## Hooking into the Python import system

Engineering

PyCon 2022 April 30, 2022

Fred Phillips (he/him)
Engineering Team Lead, News Automation

Tech At Bloomberg.com

## Introduction





TechAtBloomberg.com

Bloomberg

## What does import actually do?



TechAtBloomberg.com

Bloomberg

## **Finding & Loading**

BuiltinImporter

FrozenImporter

PathFinder

TechAtBloomberg.com



## Implementing a finder

```
def find_spec(
          self,
          fullname: str,
          path: Sequence[Union[bytes, str]] | None,
          target: ModuleType | None = None,
          ) -> ModuleSpec | None:
          pass
```

TechAtBloomberg.com



## Implementing a finder

sys.meta\_path.insert(0, MyFinder())

TechAtBloomberg.com

Bloomberg

## Implementing a loader

```
def create_module(
          self,
          spec: ModuleSpec
     ) -> ModuleType:
     pass
```

TechAtBloomberg.com



## Implementing a loader

```
def exec_module(
          self,
          module: ModuleType
     ) -> None:
     pass
```

TechAtBloomberg.com



## How to hook into the import system



TechAtBloomberg.com

Bloomberg

## **Tracing finder**

```
class TracingFinder(MetaPathFinder):
    def find_spec(
        self, fullname, path, target=None):
        print(f"Looking for: {fullname=}
        {path=}")
        return None
```

TechAtBloomberg.com



## Tracing finder

```
sys.meta_path.insert(0, TracingFinder())
import datetime
print(datetime.datetime.now())
```

TechAtBloomberg.com



## **Tracing finder**

```
$ python tracing.py
Looking for: fullname='datetime' path=None
Looking for: fullname='math' path=None
Looking for: fullname='_datetime' path=None
2022-04-30 11:40:00.385924
$
```





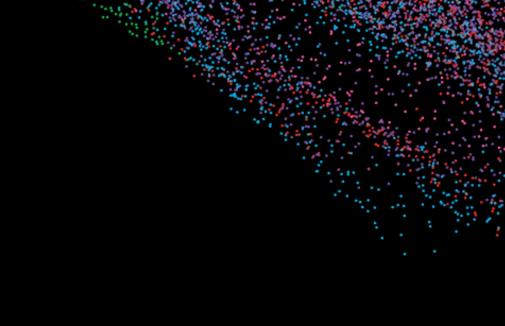




© 2022 Bloomberg Finance L.P. All rights reserved.

Bloomberg

## **Blocklist**



TechAtBloomberg.com

Engineering

```
class BlocklistFinder(MetaPathFinder):
    def __init__(
        self, blocked: List[str]):
        self._blocked = blocked
```

Tech At Bloomberg.com

**Bloomberg** 

```
def find_spec(
    self, fullname, path, target):
    if fullname in blocked:
        raise ImportError(f"Cannot import
        {fullname!r} it is blocked")
    return None
```

TechAtBloomberg.com

**Bloomberg** 

```
sys.meta_path.insert(
    0, BlocklistFinder(["socket"]))
import http.server
```

TechAtBloomberg.com

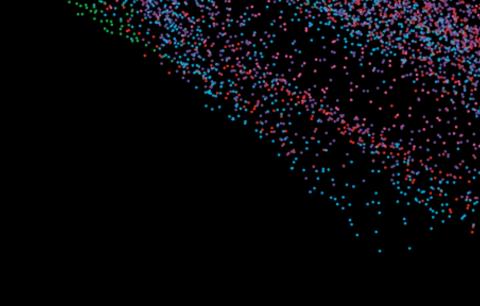
Bloomberg

ImportError: Cannot import 'socket' it is
blocked

TechAtBloomberg.com

Bloomberg

## **Database loader**



TechAtBloomberg.com

Bloomberg

```
class DbLoader(importlib.abc.Loader):
    def __init__(
        self,
        package_name: str,
        dbhandle: Connection):
        self._package_name = package_name
        self._dbhandle = dbhandle
```

Tech At Bloomberg.com



```
def create_module(
    self,
    spec: importlib.machinery.ModuleSpec
) -> types.ModuleType
```

```
def exec_module(
   self,
   module: types.ModuleType
  ) -> None
```

Tech At Bloomberg.com



```
def create_module(
    self,
    spec: importlib.machinery.ModuleSpec
) -> types.ModuleType:
    module = types.ModuleType(spec.name)
    if spec.name == self._package_name:
        module.__path__ = []
    return module
```

TechAtBloomberg.com



```
def exec module(self, module):
 if module. name == self. package name:
     return
cursor = self. dbhandle.cursor()
cursor.execute(
    "SELECT code FROM code WHERE name=?",
    [module.__name__])
code = cursor.fetchone()
exec(code[0], module. dict )
```

Tech At Bloomberg.com



```
def provides(
self,
 fullname: str
) -> bool:
 cursor.execute(
     "SELECT 1 FROM code WHERE name=?",
     [fullname])
```

TechAtBloomberg.com

Bloomberg

#### **DB Finder**

```
class
DbFinder(importlib.abc.MetaPathFinder):
    def __init__(
        self,
        loader: DbLoader):
        self._loader = loader
```

TechAtBloomberg.com



#### **DB Finder**

```
def find_spec(
self, fullname, path, target):
 if self. loader.provides(fullname):
      return importlib.util.spec from loader(
          fullname.
          self._loader)
 return None
```

Tech At Bloomberg.com



## Thank you!

<u>bloomberg.com/careers</u> <u>github.com/fophillips/import-hooks/</u>

Contact me: fphillips7@bloomberg.net or @fophillips

TechAtBloomberg.com

© 2022 Bloomberg Finance L.P. All rights reserved.

# Engineering