

# Дополнительно

## Mixin

**Mixin (примесь)** — это некоторая дополнительная функциональность, которая обычно добавляется к классу, используя принцип композиции.

Миксины бывает удобно применять, когда несколько разных классов должны выполнять одну и ту же логику, и эта логика не должна выполняться для всех наследников класса. Это лишь один из случаев использования миксинов, но достаточно часто встречающийся.

Например, в третьем домашнем задании (KTS Drive) за вас уже написан *AuthRequiredMixin*. Он, основываясь на методе класса `web.View`, переопределяет его метод `_iter` (метод, вызываемый до исполнения основных методов) и проверяет, есть ли доступ у данного пользователя к этому `View`.

Код миксина:

```
class AuthRequiredMixin:
    async def _iter(self) -> StreamResponse:
        if not getattr(self.request, "user_id", None):
            raise HTTPUnauthorized
        return await super(AuthRequiredMixin, self)._iter()
```

А вот как его можно применить:

```
class FilesListView(AuthRequiredMixin, web.View):
    ...
```

То есть класс наследуется сразу от двух родителей. Но, так как наследование происходит справа налево, то метод `_iter` класса `AuthRequiredMixin` переопределяет метод класса `web.View`.

В следующем задании за вас написан `RequestPaginationMixin`. Постарайтесь его использовать в своей разработке, хотя это опционально.