

Import

Andrew Svetlov

andrew.svetlov@gmail.com

asvetlov.blogspot.com

Basics

- Import statement
- `__import__` builtin
- `sys.path`
- WTF???

Import statement

- `import a.b.c`
- `from a.b import c`
- `from a.b import name`
- `from ..b import c`
- `from __future__ import absolute_import`

`+ - a`

`+ - b <== current`

`+ - c`

__import__

- `__import__(name:str, globals:dict=None, locals:dict=None, fromlist:list=(), level:int=0) -> Module`
- Returns module or toplevel package depending from *fromlist*.

Import lock

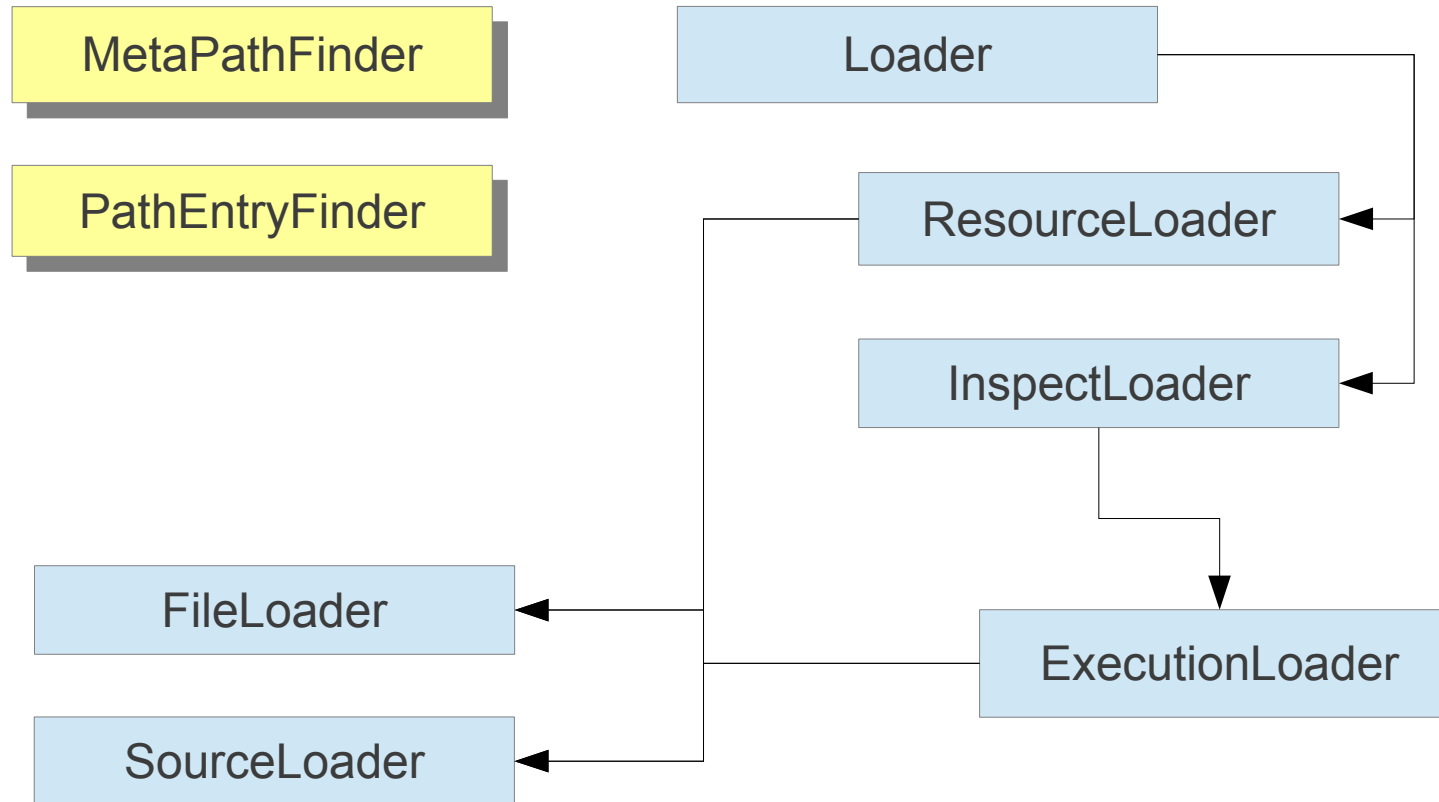
```
def f():  
    from twisted.internet import reactor  
    reactor.callLater(...)
```

- Slow
- Multithreading → Deadlock

Modules & Packages

- `__name__`: str
- `__doc__`: str
- `__file__`: str
- `__path__`: [str]
- `__package__`: str
- `__cached__`: str
- `__loader__`: Loader

ABC



sys.modules

- 'keyword': <module 'keyword' from '.../Lib/keyword.py'>
- 'unittest.signals': <module 'unittest.signals' from '.../Lib/unittest/signals.py'>
- '_frozen_importlib': <module '_frozen_importlib' from '<frozen>'>
- 'sys': <module 'sys' (built-in)>
- '_heapq': <module '_heapq' from '.../lib.linux-x86_64-3.3-pydebug/_heapq.cpython-33dm.so'>

Finder

- MetaPathFinder
 - find_module(fullname: str, path: str) -> Loader
 - invalidate_caches()
- PathEntryFinder
 - find_loader(fullname: str) -> (Loader, str)
 - invalidate_caches()

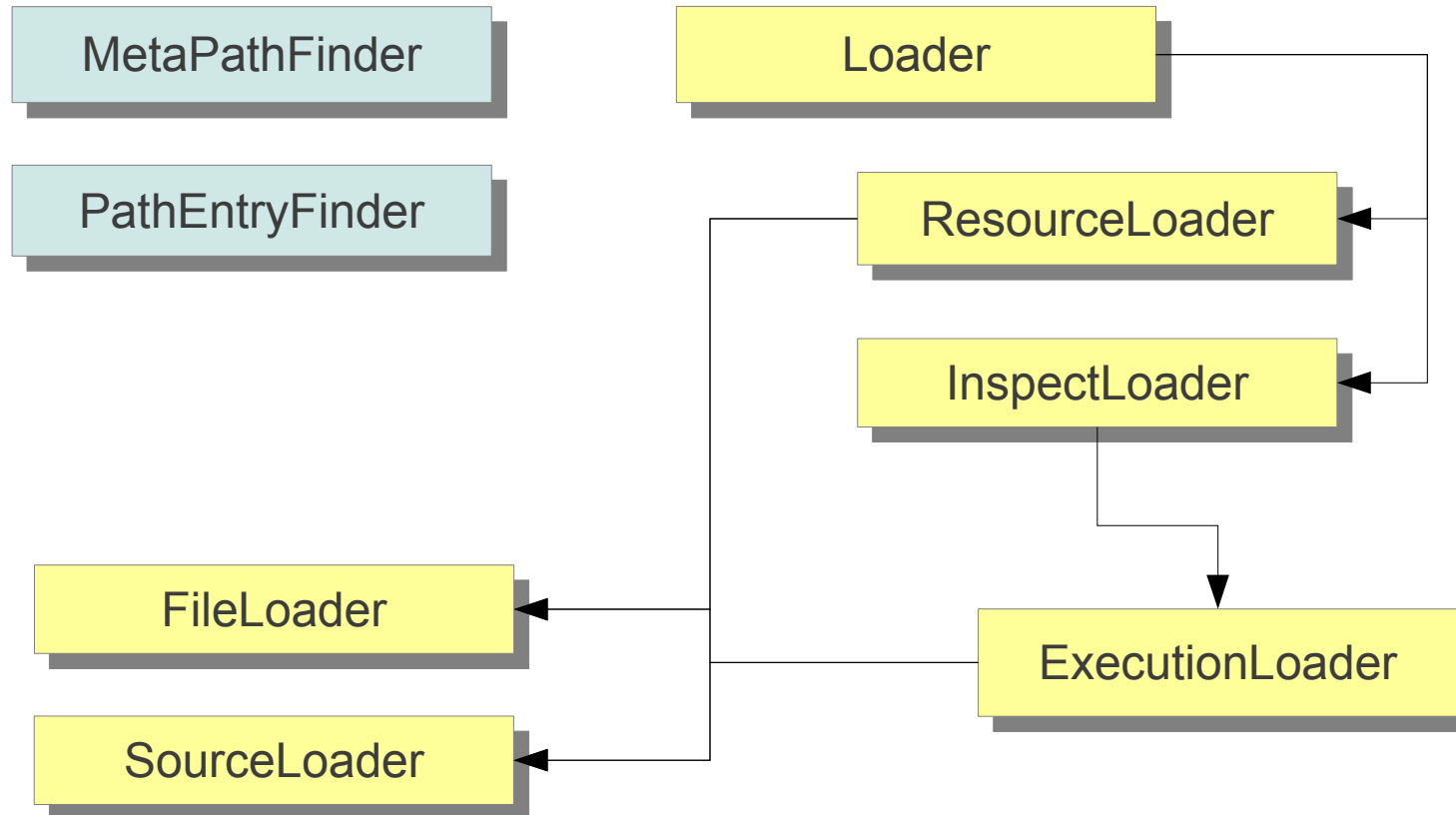
sys.meta_path

- [Finder()]
 - BuiltinImporter
 - FrozenImporter
 - PathFinder

sys.path_hooks

- [callable(path:str) -> Finder]
 - <class 'zipimport.zipimporter'>
 - <function PathEntryFinder path_hook>
- sys.path: [str]
- sys.path_importer_cache: {str:Finder}

ABC



Loader

- Base loader
- `load_module(fullname:str) -> Module`
- `module_repr(module:Module) -> str`

ResourceLoader(Loader)

- Can return data from backend storage
- *load_module*
- `get_data(path: str) -> bytes`

```
>>> folder = os.path.dirname(__file__)
```

```
>>> resname = os.path.join(folder,  
'image.png')
```

```
>>> __loader__.get_data(resname)
```

```
... <img bytes>
```

InspectLoader(Loader)

- Introspection support
- *load_module*
- `is_package(fullname:str) -> bool`
- `get_code(fullname:str) -> CodeObject`
- `get_source(fullname:str) -> str`

ExecutionLoader(InspectLoader)

- Support for execution modules as scripts
- *load_module, is_package, get_code, get_source*
- `get_filename(fullname:str) -> str`
 - `__file__`
 - `Python -m package.module`

FileLoader(ResourceLoader, ExecutionLoader)

- Just to be. Sourceless loader
- *load_module, is_package, get_code, get_data, get_source, get_filename*

SourceLoader(ResourceLoader, ExecutionLoader)

- *load_module, is_package, get_code, get_data, get_source, get_filename*
- `path_mtime(path:str) -> int`
- `path_stats(path:str) -> dict`
 - `mtime:int`
 - `size:int [optional]`
- `set_data(path:str, data:bytes)`

importlib

- `find_loader(name:str, path:str=None) -> Loader`
- `import_module(name:str, package:str=None) -> Module`

Bootstrap

- site.py
- venv
- user sitepackages
 - ~/.local/lib/python3.3/site-packages
- site-packages
- .pth files
- sitecustomize.py
- usersite.py

Shutdown

- For every module in `sys.modules` set all global attrs to `None`
- Delete modules

Вопросы?

Andrew Svetlov

andrew.svetlov@gmail.com

asvetlov.blogspot.com