

Домашнее задание 3

Проектирование модели данных

Целью домашнего задания является проектирование модели базы данных, наполнение ее тестовыми данными и отображение этих данных на сайте. Таким образом будет создана read-only версия сайта.

1. Проектирование модели

Используя Django Models спроектировать схему вашего приложения. Создать модели для основных сущностей: вопрос, ответ, тег, профиль пользователя, лайк. Важно использовать правильные типы данных и проставить необходимые связи ([ForeignKey](#)) в моделях. Для вопросов создать свой Model Manager, в котором определить типичные выборки (“лучшие” и “новые” вопросы).

Поскольку мы будем использовать встроенную авторизацию Django (**django.contrib.auth**) - у нас будет готовая модель пользователя (**django.contrib.auth.models.User**), но необходимо создать дополнительную модель (Profile) в которую разместить дополнительные поля, например avatar. Модель Profile и User должны быть связаны один к одному.

2. ModelManager и методы моделей

В коде view не должно быть кода бизнес логики (анти-паттерн fat controller). Все методы по выборке и сортировке данных - в ModelManager. Вся логика по выводу данных, например построение полного пути к вопросу - в методы модели. Пагинация - в отдельную функцию. Вьюшки должны состоять из 4-6 строк.

3. Наполнение данными

Для удобства дальнейшей разработки нужно наполнить базу тестовыми данными. Для этого нужно создать скрипт (а лучше это сделать в виде команды приложения - Management Command), в котором добавить необходимые объекты в базу. При добавлении удобно использовать уже созданные модели, а не писать сырой SQL.

Обратите внимание, что при указании связи в моделях можно использовать как id, так и объекты на которые вы ссылаетесь:

```
u = User.objects.get(pk = 1)
q = Question(author = u, text = 'wut?') # ссылка на объект
q.save()
a = Answer(question_id = 1, text = 'blabla') # ссылка по id
a.save()
```

В базе данных в любом случае будут храниться поля вида author_id и question_id.

4. Отображение данных

Разработать view для отображения следующих страниц:

- Список новых вопросов;
- Список “лучших” вопросов;
- Список вопросов по тегу;
- Страница 1 вопроса со списком ответов.

Для страниц не существующих объектов нужно возвращать 404 страницу. В коде view (контроллеров) не должно быть дублирующегося кода. Использовать функцию пагинации созданную на предыдущем занятии.

5. Полезные ссылки

- Тutorial по [проектированию моделей в Django](#);
- Подробная [документация по моделям](#);
- Скрипты - [Management Commands](#).