python / cpython Public

Code Issues 5k+ Pull requests 2.1k Actions Projects Security and q

Commit d786d59

3 people authored 11 hours ago · 41 / 44 · Verified

```
[3.14] gh-146121: Clarify security model of pkgutil.getdata (GH-148197) (GH-148206)

(cherry picked from commit cf59bf7)

Co-authored-by: Petr Viktorin <encukou@gmail.com>
Co-authored-by: Stan Ulbrych <stan@python.org>
```

3.14 (#148206) · v3.14.4

1 parent 25369a8 commit d786d59

1 file changed +26 -2 lines changed

Top

Filter files...

Doc/library

pkgutil.rst

1 file changed +26 -2 lines changed

Search within code

Doc/library/pkgutil.rst

```
@@ -151,24 +151,48 @@ support.
151 151     :meth:`get_data <importlib.abc.ResourceLoader.get_data>` API.  The
152 152     *package* argument should be the name of a package, in standard module
        format
153 153     (``foo.bar``).  The *resource* argument should be in the form of a relative
154       -   filename, using ``/`` as the path separator.  The parent directory name
155       -   ``..`` is not allowed, and nor is a rooted name (starting with a ``/``).
        154 +   filename, using ``/`` as the path separator.
156 155
157 156     The function returns a binary string that is the contents of the specified
```

```
158 157       resource.
159 158
    159 +     This function uses the :term:`loader` method
    160 +     :func:`~importlib.abc.FileLoader.get_data`
    161 +     to support modules installed in the filesystem, but also in zip files,
    162 +     databases, or elsewhere.
    163 +
160 164       For packages located in the filesystem, which have already been imported,
161 165       this is the rough equivalent of::
162 166
163 167          d = os.path.dirname(sys.modules[package].__file__)
164 168          data = open(os.path.join(d, resource), 'rb').read()
165 169
    170 +     Like the :func:`open` function, :func:`!get_data` can follow parent
    171 +     directories (``../``) and absolute paths (starting with ``/`` or ``C:/``,
    172 +     for example).
    173 +     It can open compilation/installation artifacts like ``.py`` and ``.pyc``
    174 +     files or files with :func:`reserved filenames <os.path.isreserved>`.
    175 +     To be compatible with non-filesystem loaders, avoid using these features.
    176 +
    177 +     .. warning::
    178 +
    179 +        This function is intended for trusted input.
    180 +        It does not verify that *resource* "belongs" to *package*.
    181 +
    182 +     If you use a user-provided *resource* path, consider verifying it.
    183 +     For example, require an alphanumeric filename with a known extension, or
    184 +     install and check a list of known resources.
    185 +
166 186       If the package cannot be located or loaded, or it uses a :term:`loader`
167 187       which does not support :meth:`get_data
          <importlib.abc.ResourceLoader.get_data>`,
168 188       then ``None`` is returned.  In particular, the :term:`loader` for
169 189       :term:`namespace packages <namespace package>` does not support
170 190       :meth:`get_data <importlib.abc.ResourceLoader.get_data>`.
171 191
    192 +     .. seealso::
    193 +
    194 +        The :mod:`importlib.resources` module provides structured access to
    195 +        module resources.
```

```
172  196
173  197     .. function:: resolve_name(name)
174  198
```

Comments 0