**Тема занятия № 21: Модуль 11. Шаблоны и статические файлы**

1. Настройки проекта, касающиеся шаблонов

**Шаблон** — это образец для генерирования веб-страницы, отправляемой клиенту в ответ на его запрос. Django также использует шаблоны для формирования электронных писем.

**Рендеринг** — собственно генерирование веб-страницы (или электронного письма) на основе заданного шаблона и контекста шаблона, содержащего все необходимые данные. **Шаблонизатор** — подсистема фреймворка, выполняющая рендеринг.

Все настройки проекта, касающиеся шаблонов и шаблонизатора, записываются в параметре templates модуля settings.py из пакета конфигурации. Параметру присваивается массив, каждый элемент которого является словарем, задающим параметры одного из шаблонизаторов, доступных во фреймворке.

В каждом таком словаре можно задать следующие элементы:

□ backend — путь к модулю шаблонизатора, записанный в виде строки.

В составе Django поставляются два шаблонизатора:

• django.template.backends.django.djangotemplates — стандартный шаблонизатор, применяемый в большинстве случаев;

• dj ango. Template. Backends. J inj a2. Jinj a2 — шаблонизатор Jinja2;

□ name — псевдоним для шаблонизатора. Если не указан, то для обращения к шаблонизатору используется последняя часть пути к его модулю;

□ dirs — список путей к папкам, в которых шаблонизатор будет искать шаблоны

(по умолчанию — ’’пустой” список);

□ app dirs — если True, то шаблонизатор дополнительно будет искать шаблоны в папках templates, располагающихся в пакетах приложений. Если False, то шаблонизатор станет искать шаблоны исключительно в папках из списка dirs. Значение по умолчанию — False, однако во вновь созданном проекте устанавливается В True;

□ options — дополнительные параметры, поддерживаемые конкретным шаблонизатором. Также указываются в виде словаря, элементы которого задают отдельные параметры.

Стандартный шаблонизатор dj ango. Template. Backends. Dj ango. Dj angotemplates поддерживает такие параметры:

• autoescape — если True, то все недопустимые знаки HTML (двойная кавычка, знаки ’’меньше” и ’’больше”) при их выводе будут преобразованы в соответствующие специальные символы (поведение по умолчанию). Если False, то такое преобразование выполняться не будет;

• string if invalid— строка, выводящаяся на экран в случае, если попытка доступа к переменной контекста шаблона или вычисления выражения потерпела неудачу (значение по умолчанию — ’’пустая” строка);

• file charset— обозначение кодировки, в которой записан код шаблонов, в виде строки (по умолчанию: "utf-8”);

• context processors — список имен модулей, реализующих обработчики контекста, которые должны использоваться совместно с заданным шаблонизатором. Имена модулей должны быть заданы в виде строк.

**Обработчик контекста** — это программный модуль, добавляющий в контекст шаблона какие-либо дополнительные переменные уже после его формирования контроллером. Список всех доступных в Django обработчиков контекста будет приведен позже;

• debug — если True, то будут выводиться развернутые сообщения об ошибках в коде шаблона, если False — совсем короткие сообщения. Если параметр не указан, то будет использовано значение параметра проекта debug;

• loaders —список имен модулей, выполняющих загрузку шаблонов.

Django — в зависимости от значений параметров dirs и app dirs — сам выбирает, какие загрузчики шаблонов использовать, и явно указывать их список не требуется. На всякий случай, перечень доступных во фреймворке загрузчиков шаблонов и их описания приведены на странице https://docs.

Djangoproject.eom/en/3.0/ref/templates/api/#template-loaders;

• builtins — список строк с путями к встраиваемым библиотекам тегов, которые должны использоваться с текущим шаблонизатором. Значение по умолчанию — ’’пустой” список.

**Библиотека тегов** — это программный модуль Python, расширяющий набор доступных тегов шаблонизатора. Встраиваемая библиотека тегов загружается в память непосредственно при запуске проекта, и объявленные в ней дополнительные теги доступны к использованию в любой момент времени без каких бы то ни было дополнительных действий;

• libraries — перечень загружаемых библиотек шаблонов. Записывается в виде словаря, ключами элементов которого станут псевдонимы библиотек тегов, а значениями элементов — строковые пути к модулям, реализующим эти библиотеки. Значение по умолчанию — ’’пустой" словарь.

В отличие от встраиваемой библиотеки тегов, загружаемая библиотека перед использованием должна быть явно загружена с помощью тега шаблонизатора Load.

Параметры builtins и libraries служат для указания исключительно сторонних библиотек тегов, поставляемых отдельно от Django. Библиотеки тегов,

Входящие в состав фреймворка, записывать туда не нужно.

Теперь рассмотрим обработчики контекста, доступные в Django:

□ django.template.context\_processors. Request — добавляет В kohtekct шаблона переменную request, хранящую объект текущего запроса (в виде экземпляра класса Request);

□ django.template.context processors.csrf — добавляет в контекст шаблона переменную csrf token, хранящую электронный жетон, который используется тегом шаблонизатора csrf token;

□ django.contrib.auth.context\_processors.auth — добавляет В kohtekct шаблона переменные user и perms, хранящие соответственно сведения о текущем пользователе и его правах;

□ django.template.context\_processors.static — добавляет В kohtekct шаблона переменную static url, хранящую значение одноименного параметра проекта;

П django.template.context\_processors.media— добавляет В kohtekct шаблона переменную media url, хранящую значение одноименного параметра проекта;

□ django.contrib.messages.context\_processors.messages — добавляет В kohtekct шаблона переменные messages и default message levels, хранящие соответственно список всплывающих сообщений и словарь, сопоставляющий строковые обозначения уровней сообщений с их числовыми кодами;

□ django.template.context processors.tz— добавляет в контекст шаблона переменную time zone, хранящую наименование текущей временной зоны;

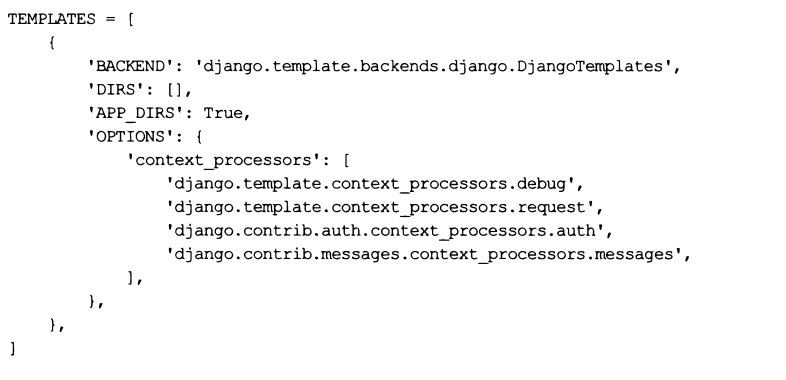
□ django.template.context\_processors.debug— добавляет в контекст шаблона переменные:

• debug —хранит значение параметра проекта debug;

• sqi queries — хранит сведения о запросах к базе данных. Представляют собой список словарей, каждый из которых представляет один запрос. Элемент Sqi такого словаря хранит SQL-код запроса, а элемент time — время его выполнения.

Может пригодиться при отладке сайта.

Код, задающий настройки шаблонов по умолчанию, которые формируются при создании нового проекта.

****

1. Вывод данных. Директивы

Для вывода данных в коде шаблона применяются директивы. Директива записывается в формате {{ <источник значения> }} и размещается в том месте шаблона, в которое нужно поместить значение из указанного источника, В его качестве можно задать:

□ переменную из контекста шаблона. Пример вставки значения из переменной Rubric:

****

□ элемент последовательности, применив синтаксис:

<переменная с последовательностью>. <индекс элемента>

Вывод первой (с индексом 0) рубрики из списка rubrics:

****

□ элемент словаря, применив синтаксис:

<переменная со словарем>. <ключ элемента>

Вывод элемента kind словаря current\_bb:

****

□ атрибут класса или экземпляра, применив привычную запись ”с точкой”. Вывод названия рубрики (атрибут паше) из переменной current rubric:

****

□ результат, возвращенный методом. Применяется запись ”с точкой”, круглые скобки не ставятся. Пример вызова метода get absolute uri о у рубрики rubric:

****

Внимание!

Шаблонизатор Django не позволяет указать параметры у вызываемого метода. Поэтому в шаблонах можно вызывать только методы, не принимающие параметров или принимающие только необязательные параметры.

□ обычные константы. Они записываются в том же виде, что и в Python-коде:

□ строки должны быть взяты в одинарные или двойные кавычки, а числа с плавающей точкой должны включать дробную часть.

Внимание!

Использовать в директивах выражения не допускается.