyclonedx.model.vulnerability.VulnerabilitySource

Instance that represents this data source.

Returns

VulnerabilitySource

static source_name() → str

Human-friendly name for this data source.

Returns

str

static source_description() → str

Human-friendly description of this data source.

Returns

str

static source_ecosystems() → Set[vexy.EcoSystem]

Which ecosystems this source has vulnerability data for.

Returns

Set[str]

static source_url() → str

Public URL for this data source

Returns

str

vexy.sources.ALL_SOURCES: Dict[str, Type[base.BaseSource]]

7.1.2 Submodules

vexy.__main__

vexy.client

Module Contents

Classes

_CLI_OUTPUT_FORMAT

Generic enumeration.

VexyCmd

Functions

```
main(→ None)
```

Attributes

```
_output_formats
```

```
_output_default_filenames
```

```
__ThisToolVersion
```

```
ThisTool
```

```
class vexy.client._CLI_OUTPUT_FORMAT
```

```
    Bases: enum.Enum
```

```
    Generic enumeration.
```

```
    Derive from this class to define new enumerations.
```

```
    XML = 'xml'
```

```
    JSON = 'json'
```

```
vexy.client._output_formats: Dict[_CLI_OUTPUT_FORMAT, cyclonedx.schema.OutputFormat]
```

```
vexy.client._output_default_filenames
```

```
vexy.client.__ThisToolVersion: str | None
```

```
vexy.client.ThisTool
```

```
class vexy.client.VexyCmd(args: argparse.Namespace)
```

```
    DEFAULT_CONFIG_FILE: str = '.vexy.config'
```

```
    _DEBUG_ENABLED: bool = False
```

```
    _arguments: argparse.Namespace
```

```
    _attempt_source_config_load(config: io.TextIOWrapper) → None
```

```
    get_cli_output_format() → _CLI_OUTPUT_FORMAT
```

```
    _get_output_format() → cyclonedx.schema.OutputFormat
```

```
    _is_quiet() → bool
```

```
    execute() → None
```

```

_get_outputter(output_format: cyclonedx.schema.OutputFormat, bom: cyclonedx.model.bom.Bom) →
    cyclonedx.output.BaseOutput

static get_arg_parser(*, prog: str | None = None) → argparse.ArgumentParser

_debug_message(message: str) → None

static _error_and_exit(message: str, exit_code: int = 1) → None

vexy.client.main(*, prog_name: str | None = None) → None

```

7.1.3 Package Contents

Classes

<i>EcoSystemType</i>	
<i>EcoSystem</i>	Languages/ecosystems to the PURL type

Attributes

<i>_ALL_ECOSYSTEMS</i>

```

class vexy.EcoSystemType(*, name: str, purl_type: str, description: str)

    property name: str
    property purl_type: str
    property description: str

vexy._ALL_ECOSYSTEMS

class vexy.EcoSystem
    Bases: enum.Enum

    Languages/ecosystems to the PURL type

    Starting list taken from https://github.com/package-url/purl-spec/blob/master/PURL-TYPES.rst

    BITBUCKET = 'BITBUCKET'

    CARGO = 'CARGO'

    COCOAPODS = 'COCOAPODS'

    COMPOSER = 'COMPOSER'

    CONAN = 'CONAN'

    CONDA = 'CONDA'

```

```
CRAN = 'CRAN'
DART = 'PUB'
DEBIAN = 'DEB'
DOCKER = 'DOCKER'
FLUTTER = 'PUB'
GENERIC = 'GENERIC'
GITHUB = 'GITHUB'
GO = 'GOLANG'
HASKELL = 'HACKAGE'
HEX = 'HEX'
MAVEN = 'MAVEN'
NPM = 'NPM'
NUGET = 'NUGET'
OCI = 'OCI'
PYPI = 'PYPI'
RPM = 'RPM'
RUBY_GEM = 'GEM'
SWIFT = 'SWIFT'
get_info() → EcoSystemType
```


PYTHON MODULE INDEX

V

- [vexy](#), 17
- [vexy.__main__](#), 25
- [vexy.client](#), 25
- [vexy.sources](#), 17
 - [vexy.sources.base](#), 17
 - [vexy.sources.ossindex](#), 18
 - [vexy.sources.osv](#), 20
 - [vexy.sources.osvdb](#), 21

Symbols

_ALL_ECOSYSTEMS (in module *vexy*), 27
 _CLI_OUTPUT_FORMAT (class in *vexy.client*), 26
 _DEBUG_ENABLED (*vexy.client.VexyCmd* attribute), 26
 __ThisToolVersion (in module *vexy.client*), 26
 _arguments (*vexy.client.VexyCmd* attribute), 26
 _attempt_source_config_load()
 (*vexy.client.VexyCmd* method), 26
 _component_complete_for_source()
 (*vexy.sources.BaseSource* method), 22
 _component_complete_for_source()
 (*vexy.sources.OssIndexSource* method), 23
 _component_complete_for_source()
 (*vexy.sources.OsvSource* method), 24
 _component_complete_for_source()
 (*vexy.sources.base.BaseSource* method),
 17
 _component_complete_for_source()
 (*vexy.sources.ossindex.OssIndexSource*
 method), 19
 _component_complete_for_source()
 (*vexy.sources.osv.OsvSource* method), 20
 _component_complete_for_source()
 (*vexy.sources.osvdb.OsvDbSource* method), 21
 _configure_source() (vexy.sources.BaseSource
 method), 23
 _configure_source() (vexy.sources.OssIndexSource
 method), 23
 _configure_source() (vexy.sources.OsvSource
 method), 24
 _configure_source() (vexy.sources.base.BaseSource
 method), 18
 _configure_source()
 (*vexy.sources.ossindex.OssIndexSource*
 method), 19
 _configure_source() (vexy.sources.osv.OsvSource
 method), 20
 _configure_source()
 (*vexy.sources.osvdb.OsvDbSource* method), 21
 _debug_message() (*vexy.client.VexyCmd* method), 27
 _error_and_exit() (*vexy.client.VexyCmd* static
 method), 27

_get_output_format() (*vexy.client.VexyCmd* method),
 26
 _get_outputter() (*vexy.client.VexyCmd* method), 26
 _is_quiet() (*vexy.client.VexyCmd* method), 26
 _output_default_filenames (in module *vexy.client*),
 26
 _output_formats (in module *vexy.client*), 26

A

all_components (*vexy.sources.base.BaseSource* prop-
 erty), 17
 all_components (*vexy.sources.BaseSource* property),
 22
 ALL_SOURCES (in module *vexy.sources*), 25

B

BaseSource (class in *vexy.sources*), 22
 BaseSource (class in *vexy.sources.base*), 17
 BITBUCKET (*vexy.EcoSystem* attribute), 27

C

CARGO (*vexy.EcoSystem* attribute), 27
 COCOAPODS (*vexy.EcoSystem* attribute), 27
 COMPOSER (*vexy.EcoSystem* attribute), 27
 CONAN (*vexy.EcoSystem* attribute), 27
 CONDA (*vexy.EcoSystem* attribute), 27
 CRAN (*vexy.EcoSystem* attribute), 27

D

DART (*vexy.EcoSystem* attribute), 28
 DEBIAN (*vexy.EcoSystem* attribute), 28
 DEFAULT_CONFIG_FILE (*vexy.client.VexyCmd* attribute),
 26
 description (*vexy.EcoSystemType* property), 27
 DOCKER (*vexy.EcoSystem* attribute), 28

E

EcoSystem (class in *vexy*), 27
 EcoSystemType (class in *vexy*), 27
 execute() (*vexy.client.VexyCmd* method), 26

F

FLUTTER (*vexy.EcoSystem* attribute), 28

G

GENERIC (*vexy.EcoSystem* attribute), 28

get_arg_parser() (*vexy.client.VexyCmd* static method), 27

get_cli_output_format() (*vexy.client.VexyCmd* method), 26

get_info() (*vexy.EcoSystem* method), 28

get_vulnerabilities() (*vexy.sources.base.BaseSource* method), 17

get_vulnerabilities() (*vexy.sources.BaseSource* method), 22

get_vulnerabilities() (*vexy.sources.ossindex.OssIndexSource* method), 19

get_vulnerabilities() (*vexy.sources.OssIndexSource* method), 24

get_vulnerabilities() (*vexy.sources.osv.OsvSource* method), 20

get_vulnerabilities() (*vexy.sources.osvdb.OsvDbSource* method), 21

get_vulnerabilities() (*vexy.sources.OsvSource* method), 24

GITHUB (*vexy.EcoSystem* attribute), 28

GO (*vexy.EcoSystem* attribute), 28

H

HASKELL (*vexy.EcoSystem* attribute), 28

HEX (*vexy.EcoSystem* attribute), 28

J

JSON (*vexy.client._CLI_OUTPUT_FORMAT* attribute), 26

M

main() (in module *vexy.client*), 27

MAVEN (*vexy.EcoSystem* attribute), 28

module

vexy, 17

vexy.__main__, 25

vexy.client, 25

vexy.sources, 17

vexy.sources.base, 17

vexy.sources.ossindex, 18

vexy.sources.osv, 20

vexy.sources.osvdb, 21

N

name (*vexy.EcoSystemType* property), 27

NPM (*vexy.EcoSystem* attribute), 28

NUGET (*vexy.EcoSystem* attribute), 28

O

OCI (*vexy.EcoSystem* attribute), 28

OssIndexSource (class in *vexy.sources*), 23

OssIndexSource (class in *vexy.sources.ossindex*), 18

OsvDbSource (class in *vexy.sources.osvdb*), 21

OsvSource (class in *vexy.sources*), 24

OsvSource (class in *vexy.sources.osv*), 20

P

process_components() (*vexy.sources.base.BaseSource* method), 17

process_components() (*vexy.sources.BaseSource* method), 22

purl_type (*vexy.EcoSystemType* property), 27

PYPI (*vexy.EcoSystem* attribute), 28

R

RPM (*vexy.EcoSystem* attribute), 28

RUBY_GEM (*vexy.EcoSystem* attribute), 28

S

source() (*vexy.sources.base.BaseSource* static method), 18

source() (*vexy.sources.BaseSource* static method), 23

source() (*vexy.sources.ossindex.OssIndexSource* static method), 19

source() (*vexy.sources.OssIndexSource* static method), 24

source() (*vexy.sources.osv.OsvSource* static method), 20

source() (*vexy.sources.osvdb.OsvDbSource* static method), 21

source() (*vexy.sources.OsvSource* static method), 24

source_description() (*vexy.sources.base.BaseSource* static method), 18

source_description() (*vexy.sources.BaseSource* static method), 23

source_description() (*vexy.sources.ossindex.OssIndexSource* static method), 19

source_description() (*vexy.sources.OssIndexSource* static method), 24

source_description() (*vexy.sources.osv.OsvSource* static method), 20

source_description() (*vexy.sources.osvdb.OsvDbSource* static method), 21

source_description() (*vexy.sources.OsvSource* static method), 25

[source_ecosystems\(\)](#) (*vexy.sources.base.BaseSource static method*), 18
[source_ecosystems\(\)](#) (*vexy.sources.BaseSource static method*), 23
[source_ecosystems\(\)](#) (*vexy.sources.ossindex.OssIndexSource static method*), 19
[source_ecosystems\(\)](#) (*vexy.sources.OssIndexSource static method*), 24
[source_ecosystems\(\)](#) (*vexy.sources.osv.OsvSource static method*), 20
[source_ecosystems\(\)](#) (*vexy.sources.osvdb.OsvDbSource static method*), 22
[source_ecosystems\(\)](#) (*vexy.sources.OsvSource static method*), 25
[source_name\(\)](#) (*vexy.sources.base.BaseSource static method*), 18
[source_name\(\)](#) (*vexy.sources.BaseSource static method*), 23
[source_name\(\)](#) (*vexy.sources.ossindex.OssIndexSource static method*), 19
[source_name\(\)](#) (*vexy.sources.OssIndexSource static method*), 24
[source_name\(\)](#) (*vexy.sources.osv.OsvSource static method*), 20
[source_name\(\)](#) (*vexy.sources.osvdb.OsvDbSource static method*), 21
[source_name\(\)](#) (*vexy.sources.OsvSource static method*), 25
[source_url\(\)](#) (*vexy.sources.base.BaseSource static method*), 18
[source_url\(\)](#) (*vexy.sources.BaseSource static method*), 23
[source_url\(\)](#) (*vexy.sources.ossindex.OssIndexSource static method*), 19
[source_url\(\)](#) (*vexy.sources.OssIndexSource static method*), 24
[source_url\(\)](#) (*vexy.sources.osv.OsvSource static method*), 21
[source_url\(\)](#) (*vexy.sources.osvdb.OsvDbSource static method*), 22
[source_url\(\)](#) (*vexy.sources.OsvSource static method*), 25
[SWIFT](#) (*vexy.EcoSystem attribute*), 28

T

[ThisTool](#) (*in module vexy.client*), 26

V

[valid_components](#) (*vexy.sources.base.BaseSource property*), 17
[valid_components](#) (*vexy.sources.BaseSource property*), 22

vexy
[module](#), 17
[vexy.__main__](#)
[module](#), 25
[vexy.client](#)
[module](#), 25
[vexy.sources](#)
[module](#), 17
[vexy.sources.base](#)
[module](#), 17
[vexy.sources.ossindex](#)
[module](#), 18
[vexy.sources.osv](#)
[module](#), 20
[vexy.sources.osvdb](#)
[module](#), 21
[VexyCmd](#) (*class in vexy.client*), 26

X

[XML](#) (*vexy.client._CLI_OUTPUT_FORMAT attribute*), 26