

Конфигурация

Проблема

```
DEBUG = False
```

```
try:  
    from some.other.settings import * # All, Karl!  
except ImportError:  
    pass    # yeah, whatever.
```

```
import os
```

```
MY_TEMPLATES = os.path.join(os.path.dirname(__file__), 'templates')
```

Проблема

```
import requests

globals().update(
    requests
        .get('http://example.com/settings.json')
        .json()
)

# In your code

def get_template(name):
    dir = pathlib.Path(settings.MY_TEMPLATES)
    with (dir / name).open('rt') as f:
        return f.read()
```

Что делать?

Что делать?

- Конфиги не должны быть кодом!
- Используйте текстовые форматы с поддержкой типов данных.

Например Yaml или Toml

Yaml

```
app:  
  debug: &DEBUG true  
  host: localhost  
  port: 5000  
  
  postgres:  
    host: localhost  
    echo: *DEBUG  
  redis:  
    address:  
    - localhost  
    - 6380
```

Toml

```
[app]
debug = true
host = "localhost"
port = 5000

postgres = { host = "localhost", echo = true }

[app.redis.address]      # {'app': {'redis': {'address': {
host = "localhost"      #      'host': "localhost",
port = 6380              #      'port': 6380,
                        #  }}}}]

# use = "/path/to/some/other/config.toml"
~
```

Валидация и схема!

- Схема:
 - не позволит добавить случайные или забыть необходимые параметры.
 - Значения по-умолчанию.
 - Преобразовать в другие структуры
- Валидация:
 - Правильность данных.

Trafaret

github.com/deepwalker/trafaret

```
import yaml
import trafaret as t

HostPort = t.Tuple(t.String, t.Int)
Unixsocket = t.String
HostPortDict = t.Dict({
    t.Key('host'): t.String,
    t.Key('port'): t.Int,
})

postgres_trafaret = t.Forward()

config_trafaret = t.Dict({
    t.Key('debug'): t.Bool,
    t.Key('host', default='localhost'): t.String,
    t.Key('port', default=5000): t.Int[1025:65535],
    t.Key('postgres'): postgres_trafaret,
    t.Key('redis'): t.Dict({
        t.Key('address', default=('localhost', 6379)): t.Or(
            HostPort,
            HostPortDict,
            Unixsocket,
        )
    })
})

postgres_trafaret <= t.Dict({
    t.Key('host', default='localhost'): t.String,
    t.Key('port', default=5432): t.Int,
    t.Key('user', default='test'): t.String,
    t.Key('password', default='test'): t.String,
    t.Key('db_name', default='test'): t.String,
    t.Key('echo', default=False): t.Bool,
})

def load_config(fname):
    with open(fname, 'rt') as f:
        return config_trafaret(yaml.load(f))
```