

Внешние источники

Для работы с внешними источниками обычно выделяются три сущности.

Accessor

Умеет обращаться к сторонним источникам данных и умеет преобразовывать полученные данные в нужный вид.

Например, можно сделать accessor для доступа к PostgreSQL или к внешнему API. Также, accessor обычно умеет выполнять подключение и отключение от внешнего источника данных.

Пример метода rfc-аксессора, который ходит к внешнему API по RFC, получает данные и преобразует их к заданной схеме:

```
async def get_date(self, proxy_number: str):
    result = await self.store.rfc.call(
        "Z_F_HR_LK2_GET_CALC_DATE", IV_PERNR=proxy_number
    )
    return GetDateInfo.Schema().load(result)
```

Store

Набор всех accessor, доступный из всех мест в приложении. Он инициализируется при старте приложения и по очереди подключает каждый accessor, тем самым обеспечивая их боеготовность.

Пример кода функции __init__ в store:

```
self.db = AioMongoAccessor(self, name='db')
self.qdb = TarantoolAccessor(self, name='qdb')
self.pgdb = PgAccessor(self, name='pgdb')
self.consul = ConsulAccessor(self, name='consul')
```

Manager

Помогает вынести большие куски логики из View и переиспользовать их. Обычно manager включает в себя обращение сразу к нескольким accessor. Например, при регистрации нового пользователя manager может сначала создать пользователя с помощью postgresql-аксессора, а потом добавить ему сессию, воспользовавшись redis-аксессором.

Пример метода manager, который выполняет кусок логики, вызываемый из разных view:

```
async def supply_manager_linked(self, manager_email: str, channels: list[dict]) -> list[dict]:
    manager = await self.store.managers.get_by_email(email=manager_email)
    external_ids = manager.get('external_ids', [])
    linked_channel_kinds = set()
    for external_id in external_ids:
        linked_channel_kinds.add(external_id.get('kind'))
    for channel in channels:
        channel['manager_linked'] = channel['kind'] in linked_channel_kinds
    return channels
```

Примеры

1. [Организация доступа к Store и другими глобальным объектам из Application](#)