

ДЗ № 3 - “Обработка HTTP запросов”

Целью домашнего задания является:

- Создание основных view сайта (отображающих “рыбу”)
- Настройка маршрутизации URL
- Использование шаблонизатора для генерации HTML
- Постраничное отображение (пагинация)

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 1 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.

Маршрутизация URL

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

Шаблонизация

Для каждой из вышеперчисленных 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 %}

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
```

Полезные ссылки

1. Тьюториал по Django Views
<http://djbook.ru/rel1.7/intro/tutorial03.html>
2. Документация по шаблонам
<http://djbook.ru/rel1.7/topics/templates.html>
3. Наследование шаблонов
<http://djbook.ru/rel1.7/topics/templates.html#template-inheritance>
4. Постраничный вывод (пагинация)
<http://djbook.ru/rel1.7/topics/pagination.html>