skypro

Шаблонизация. Шпаргалка

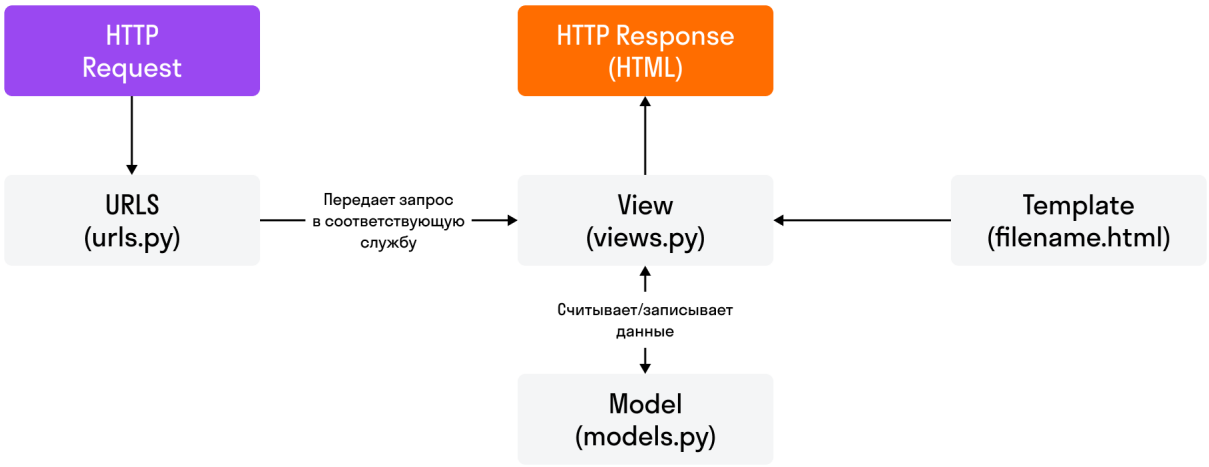
Шаблонизация в Django

Шаблонизация в Django — это процесс создания HTML-страниц на основе шаблонов, которые содержат как статический контент (HTML, CSS, JavaScript), так и динамический контент, полученный с сервера.

Шаблон в Django — это текстовый документ, внутри которого находятся:

- HTML-разметка,
- подключаемые стили,
- JavaScript-файлы.

Порядок обработки запроса в Django



Передача данных из контроллера в шаблон

Контекст в Django — это структура данных (словарь), которая передается в шаблон для отображения данных. Контекст позволяет передавать переменные из контроллера в шаблон, где они могут быть использованы для динамического формирования содержимого страницы. Такие переменные называются **шаблонными переменными**.

Использование шаблонных переменных

Контроллер, в котором вызывается шаблон:

```python
# views.py
from django.shortcuts import render
from .models import Student

def index(request):
    student = Student.objects.get(id=1)
    context = {
        'student_name': f'{student.first_name} {student.last_name}',
        'student_year': student.get_year_display(),
    }
    return render(request, 'students/index.html', context)
```

Шаблон:

```
<!-- students/index.html -->
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Student Info</title>
</head>
<body>
    <h1>Информация о студенте</h1>
    <p>Имя: {{ student_name }}</p>
    <p>Курс: {{ student_year }}</p>
</body>
</html>
```

Обращение к полям объектов в шаблонах

Точечная нотация (dot notation) — это способ доступа к атрибутам или методам объекта в объектно-ориентированном программировании. В точечной нотации используется точка (`.`) для разделения имени объекта и имени атрибута или метода.

Контроллер, в котором вызывается шаблон:

```
# views.py
from django.shortcuts import render
from .models import Student

def student_detail(request, student_id):
    student = Student.objects.get(id=student_id)
    context = {'student': student}
    return render(request, 'students/student_detail.html', context)
```

Шаблон:

```
<!-- students/student_detail.html -->
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Детали студента</title>
</head>
<body>
    <h1>Детали студента</h1>
    <p>Имя: {{ student.first_name }}</p>
    <p>Фамилия: {{ student.last_name }}</p>
    <p>Курс: {{ student.get_year_display }}</p>
    <p>Фотография: <img src="{{ student.photo.url }}" alt="{{ student.fir
st_name }} {{ student.last_name }}"></p>
</body>
</html>
```

Шаблонные теги

Шаблонные теги в Django — используются для выполнения различных операций в шаблонах, таких как условия, циклы, включение других шаблонов и многое другое. Они обрабатываются при рендеринге шаблона и позволяют добавлять логику в HTML код.

Тег if

Синтаксис тега `{% if %}`:

```
{% if условие %}
    <!-- Контент, отображаемый, если условие истинно -->
{% elif другое_условие %}
    <!-- Контент, отображаемый, если другое условие истинно -->
{% else %}
    <!-- Контент, отображаемый, если все условия ложны -->
{% endif %}
```

Пример:

```
{% if student.year == 'first' %}
    <p>Студент первого курса</p>
{% elif student.year == 'second' %}
    <p>Студент второго курса</p>
{% else %}
    <p>Студент не первого и не второго курса</p>
{% endif %}
```

Тег for

Синтаксис тега `{% for %}`:

```
{% for элемент in список %}
    <!-- Контент, отображаемый для каждого элемента в списке -->
{% endfor %}
```

Пример:

```
<ul>
    {% for student in students %}
        <li>{{ student.first_name }} {{ student.last_name }}</li>
    {% endfor %}
</ul>
```

Тег url

Пример тега `{% url %}`:

```
<a href="{% url 'mymodel_edit' mymodel.pk %}">Редактировать</a>
```

Шаблонные фильтры

Шаблонные фильтры в Django — позволяют изменять значения переменных непосредственно в шаблонах. Фильтры используются для форматирования данных, выполнения простых преобразований или для обработки строк, чисел и других типов данных перед их выводом на страницу. Они применяются с использованием символа вертикальной черты `|`.

Основные шаблонные фильтры

1. **Фильтр** `upper` — преобразует строку в верхний регистр.

   ```
   <p>{{ student.first_name|upper }}</p>
   ```

2. **Фильтр** `lower` — преобразует строку в нижний регистр.

   ```
   <p>{{ student.last_name|lower }}</p>
   ```

3. **Фильтр** `date` — форматирует дату в заданном формате.

   ```
   <p>Дата рождения: {{ student.birth_date|date:"d M Y" }}</p>
   ```

4. **Фильтр** `length` — возвращает длину списка или строки.

   ```
   <p>Количество студентов: {{ students|length }}</p>
   ```

5. Фильтр `default` — возвращает значение по умолчанию, если переменная пуста или отсутствует.

```
<p>Комментарий: {{ student.comment|default:"Комментарий отсутствует"
}}</p>
```

6. Фильтр `truncatechars` — сокращает текст до указанного количества символов.

```
<p>Описание: {{ student.description|truncatechars:100 }}</p>
```

Пример использования фильтров

Контроллер, в котором вызывается шаблон:

```
# views.py
from django.shortcuts import render
from .models import Student

def student_detail(request, student_id):
    student = Student.objects.get(id=student_id)
    context = {'student': student}
    return render(request, 'students/student_detail.html', context)
```

Шаблон:

```
<!-- students/student_detail.html -->
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.
0">
    <title>Детали студента</title>
    <!-- Подключение Bootstrap CSS -->
    <link rel="stylesheet" href="/static/css/bootstrap.min.css">
</head>
<body>
    <div class="container mt-5">
        <h1 class="mb-4">Детали студента</h1>
        <p>Имя: {{ student.first_name|upper }}</p>
        <p>Фамилия: {{ student.last_name|lower }}</p>
        <p>Дата рождения: {{ student.birth_date|date:"d M Y" }}</p>
```

```
        <p>Комментарий: {{ student.comment|default:"Комментарий отсутствует" }}</p>
        <p>Описание: {{ student.description|truncatechars:100 }}</p>
        <p>Фото:</p>
        <img src="{{ student.photo.url }}" alt="{{ student.first_name }} {{ student.last_name }}" class="img-thumbnail">
    </div>
    <!-- Подключение Bootstrap JS -->
    <script src="/static/js/bootstrap.bundle.min.js"></script>
</body>
</html>
```

Базовые шаблоны и подшаблоны

Базовый шаблон в Django — это шаблон, который служит основой для других шаблонов. Он обычно содержит общие элементы, такие как заголовок, навигационное меню, подвал и другие повторяющиеся элементы интерфейса.

Пример базового шаблона:

```
<!-- base.html -->
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>{% block title %}Мой сайт{% endblock %}</title>
    <!-- Подключение Bootstrap CSS -->
    <link rel="stylesheet" href="/static/css/bootstrap.min.css">
</head>
<body>
    <nav class="navbar navbar-expand-lg navbar-light bg-light">
        <a class="navbar-brand" href="#">Мой сайт</a>
        <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">
            <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarNav">
```

```
            <ul class="navbar-nav">
                <li class="nav-item active">
                    <a class="nav-link" href="#">Главная</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="#">О нас</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="#">Контакты</a>
                </li>
            </ul>
        </div>
    </nav>
    <div class="container mt-5">
        {% block content %}{% endblock %}
    </div>
    <!-- Подключение Bootstrap JS -->
    <script src="/static/js/bootstrap.bundle.min.js"></script>
</body>
</html>
```

Использование тегов {% extends %} и {% block %}

Для использования базового шаблона в других шаблонах используются теги `{% extends %}` и `{% block %}`.

> Шаблоны, которые используют (расширяют) другие шаблоны, называются **дочерними**.

Пример дочернего шаблона:

```
<!-- home.html -->
{% extends 'base.html' %}

{% block title %}Главная страница{% endblock %}

{% block content %}
<h1>Добро пожаловать на наш сайт!</h1>
<p>Это главная страница нашего сайта.</p>
{% endblock %}
```

Подшаблоны и использование тега {% include %}

Подшаблоны позволяют включать небольшие фрагменты шаблона в другие шаблоны. Это полезно для повторного использования общих частей страницы, таких как формы, таблицы, меню и т. д.

Пример подшаблона:

```
<!-- header.html -->
<header>
    <h1>Заголовок сайта</h1>
    <p>Подзаголовок сайта</p>
</header>
```

Использование подшаблона в основном шаблоне:

```
<!-- base.html -->
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.
0">
    <title>{% block title %}Мой сайт{% endblock %}</title>
    <!-- Подключение Bootstrap CSS -->
    <link rel="stylesheet" href="/static/css/bootstrap.min.css">
</head>
<body>
    {% include 'header.html' %}
    <nav class="navbar navbar-expand-lg navbar-light bg-light">
        <a class="navbar-brand" href="#">Мой сайт</a>
        <button class="navbar-toggler" type="button" data-toggle="collaps
e" data-target="#navbarNav" aria-controls="navbarNav" aria-expanded="fals
e" aria-label="Toggle navigation">
            <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarNav">
            <ul class="navbar-nav">
                <li class="nav-item active">
                    <a class="nav-link" href="#">Главная</a>
                </li>
```

```
                <li class="nav-item">
                    <a class="nav-link" href="#">О нас</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="#">Контакты</a>
                </li>
            </ul>
        </div>
    </nav>
    <div class="container mt-5">
        {% block content %}{% endblock %}
    </div>
    <!-- Подключение Bootstrap JS -->
    <script src="/static/js/bootstrap.bundle.min.js"></script>
</body>
</html>
```