

Использование API подписки SkyDNS

Провайдеру выдается публичный ключ, который используется для аутентификации и подписывания запросов.

Все запросы выполняются через протокол HTTPS. С помощью GET запросов. Каждый запрос должен содержать публичный ключ провайдера с именем параметра **key**.

Например:

```
curl "https://www.skydns.ru/provider_api/{имя метода}?key=abcd&a=1&b=1"
```

Предупреждение

Публичный ключ передается отдельным письмом и не должен публиковаться в открытом доступе!

На данный момент доступны следующие методы API:

- [subscribe](#) - зарегистрировать пользователя в системе с переданными реквизитами и установить тариф провайдера по умолчанию.
- [subscribe_plans](#) - получить список доступных тарифов для создания/изменения пользователя реселлером.
- [activate](#) - активировать пользователя провайдера.
- [deactivate](#) - деактивировать пользователя провайдера.
- [update_email](#) - обновить email пользователя для восстановления пароля и получения информационных сообщений.
- [update_password](#) - обновить пароль пользователя.
- [prolongate](#) - включить или изменить платный тариф для пользователя. Этот метод всегда должен вызываться после метода [subscribe](#) для включения услуги пользователю.
- [unsubscribe](#) - отключить пользователя и переключить на бесплатный тариф.
- [subscription_info](#) - получить информацию о дате окончания подписки.
- [profiles](#) - получить список активных профилей пользователя.
- [create_profile](#) - создать профиль.

- [update_profile](#) - изменить параметры профиля пользователя реселлера.
- [add_ip](#) - добавить один или несколько статических IP адресов пользователя.
- [clear_ip](#) - удалить все статические IP адреса на профиле пользователя.
- [list_ip](#) - получить список всех статических IP адресов пользователя.
- [update_ip](#) - создать или обновить динамический IP адрес пользователя и привязать его к необходимому профилю фильтрации.
- [remove_ip](#) - удалить IP адрес пользователя.
- [add_vpn](#) - создать VPN-сертификат для профиля и получить профиль настройки OpenVPN.
- [clear_vpn_for_profile](#) - удалить все vpn-соединения на профиле пользователя.
- [clear_vpn_for_user](#) - удалить все vpn-соединения на всех профилях пользователя.
- [get_vpn_list](#) - получить список всех vpn-соединений пользователя.
- [remove_vpn](#) - удалить vpn-соединение пользователя.
- [update_nat](#) - добавить или обновить привязку профиля к DNS адресам для пользователей за NAT.
- [get_activity](#) - получить информацию об активности пользователя за определенную дату.
- [get_activity_report](#) - получить отчет об активности пользователя за определенный период времени.
- [get_list_activity_report](#) - отчет об активности пользователя за определенный период времени.
- [get_popular_report](#) - получить отчет о популярных запросах пользователя за определенный период времени.
- [get_category_report](#) - получить отчет по категориям пользователя за определенный период времени.
- [send_monthly_stat](#) - отправить статистику за месяц пользователю по электронной почте.
- [get_daily_stat](#) - получить статистику пользователя за день.
- [get_active_users](#) - получить список активных пользователей и их тарифы.

Внимание

API доступно по адресу: https://www.skydns.ru/provider_api/

Примечание

Вызывается по схеме: https://www.skydns.ru/provider_api/{имя метода}?{GET параметры запроса}

Доступные методы

subscribe

Зарегистрировать пользователя в системе с переданными реквизитами и установить тариф по умолчанию.

Передается 1 обязательный параметр:

- **password** - пароль создаваемого пользователя.

Может передаваться 4 необязательных параметра:

- **login** - логин, по умолчанию стандартное именование с уникальным префиксом провайдера.
- **plan** - тарифный план. Список доступных для вас планов уточняйте у менеджера СкайДНС или с помощью метода `subscribe_plans`. Если вы используете недопустимый код, то в ответ будет выдана ошибка.
- **email** - адрес электронной почты пользователя (может использоваться для самостоятельного восстановления пароля пользователем через личный кабинет и получения служебных уведомлений).
- **customer** - наименование договора.

Ответ возвращается в формате JSON следующего вида:

```
{
  "response": {
    "status": "ok",
    "data": {
      "username": "username",
      "email": "email",
      "plan": "tariff_name"
    }
  }
}
```

subscribe_plans

Получить список доступных тарифов для создания/изменения пользователя реселлером.

Ответ возвращается в формате JSON следующего вида:

```
{
  "response": {
    "status": "ok",
    "data": {
      "plan1": "tariff_name1",
      "plan2": "tariff_name2"
    }
  }
}
```

activate

Активировать пользователя провайдера.

Передается 1 обязательный параметр:

- **ident** - login пользователя, которого необходимо активировать.

Ответ возвращается в формате JSON следующего вида:

```
{
  "response": {
    "status": "ok"
  }
}
```

deactivate

Деактивировать пользователя провайдера.

Передается 1 обязательный параметр:

- **ident** - login пользователя, которого необходимо деактивировать.

Ответ возвращается в формате JSON следующего вида:

```
{
  "response": {
    "status": "ok"
  }
}
```

```
}
```

update_email

Обновить email пользователя для восстановления пароля и получения информационных сообщений.

Передается 2 обязательных параметра:

- **ident** - login пользователя, у которого изменяется email.
- **email** - новый email пользователя. Если в базе уже есть пользователь с таким email, будет возвращена ошибка.

Ответ возвращается в формате JSON следующего вида:

```
{
  "response": {
    "status": "ok"
  }
}
```

update_password

Обновить пароль пользователя.

Передается 2 обязательных параметра:

- **ident** - login пользователя.
- **password** - новый пароль пользователя.

Ответ возвращается в формате JSON следующего вида:

```
{
  "response": {
    "status": "ok"
  }
}
```

prolongate

Включить или изменить платный тариф для пользователя. Этот метод всегда должен вызываться после метода `subscribe` для включения услуги пользователю.

Передается 1 обязательный параметр:

- **ident** - login пользователя, у которого включается платная подписка.

Может передаваться 1 необязательный параметр:

- **plan** с доступными значениями: PREMIUM, SCHOOL, BUSINESS. Список доступных для вас планов уточняйте у менеджера СкайДНС или с помощью метода `subscribe_plans`. Если вы используете недопустимый код, то в ответ будет выдана ошибка.

Если параметр **plan** не передан, то по умолчанию включается план PREMIUM.

Ответ возвращается в формате JSON следующего вида:

```
{
  "response": {
    "status": "ok"
  }
}
```

unsubscribe

Отключить пользователя и переключить на бесплатный тариф.

Передается 1 параметр:

- **ident** - login пользователя, аккаунт которого переключается в бесплатный режим.

Ответ возвращается в формате JSON следующего вида:

```
{
  "response": {
    "status": "ok"
  }
}
```

subscription_info

Получить информацию о дате окончания подписки.

Передается 1 параметр:

- **ident** - login пользователя, у которого проверяется срок подписки.

Ответ возвращается в формате JSON следующего вида:

```
{
  "response": {
    "status": "ok",
    "data": {
      "date_end": 1390930655
    }
  }
}
```

Примечание

Дата окончания в unix time.

Предупреждение

В настоящее время не используется, зарезервировано для будущего использования.

profiles

Получить список активных профилей пользователя.

Передается 1 обязательный параметр:

- **ident** - login пользователя, у которого необходимо получить список профилей.

Ответ возвращается в формате JSON следующего вида:

```
{
  "response": {
    "status": "ok",
```

```
    "data": {
      "ид профиля 1": "имя профиля",
      "ид профиля 2": "имя профиля",
      "ид профиля 3": "имя профиля"
    }
  }
}
```

Примечание

“имя профиля” передается в кодировке utf-8.

create_profile

Создать профиль.

Передается 2 обязательных параметра:

- **ident** - login пользователя.
- **name** - имя профиля.

Дополнительно можно передать 2 необязательных параметра:

- **tls** - будет ли использоваться TLS. Принимает значения: True, False.
- **blockpage_id** - id страницы блокировки, которую необходимо установить на профиль.

Ответ возвращается в формате JSON следующего вида:

```
{
  "response": {
    "status": "ok"
  }
}
```

update_profile

Изменить параметры профиля пользователя реселлера.

Передается 1 обязательный параметр:

- **profile_id** – id профиля пользователя, который необходимо изменить.

Дополнительно можно передать 2 необязательных параметра:

- **name** – новое имя профиля пользователя.
- **tls** – булево значение для установки профилю, будет ли использоваться TLS.
Принимает значения: True, False.

Ответ возвращается в формате JSON следующего вида:

```
{
  "response": {
    "status": "ok",
    "data": {
      "id": "profile_id",
      "name": "profile_name",
      "tls": "True|False"
    }
  }
}
```

add_ip

Добавить один или несколько статических IP адресов пользователя.

Передается 2 обязательных параметра:

- **ident** - login пользователя.
- **ip** - один или несколько IP адресов. При передаче нескольких адресов, каждое значение нужно указывать отдельным параметром.

Дополнительно можно передать 2 необязательных параметра:

- **profile** - числовой id профиля, к которому необходимо привязать IP адреса.
- **comment** - комментарий к IP адресу.

Если параметр **profile** не будет передан, то IP адреса будут привязаны к профилю “По умолчанию”.

Ответ возвращается в формате JSON следующего вида:

```
{
  "response": {
    "status": "ok",
    "data": {
      "added_addresses": ["1.1.1.1", "1.1.1.2"]
    }
  }
}
```

Если какие-то из переданных адресов не удалось добавить, то дополнительно будет выведен их список:

```
{
  "response": {
    "status": "ok",
    "data": {
      "added_addresses": ["1.1.1.1", "1.1.1.2"],
      "invalid_addresses": [
        {"1.invalid.ip.address": "IP address is invalid"},
        {"0.0.0.0": "This address is not public"}
      ]
    }
  }
}
```

clear_ip

Удалить все статические IP адреса на профиле пользователя.

Передается 2 обязательных параметра:

- **ident** - login пользователя.
- **profile** - числовой id профиля.

Ответ возвращается в формате JSON следующего вида:

```
{
  "response": {
    "status": "ok"
  }
}
```

list_ip

Получить список всех статических IP адресов пользователя.

Передается 1 обязательный параметр:

- **ident** - login пользователя, у которого необходимо получить IP адреса.

Дополнительно можно передать 1 необязательный параметр:

- **profile** - числовой id профиля, для которого необходимо получить IP адреса.

Если параметр **profile** не будет передан, то IP адреса будут получены для всех профилей.

Ответ возвращается в формате JSON следующего вида:

```
{
  "response": {
    "status": "ok",
    "data": {
      "ip": [
        {"address": "1.2.3.4", "comment": "example.com"},
        {"address": "1.2.3.5", "comment": "example.com"}
      ],
    }
  }
}
```

update_ip

Создать или обновить динамический IP адрес пользователя и привязать его к необходимому профилю фильтрации.

Передается 2 обязательных параметра:

- **ident** - login пользователя.
- **ip** - новый динамический IP адрес пользователя.

Предоставленный IP адрес будет установлен на профиле Основной указанного пользователя. Если предоставленный IP уже привязан к другому пользователю провайдера, то привязка будет убрана и IP адрес будет привязан к указанному в методе пользователю.

Дополнительно можно передать 2 необязательных параметра:

- **hostname** - позволяет привязать несколько IP адресов к одному профилю, а также профилям, отличным от профиля Основной. При последующих обновлениях адреса

привязка к профилю будет сохраняться.

- **profile** - числовой id профиля к которому необходимо привязать IP адрес пользователя. Если данный параметр будет опущен, то по умолчанию привязка произойдет на профиль Основной. Числовые коды профилей доступны в личном кабинете в списке профилей.

Ответ возвращается в формате JSON следующего вида:

```
{
  "response": {
    "status": "ok"
  }
}
```

remove_ip

Удалить IP адрес пользователя.

Передается 2 обязательных параметра:

- **ident** - login пользователя.
- **ip** - IP адрес пользователя, который необходимо удалить.

Ответ возвращается в формате JSON следующего вида:

```
{
  "response": {
    "status": "ok"
  }
}
```

add_vpn

Создать VPN-сертификат для профиля и получить профиль настройки OpenVPN.

Передается 3 обязательных параметра:

- **ident** - login пользователя, которому запрашивается профиль настройки OpenVPN.
- **name** - уникальное для пользователя название создаваемого VPN-подключения.

- **profile_id** - идентификатор профиля пользователя, для которого запрашивается профиль настройки OpenVPN. Если **profile_id** не соответствует ни одному из профилей пользователя, будет получена соответствующая ошибка.

Ответ возвращается в формате JSON следующего вида:

```
{
  "response": {
    "status": "ok",
    "data": {
      "ovpn": "client\nremote vpn.skydns.ru 1194 udp\nnobind\ndev tun\npersist-tun\npersist-key"
    }
  }
}
```

Примечание

Если лимит VPN-соединений исчерпан, попытка добавить новое завершится ошибкой.

clear_vpn_for_profile

Удалить все vpn-соединения на профиле пользователя.

Передается 1 обязательный параметр:

- **profile_id** - числовой id профиля, у которого необходимо удалить все vpn-соединения.

Ответ возвращается в формате JSON следующего вида:

```
{
  "response": {
    "status": "ok"
  }
}
```

clear_vpn_for_user

Удалить все vpn-соединения на всех профилях пользователя.

Передается 1 обязательный параметр:

- **ident** - login пользователя, у которого необходимо удалить vpn-соединения.

Ответ возвращается в формате JSON следующего вида:

```
{
  "response": {
    "status": "ok"
  }
}
```

get_vpn_list

Получить список всех vpn-соединений пользователя.

Передается 1 обязательный параметр:

- **ident** - login пользователя, у которого необходимо получить список vpn соединений.

Ответ возвращается в формате JSON следующего вида:

```
{
  "response": {
    "status": "ok",
    "data": [
      {"id": "vpn_id1", "profile": "profile_name", "ip": "vpn_hostname1", "name": "vpn_name"},
      {"id": "vpn_id2", "profile": "profile_name", "ip": "vpn_hostname2", "name": "vpn_name"}
    ]
  }
}
```

remove_vpn

Удалить vpn-соединение пользователя.

Передается 1 обязательный параметр:

- **id** - id vpn соединения, которое необходимо удалить.

Ответ возвращается в формате JSON следующего вида:

```
{
  "response": {
    "status": "ok"
  }
}
```

update_nat

Добавить или обновить привязку профиля к DNS адресам для пользователей за NAT.

Передается 2 обязательных параметра:

- **profile_id** - id профиля пользователя.
- **address** - IP адрес NAT DNS.

Ответ возвращается в формате JSON следующего вида:

```
{
  "response": {
    "status": "ok"
  }
}
```

get_activity

Получить информацию об активности пользователя за определенную дату.

Передается 1 обязательный параметр:

- **ident** - login пользователя, для которого необходимо получить информацию.

Дополнительно можно передать 2 необязательных параметра:

- **date** - дата, за которую необходимо получить отчет о действиях пользователя в формате YYYY-MM-DD. Если параметр не задан, то отчет будет сформирован за сегодня.
- **profile_id** - идентификатор профиля пользователя, для которого запрашивается отчет.

Значение параметра **date** должно быть не ранее, чем 1 января предыдущего года.

Ответ возвращается в формате JSON следующего вида:

```
{
  "response": {
    "status": "ok",
    "data": {
      "requests": "2600",
      "blocks": "581"
    }
  }
}
```

Примечание

Где **requests** - это общее количество запросов за указанную дату, **blocks** - количество блокировок.

get_activity_report

Получить отчет об активности пользователя за определенный период времени.

Передается 3 обязательных параметра:

- **ident** - login пользователя, для которого необходимо получить информацию.
- **start** - дата начала периода для получения статистики в формате YYYY-MM-DD.
- **end** - дата окончания периода для получения статистики в формате YYYY-MM-DD.

Дополнительно можно передать 1 необязательный параметр:

- **profile_id** - идентификатор профиля пользователя, для которого запрашивается отчет.

Если диапазон **start** - **end** охватывает более чем 30 суток, то в отчет попадают только последние 30 из них.

Начало периода **start** должно быть не ранее, чем 1 января предыдущего года.

Ответ возвращается в формате JSON следующего вида:

```
{
  "response": {
    "status": "ok",
```

```
    "data": {
      "labels": [
        "2016-06-29 14: 00: 00",
        "2016-06-29 15: 00: 00"
      ],
      "datasets": [
        {
          "label": "Requests",
          "data": [375, 275]
        },
        {
          "label": "Blocks",
          "data": [13, 0]
        }
      ]
    }
  }
}
```

Примечание

Где **labels** - список значений для оси времени. **datasets** - список словарей содержащих наборы данных по определенным параметрам отчета, *label* - название параметра отчета, *data* - список значений соответствующих датам из списка **labels**.

Примечание

В **datasets** используются следующие наборы данных: **Requests** - количество запросов, **Blocks** - количество блокировок.

get_list_activity_report

Получить отчет об активности пользователя за определенный период времени.

Передается 3 обязательных параметра:

- **ident** - login пользователя, для которого необходимо получить информацию (в запросе может быть несколько).
- **start** - дата начала периода для получения статистики в формате YYYY-MM-DD.
- **end** - дата окончания периода для получения статистики в формате YYYY-MM-DD.

Пример запроса:

https://skydns.ru/provider_api/get_list_activity_report/?key=supersecretkey&start=2023-04-01&end=2023-04-17&ident=username1&ident=username2

Ответ возвращается в формате JSON следующего вида:

```
{
  "data": {
    "2023-04-03": {
      "username1": {
        "visits": 28251,
        "blocks": 1106
      },
      "username2": {
        "visits": 4854,
        "blocks": 760
      }
    },
  }
}
```

get_popular_report

Получить отчет о популярных запросах пользователя за определенный период времени.

Передается 3 обязательных параметра:

- **ident** - login пользователя, для которого необходимо получить информацию.
- **start** - дата начала периода для получения статистики в формате YYYY-MM-DD.
- **end** - дата окончания периода для получения статистики в формате YYYY-MM-DD.

Дополнительно можно передать 1 необязательный параметр:

- **profile_id** - идентификатор профиля пользователя, для которого запрашивается отчет.

Если диапазон **start** - **end** охватывает более чем 30 суток, то в отчет попадают только последние 30 из них.

Начало периода **start** должно быть не ранее, чем 1 января предыдущего года.

Ответ возвращается в формате JSON следующего вида:

```
{
  "response": {
```

```
"status": "ok",
"data": {
  "labels": ["example.com", "google.com", "asdfg.com"],
  "datasets": [
    {
      "label": "Requests",
      "data": [630, 474, 290]
    },
    {
      "label": "NXdomain",
      "data": [0, 0, 290]
    },
    {
      "label": "Blocks",
      "data": [630, 0, 0]
    }
  ]
}
```

Примечание

Где **labels** - список значений для оси времени. **datasets** - список словарей содержащих наборы данных по определенным параметрам отчета, *label* - название параметра отчета, *data* - список значений соответствующих датам из списка **labels**.

Примечание

в **datasets** используются следующие наборы данных: **Requests** - количество запросов, **Blocks** - количество блокировок, **NXdomain** - количество неразрешенных доменов.

get_category_report

Получить отчет по категориям пользователя за определенный период времени.

Передается 3 обязательных параметра:

- **ident** - login пользователя, для которого необходимо получить информацию.
- **start** - дата начала периода для получения статистики в формате YYYY-MM-DD.
- **end** - дата окончания периода для получения статистики в формате YYYY-MM-DD.

Дополнительно можно передать 1 необязательный параметр:

- **profile_id** - идентификатор профиля пользователя, для которого запрашивается отчет.

Если диапазон **start** - **end** охватывает более чем 30 суток, то в отчет попадают только последние 30 из них.

Начало периода **start** должно быть не ранее, чем 1 января предыдущего года.

Ответ возвращается в формате JSON следующего вида:

```
{
  "response": {
    "status": "ok",
    "data": {
      "Фильмы и видео онлайн": "5",
      "Файловые архивы": "2",
      "Дом, семья, хобби": "3"
    }
  }
}
```

Примечание

Где **data** - представляет собой словарь, *ключ* - название категории, *значение* - количество посещений.

send_monthly_stat

Отправить пользователю по электронной почте статистику за месяц.

Передается 3 обязательных параметра:

- **ident** - login пользователя.
- **year** - год в формате YYYY.
- **month** - месяц в формате MM, за который необходимо сформировать отчет.

Дополнительно можно передать 1 необязательный параметр:

- **profile_id** - идентификатор профиля пользователя, для которого запрашивается статистика.

На email пользователя будет выслан отчет в формате CSV со статистикой за указанный месяц. Отчет содержит следующие столбцы: "Timestamp", "Domain name", "Visits", "Blocks", "Profile", "Categories".

Значение параметра **year** должно быть не меньше прошлого года.

Ответ возвращается в формате JSON следующего вида:

```
{
  "response": {
    "status": "ok"
  }
}
```

get_daily_stat

Получить статистику пользователя за день.

Передается 2 обязательных параметра:

- **date** - дата в формате 'YYYY-MM-DD' определяет день, за который необходима статистика.
- **ident** - имя пользователя, для которого необходима статистика.

Дополнительно можно передать 2 необязательных параметра:

- **email_to** - email, на который будет отправлена статистика.
- **profile_id** - идентификатор профиля пользователя, для которого запрашивается статистика.

Значение параметра **date** должно быть не ранее, чем 1 января предыдущего года.

Ответ возвращается в формате JSON следующего вида:

```
{
  "response": {
    "status": "ok",
    "data": {
      "result": "ссылка на файл"
    }
  }
}
```

get_active_users

Получить список активных пользователей и их тарифы.

Ответ возвращается в формате JSON следующего вида:

```
{
  "data": [
    {
      "username": "username1",
      "plan_name": "Бизнес"
    },
    {
      "username": "username1",
      "plan_name": "Школа"
    },
  ]
}
```

Revision #2

Created 8 December 2023 14:03:23 by Виктор

Updated 8 October 2024 06:26:01 by Leo