

Hooking into the Python import system

Engineering

Bloomberg

PyCon 2022
April 30, 2022

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

TechAtBloomberg.com

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



Introduction

TechAtBloomberg.com

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

Bloomberg

Engineering

import



TechAtBloomberg.com

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

Bloomberg

Engineering



What does **import** actually do?

TechAtBloomberg.com

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

Bloomberg

Engineering

Finding & Loading

BuiltinImporter

FrozenImporter

PathFinder

TechAtBloomberg.com

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

Bloomberg

Engineering

Implementing a finder

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

Implementing a finder

```
sys.meta_path.insert(0, MyFinder())
```

Implementing a loader

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


Implementing a loader

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

How to hook into the import system



TechAtBloomberg.com

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

Bloomberg

Engineering

Tracing finder

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

Tracing finder

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


Tracing finder

```
bash
$ 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
$ █
```



Why?

TechAtBloomberg.com

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

Bloomberg

Engineering



Blocklist

TechAtBloomberg.com

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

Bloomberg

Engineering

Blocklist finder

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


Blocklist finder

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

Blocklist finder

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

Blocklist finder

`ImportError: Cannot import 'socket' it is blocked`

Database loader

TechAtBloomberg.com

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

Bloomberg

Engineering

DB Loader

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

DB Loader

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

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

DB Loader

```
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
```

DB Loader

```
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__)
```


DB Loader

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

DB Finder

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

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
```


Thank you!

bloomberg.com/careers

github.com/fophillips/import-hooks/

Contact me: fphillips7@bloomberg.net or [@fophillips](https://twitter.com/fophillips)

Engineering

Bloomberg

TechAtBloomberg.com

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