

An `term` abstract base class to provide the ability to read *resources*.

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]importlib.resources.abc.rst, line 5); [backlink](#)

Unknown interpreted text role "term".

From the perspective of this ABC, a *resource* is a binary artifact that is shipped within a package. Typically this is something like a data file that lives next to the `__init__.py` file of the package. The purpose of this class is to help abstract out the accessing of such data files so that it does not matter if the package and its data file(s) are stored in a e.g. zip file versus on the file system.

For any of methods of this class, a *resource* argument is expected to be a `term` *path-like object* which represents conceptually just a file name. This means that no subdirectory paths should be included in the *resource* argument. This is because the location of the package the reader is for, acts as the "directory". Hence the metaphor for directories and file names is packages and resources, respectively. This is also why instances of this class are expected to directly correlate to a specific package (instead of potentially representing multiple packages or a module).

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]importlib.resources.abc.rst, line 16); [backlink](#)

Unknown interpreted text role "term".

Loaders that wish to support resource reading are expected to provide a method called `get_resource_reader(fullname)` which returns an object implementing this ABC's interface. If the module specified by `fullname` is not a package, this method should return `const:None`. An object compatible with this ABC should only be returned when the specified module is a package.

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]importlib.resources.abc.rst, line 27); [backlink](#)

Unknown interpreted text role "const".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]importlib.resources.abc.rst, line 34)

Unknown directive type "versionadded".

```
.. versionadded:: 3.7
```

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]importlib.resources.abc.rst, line 36)

Unknown directive type "abstractmethod".

```
.. abstractmethod:: open_resource(resource)

    Returns an opened, :term:`file-like object` for binary reading
    of the *resource*.

    If the resource cannot be found, :exc:`FileNotFoundError` is
    raised.
```

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]importlib.resources.abc.rst, line 44)

Unknown directive type "abstractmethod".

```
.. abstractmethod:: resource_path(resource)

    Returns the file system path to the *resource*.

    If the resource does not concretely exist on the file system,
    raise :exc:`FileNotFoundError`.
```

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]importlib.resources.abc.rst, line 51)

Unknown directive type "abstractmethod".

```
.. abstractmethod:: is_resource(name)
```

Returns ``True`` if the named \*name\* is considered a resource.  
:exc:`FileNotFoundError` is raised if \*name\* does not exist.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]importlib.resources.abc.rst, line 56)**

Unknown directive type "abstractmethod".

```
.. abstractmethod:: contents()
```

Returns an :term:`iterable` of strings over the contents of the package. Do note that it is not required that all names returned by the iterator be actual resources, e.g. it is acceptable to return names for which :meth:`is\_resource` would be false.

Allowing non-resource names to be returned is to allow for situations where how a package and its resources are stored are known a priori and the non-resource names would be useful. For instance, returning subdirectory names is allowed so that when it is known that the package and resources are stored on the file system then those subdirectory names can be used directly.

The abstract method returns an iterable of no items.

An abstract base class for a `term`loader`` which implements the optional [PEP 302](#) protocol for loading arbitrary resources from the storage back-end.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]importlib.resources.abc.rst, line 77); [backlink](#)**

Unknown interpreted text role "term".

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]importlib.resources.abc.rst, line 81)**

Unknown directive type "deprecated".

```
.. deprecated:: 3.7
```

This ABC is deprecated in favour of supporting resource loading through :class:`importlib.abc.ResourceReader`.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]importlib.resources.abc.rst, line 85)**

Unknown directive type "abstractmethod".

```
.. abstractmethod:: get_data(path)
```

An abstract method to return the bytes for the data located at \*path\*. Loaders that have a file-like storage back-end that allows storing arbitrary data can implement this abstract method to give direct access to the data stored. :exc:`OSError` is to be raised if the \*path\* cannot be found. The \*path\* is expected to be constructed using a module's :attr:`\_\_file\_\_` attribute or an item from a package's :attr:`\_\_path\_\_`.

```
.. versionchanged:: 3.4
```

Raises :exc:`OSError` instead of :exc:`NotImplementedError`.

An abstract base class for a `term`loader`` which implements the optional [PEP 302](#) protocol for loaders that inspect modules.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]importlib.resources.abc.rst, line 101); [backlink](#)**

Unknown interpreted text role "term".

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]importlib.resources.abc.rst, line 104)**

Unknown directive type "method".

```
.. method:: get_code(fullname)
```

Return the code object for a module, or ``None`` if the module does not have a code object (as would be the case, for example, for a built-in module). Raise an :exc:`ImportError` if loader cannot find the requested module.

```
.. note::
```

While the method has a default implementation, it is suggested that it be overridden if possible for performance.

```
.. index::
```

single: universal newlines; importlib.abc.InspectLoader.get\_source method

```
.. versionchanged:: 3.4
```

No longer abstract and a concrete implementation is provided.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]importlib.resources.abc.rst, line 121)**

Unknown directive type "abstractmethod".

```
.. abstractmethod:: get_source(fullname)
```

An abstract method to return the source of a module. It is returned as a text string using :term:`universal newlines`, translating all recognized line separators into ``'\n'`` characters. Returns ``None`` if no source is available (e.g. a built-in module). Raises :exc:`ImportError` if the loader cannot find the module specified.

```
.. versionchanged:: 3.4
```

Raises :exc:`ImportError` instead of :exc:`NotImplementedError`.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]importlib.resources.abc.rst, line 132)**

Unknown directive type "method".

```
.. method:: is_package(fullname)
```

An optional method to return a true value if the module is a package, a false value otherwise. :exc:`ImportError` is raised if the :term:`loader` cannot find the module.

```
.. versionchanged:: 3.4
```

Raises :exc:`ImportError` instead of :exc:`NotImplementedError`.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]importlib.resources.abc.rst, line 141)**

Unknown directive type "staticmethod".

```
.. staticmethod:: source_to_code(data, path='<string>')
```

Create a code object from Python source.

The \*data\* argument can be whatever the :func:`compile` function supports (i.e. string or bytes). The \*path\* argument should be the "path" to where the source code originated from, which can be an abstract concept (e.g. location in a zip file).

With the subsequent code object one can execute it in a module by running ``exec(code, module.\_\_dict\_\_)``.

```
.. versionadded:: 3.4
```

```
.. versionchanged:: 3.5
```

Made the method static.

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]importlib.resources.abc.rst, line 158)

Unknown directive type "method".

```
.. method:: exec_module(module)

Implementation of :meth:`Loader.exec_module`.

.. versionadded:: 3.4
```

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]importlib.resources.abc.rst, line 164)

Unknown directive type "method".

```
.. method:: load_module(fullname)

Implementation of :meth:`Loader.load_module`.

.. deprecated:: 3.4
   use :meth:`exec_module` instead.
```

An abstract base class which inherits from :class:`InspectLoader` that, when implemented, helps a module to be executed as a script. The ABC represents an optional [PEP 302](#) protocol.

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]importlib.resources.abc.rst, line 174);

[backlink](#)

Unknown interpreted text role "class".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]importlib.resources.abc.rst, line 178)

Unknown directive type "abstractmethod".

```
.. abstractmethod:: get_filename(fullname)

An abstract method that is to return the value of :attr:`__file__` for
the specified module. If no path is available, :exc:`ImportError` is
raised.

If source code is available, then the method should return the path to
the source file, regardless of whether a bytecode was used to load the
module.

.. versionchanged:: 3.4
   Raises :exc:`ImportError` instead of :exc:`NotImplementedError`.
```

An abstract base class which inherits from :class:`ResourceLoader` and :class:`ExecutionLoader`, providing concrete implementations of :meth:`ResourceLoader.get\_data` and :meth:`ExecutionLoader.get\_filename`.

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]importlib.resources.abc.rst, line 194);

[backlink](#)

Unknown interpreted text role "class".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]importlib.resources.abc.rst, line 194);

[backlink](#)

Unknown interpreted text role "class".

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]importlib.resources.abc.rst, line 194);**  
[backlink](#)

Unknown interpreted text role "meth".

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]importlib.resources.abc.rst, line 194);**  
[backlink](#)

Unknown interpreted text role "meth".

The *fullname* argument is a fully resolved name of the module the loader is to handle. The *path* argument is the path to the file for the module.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]importlib.resources.abc.rst, line 201)**

Unknown directive type "versionadded".

```
.. versionadded:: 3.3
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]importlib.resources.abc.rst, line 203)**

Unknown directive type "attribute".

```
.. attribute:: name
```

The name of the module the loader can handle.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]importlib.resources.abc.rst, line 207)**

Unknown directive type "attribute".

```
.. attribute:: path
```

Path to the file of the module.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]importlib.resources.abc.rst, line 211)**

Unknown directive type "method".

```
.. method:: load_module(fullname)
```

Calls `super's ``load_module()```.

```
.. deprecated:: 3.4
```

Use `:meth:`Loader.exec_module`` instead.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]importlib.resources.abc.rst, line 218)**

Unknown directive type "abstractmethod".

```
.. abstractmethod:: get_filename(fullname)
```

Returns `:attr:`path``.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]importlib.resources.abc.rst, line 222)**

Unknown directive type "abstractmethod".

```
.. abstractmethod:: get_data(path)
```

Reads `*path*` as a binary file and returns the bytes from it.

An abstract base class for implementing source (and optionally bytecode) file loading. The class inherits from both `:class:`ResourceLoader`` and `:class:`ExecutionLoader``, requiring the implementation of:

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]importlib.resources.abc.rst, line 229); [backlink](#)

Unknown interpreted text role "class".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]importlib.resources.abc.rst, line 229); [backlink](#)

Unknown interpreted text role "class".

- `:meth:`ResourceLoader.get_data``

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]importlib.resources.abc.rst, line 233); [backlink](#)

Unknown interpreted text role "meth".

- `:meth:`ExecutionLoader.get_filename``

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]importlib.resources.abc.rst, line 236); [backlink](#)

Unknown interpreted text role "meth".

Should only return the path to the source file; sourceless loading is not supported.

The abstract methods defined by this class are to add optional bytecode file support. Not implementing these optional methods (or causing them to raise `:exc:`NotImplementedError``) causes the loader to only work with source code. Implementing the methods allows the loader to work with source *and* bytecode files; it does not allow for *sourceless* loading where only bytecode is provided. Bytecode files are an optimization to speed up loading by removing the parsing step of Python's compiler, and so no bytecode-specific API is exposed.

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]importlib.resources.abc.rst, line 238); [backlink](#)

Unknown interpreted text role "exc".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]importlib.resources.abc.rst, line 247)

Unknown directive type "method".

```
.. method:: path_stats(path)
```

Optional abstract method which returns a `:class:`dict`` containing metadata about the specified path. Supported dictionary keys are:

- ```mtime``` (mandatory): an integer or floating-point number representing the modification time of the source code;
- ```size``` (optional): the size in bytes of the source code.

Any other keys in the dictionary are ignored, to allow for future extensions. If the path cannot be handled, `:exc:`OSError`` is raised.

```
.. versionadded:: 3.3
```

```
.. versionchanged:: 3.4
```

Raise `:exc:`OSError`` instead of `:exc:`NotImplementedError``.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]importlib.resources.abc.rst, line 264)**

Unknown directive type "method".

```
.. method:: path_mtime(path)
```

Optional abstract method which returns the modification time for the specified path.

```
.. deprecated:: 3.3
```

This method is deprecated in favour of :meth:`path\_stats`. You don't have to implement it, but it is still available for compatibility purposes. Raise :exc:`OSError` if the path cannot be handled.

```
.. versionchanged:: 3.4
```

Raise :exc:`OSError` instead of :exc:`NotImplementedError`.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]importlib.resources.abc.rst, line 277)**

Unknown directive type "method".

```
.. method:: set_data(path, data)
```

Optional abstract method which writes the specified bytes to a file path. Any intermediate directories which do not exist are to be created automatically.

When writing to the path fails because the path is read-only (:attr:`errno.EACCES`/:exc:`PermissionError`), do not propagate the exception.

```
.. versionchanged:: 3.4
```

No longer raises :exc:`NotImplementedError` when called.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]importlib.resources.abc.rst, line 290)**

Unknown directive type "method".

```
.. method:: get_code(fullname)
```

Concrete implementation of :meth:`InspectLoader.get\_code`.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]importlib.resources.abc.rst, line 294)**

Unknown directive type "method".

```
.. method:: exec_module(module)
```

Concrete implementation of :meth:`Loader.exec\_module`.

```
.. versionadded:: 3.4
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]importlib.resources.abc.rst, line 300)**

Unknown directive type "method".

```
.. method:: load_module(fullname)
```

Concrete implementation of :meth:`Loader.load\_module`.

```
.. deprecated:: 3.4
```

Use :meth:`exec\_module` instead.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-**

**main\Doc\library\[cpython-main] [Doc] [library]importlib.resources.abc.rst, line 307)**

Unknown directive type "method".

```
.. method:: get_source(fullname)
```

Concrete implementation of :meth:`InspectLoader.get\_source`.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]importlib.resources.abc.rst, line 311)**

Unknown directive type "method".

```
.. method:: is_package(fullname)
```

Concrete implementation of :meth:`InspectLoader.is\_package`. A module is determined to be a package if its file path (as provided by :meth:`ExecutionLoader.get\_filename`) is a file named ``\_\_init\_\_`` when the file extension is removed **\*\*and\*\*** the module name itself does not end in ``\_\_init\_\_``.

An object with a subset of pathlib.Path methods suitable for traversing directories and opening files.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]importlib.resources.abc.rst, line 325)**

Unknown directive type "versionadded".

```
.. versionadded:: 3.9
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]importlib.resources.abc.rst, line 327)**

Unknown directive type "attribute".

```
.. attribute:: name
```

Abstract. The base name of this object without any parent references.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]importlib.resources.abc.rst, line 331)**

Unknown directive type "abstractmethod".

```
.. abstractmethod:: iterdir()
```

Yield Traversable objects in self.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]importlib.resources.abc.rst, line 335)**

Unknown directive type "abstractmethod".

```
.. abstractmethod:: is_dir()
```

Return True if self is a directory.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]importlib.resources.abc.rst, line 339)**

Unknown directive type "abstractmethod".

```
.. abstractmethod:: is_file()
```

Return True if self is a file.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-**



**main\Doc\library\[cpython-main] [Doc] [library]importlib.resources.abc.rst, line 343)**

Unknown directive type "abstractmethod".

```
.. abstractmethod:: joinpath(child)

    Return Traversable child in self.
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]importlib.resources.abc.rst, line 347)**

Unknown directive type "abstractmethod".

```
.. abstractmethod:: __truediv__(child)

    Return Traversable child in self.
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]importlib.resources.abc.rst, line 351)**

Unknown directive type "abstractmethod".

```
.. abstractmethod:: open(mode='r', *args, **kwargs)

    *mode* may be 'r' or 'rb' to open as text or binary. Return a handle
    suitable for reading (same as :attr:`pathlib.Path.open`).

    When opening as text, accepts encoding parameters such as those
    accepted by :attr:`io.TextIOWrapper`.
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]importlib.resources.abc.rst, line 359)**

Unknown directive type "method".

```
.. method:: read_bytes()

    Read contents of self as bytes.
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]importlib.resources.abc.rst, line 363)**

Unknown directive type "method".

```
.. method:: read_text(encoding=None)

    Read contents of self as text.
```

An abstract base class for resource readers capable of serving the `meth`importlib.resources.files`` interface. Subclasses `class`importlib.abc.ResourceReader`` and provides concrete implementations of the `class`importlib.abc.ResourceReader``'s abstract methods. Therefore, any loader supplying `class`importlib.abc.TraversableReader`` also supplies `ResourceReader`.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]importlib.resources.abc.rst, line 370);**

[backlink](#)

Unknown interpreted text role "meth".

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]importlib.resources.abc.rst, line 370);**

[backlink](#)

Unknown interpreted text role "class".

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]importlib.resources.abc.rst, line 370);**

[backlink](#)

Unknown interpreted text role "class".

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]importlib.resources.abc.rst, line 370);**  
[backlink](#)

Unknown interpreted text role "class".

Loaders that wish to support resource reading are expected to implement this interface.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]importlib.resources.abc.rst, line 380)**

Unknown directive type "versionadded".

```
.. versionadded:: 3.9
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]importlib.resources.abc.rst, line 382)**

Unknown directive type "abstractmethod".

```
.. abstractmethod:: files()
```

```
Returns a :class:`importlib.abc.Traversable` object for the loaded
package.
```