

django

Web-разработка на батарейках.

- Где взять: <https://www.djangoproject.com/>
- Документация: <https://docs.djangoproject.com/en/2.0/>

Шередеко Василий, piphon@gmail.com

Батарейки

- ORM для работы с БД: PostgreSQL, MySQL и SQLite.
- Встроенный шаблонизатор и поддержка шаблонизатора Jinja2.
- Инструменты для валидации пользовательских данных.
- Административный интерфейс.
- Аутентификация пользователя.

Создание проекта

Виртуальное окружение

```
mkvirtualenv --python=`which python3` example
```

Создание проекта

Устанавливаем Django

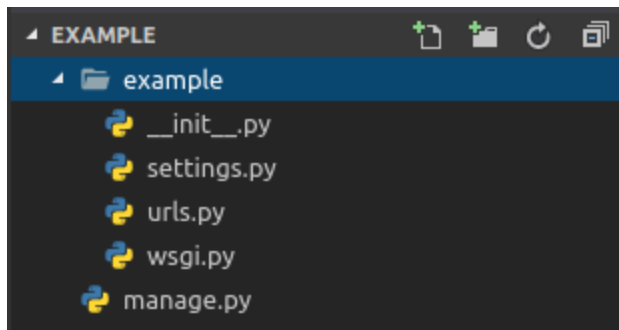
```
pip install Django
```

Создание проекта

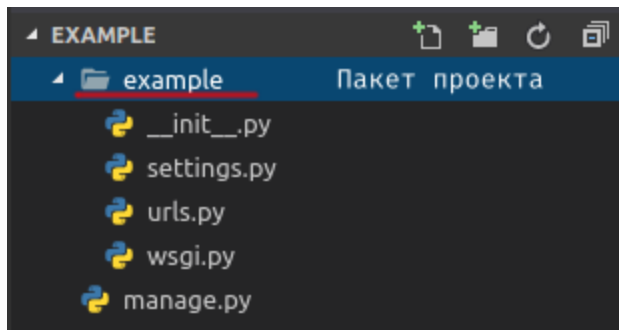
Создаем проект Django

```
django-admin startproject example
```

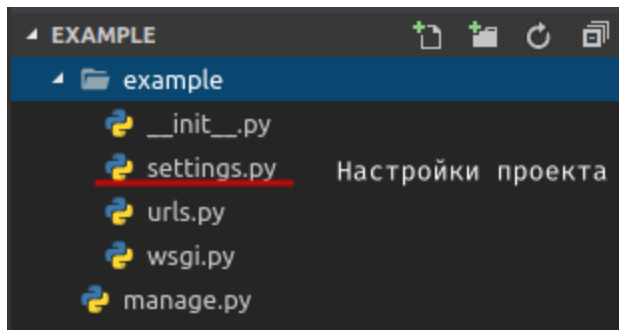
Структура проекта



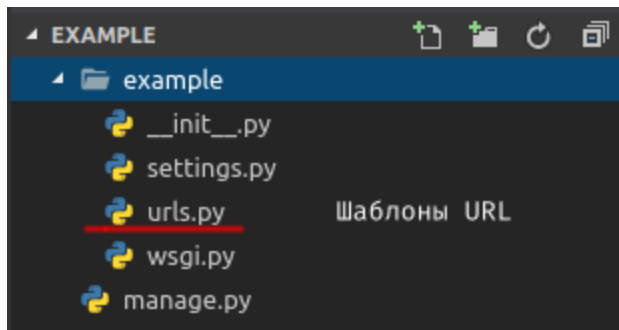
Структура проекта



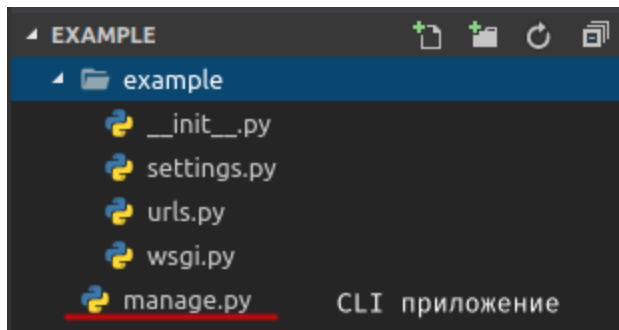
Структура проекта



Структура проекта



Структура проекта



Консольные команды

Список команд

```
python manage.py
```

Консольные команды

Запуск Web-сервера

```
python manage.py runserver
```

Консольные команды

Обновление таблиц в базе данных

```
python manage.py migrate
```

Консольные команды

Создание пользователя для админки

```
python manage.py createsuperuser
```

Консольные команды

Интерактивная консоль

```
python manage.py shell
```

Консольные команды

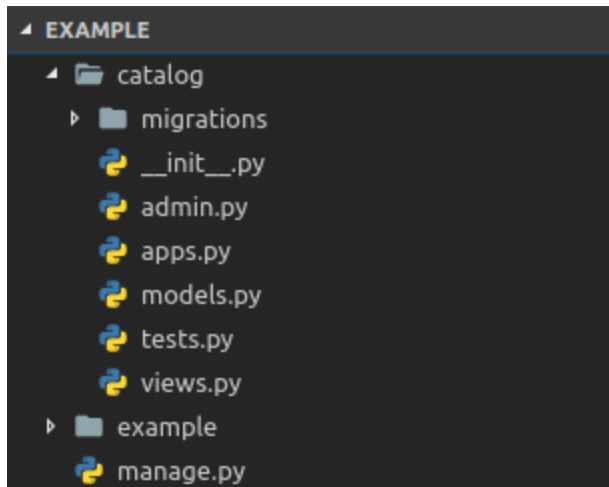
Интерактивная консоль базы данных

```
python manage.py dbshell
```

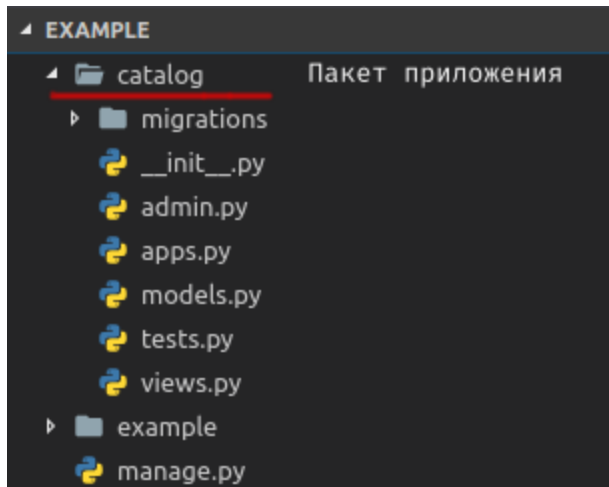

Создание приложения Django

```
python manage.py startapp catalog
```

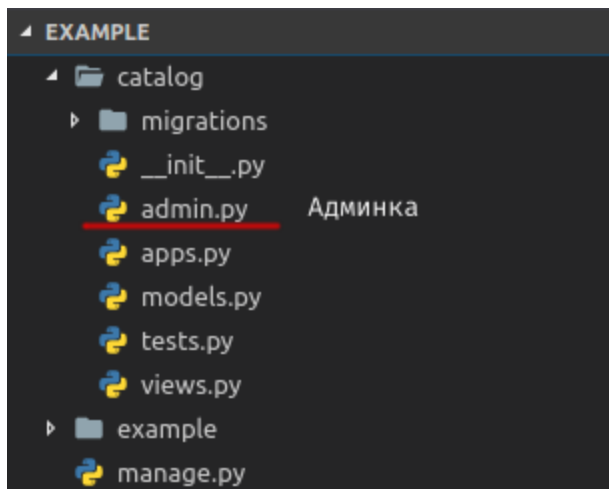
Структура приложения



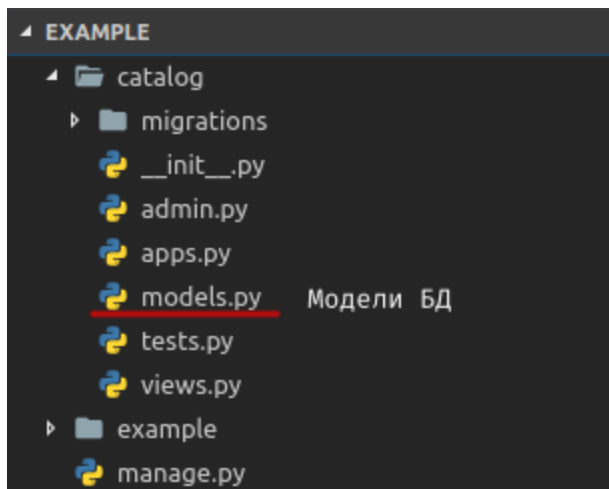
Структура приложения



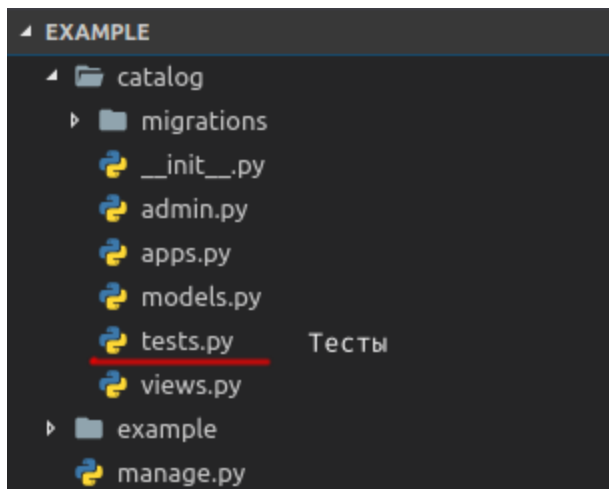
Структура приложения



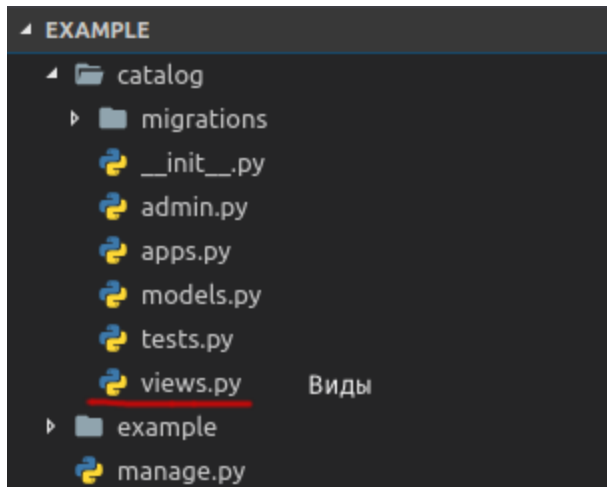
Структура приложения



Структура приложения



Структура приложения



Список приложений

Добавим новое предложение настройку `INSTALLED_APPS`

```
INSTALLED_APPS = [  
    ...,  
    'catalog'  
]
```

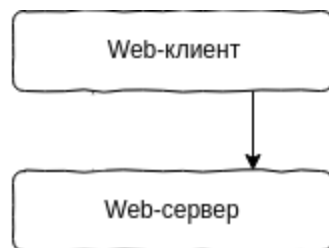

Настройки проекта Django

- `DEBUG` - показывать отладочную информацию при ошибках.
- `ALLOWED_HOSTS` - под какими доменами работает сайт в продакшене.
- `INSTALLED_APPS` - установленные Django-приложения
- `LANGUAGE_CODE` - язык проекта
- `USE_I18N` и `USE_L10N` - для локализация и интернационализация

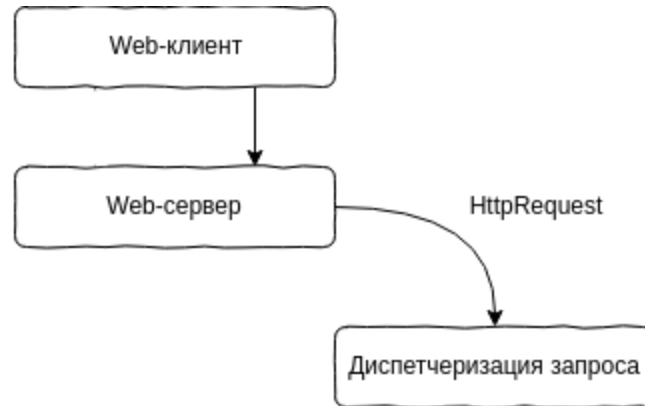
Обработка запросов

Web-клиент

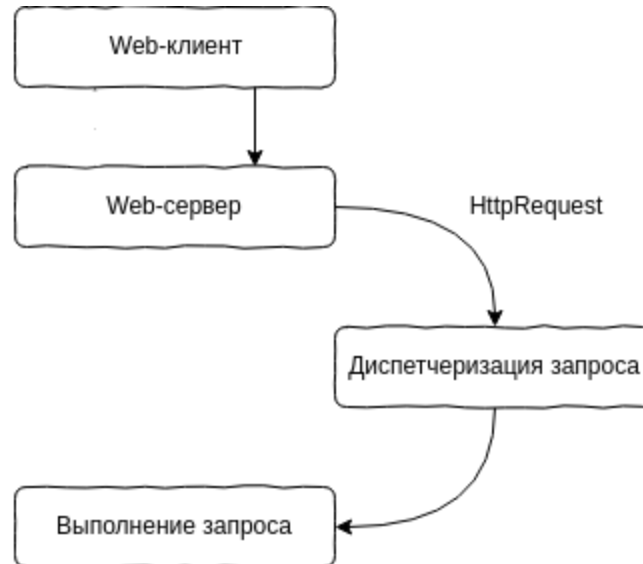
Обработка запросов



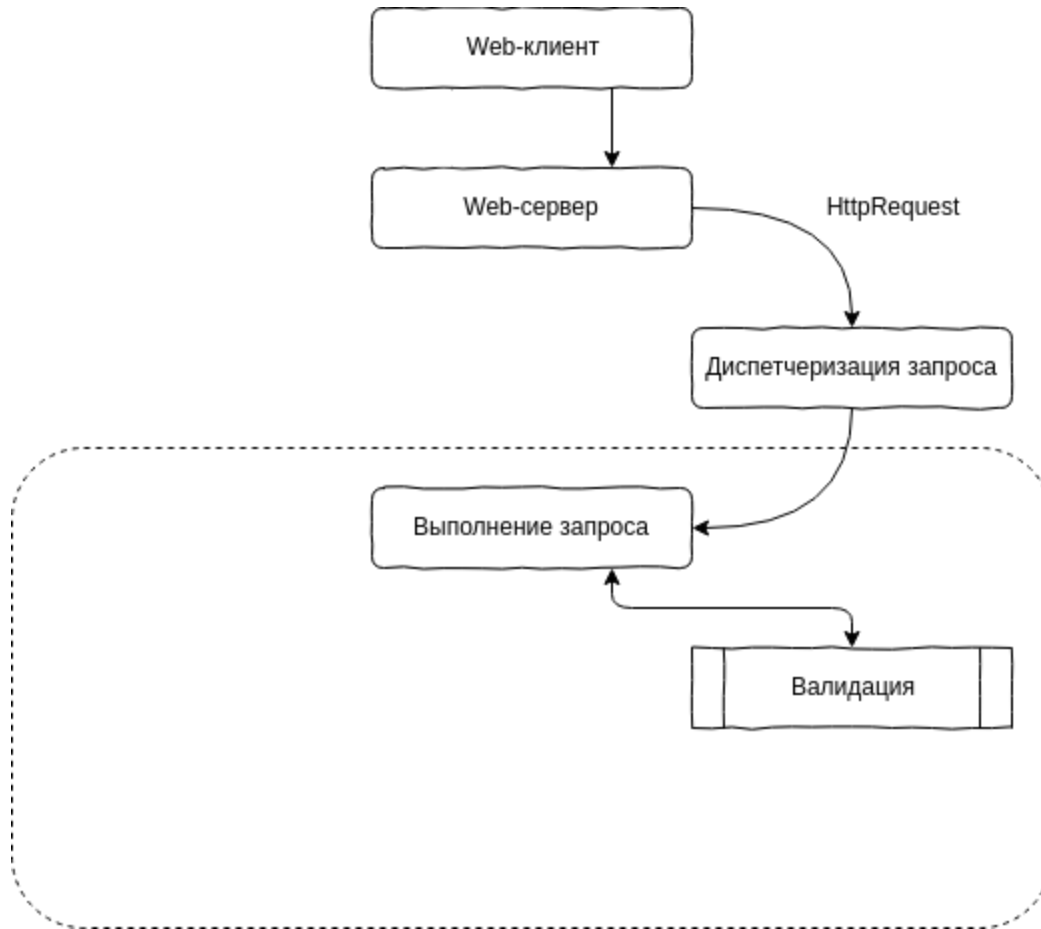
Обработка запросов



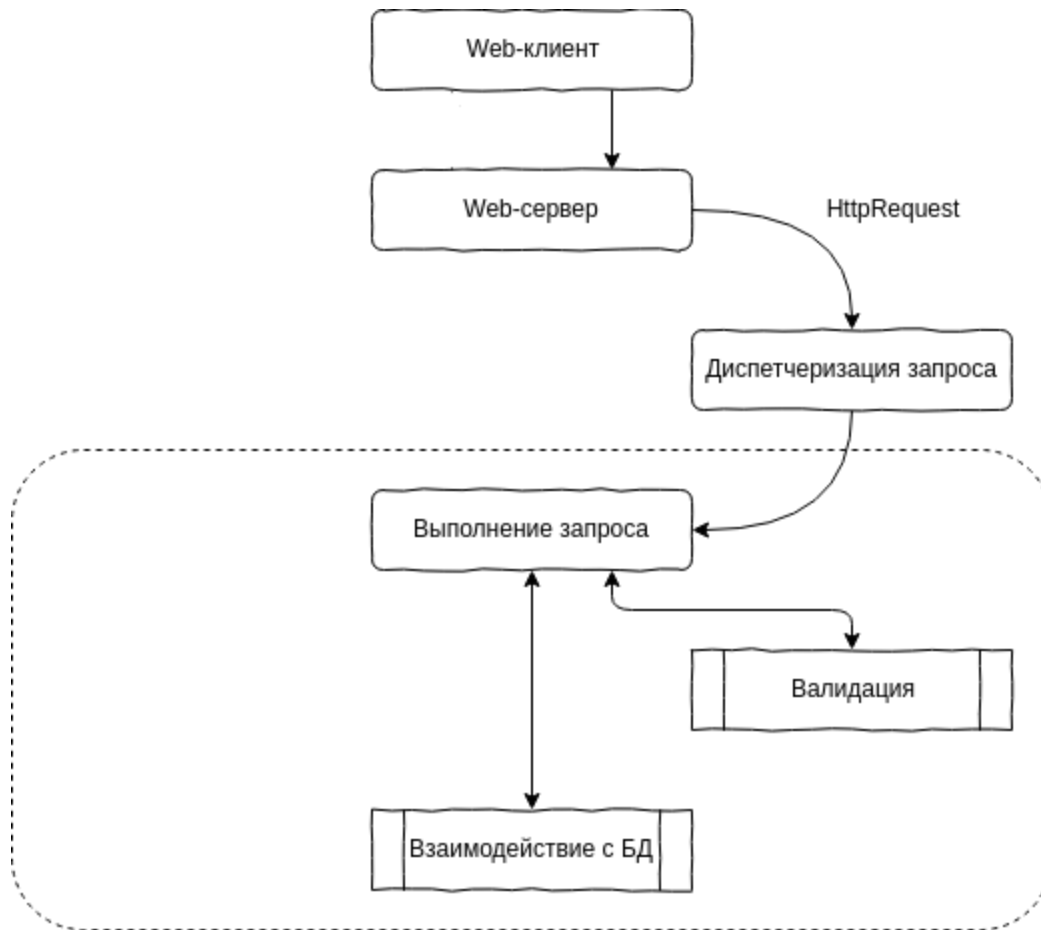
Обработка запросов



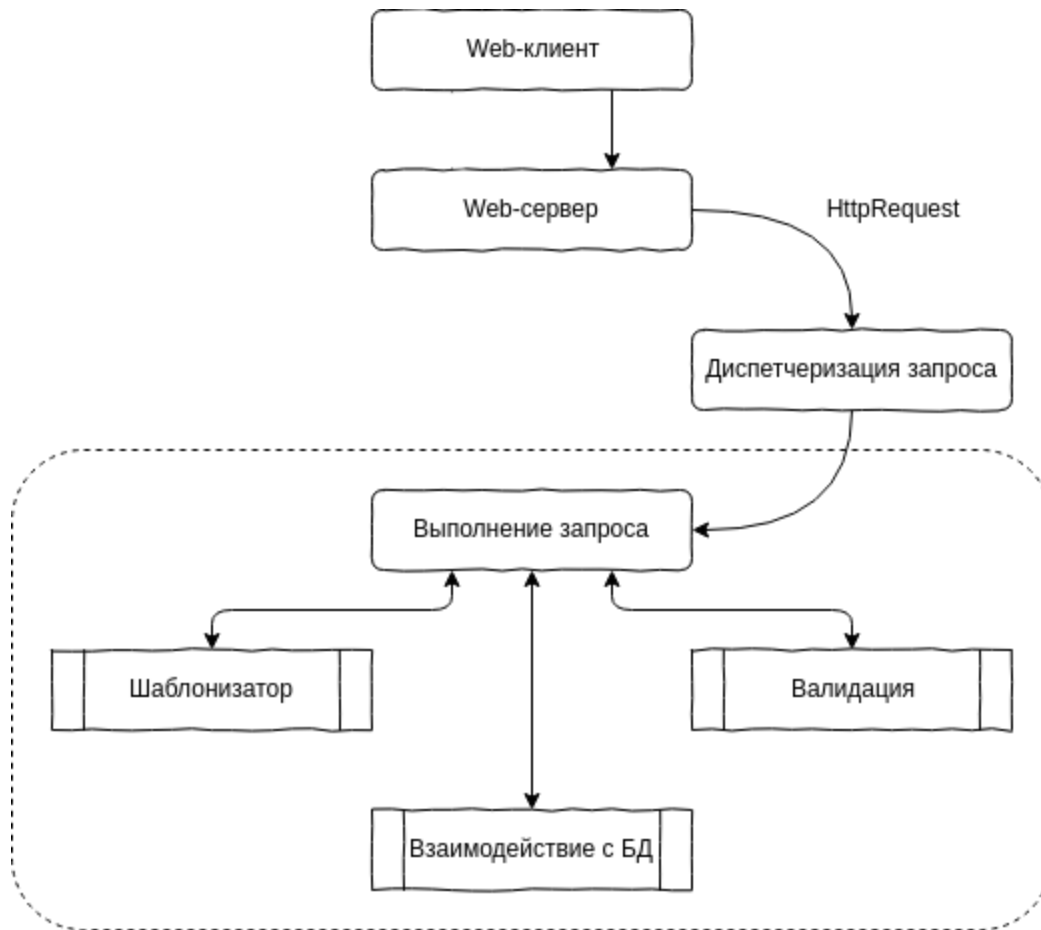
Обработка запросов



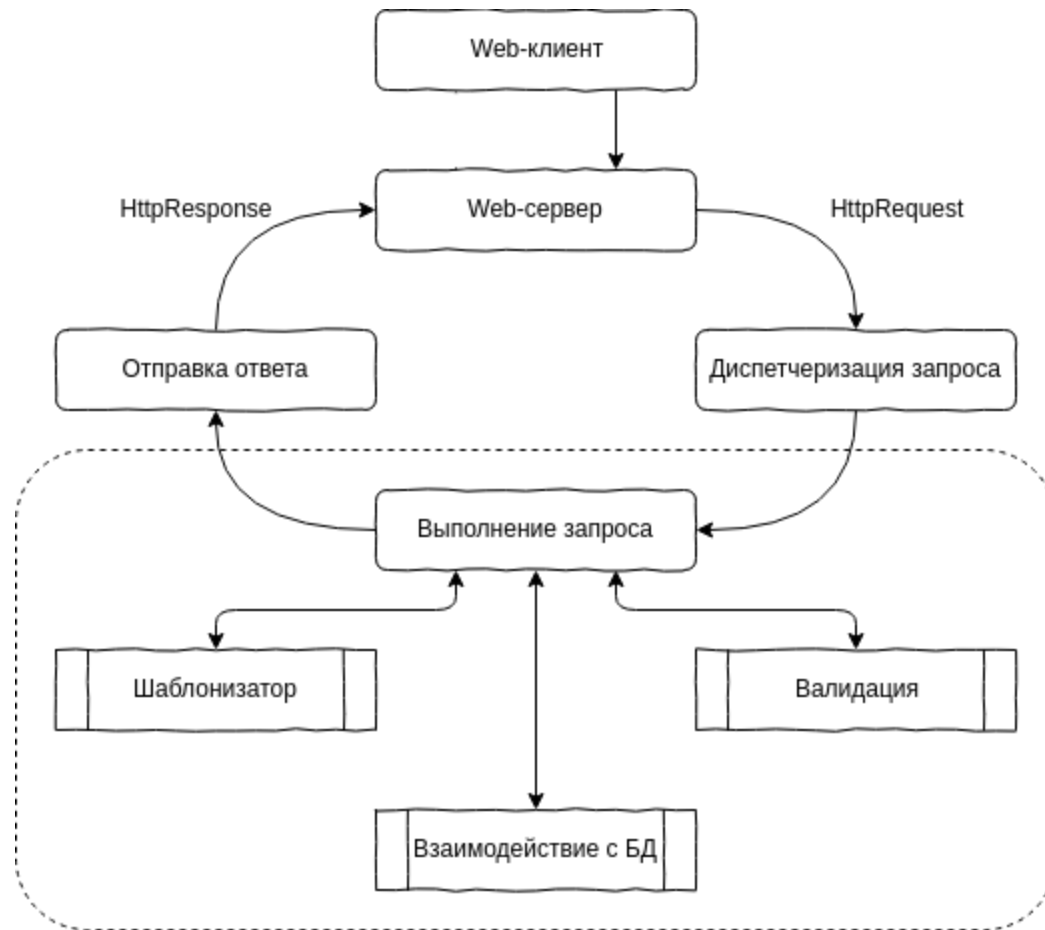
Обработка запросов



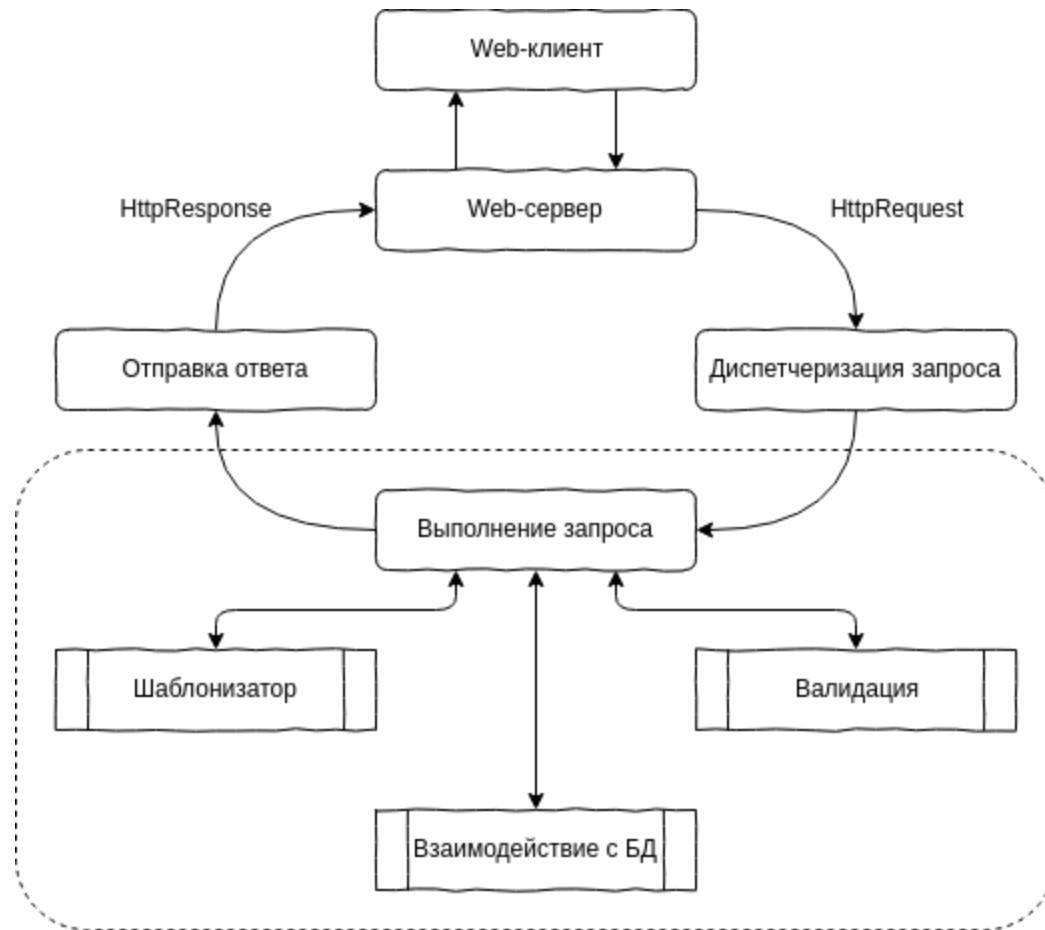
Обработка запросов



Обработка запросов



Обработка запросов



Наша первая страница: вид

`views.py`

Обработка HTTP запроса и возвращаем

```
from django.http import HttpRequest, HttpResponse

def home(request: HttpRequest) -> HttpResponse:
    return HttpResponse(
        '<b>Hello world!</b>', content_type='text/html'
    )
```

Наша первая страница: URL

`urls.py`

```
...  
from catalog import views  
  
urlpatterns = [  
    path('/', views.home)  
    ...  
]
```

Наша первая модель: описание

```
from django.db import models

class Article(models.Model):
    name = models.CharField()
    content = models.TextField()
```

Наша первая модель: БД

```
python manage.py makemigrations catalog
```

```
python manage.py migrate
```

Наша первая модель: админка

```
from django.contrib import admin
from catalog.models import Article

class ArticleAdmin(admin.ModelAdmin):
    pass

admin.site.register(Article, ArticleAdmin)
```

Наша первая модель: вид

```
def home(request):  
    article = Article.objects.get(pk=1)  
  
    return HttpResponse(  
        f'<title>{ article.title }!</title>'  
        f'<b>{ article.content }!</b>',  
        content_type='text/html'  
    )
```