djangobook

djangobook

Документация Django на русском (/rel1.5/)

Внимание!

Книга написана для очень старой версии Django. Надеемся автор обновит ее и мы сможем обновить перевод. Пока советуем читать **перевод официальной документации** (/rel1.5/).

Меню (#secondary-nav)

Статические страницы

Пред. (ch14s02.html) **Глава 14. Средства от других разработчиков** След. (ch14s04.html)

Статические страницы

Часто, при использовании веб приложений с базой данных, появляется необходимость вывода нескольких статических страниц, таких как страница с информацией о сайте или страница политики безопасности. В таком случае можно использовать обычный веб сервер, например, Apache, для вывода этих страниц в виде обычных HTML файлов. Но такой подход добавит дополнительный уровень сложности в ваше приложение, потому что потребуется заботиться о конфигурации Apache, предоставлении доступа на редактирование для вашей команды и вы не сможете воспользоваться преимуществами использования шаблонной системы Django для управления видом страниц.

Решением этой проблемы является использование специального приложения, которое находится в пакете django.contrib.flatpages. Это приложение позволяет вам управлять статическими страницами через интерфейс администратора Django и указывать шаблоны для таких страниц с помощью шаблонной системы Django. Это приложение использует модели Django для своей работы, значит оно хранит страницы в базе данных, как и остальные ваши данные. Вы можете получить доступ к статическим страницам с помощью стандартного API для взаимодействия с базой данных.

Статические страницы выбираются по URL и сайту. При создании статической страницы вы указываете с каким URL она ассоциирована и с каким сайтом. (Более подробно о системе управления сайтами рассказано в секции «Сайты (ch14s02.html)».)

Использование

Для установки приложения выполните следующие шаги:

1. Добавьте django.contrib.flatpages в параметр конфигурации INSTALLED_APPS. Пакет django.contrib.flatpages зависит от пакета django.contrib.sites, удостоверьтесь, что оба этих пакета указаны в INSTALLED_APPS.

2. Добавьте django.contrib.flatpages.middleware.FlatpageFallbackMiddleware в параметр конфигурации MIDDLEWARE_CLASSES.

3. Выполните команду **manage.py syncdb** для установки двух необходимых таблиц в вашу базу данных.

Приложение создаёт две таблицы в базе данных: django_flatpage и django_flatpage_sites. Таблица django_flatpage содержит соответствия URL и страницы. Таблица django_flatpage_sites является таблицей типа «многие-ко-многим», которая связывает страницу с одним или более сайтами.

Приложение поставляется с единственной моделью `FlatPage`, определённой в `django/contrib/flatpages/models.py`:

```
from django.db import models
from django.contrib.sites.models import Site

class FlatPage(models.Model):
    url = models.CharField(max_length=100)
    title = models.CharField(max_length=200)
    content = models.TextField()
    enable_comments = models.BooleanField()
    template_name = models.CharField(max_length=70, blank=True)
    registration_required = models.BooleanField()
    sites = models.ManyToManyField(Site)
```

Рассмотрим поля модели:

url: URL, с которым ассоциирована страница, исключая доменное имя и включая начальный слэш (т.е., /about/contact/).

title: Заголовок страницы. Приложение не делает ничего особенного с заголовком. Это вам решать как отобразить его в шаблоне.

content: Содержимое страницы (т.е., HTML код страницы). Приложение не делает ничего особенного с содержимым. Это вам решать как отобразить его в шаблоне.

enable_comments: Разрешение создания комментариев для страницы. Приложение никак не реагирует на значение этого поля. Вы можете проверять значение этого поля в вашем шаблоне и отображать форму для комментария по необходимости.

template_name: Имя шаблона, который используется для отображения страницы. Необязательное поле. Если такого поля нет, то приложение будет использовать шаблон `flatpages/default.html`.

registration_required: Указание на необходимость регистрации пользователя для посещения страницы. Интегрируется с системой аутентификации пользователя, которая была описана в разделе «Аутентификация пользователей (ch12s03.html)».

sites: Сайт, для которого предназначена страницы. Интегрируется с системой управления сайтами, которая рассмотрена в разделе «Сайты (ch14s02.html)».

Вы можете создавать страницы как через интерфейс администратора Django, так и через API работы с базой данных. Более подробно это описано в разделе «Добавление, изменение и удаление (ch14s03.html#djangobook.chap14.flatpages.manage)».

После установки приложения вся работа будет выполняться с помощью класса FlatpageFallbackMiddleware. При каждом вызове ошибки 404, в качестве последнего действия, этот класс будет проверять наличие в базе данных статических страниц для требуемого URL. Главное, проверка производится с учётом сайта.

Если будет найдено совпадение, то будет произведена загрузка шаблона страницы (в крайнем случае будет загружен `flatpages/default.html`). В шаблон будет передана одна контекстная переменная, `flatpage`, которая будет ссылаться на объект. При обработке шаблона будет использоваться RequestContext.

Если не будет найдено совпадение, запрос будет обработан как обычно.

Замечание

Данный обработчик активируется только при ошибке 404 (страница не найдена), а не в случае ошибки 500 (ошибка сервера) или каких-то других. Следует отметить, что порядок значений MIDDLEWARE_CLASSES важен. Рекомендуется размещать FlatpageFallbackMiddleware в конце списка.

Добавление, изменение и удаление

Вы можете добавлять, изменять и удалять страницы двумя способами:

Через интерфейс администратора

Если вы активировали автоматический интерфейс администратора Django, вы должны увидеть на его главной странице раздел «Flatpages». Редактирование статических страниц производится аналогично редактированию любого объекта системы.

Через Python API

Как было описано ранее, статические страницы представлены в виде стандартной Django модели, которая расположена в `django/contrib/flatpages/models.py`. Следовательно, вы можете получить доступ к объектам через API работы с базой данных, например:

```
>>> from django.contrib.flatpages.models import FlatPage
>>> from django.contrib.sites.models import Site
>>> fp = FlatPage(
...     url='/about/',
...     title='About',
...     content='<p>About this site...</p>',
...     enable_comments=False,
...     template_name='',
...     registration_required=False,
... )
>>> fp.save()
>>> fp.sites.add(Site.objects.get(id=1))
>>> FlatPage.objects.get(url='/about/')
<FlatPage: /about/ -- About>
```

Использование шаблонов

По умолчанию статические страницы генерируются с использованием шаблона `flatpages/default.html`, но вы можете указать свой шаблон с помощью поля template_name объекта `FlatPage`.

Ответственность за создание шаблона `flatpages/default.html` лежит на вас. Необходимо создать каталог `flatpages` в вашем каталоге шаблонов.

Шаблоны статических страниц получают одну контекстную переменную, `flatpage`, которая содержит объект.

Пример шаблона `flatpages/default.html`:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN"
    "http://www.w3.org/TR/REC-html40/loose.dtd">
<html>
<head>
<title>{{ flatpage.title }}</title>
</head>
<body>
{{ flatpage.content }}
</body>
</html>
```

Пред. (ch14s02.html) Уровень выше (ch14.html) След. (ch14s04.html)

Сайты Начало (index.html) Перенаправления

0 comments | Оставьте комментарий (#comment-form)

Пожалуйста, произведите вход для отправки комментария (/auth/login/?next=/ch14s03.html)

Пожалуйста, дайте нам знать, если что-то не работает. (/feedback/)