**Курс: Разработка Web-приложений на Python, с применением Фреймворка Django**

**Дисциплина: Создание web-приложений с использованием фреймворка Django**

**Занятие № 50: Экзамен**

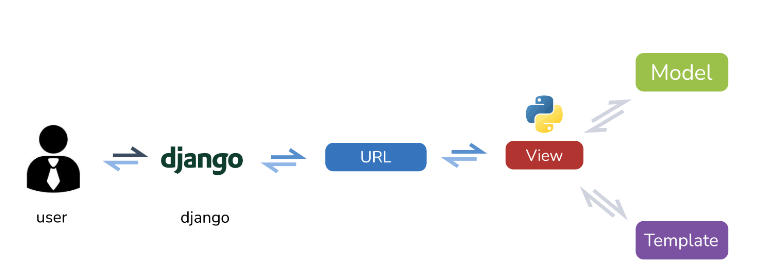
**Теоретическая часть**

**Вопросы:**

**Лёгкий уровень:**

1. Как выглядит структура django MVT паттерна (можно упрощённо)? Можно словами, можно нарисовать схему.

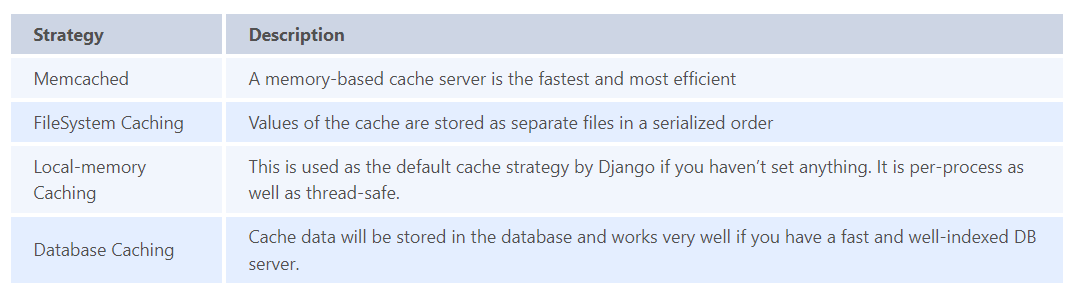
Ответ:



1. Как выглядит структура проекта и приложения в Django? Если конкретнее, то что за файлы внутри каждого из них и для чего?
2. Что такое модели в Django? Покажите пример модели и расскажите об их структуре.
3. Что такое шаблоны в Django? Покажите пример шаблона и расскажите об их структуре.
4. Что такое контроллеры или вью в Django? Покажите пример функции-контроллера и класса-контроллера и расскажите об их структуре.
5. Каким образом к Django-backend можно подключить frontend react или vue? Что за библиотека и её предназначение?
6. Что такое статические файлы и какие настройки определяют место их сбора и хранения?
7. Что такое маршруты? Каким образом можно их строить в Django? Регулярные выражения?
8. Какова функция параметра on\_delete в поле ForeignKey?
9. Что такое csrf? Как его подключить к форме?
10. Можно ли использовать команду runserver для запуска «боевого проекта» и почему нет?

**Средний уровень:**

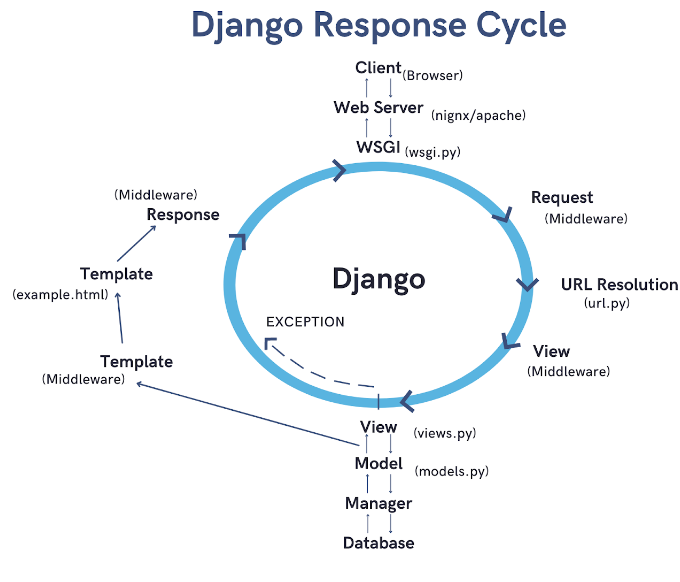
1. Что такое django сигналы? Для чего их можно использовать?
2. Что такое кеширование? Какие способы кеширования есть в Django?

Ответ:  


1. Что такое авторизация и аутентификация? Какие способы их использования есть в Django?
2. Перечислите какие базы данных поддерживает Django? Как выполнить переключение между базой PostgreSQL и SLQite?
3. Что за функция render (внутри django.shortcuts)? Что такое контекст? И что он передаёт и куда?
4. Как использовать django ORM для создания данных партиями / пачкой?
5. В чём суть промежуточного(middleware) ПО в Django? Какие есть стандартные middleware и для чего они?

**Сложный уровень**

1. Что такое django ORM? Как получить все записи с модели? Как их отфильтровать или искать внутри? Перечислите все методы ORM.
2. Разница между связью ForeignKey и OneToOne? Расскажите о примере использования обеих.
3. Что такое миксин? Напишите новый контроллер-класс и добавьте к нему одну из примесей.
4. Как выполнить чистый SQL код внутри Python и Django?
5. В чём разница между select\_related и prefetc\_related? Покажите синонемичный код на SQL.
6. Что такое объект Q? Для чего он применяется и пару вариантов его использования.
7. Какой цикл ответа и запроса в Django?

Ответ:   


**Практическая часть**

**Задание 1.**

[1 балл]

1) Создайте новое виртуальное окружение, новый проект и приложение, выполните первичные настройки проекта. Объясните параметры настройки проекта.

2) Выполните миграции, создайте суперпользователя. Накатите библиотеки django-grappelli, Pillow, cors-headers, django-environ и их настройки. Запустите приложение.

**Задание 2.**

[2 балла]

1) Создайте модель поста и комментария к посту. Объясните структуру кода моделей. Выполните миграции и покажите в админке как Вы можете создавать и удалять обе модели.

2) Выполните расширение модели пользователя через связь один к одному. Подключите сигнал для автоматического создания модели профиля.

**Задание 3.**

[2 балла]

1) Создайте шаблон, в котором задействуйте 1 встроенный фильтр и 1 тэг Django. Затем реализуйте по 1 тэгу и фильтру самостоятельно и задействуйте их. Один раз воспользуйтесь переменной и псевдонимом.

2) Выполните наследование и включение шаблонов в другие шаблоны. Объясните структуру кода шаблонов.

**Задание 4.**

[3 балла]

1) Создайте 2 маршрута, используя регулярные выражения – для постов и комментариев соответственно.

Маршрут постов должен вести на контроллер-функцию с POST и GET методами, максимально полно реализуйте функционал этого контроллера.

Маршрут комментариев должен вести на контроллер-класс с POST и GET методами, максимально полно реализуйте функционал этого контроллера.

2) Выполните вывод и отправку данных для каждого из этих контроллеров. Объясните структуру кода обоих контроллеров.

**Задание 4.**

[4 балла]

1) Создайте новый маршрут для возврата списка «незабаненных» пользователей, без регулярного выражения, также маршрут для возврата одного пользователя. Подключите библиотеку django-rest-framework и её примените её настройки. Напишите одну контроллер-функцию, которая будет реагировать на оба маршрута и выполнять методы: получить одиночного пользователя, изменить одиночного пользователя, удалить одиночного пользователя, получить массив пользователей с пагинацией и создать нового пользователя.

2) Выполните подключение библиотеки JWT для Django и её настроек, включая маршруты. Покажите в админке, как Вы получаете и используете токен.

**Комплексное задание:**

В качестве практической части экзамена выполните реализацию проекта, техническое задание которого описано ниже:

**Тема:** Веб-сайт с публикациями и комментариями к ним.

**Описание:** Весь веб-сайт нужно реализовать на паттерне Django MVT, с полного нуля. Нужна постраничная пагинация публикаций и комментариев. Подключить bootstrap, иконки и дополнительные шрифты. Реализовать «футер» и «навбар».

**Страницы:** домашняя, регистрация + вход, все публикации, детально о публикации + форма для отправки комментария, отправка публикации и редактирование публикации.

**Итогом проекта** должен быть опубликованный и рабочий проект на Heroku.

**Разрешено:**

* Смотреть код и реализацию функционала в своих старых проектах, частично брать оттуда код для изменения.
* Смотреть реализацию функционала в интернете.
* Реализовывать дополнительный функционал, ради увеличения баллов.

**Запрещено:**

* Брать код преподавателя с «репозиториев» или полностью копировать чужие проекты.