

Страница 1 из 1

Модуль email-validate в Python, валидация и существование email

отф

Важно

Установка

На кассе — не страшно

0+

Подробнее на tinkoff.ru.  
АО «Тинькофф Банк», лицензия №2673.

3. / Модуль email-validate в Python, валидация и существование email

ет проверять, является ли EMAIL-адрес (адрес электронной почты) действительным, правильно  
но ли существует, а так же не занесен в черный список.

ен для Python 3.6 и выше!

-validate в виртуальное окружение:

## Валидация и проверка существования EMAIL-адреса.

### Базовое использование:

### Параметры:

- `email_address`: адрес электронной почты для проверки.
- `check_format`: проверяет правильность структуры адреса электронной почты. По умолчанию True.
- `check_blacklist`: проверяет электронную почту по черному списку доменов, загруженных из временных данных электронной почты DadouData (ежедневное обновление!!!); По умолчанию True.
- `check_dns`: проверяет mx-записи DNS, по умолчанию True.
- `dns_timeout`: секунды до тайм-аута DNS, по умолчанию 10 секунд.
- `check_smtp`: проверяет, действительно ли существует электронная почта, инициировав SMTP-диалог, по умолчанию True.
- `smtp_timeout`: секунды до истечения времени ожидания SMTP, по умолчанию 10 секунд
- `smtp_helo_host`: имя хоста для использования в SMTP HELO/EHLO; если установлено значение None (по умолчанию), то используется полное доменное имя локального хоста.
- `smtp_from_address`: адрес электронной почты, используемый отправителем в SMTP-опросе. Если установлено значение None (по умолчанию), то аргумент `email_address` также используется в качестве отправителя.

- `smtp_debug`: активирует отладочный вывод `smtplib`, который всегда направляется на [stderr](#), по умолчанию `False`.

## Результаты функций `validate()` и `validate_or_fail()`.

Функция `email_validate.validate()` возвращает следующие результаты:

РЕКЛАМА

⋮

- `True`: Все запрошенные проверки для данного адреса электронной почты прошли успешно.
- `False`: По крайней мере одна из запрошенных проверок не удалась для данного адреса электронной почты.
- `None`: Ни одна из запрошенных проверок не дала сбоев, но хотя бы одна из них дала неоднозначный результат. В настоящее время проверка SMTP является единственной проверкой, которая может дать неоднозначный результат.

Функция `email_validate.validate_or_fail()` работает точно так же, как `validate()`, за исключением того, что она вызывает исключение в случае сбоя проверки и неоднозначного результата вместо возврата `False` или `None` соответственно.

Все исключения в модуле происходят от `EmailValidationError`. Ниже приведены точные исключения, возникающие при различных проверках. Обратите внимание, что все классы исключений определены в модуле `validate_email.exceptions`.

Обратите внимание, что `SMTPTemporaryError` указывает на неоднозначный результат проверки, а не на сбой проверки, поэтому, если используется функция `validate_or_fail()`, то вероятно, захотите перехватить это исключение.

## Появление исключений при использовании `validate_or_fail()`.

По умолчанию, у `validate_or_fail()` все проверки включены, но каждую из них можно отключить одним из аргументов `check_...`. Обратите внимание, что аргумент `check_smtp` подразумевает `check_dns`.

- Аргумент `check_format` проверяет, соответствует ли данный адрес электронной почты общим требованиям к формату действительных адресов электронной почты. Вызывает `AddressFormatError` при любом сбое этого теста.
- Аргумент `check_blacklist` проверяет, известна ли доменная часть данного адреса электронной почты (часть за @) как одноразовый и временный домен адреса электронной почты. Они часто используются для регистрации фиктивных пользователей с целью рассылки спама или злоупотребления некоторыми услугами. Список таких доменов хранится во временных данных электронной почты `DadouData` (ежедневное обновление!!!), и этот модуль использует этот список. Вызывает ошибку `DomainBlacklistedError`, если адрес электронной почты принадлежит домену, внесенному в черный список.
- Аргумент `check_dns` проверяет, существует ли действующий список серверов, ответственных за доставку писем на указанный адрес электронной почты. Сначала выполняется DNS-запрос для домена адреса электронной почты, чтобы получить список всех записей MX. Затем этот список удаляется из дубликатов и неправильно отформатированных записей. Если по окончании этой процедуры останется хотя бы одна действующая MX-запись, проверка считается успешной.

В случае неудачи этой проверки, `validate_or_fail()` вызывает одно из следующих исключений, все из которых происходят от `DNSError`:

- `DomainNotFoundError`: Домен адреса электронной почты вообще не может быть найден.
- `NoNameserverError`: Для домена нет сервера имен.
- `DNSTimeoutError`: Истекло время ожидания при запросе сервера имен. Обратите внимание, что период ожидания можно изменить с помощью аргумента `dns_timeout`.
- `DNSConfigurationError`: Сервер имен настроен неправильно.
- `NoMXError`: Сервер имен не содержит записей MX для домена.
- `NoValidMXError`: Сервер имен перечисляет записи MX для домена, но ни одна из них не является допустимой.
- Аргумент `check_smtp` проверяет, существует ли данный адрес электронной почты, имитируя реальную доставку электронной почты. Устанавливается соединение с SMTP-сервером, идентифицированным через запись MX домена, и инициируется SMTP-диалог до момента, когда сервер подтверждает существование адреса электронной почты. После этого вместо фактической отправки электронного письма приветствие отменяется.

Модуль попытается согласовать TLS-соединение с STARTTLS и молча переключится на незашифрованное SMTP-соединение, если сервер его не поддерживает.

- Если SMTP-сервер отвечает на команду RCPT TO ответом с кодом 250 (успех), проверка считается успешной.
- Если SMTP-сервер отвечает кодом 5xx (постоянная ошибка) в любой момент диалога, проверка считается неудачной.
- Если SMTP-сервер не может подключиться, неожиданно закрывает соединение или отвечает кодом 4xx (временная ошибка) на любом этапе разговора, проверка считается неоднозначной.

Если для домена существует более одной действительной записи MX, то они проверяются в порядке приоритета до тех пор, пока в первый раз проверка не будет успешной или неудачной. Только в случае неоднозначного результата проверки пробуются следующий сервер, и только если результат проверки неоднозначен для всех серверов, неоднозначной считается и вся проверка в целом.

В случае сбоя этой проверки или неоднозначного результата, функция `validate_or_fail()` вызывает одно из следующих исключений, все из которых происходят от `SMTPError`:

- `AddressNotDeliverableError`: Сервер SMTP отказался от проверяемого адреса электронной почты. Технически это означает, что сервер ответил на команду `RCPT TO` ответом с кодом `5xx`.
- `SMTPCommunicationError`: Сервер SMTP не дает добраться до точки, где можно спросить его об адресе электронной почты. Технически это означает, что сервер отправил ответ с кодом `5xx` либо сразу после подключения, либо в ответ на команды `EHLO` (или `HELO`) или `MAIL FROM`.
- `SMTPTemporaryError`: Произошла ошибка при проверке всех доступных MX-серверов. Это считается неоднозначным результатом проверки. Например, частой причиной этого является занесение в серый список.

Все три вышеупомянутых исключения предоставляют дополнительные сведения об ответах на ошибки в переменной экземпляра исключения `error_messages`.

## Обновление источника данных `blacklist`.

[Модуль `email-validate`](#) содержит автоапдейтер для скачивания и обновления встроенного файла `blacklist.txt`. Он будет запускаться при каждой загрузке модуля (и установке), но будет пытаться обновить содержимое только в том случае, если файл старше 5 дней и если содержимое не совпадает с уже загруженным.

Обновление можно запустить вручную:

```
from validate_email.updater import update_builtin_blacklist

update_builtin_blacklist(force=False, background=True, callback=None)
```

- `force`: форсирует обновление, даже если загруженный/установленный файл достаточно свежий.
- `background`: запускает обновление в потоке, чтобы ваш код не зависал во время его обновления. Если установить для этого аргумента значение `True`, то функция вернет поток, используемый для запуска обновления, чтобы можно было присоединиться к нему, если это необходимо.
- `callback`: необязательная функция/метод, который будет вызываться после завершения обновления.