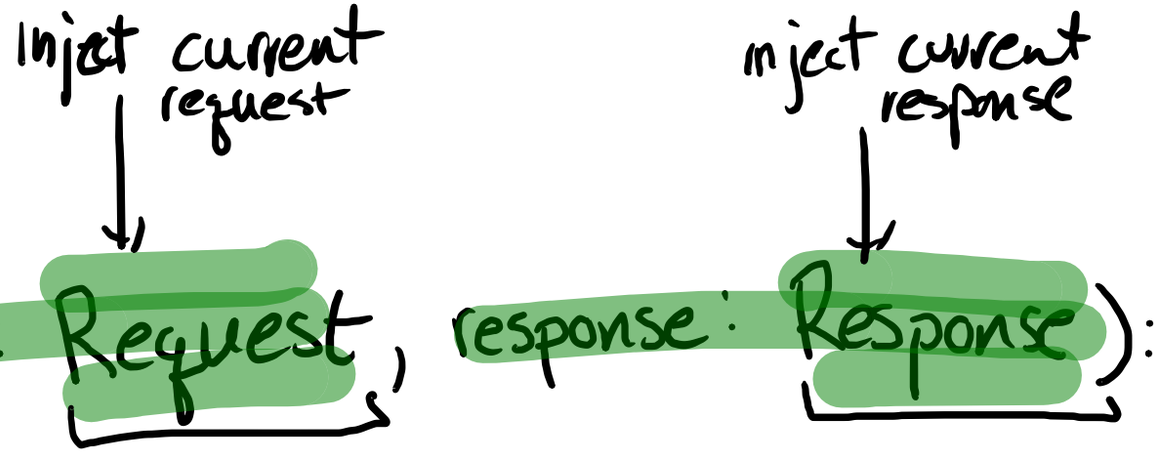
Dependency Injection (the "D" in SOLID, kind of)

Robert "Uncle Bob" Martin

Dependency Inversion

FastAPI




```
@app.put("/api/users/{user_id}")
def update_user(user_id: int):
    #code
    pass
```

← missing HTTP request object!!

"repository pattern"

```
class AccountQueries:
    def create_account(self, ...):
        sql.insert(...)
        return new_account
        # totally db-bound
```

"kind of like DAO (data access object)"

Without DI:

```
def create_account(account):
    repo = AccountQueries()
    return repo.create_account(account)
    # hard to write unit tests
    # does not need: network, db, fs
```

Unit test with DI version →

```
def test_create_account():
    app.dependency_overrides[AccountQueries] =
        FakeAccountQueries
    result = app.post("/...", data)
    # when it runs
    # FakeAccountQueries
    assert result.status_code == 201
```

it uses

With DI:

```
def create_account(
    account,
    repo: AccountQueries = Depends(),
):
    return repo.create_account(account)
```

Class-based version

```
def f(repo : AccountQueries = Depends()):
```

uses the type hints

```
def f(repo = Depends(AccountQueries)):
```

← uses the actual class

```
def f(repo : AccountQueries = Depends(AccountQueries)):
```

ignored

Method results version

```
def f(user : Account = Depends(get_current_user)):
```

uses the fn name

Dependencies with dependencies

```
class BlogQueries:
    def __init__(self, user: Account = Depends(get_current_user)):
        self.user = user

    def create(self, blog_content):
        db.create({"text": blog_content, "author": self.user})


@app.post("/api/blogs")
def create_blog(blog: Blog, repo: BlogQueries = Depends()):
    repo.create(blog)
    print(repo.user)
```

"hints"

3: invoke get_current_user result

2: invoke ctor, look at deps

4 Actually invoking

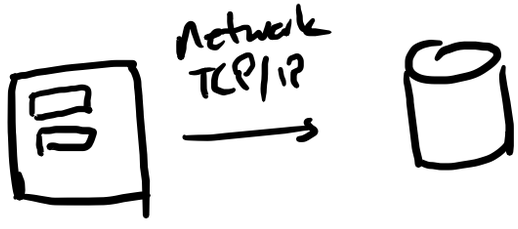
returns an obj of type Account

1 invoke this fn, look at deps: BlogQueries

5 actual value repo parameter

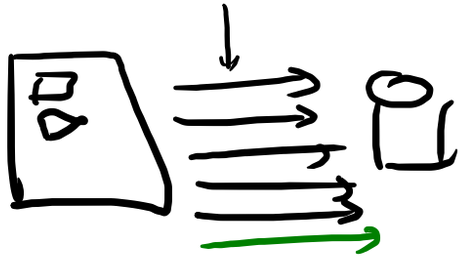
Pydantic Model "structural validation"

Thread or I/O "Pools" > Resource pools



Creating the connection takes "a long time"

- TCP/IP handshake
- Db handshake
- Conn string
- ✓ Hooray



pool.connection()

- assigns an existing connection
- ✓ Query!

Need sixth connection!

- go thru dumb steps
- ✓ Query!