**Тема занятия № 43: Модуль 25. Отправка электронных писем**

1. Настройка подсистемы отправки электронных писем

Django предоставляет развитые средства для отправки электронных писем, в том числе составных и с вложениями.

Настройки подсистемы рассылки писем записываются в модуле settings.py пакета конфигурации:

□ email backend— строка с именем класса, который реализует отправку писем.

В составе Django поставляются следующие классы-отправители писем:

• dj ango. Core, mail .backends. Smtp. Emailbackend — отправляет письма на ПОЧТОВЫЙ сервер по протоколу SMTP. Может использоваться как при разработке сайта, так и при его эксплуатации;

Следующие классы применяются исключительно при разработке и отладке сайта:

• django.core.mail.backends.filebased.emailbackend — сохраняет письма В файлах;

• django.core.mail.backends.console.emailbackend — ВЫВОДИТ письма В командной строке;

• django. Core .mail.backends. Locmem. Emailbackend — Сохраняет письма В оперативной памяти. В модуле django.core.mail создается переменная outbox, и все отправленные письма записываются в нее в виде списка;

• django. Core.mail .backends .dummy.emailbackend — никуда не отправляет, нигде не сохраняет и не выводит письма.

Значение параметра ПО умолчанию: "django. Core.mail.backends. Smtp. Emailbackend";

□ default from email— адрес электронной почты отправителя, по умолчанию указываемый в отправляемых письмах (по умолчанию: "webmasterqlocalhost").

Следующие параметры принимаются во внимание, только если в качестве класса-отправителя писем был выбран django.core.mail.backends.smtp.emailbackend:

• email host— интернет-адрес SMTP-сервера, которому будут отправляться письма (по умолчанию: "localhost");

• email port — номер TCP-порта, через который работает SMTP-сервер, в виде числа (по умолчанию: 25);

• email host user— имя пользователя для аутентификации на SMTP-сервере (по умолчанию — "пустая" строка);

• email host password— пароль для аутентификации на SMTP-сервере (по умолчанию — "пустая" строка).

Если значение хотя бы одного из параметров emailhostuser и Email host password равно "пустой" строке, то аутентификация на smtpсервере выполняться не будет;

• email use ssl — если True, то взаимодействие с SMTP-сервером будет происходить через протокол SSL (Secure Sockets Layer, уровень защищенных сокетов), если False, то таковой применяться не будет. Протокол SSL работает через ТСР-порт 465. По умолчанию — False;

• email use tls — если True, то взаимодействие с SMTP-сервером будет происходить через протокол TLS (Transport Layer Security, протокол защиты транспортного уровня), если False, то таковой применяться не будет. Протокол TLS работает через ТСР-порт 587. По умолчанию — False;

Внимание!

Можно указать значение True только для одного из параметров: email use ssl или EMAIL\_USE\_TLS.

• email ssl certfile — строка с путем к файлу сертификата. Принимается во внимание, только если для одного из параметров: email use ssl или Email use tls — было указано значение True. По умолчанию — None;

• email ssl keyfile — строка с путем к файлу с закрытым ключом. Принимается во внимание, только если для одного из параметров: email use ssl или EMAIL USE TLS-- было указано значение True. По умолчанию----None;

• email timeout — промежуток времени, в течение которого класс-отправитель будет пытаться установить соединение с SMTP-сервером, в виде целого числа в секундах. Если соединение установить не удастся, будет возбуждено исключение timeout из модуля socket. Если для параметра указать значение None, то будет использовано значение промежутка времени по умолчанию.

Значение параметра по умолчанию — None;

• email use localtime — если True, то в заголовках отправляемых писем будет указано локальное время, если False — всемирное координированное время (UTC). По умолчанию — False;

• email file path — полный путь к папке, в которой будут сохраняться файлы с письмами. Принимается во внимание, только если в качестве классаотправителя писем был выбран django.core.mail.backends, filebased.

Emailbackend. По умолчанию — "пустая" строка.

1. Низкоуровневые инструменты для отправки писем

Инструменты низкого уровня для отправки электронных писем имеет смысл применять лишь в том случае, если нужно отправить письмо с вложениями.

**Класс emailmessage: обычное электронное письмо**

Класс emailmessage из модуля django. Core .mail позволяет отправить обычное текстовое электронное письмо, возможно, с вложениями.

Конструктор этого класса принимает довольно много параметров. Все они являются именованными и необязательными:

□ subject —тема письма;

□ body —тело письма;

□ from emaii— адрес отправителя в виде строки. Также может быть записан в формате <имя отправителя> «адрес электронной почты», например: "Admin<admin@supersite. Ru>". Если не указан, то адрес будет взят из параметра Default\_from\_email настроек сайта;

О to —список или кортеж адресов получателей письма;

□ сс —список или кортеж адресов получателей копии письма;

□ Ьсс —список или кортеж адресов получателей скрытой копии письма;

□ repiy to —список или кортеж адресов для отправки ответа на письмо;

□ attachments — список вложений, которые нужно добавить в письмо. Каждое вложение может быть задано в виде:

• экземпляра класса mimebase из модуля email.mime.base Python или одного из его подклассов;

• кортежа из трех элементов: строки с именем файла, строки или объекта bytes с содержимым файла и строки с MIME-типом файла;

□ headers — словарь с дополнительными заголовками, которые нужно добавить в письмо. Ключи элементов этого словаря задают имена заголовков, а значения элементов — значения заголовков;

□ connection — объект соединения для отправки письма (его использование будет описано позже). Если не указан, то для отправки каждого письма будет установлено отдельное соединение.

Для работы с письмами класс emaiimessage предоставляет ряд методов:

□ attach () — добавляет вложение к письму. Форматы вызова метода: Attach (<объект вложения>)

Attach(<имя файла>, <содержимое файла>[, <М1МЕ-тип содержимого>])

Первый формат принимает в качестве параметра объект вложения, представленный экземпляром класса mimebase или одного из его подклассов.

Второй формат принимает имя файла, который будет сформирован в письме в качестве вложения, содержимое этого файла в виде строки или объекта bytes и строку с mime-типом содержимого файла, которую можно не указывать. Если MIME-тип не указан, то Django определит его по расширению из имени файла;

□ attach\_file (<путь к файлу>[, <М1МЕ-тип файла>]) — добавляет файл К письму в качестве вложения. Если mime-тип не указан, то Django определит его по расширению файла;

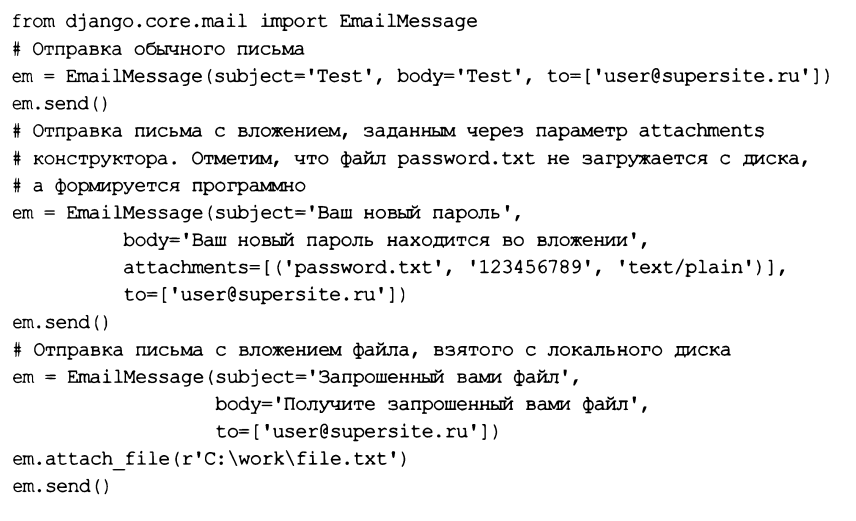
□ send( [faii\_siientiy=Faise])— отправляет письмо. Если параметру fail\_Silently дать значение True, то в случае возникновения нештатной ситуации при отправке письма никаких исключений возбуждено не будет (по умолчанию в таких случаях возбуждается исключение smtpexception из модуля smtpiib);

□ message о — возвращает экземпляр класса safemimetext ИЗ модуля django.

Core.mail, представляющий текущее письмо. Этот класс является производным от класса mimebase, следовательно, может быть указан в списке вложений, задаваемых параметром attachments конструктора класса, или в вызове метода attach ();

□ recipients () — возвращает список адресов всех получателей письма и его копий, которые указаны в параметрах to, сс и Ьсс конструктора класса.

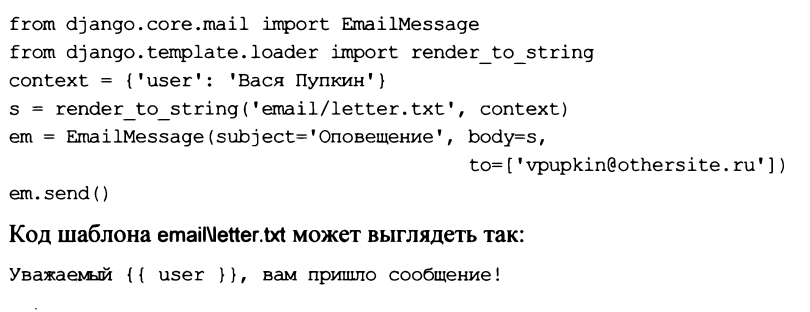
Вот примеры, демонстрирующие наиболее часто встречающиеся на практике случаи отправки электронных писем:

****

**Формирование писем на основе шаблонов**

Электронные письма могут быть сформированы на основе обычных шаблонов Django, Для этого применяется функция render\_to\_string() из модуля django. Template, loader.

Вот пример отправки электронного письма, которое формируется на основе шаблона email\letter.txt:

****

Код шаблона email\letter.txt может выглядеть так:

Уважаемый {{ user }}, вам пришло сообщение!

**Использование соединений. Массовая рассылка писем**

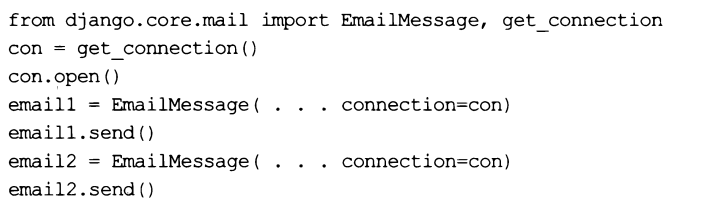
При отправке каждого письма, представленного экземпляром класса emailmessage, устанавливается отдельное соединение с SMTP-сервером. Установление соединения отнимает заметное время, которое может достигать нескольких секунд. Если отправляется одно письмо, с такой задержкой еще можно смириться, но при массовой рассылке писем это совершенно неприемлемо.

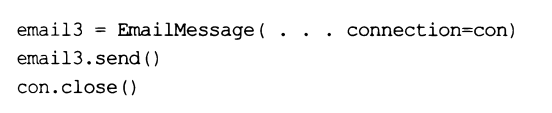
Параметр connection конструктора класса emailmessage позволяет указать объект соединения, используемый для отправки текущего письма. Мы можем задействовать одно и то же соединение для отправки произвольного количества писем, тем самым устранив необходимость каждый раз устанавливать соединение.

Получить соединение мы можем вызовом функции get\_connection() из модуля Django.core.mail. Функция в качестве результата вернет то, что нам нужно, — объект соединения.

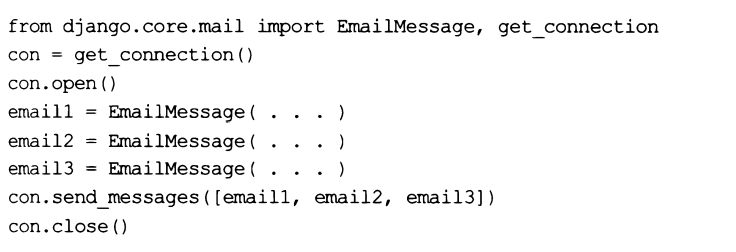
Для открытия соединения мы вызовем у полученного объекта метод open (), а для закрытия — метод close ().

Вот пример отправки трех писем с применением одного и того же соединения:

****



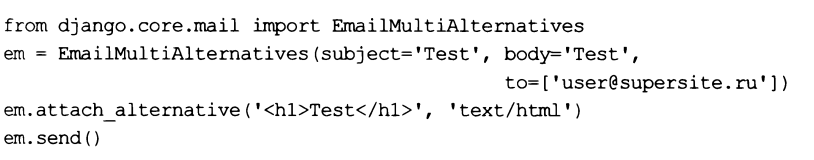
Недостатком здесь может стать то, что соединение придется указывать при создании каждого письма. Но существует удобная альтернатива — использование метода send\_messages (<отправляемые письма>) объекта соединения. СПИСОК отправляемых писем, представленных экземплярами класса emailmessage, передается методу единственным параметром. Пример:

** Класс emailmultialternatives: составное письмо**

Класс emailmultialternatives ИЗ ТОГО же модуля django.core.mail представляет составное письмо, которое состоит из нескольких частей, записанных в разных форматах. Обычно такое письмо содержит основную часть, представляющую собой обычный текст, и дополнительную часть, которая написана на языке HTML.

Класс emailmultialternatives является производным ОТ класса emailmessage и имеет тот же формат вызова конструктора. Объявленный в нем метод attach\_Alternative (<содержимое части>, <М1МЕ-тип части>) добавляет К письму новую часть с указанным в виде строки содержимым.

Пример отправки письма, которое, помимо текстовой части, содержит еще и фрагмент, написанный на HTML:

****

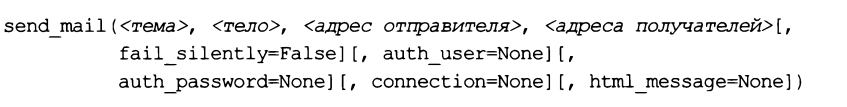
Разумеется, для формирования дополнительных частей в таких письмах можно использовать шаблоны.

1. Высокоуровневые инструменты для отправки писем

Если нет необходимости создавать письма с вложениями, то можно воспользоваться высокоуровневыми средствами отправки писем.

Отправку писем по произвольным адресам выполняют две функции из модуля Dj ango.core.mail:

□ send maiio — отправляет одно письмо, возможно, включающее HTML-часть, по указанным адресам. Формат вызова функции:

****

Адрес отправителя указывается в виде строки. Его также можно записать в формате <имя отправителя> «адрес электронной почты», например: "Admin<admin@supersite.ru>". Адреса получателей указываются В виде списка ИЛИ кортежа.

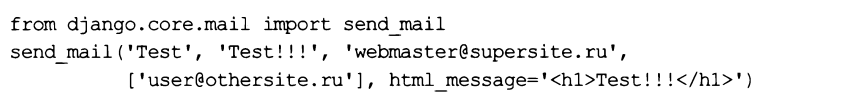
Если параметру faii silentiy задать значение True, то в случае возникновения нештатной ситуации при отправке письма никаких исключений возбуждено не будет. Значение False указывает возбудить в таких случаях исключение Smtpexception ИЗ модуля smtplib.

Параметры auth user и auth password задают соответственно имя пользователя и пароль для подключения к SMTP-серверу. Если эти параметры не заданы, то их значения будут взяты из параметров email\_host\_user и email\_host\_password настроек проекта.

Параметр connection указывает объект соединения, применяемого для отправки письма. Если он не задан, то для отправки письма будет установлено отдельное соединение.

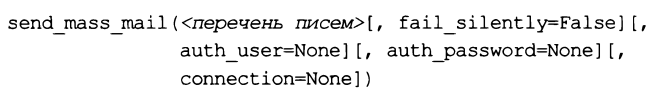
Параметр htmljnessage задает строку с HTML-кодом второй части. Если он отсутствует, то письмо будет содержать всего одну часть — текстовую.

Функция возвращает 1 в случае успешной отправки письма и о, если письмо по какой-то причине отправить не удалось. Пример:

****

□ send mass maii () — выполняет отправку писем из указанного перечня. Формат

Вызова:

****

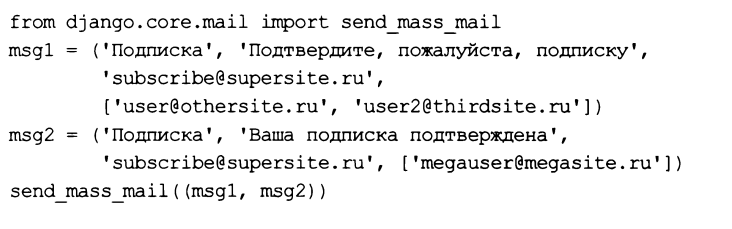
Перечень писем должен представлять собой кортеж, каждый элемент которого описывает одно из отправляемых писем и также представляет собой кортеж из четырех элементов: темы письма, тела письма, адреса отправителя и списка или кортежа адресов получателей.

Назначение остальных параметров этого метода было описано ранее, при рассмотрении метода sendmail ().

Для отправки всех писем из перечня используется всего одно соединение с SMTP-сервером.

В качестве результата метод send\_mass\_mail () возвращает количество успешно отправленных писем.

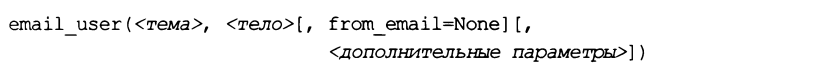
Пример:

****

**Отправка писем зарегистрированным пользователям**

Класс user, реализующий модель пользователя, предлагает метод email user (),

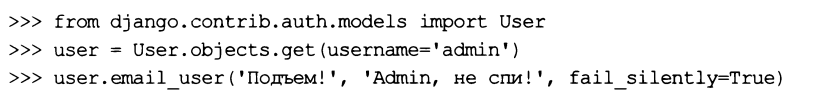
Отправляющий письмо текущему зарегистрированному пользователю:

****

Параметр from emaii указывает адрес отправителя. Если он опущен, то адрес отправителя будет взят из параметра default\_from\_email настроек проекта.

Все дополнительные параметры, указанные в вызове метода, будут без изменений переданы функции send maiio, которая служит для выполнения отправки.

Пример:

****

В этом примере параметр faii siientiy будет передан функции send mail (), используемой методом email user().

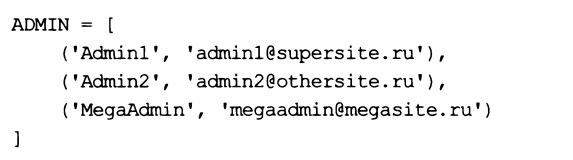
**Отправка писем администраторам и редакторам сайта**

Django может рассылать письма по адресам, указанным в двух параметрах настроек сайта. Один из этих параметров задает список адресов администраторов, получающих сообщения об ошибках в программном коде сайта, а второй — список адресов редакторов, которым будут отсылаться уведомления об отсутствующих страницах.

Обычно инструменты, рассылающие письма по этим адресам, применяются в подсистеме журналирования.

Параметры, управляющие этими инструментами, записываются в модуле settings.py пакета конфигурации:

□ admins — список администраторов. Каждый его элемент обозначает одного из администраторов и должен представлять собой кортеж из двух элементов: имени администратора и его адреса электронной почты. Пример:

****

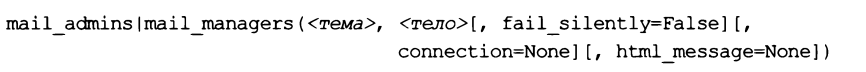
Значение по умолчанию — "пустой" список;

□ managers — список редакторов. Задается в том же формате, что и список администраторов из параметра admin. По умолчанию — "пустой" список;

□ server email — адрес электронной почты отправителя, указываемый в письмах, что отправляются администраторам и редакторам (по умолчанию: "roots Localhost");

□ email subject prefix — префикс, который добавляется к теме каждого письма, отправляемого администраторам и редакторам (по умолчанию: " [Django]").

Для отправки писем администраторам применяется функция maii admins (), для отправки писем редакторам— функция maii managers (). Обе функции объявлены в модуле django.core.mail и имеют одинаковый формат вызова:

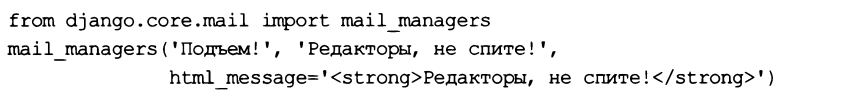
****

Если параметру faii siientiy задать значение True, то в случае возникновения нештатной ситуации при отправке письма никаких исключений возбуждено не будет.

Значение False указывает возбудить в таких случаях исключение smtpexception из модуля smtplib.

Параметр connection указывает объект соединения, применяемого для отправки письма. Если он не задан, то для отправки письма будет установлено отдельное соединение.

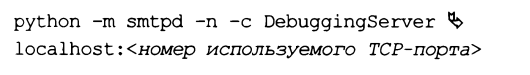
Параметр htmi message задает строку с HTML-кодом второй части. Если он отсутствует, то письмо будет содержать всего одну часть — текстовую. Пример:

****

1. Отладочный SMTP-сервер

**Отладочный SMTP-сервер**

Чтобы проверить в работе рассылку электронных писем, можно воспользоваться встроенным в Python отладочным SMTP-сервером, выводящим содержимое отправленных ему писем непосредственно в командной строке. Он запускается подачей команды формата:



Например, чтобы запустить отладочный сервер и указать ему использовать для приема писем порт № 1025, следует задать команду:



Для использования отладочного SMTP-сервера в модуле settings.py пакета конфигурации необходимо лишь указать номер требуемого ТСР-порта — в параметре Email port:



Завершить работу отладочного сервера можно, закрыв окно командной строки, в которой он был запущен.