

Problem #01

Configuration Problem

In real world, we cannot hard-code configuration values (DB name, passwords, timeout etc).

→ Different Environment need different configuration.

For Examples

→ Developer uses his local Database To Test the system, so he configures the local database to program.

→ Tester may use data on another server to test the system.

→ When software is in production, it uses different data.

So Hardcoding the configuration data in Development stage causes the system to crash in testing and production because db does not exist locally there.

That is why we create single configuration loader that

- Read From YAML Files
- start config based on environment type
- Map config values into a python object.

Folder structure

Project /

config/

development.yaml

staging.yaml

prod.yaml

config.py

main.py

YAML FILE STRUCTURE

Example (Prod.yaml)

db:

DB: "ProductionDB"

TableName: "Users"

Password: "secrets"

timeout: 30

YAML → automatically parsed into python dictionary.

Config class Implementation

```
import yaml
```

```
class config():
```

```
    DB: str = " "
```

Date . .

```
TableName: str = ""
```

```
Password: str = ""
```

```
Timeout: int = ""
```

- Acts As typed container
- Default values prevent crashes if keys are missing
- Makes config self-documenting

load()

```
def load(self, envType) ← Determines which environment (development, staging, prod)
```

reading YAML file

```
with open(f'config/{envType}.yaml', 'r')  
    as file:
```

```
    configData = yaml.safe_load(file)
```

→ with open(...) → File auto closes (even on errors)

yaml.safe_load() → Safe parsing into python dict

extracting only what we need

```
dbconfig = configData['db']
```

Assigning values

```
self.db = dbconfig.get("DB", "")
```

```
self.TableName = dbconfig.get("TableName", "")
```

```
self.Password = dbconfig.get("Password", " ")
```

```
self.Timeout = dbconfig.get('timeout', 15)
```

`get()` prevent key errors, Allows defaults, makes config backward compatible.

```
return self.
```

```
# main.py
```

```
from configpy import config
```

```
cfg = config().load("Prod")
```

```
print(cfg.DB)
```

```
print(cfg.TableName)
```

```
print(cfg.Password)
```

```
print(cfg.Timeout)
```