

Домашнее задание 2

Обработка HTTP запросов

Целью домашнего задания является:

- создание основных view сайта (отображающих “рыбу”);
- настройка маршрутизации URL;
- использование шаблонизатора для генерации HTML;
- постраничное отображение (пагинация).

1. Отображение данных

Разработать view для отображения следующих страниц:

- список новых вопросов (главная страница) (URL = /)
- список “лучших” вопросов (URL = /hot/)
- список вопросов по тэгу (URL = /tag/blablabla/)
- страница одного вопроса со списком ответов (URL = /question/35/)
- форма логина (URL = /login/)
- форма регистрации (URL = /signup/)
- форма создания вопроса (URL = /ask/)

На данном этапе view не обязаны выбирать данные из базы, их задача - подготовить данные-заглушки (“рыбу”) и передать на отрисовку в соответствующий шаблон. Например таким образом:

```
questions=[]
for i in xrange(1,30):
    questions.append({
        'title': 'title'+str(i),
        'id': i,
        'text': 'text'+str(i)
    })
```

Далее список questions можно будет легко заменить на выборку из базы данных. Вьюшки отображающие формы (например форму логина) - могут просто вызывать шаблон.

2. Маршрутизация URL

Для всех вышеперечисленных view нужно прописать маршруты в urls.py. Маршруты должны быть именованными.

3. Шаблонизация

Для каждой из вышеперечисленных view нужно создать отдельный шаблон, отрисовывающий HTML данной страницы. В шаблонах **не должно быть** дублирования HTML - для этого следует использовать наследование шаблонов и подшаблоны (include). В шаблонах должны быть прописаны гиперссылки для переходов на другие страницы. URL **не хардкодить**, вместо этого использовать тэг url.

4. Функция пагинации

Общим функционалом для этих страниц является пагинация. Поэтому функционал пагинации следует вынести в отдельную функцию, которая будет принимать список объектов (или query-set),

объект запроса (request) и будет возвращать нужную страницу и данные для шаблона пагинатора. Некорректные параметры пагинации нужно заменять на значения по-умолчанию. Примерная сигнатура функции такова:

```
def paginate(objects_list, request):  
    #do smth with Paginator, etc...  
    return objects_page, paginator
```

5. Полезные ссылки

- Тьюториал по [Django Views](#);
- Документация по [шаблонам](#);
- [Наследование шаблонов](#);
- [Постраничный вывод \(пагинация\)](#).