

# Публичный API VDSina

Общая информация .....	1
Получение и использование токена .....	2
Описание параметров и возвращаемых ошибок .....	3
Список доступных действий .....	4
<b>Информация об аккаунте и прогнозе отключения, GET /v1/account</b> .....	4
<b>Баланс аккаунта, GET /v1/account.balance</b> .....	4
<b>Лимиты аккаунта, GET /v1/account.limit</b> .....	4
<b>Регистрация нового аккаунта, POST /v1/register</b> .....	5
<b>Список групп тарифных планов, GET /v1/server-group</b> .....	6
<b>Список дата-центров, GET /v1/datacenter</b> .....	6
<b>Список шаблонов ОС, GET /v1/template</b> .....	6
<b>Список тарифных планов для серверов, GET /v1/server-plan/ID</b> .....	8
<b>SSH-ключи для серверов</b> .....	10
<b>ISO для серверов</b> .....	10
<b>Серверы</b> .....	11
<b>Перезагрузка сервера, PUT /v1/server.reboot/ID</b> .....	14
<b>Переустановка сервера, PUT /v1/server.reinstall/ID</b> .....	15
<b>Установка пароля сервера, PUT /v1/server.password/ID</b> .....	15
<b>Изменение тарифного плана сервера, PUT /v1/server.plan/ID</b> .....	15
<b>Продление сервера, PUT /v1/server.prolong/ID</b> .....	16
<b>VNC-подключение, GET /v1/server.vnc/ID</b> .....	16
<b>Резервная копия сервера</b> .....	16
<b>Настройка резервных копий сервера по расписанию</b> .....	17
<b>Подключение и отключение ISO для сервера</b> .....	17
<b>Дополнительные IP-адреса для сервера</b> .....	18
<b>Локальный IP-адрес для сервера</b> .....	18
<b>Статистика сервера, GET /v1/server.stat/ID</b> .....	19
<b>Операции по балансам аккаунта</b> .....	19
<b>Управление DNS</b> .....	21

## Общая информация

URL для подключения: <https://userapi.vdsina.ru>.

Формат данных входящего запроса и возвращаемых данных: JSON.

Поддерживаемые методы запросов: GET, POST, PUT, DELETE.

Стандартное применение: GET – для получения, POST – для создания, PUT – для изменения и DELETE – для удаления объекта.

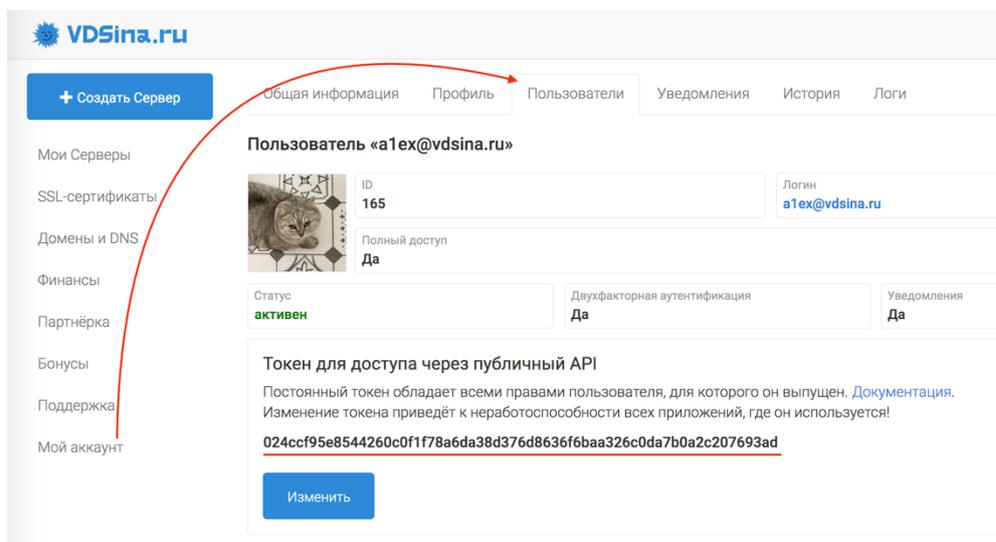
Авторизация: токен в HTTP-заголовке Authorization.

Все даты и метки времени возвращаются в зоне Europe/Moscow (часовом поясе, в котором располагается сервер API).

Существуют ограничения на количество запросов на аутентификацию по паролю с одного IP-адреса, при частых безуспешных попытках аутентификации IP-адрес блокируется на 4 часа. Также осуществляется проверка IP-адреса по чёрным спискам Spamhaus (SBL, SBL CSS, XBL, SBL DROP, TOR), аутентификация с таких адресов запрещена.

## Получение и использование токена

Постоянный токен авторизации можно получить в личном кабинете в просмотре информации пользователя аккаунта:



The screenshot shows the VDSina.ru user interface. On the left is a sidebar with navigation links: '+ Создать Сервер', 'Мои Серверы', 'SSL-сертификаты', 'Домены и DNS', 'Финансы', 'Партнёрка', 'Бонусы', 'Поддержка', and 'Мой аккаунт'. The main content area is titled 'Пользователь «a1ex@vdsina.ru»' and contains the following information:

- Общая информация: ID 165, Логин a1ex@vdsina.ru
- Полный доступ: Да
- Статус: активен
- Двухфакторная аутентификация: Да
- Уведомления: Да

Below this information, there is a section for the API token:

Токен для доступа через публичный API  
Постоянный токен обладает всеми правами пользователя, для которого он выпущен. [Документация.](#)  
Изменение токена приведёт к неработоспособности всех приложений, где он используется!

024ccf95e8544260c0f1f78a6da38d376d8636f6baa326c0da7b0a2c207693ad

There is an 'Изменить' button below the token.

Токен меняется при изменении пароля пользователя.

Для получения токена авторизации с помощью API необходимо пройти аутентификацию, отправив POST-запрос в локацию /v1/auth с JSON-объектом, внутри которого будут указаны параметры email и password (данные пользователя, с которым вы входите в панель управления), например, так:

```
curl -X POST -H 'Content-Type: application/json' "https://userapi.vdsina.ru/v1/auth" -d '{"email": "admin@domain.ru", "password": "Pas$W0rD"}'
```

Результатом вернётся JSON-объект с токеном:

```
{
  "status": "ok",
  "status_msg": "Token info",
  "data": {
    "token": "024ccf95e8544260c0f1f78a6da38d376d8636f6baa326c0da7b0a2c207693ad"
  }
}
```

После этого полученный токен можно использовать для любых запросов к API, например, чтобы получить баланс аккаунта:

```
curl -X GET -H 'Authorization:
024ccf95e8544260c0f1f78a6da38d376d8636f6baa326c0da7b0a2c207693ad '
"https://userapi.vdsina.ru/v1/account.balance"
```

Результатом вернётся JSON-объект с данными о балансе:

```
{
  "status": "ok",
  "status_msg": "Balance information",
  "data": {
    "real": 2818,
    "bonus": 1240,
    "partner": 790
  }
}
```

Токен будет иметь те же права доступа, что и указанный пользователь, от чьего имени делался запрос на получение токена. Если вам нужно ограничить действия для запросов через API, необходимо создать отдельного пользователя в аккаунте с необходимым набором прав и выполнять запросы с токеном этого пользователя.

## Описание параметров и возвращаемых ошибок

Предусмотрен возврат стандартных HTTP-статусов, как успешных, так и ошибочных, например:

- 200 – запрос завершился без ошибки,
- 400 – данные переданы неверно или запрос сформирован неправильно,
- 401 – необходима аутентификация,
- 403 – доступ запрещён,
- 500 – внутренняя ошибка сервера и другие.

В результате всегда должен возвращаться JSON-объект, некоторые его обязательные поля:

- `status` – статус выполнения запроса, обычно «ok» или «error»,
- `status_msg` – текстовая расшифровка статуса ошибки или успеха, может быть пустой строкой,
- `data` – объект с возвращаемыми данными.

В качестве необязательных полей могут присутствовать:

- `status_code` – числовой статус выполнения запроса, обычно из списка стандартных HTTP-статусов,
- `description` – подробное описание возникшей ошибки, может быть пустой строкой.

## Список доступных действий

### Информация об аккаунте и прогнозе отключения, GET /v1/account

Возвращается ID и название аккаунта, дата создания и прогноз отключения (дата, до которой достаточно средств на оплату всех услуг):

```
curl -X GET -H 'Authorization:
024ccf95e8544260c0f1f78a6da38d376d8636f6baa326c0da7b0a2c207693ad'
"https://userapi.vdsina.ru/v1/account"
```

Например, в результате возвращается JSON-объект:

```
{
  "status": "ok",
  "status_msg": "Account information",
  "data": {
    "account": {
      "id": 7,
      "name": "a7"
    },
    "created": "2014-11-19 15:28:42",
    "forecast": "2020-08-12",
    "can": {
      "add_user": true,
      "add_service": true,
      "convert_to_cash": true
    }
  }
}
```

Данные прогноза отключения и баланса кэшируются и изменяются только в случае реальных изменений по счетам или услугам. В объекте can будут перечислены некоторые возможности аккаунта: возможность создавать новых пользователей, заказывать новые услуги, выводить деньги со счетов.

### Баланс аккаунта, GET /v1/account.balance

Возвращаются все доступные балансы, основной, бонусный и партнёрский, если операций по счёту не было, то баланс не возвращается:

```
curl -X GET -H 'Authorization:
024ccf95e8544260c0f1f78a6da38d376d8636f6baa326c0da7b0a2c207693ad'
"https://userapi.vdsina.ru/v1/account.balance"
```

### Лимиты аккаунта, GET /v1/account.limit

Возвращаются все доступные лимиты по типам услуг:

```
curl -X GET -H 'Authorization:
024ccf95e8544260c0f1f78a6da38d376d8636f6baa326c0da7b0a2c207693ad'
"https://userapi.vdsina.ru/v1/account.limit"
```

Например, в результате возвращается JSON-объект:

```
{
  "status": "ok",
  "status_msg": "Account limits information",
  "data": {
    "server": {
      "max": 1,
      "now": 1
    },
    "server-ip4": {
      "max": 100,
      "child_max": 10,
      "now": 1
    },
    ...
  }
}
```

Для каждого типа услуги будет возвращён объект, в котором указаны ограничения: `max` – максимум услуг такого типа в аккаунте, `child_max` – максимум услуг такого типа в родительской услуге (например, количество IPv4-адресов для одного сервера), `now` – количество заказанных услуг такого типа в аккаунте на данный момент.

#### Регистрация нового аккаунта, `POST /v1/register`

Создаётся новый клиентский аккаунт и пользователь с доступом в панель управления.

Пример запроса:

```
curl -X POST -H 'Authorization:
024ccf95e8544260c0f1f78a6da38d376d8636f6baa326c0da7b0a2c207693ad' -H 'Content-Type:
application/json' "https://userapi.vdsina.ru/v1/register" -d '{"email":
"admin@domain.ru", "code": "SuperPartner"}'
```

Для регистрации нового аккаунта необходимо передать токен авторизации существующего клиента. Для регистрации по партнёрской программе необходимо передать свой партнёрский код в поле `code`. Результатом запроса будет подобный объект с данными о новом аккаунте и пользователе:

```
{
  "status": "ok",
  "status_msg": "New account created",
  "data": {
    "account": {
      "id": 190
    },
    "user": {
      "id": 347,
      "name": "admin@domain.ru",
      "token": "024ccf95e8544260c0f1f78a6deadbeefd8636f6baa326c0da7b0a2c207693ad"
    }
  }
}
```

## Список групп тарифных планов, GET /v1/server-group

Возвращается список групп тарифных планов с кратким описанием, например:

```
curl -X GET -H 'Authorization:
024ccf95e8544260c0f1f78a6da38d376d8636f6baa326c0da7b0a2c207693ad '
"https://userapi.vdsina.ru/v1/server-group"
```

Полученные ID групп должны использоваться в дальнейшем при запросе информации по дополнительным объектам.

## Список дата-центров, GET /v1/datacenter

Возвращается список дата-центров, например:

```
curl -X GET -H 'Authorization:
024ccf95e8544260c0f1f78a6da38d376d8636f6baa326c0da7b0a2c207693ad '
"https://userapi.vdsina.ru/v1/datacenter"
```

Например, в результате возвращается JSON-объект:

```
{
  "status": "ok",
  "status_msg": "Datacenters list",
  "data": [
    {
      "id": 1,
      "name": "Дата-центр RU",
      "country": "ru",
      "active": true
    },
    {
      "id": 2,
      "name": "Дата-центр NL",
      "country": "nl",
      "active": false
    }
  ]
}
```

Данные представляют из себя массив объектов, флаг active указывает на возможность заказа сервера в конкретном дата-центре.

## Список шаблонов ОС, GET /v1/template

Список шаблонов операционных систем, доступных для установки или переустановки сервера, например:

```
curl -X GET -H 'Authorization:
024ccf95e8544260c0f1f78a6da38d376d8636f6baa326c0da7b0a2c207693ad '
"https://userapi.vdsina.ru/v1/template"
```

Например, в результате возвращается JSON-объект:

```

{
  "status": "ok",
  "status_msg": "OS templates list",
  "data": [
    {
      "id": 1,
      "name": "CentOS 7 x64",
      "image":
"http://api2.vdsina.ru/uploads/template/43a7db318bd2ea21eacabe44abf62fff.png",
      "active": true,
      "has_instruction": false,
      "ssh-key": true,
      "server-plan": [
        1,
        2
      ],
      "limits": {
        "cpu": {
          "min": 1
        },
        "ram": {
          "min": 1
        },
        "disk": {
          "min": 5
        }
      }
    },
    {
      "id": 2,
      "name": "Windows Server 2019",
      "image":
"http://api2.vdsina.ru/uploads/template/2c96181313b73d1ee22075fc41c05fef.png",
      "active": true,
      "has_instruction": false,
      "ssh-key": false,
      "server-plan": [
        2,
        3
      ],
      "limits": {
        "cpu": {
          "min": 2
        },
        "ram": {
          "min": 4
        },
        "disk": {
          "min": 20
        }
      }
    }
  ]
}

```

Данные представляют из себя массив объектов, флаг active указывает на возможность заказа сервера с конкретным шаблоном ОС. Флаг ssh-key указывает на возможность использовать авторизацию по пользовательскому ключу SSH в конкретном шаблоне. Массив server-plan содержит в себе ID тарифных планов, для которых доступна установка

сервера с конкретным шаблоном ОС. Также необходимо обратить внимание на минимальные системные требования шаблона ОС: в объекте `limits` указаны следующие параметры: минимальное количество процессоров/ядер – `cpu`, минимальное количество оперативной памяти в ГБ – `ram`, минимальное количество места для дискового раздела в ГБ – `disk`.

### Список тарифных планов для серверов, GET /v1/server-plan/ID

Список тарифных планов для серверов возвращается по *ID группы тарифных планов*, например:

```
curl -X GET -H 'Authorization:
024ccf95e8544260c0f1f78a6da38d376d8636f6baa326c0da7b0a2c207693ad'
"https://userapi.vdsina.ru/v1/server-plan/1"
```

Например, в результате возвращается JSON-объект:

```
{
  "status": "ok",
  "status_msg": "Server plans list",
  "data": [
    {
      "id": 2,
      "name": "Plan-2",
      "cost": 20,
      "period": "day",
      "period_name": "день",
      "min_money": 100,
      "can_bonus": true,
      "description": "Test Plan",
      "data": {
        "cpu": {
          "type": "integer",
          "title": "vCPU",
          "value": 1,
          "for": "core"
        },
        "ram": {
          "type": "float",
          "title": "RAM",
          "value": 1,
          "bytes": 1073741824,
          "for": "ГБ"
        },
        "disk": {
          "type": "integer",
          "title": "NVMe",
          "value": 30,
          "bytes": 32212254720,
          "for": "ГБ"
        },
        "traff": {
          "type": "float",
          "title": "Трафик",
          "value": 32,
          "bytes": 35184372088832,
          "for": "ТБ"
        }
      }
    }
  ]
}
```

```

    }
  },
  "server-group": 1,
  "selected": false,
  "active": true,
  "enable": true,
  "has_params": false,
  "backup": {
    "cost": 0.1,
    "full_cost": 0.1,
    "period": "day",
    "period_name": "день",
    "for": "ГБ"
  }
},
{
  "id": 4,
  "name": "Plan-4",
  "cost": 30,
  "period": "day",
  "period_name": "день",
  "min_money": 1000,
  "can_bonus": true,
  "description": "TEST",
  "data": {
    "cpu": {
      "type": "integer",
      "title": "vCPU",
      "value": 2,
      "for": "core"
    },
    "ram": {
      "type": "float",
      "title": "RAM",
      "value": 1,
      "bytes": 1073741824,
      "for": "ГБ"
    },
    "disk": {
      "type": "integer",
      "title": "NVMe",
      "value": 30,
      "bytes": 32212254720,
      "for": "ГБ"
    },
    "traff": {
      "type": "float",
      "title": "Трафик",
      "value": 32,
      "bytes": 35184372088832,
      "for": "ТБ"
    }
  },
  "server-group": 1,
  "selected": true,
  "active": false,
  "enable": false,
  "has_params": false,
  "backup": {
    "cost": 0.1,
    "full_cost": 0.1,

```

```

    "period": "day",
    "period_name": "день",
    "for": "ГБ"
  }
]
}

```

Данные представляют из себя массив объектов, флаги `active` и `enable` указывают на возможность заказа сервера с конкретным тарифным планом. Объект `data` содержит в себе краткие характеристики тарифного плана. Поля `cost` и `period` содержат в себе цену тарифа за указанный период (обычно 1 день). Поле `min_money` указывает, сколько средств нужно иметь на основном балансе для заказа тарифа, флаг `can_bonus` указывает на возможность оплачивать тариф средствами с бонусного баланса.

В объекте `data` хранится полный состав тарифного плана: `cpu` – количество процессоров/ядер, `ram` – количество оперативной памяти в ГБ, `disk` – количество дискового пространства в ГБ, `traff` – количество включённого в тарифный план трафика на 1 календарный месяц в ГБ.

В параметре `has_params` хранится признак тарифа-конструктора, в таком случае дополнительно у тарифа может быть объект `params` с данными о возможном составе итогового тарифа и стоимости отдельных параметров тарифного плана. Общая стоимость такого тарифа складывается из стоимости самого тарифного плана и совокупной стоимости всех добавленных параметров тарифа.

## SSH-ключи для серверов

Доступные методы:

- GET `/v1/ssh-key` – список всех доступных клиенту SSH-ключей
- GET `/v1/ssh-key/ID` – просмотр данных SSH-ключа, *ID ключа*
- PUT `/v1/ssh-key/ID` – правка данных, *ID ключа*, доступные поля: `name`, `data`
- POST `/v1/ssh-key` – создание нового ключа, доступные поля: `name`, `data`
- DELETE `/v1/ssh-key/ID` – удаление SSH-ключа, *ID ключа*

Поле	Признак	Описание, тип
<code>name</code>	обязательно	Строка, название ключа
<code>data</code>	обязательно	Строка, текстовое представление ключа (base64-кодировка)

## ISO для серверов

Доступные методы:

- GET `/v1/iso` – список всех загруженных клиентом ISO-образов
- GET `/v1/iso/ID` – просмотр данных ISO, *ID услуги ISO*
- GET `/v1/iso/KEY` – просмотр состояния загрузки ISO с ключом `KEY`
- POST `/v1/iso` – начало загрузки нового файла ISO, доступные поля: `url`
- POST `/v1/iso/KEY` – создание нового ISO с ключом загруженного файла `KEY`
- DELETE `/v1/iso/ID` – удаление ISO, *ID услуги ISO*

Поле	Признак	Описание, тип
------	---------	---------------

url	обязательно	Строка, URL для скачивания файла образа, прямая ссылка на файл MIME-формата application/x-iso9660-image или application/x-iso-image, максимальный размер файла – 8 гигабайт, протоколы: http, https, ftp, ftps
-----	-------------	--

Возможные статусы ISO:

- new – ISO заказан, но не ещё не создан
- active – ISO активен и работает
- block – ISO заблокирован администрацией
- notpaid – ISO заблокирован за неуплату
- deleted – ISO удалён

Дополнительно к статусу ISO может быть дополнительный статус в поле status\_text. В нём содержится описание действия, которое выполняется с ISO в данный момент.

Процесс загрузки нового файла и создания услуги ISO:

- передаём в POST /v1/iso новый URL для загрузки в доступном поле url, в результате, в объекте data в поле id получаем идентификатор загрузки KEY, либо ошибочные статусы с указанием текста ошибки
- делаем периодический запрос GET /v1/iso/KEY с идентификатором загрузки KEY, проверяя, загрузился ли файл: в объекте data проверяем поле status на значение «done», это означает, что файл скачан, либо возвращается значение «processing» с указанием прогресса загрузки, либо ошибочные статусы с указанием текста ошибки
- создаём услугу ISO из загруженного файла POST /v1/iso/KEY с идентификатором загрузки KEY, в результате, в объекте data в поле id получаем идентификатор ID услуги ISO

## Серверы

Доступные методы:

- GET /v1/server – список всех серверов клиента, в случае, если не хватает каких-либо данных о сервере в списке, необходимо получить данные о конкретном сервере отдельным запросом по его ID
- GET /v1/server/ID – просмотр сервера и его параметров, ID услуги сервера
- PUT /v1/server/ID – правка данных, ID услуги сервера, доступные поля: name, autoprolong
- POST /v1/server – создание нового сервера, доступные поля: datacenter, server-plan, template, ssh-key, iso, backup, host, name, cpu, ram, disk, ip
- DELETE /v1/server/ID – удаление сервера со всеми его данными и зависимыми услугами, ID услуги сервера

Поле	Признак	Описание, тип
name	необязательно	Строка, текстовое название услуги для удобства клиента
autoprolong	необязательно	0 или 1, признак автоматического продления услуги
datacenter	обязательно	Число, ID дата-центра, может быть недоступен для заказа при отсутствии ресурсов

server-plan	обязательно	Число, ID тарифного плана, должен быть из нужной группы
template	необязательно	Число, ID шаблона ОС для установки, взаимоисключающий параметр с backup и iso, шаблон ОС может быть недоступен для конкретных тарифных планов
ssh-key	необязательно	Число, ID SSH-ключа для пользователя root в поддерживаемых ОС, игнорируется в случае восстановления из резервной копии
backup	необязательно	Число, ID услуги резервной копии, из которой нужно произвести восстановление после создания услуги сервера, взаимоисключающий параметр с template, резервная копия должна находиться в том же дата-центре, что и создаваемый сервер и быть не более размера диска выбранного тарифного плана
iso	необязательно	Число, ID услуги ISO, из которой нужно произвести установку после создания услуги сервера, взаимоисключающий параметр с template
host	необязательно	Строка, текстовое значение для hostname сервера, должно быть правильным доменным именем, игнорируется в случае восстановления из резервной копии
cpu	необязательно	Число, количество виртуальных процессоров для создаваемого сервера, если тарифный план поддерживает настройку параметров
ram	необязательно	Число, количество ГБ оперативной памяти для создаваемого сервера, если тарифный план поддерживает настройку параметров
disk	необязательно	Число, количество ГБ для дискового раздела для создаваемого сервера, если тарифный план поддерживает настройку параметров
ip4	необязательно	0 или 1, признак подключения IPv4-адреса при создании сервера, если тарифный план поддерживает настройку параметров

При создании сервера, необходимо учитывать минимальные системные требования шаблона ОС для установки на сервере, сравнивая параметры из состава тарифа с минимальными параметрами шаблона ОС.

Пример получения информации о сервере:

```
curl -X GET -H 'Authorization:
024ccf95e8544260c0f1f78a6da38d376d8636f6baa326c0da7b0a2c207693ad '
"https://userapi.vdsina.ru/v1/server/1345"
```

И пример возвращаемого ответа в формате JSON:

```
{
```

```

"status": "ok",
"status_msg": "Server information",
"data": {
  "id": 1345,
  "name": "v1301.hosted-by-vdsina.ru",
  "status": "active",
  "created": "2019-07-11 20:05:49",
  "updated": "2019-07-12 17:50:02",
  "end": "2019-07-12 20:08:27",
  "autoprolong": true,
  "ip": [
    {
      "id": 1574,
      "ip": "185.251.37.62",
      "type": "4",
      "host": "host-185-251-37-62.hosted-by-vdsina.ru",
      "gateway": "185.251.37.1",
      "netmask": "255.255.255.0",
      "mac": "52:54:00:00:05:41"
    }
  ],
  "ip_local": null,
  "host": "v1301.hosted-by-vdsina.ru",
  "data": {
    "cpu": {
      "type": "integer",
      "title": "vCPU",
      "value": 1,
      "for": "core",
      "total": 4
    },
    "ram": {
      "type": "float",
      "title": "RAM",
      "value": 1,
      "bytes": 1073741824,
      "for": "ГБ",
      "total": 8,
      "total_bytes": 8589934592
    },
    "disk": {
      "type": "integer",
      "title": "NVMe",
      "value": 1,
      "bytes": 1073741824,
      "for": "ГБ",
      "total": 10,
      "total_bytes": 10737418240
    },
    "traff": {
      "type": "float",
      "title": "Трафик",
      "value": 32,
      "bytes": 35184372088832,
      "for": "ТБ"
    }
  },
  "server-plan": {
    "id": 44,
    "name": "Минималь+"
  },
},

```

```

"server-group": {
  "id": 5,
  "name": "VDS"
},
"template": {
  "id": 22,
  "name": "Ubuntu 16.04"
},
"datacenter": {
  "id": 29,
  "name": "Serverius",
  "country": "nl"
},
"ssh-key": null,
"can": {
  "reboot": true,
  "update": true,
  "delete": true,
  "prolong": false,
  "backup": true,
  "ip_local": true
},
"bandwidth": {
  "current_month": "20987612334",
  "past_month": "0"
}
}
}

```

Возможные статусы сервера:

- new – сервер заказан, но не ещё не создан
- active – сервер активен и работает
- block – сервер заблокирован администрацией
- notpaid – сервер остановлен за неуплату
- deleted – сервер удалён

Дополнительно к статусу сервера может быть дополнительный статус в поле `status_text`. В нём содержится описание действия, которое выполняется с сервером в данный момент. В объекте `data` для сервера показывается его полная конфигурация согласно тарифу и содержимому тарифа: `cpu` – количество процессоров/ядер, `ram` – количество оперативной памяти в ГБ, `disk` – количество дискового пространства в ГБ, `traff` – количество включённого в тарифный план трафика на 1 календарный месяц в ГБ. В параметрах `value` и `total` указаны значения из базового тарифного плана и общие значения, соответственно, если тарифный план был настроен с дополнительными параметрами.

### Перезагрузка сервера, `PUT /v1/server.reboot/ID`

Перезагрузка сервера по *ID услуги сервера*. Дополнительно можно передать поле `type`, который может принимать значения `soft` или `hard`. При установке `type=soft` (по умолчанию), серверу будет отправлен сигнал на перезагрузку. При установке `type=hard`, операционной системе будет отправлен сигнал завершения работы, через некоторое время будет произведена проверка статуса сервера, если он выключен, то он снова будет запущен, в противном случае сервер будет выключен принудительно (с возможной потерей данных в работающей системе), а после этого снова запущен.

Поле	Признак	Описание, тип
type	необязательно	Строка, soft или hard

### Переустановка сервера, PUT /v1/server.reinstall/ID

Переустановка сервера с новым шаблоном ОС по *ID услуги сервера*. Передаваемые поля: template, ssh-key, host.

Поле	Признак	Описание, тип
template	необязательно	Число, <i>ID шаблона ОС</i> для установки, шаблон ОС может быть недоступен для конкретных тарифных планов
ssh-key	необязательно	Число, <i>ID SSH-ключа</i> для пользователя root в поддерживаемых ОС
host	необязательно	Строка, текстовое значение для hostname сервера, должно быть правильным доменным именем

### Установка пароля сервера, PUT /v1/server.password/ID

Установка пароля сервера по *ID услуги сервера*. Дополнительно нужно передать поле password с новым паролем. Сервер будет перезапущен, устанавливается пароль VNC и пароль пользователя root в поддерживаемых Linux-дистрибутивах. Внимание, если вы используете ОС, установленную из своего ISO или на сервере установлена ОС Windows или FreeBSD, пароль администратора ОС изменён не будет, в таком случае меняется только пароль VNC.

Поле	Признак	Описание, тип
password	обязательно	Строка, новый пароль

### Изменение тарифного плана сервера, PUT /v1/server.plan/ID

Изменение (расширение) тарифного плана сервера по *ID услуги сервера*. Изменение тарифного плана на младший технически невозможно. Дополнительно нужно передать поле server-plan с числовым ID нового тарифного плана. Сервер будет перезапущен. После смены тарифного плана ресурсы будут добавлены автоматически, расширить файловую систему на весь новый раздел диска необходимо вручную средствами ОС.

Поле	Признак	Описание, тип
server-plan	обязательно	Число, <i>ID тарифного плана</i> , должен быть из нужной группы
cpu	необязательно	Число, количество виртуальных процессоров для сервера, если тарифный план поддерживает настройку параметров
ram	необязательно	Число, количество ГБ оперативной памяти для сервера, если тарифный план поддерживает настройку параметров
disk	необязательно	Число, количество ГБ для дискового раздела для сервера, если тарифный план поддерживает настройку параметров

## Продление сервера, PUT /v1/server.prolong/ID

Продление сервера по *ID услуги сервера* в случае, если сервер не был запущен после оплаты, например, из-за отключённого автоматического продления. Дополнительные поля не передаются.

## VNC-подключение, GET /v1/server.vnc/ID

Получение данных VNC-подключения для сервера по *ID услуги*. Пример получения информации:

```
curl -X GET -H 'Authorization:
024ccf95e8544260c0f1f78a6da38d376d8636f6baa326c0da7b0a2c207693ad '
"https://userapi.vdsina.ru/v1/server.vnc/1345"
```

И пример возвращаемого ответа в формате JSON:

```
{
  "status": "ok",
  "status_msg": "Server VNC information",
  "data": {
    "host": "kvm9249.vdsina.ru",
    "port": 5907,
    "password": "XAqc18T8sv7J1keZ"
  }
}
```

## Резервная копия сервера

Доступные методы:

- GET /v1/backup – список всех услуг с резервными копиями
- GET /v1/backup/ID – подробная информация услуги с резервной копией, *ID услуги резервной копии*
- POST /v1/backup/ID – создание новой резервной копии сервера, *ID услуги сервера*, без дополнительных полей
- PUT /v1/backup/ID – правка данных услуги резервной копии сервера, *ID услуги резервной копии*, доступны поля: name, autoprolong
- DELETE /v1/backup/ID – удаление услуги резервной копии со всеми данными, *ID услуги резервной копии*
- PUT /v1/server.backup/ID – восстановление резервной копии на сервер, *ID услуги сервера*, необходимое поле: backup
- POST /v1/server.backup/ID – создание новой резервной копии сервера, *ID услуги сервера*, без дополнительных полей
- PUT /v1/backup.cory/ID – копирование резервной копии в другой дата-центр, *ID услуги резервной копии*, необходимое поле: datacenter

Поле	Признак	Описание, тип
name	необязательно	Строка, текстовое название услуги для удобства клиента
autoprolong	необязательно	0 или 1, признак автоматического продления услуги

backup	обязательно	Число, <i>ID услуги резервной копии</i> , из которой нужно произвести восстановление услуги сервера, резервная копия должна находиться в том же дата-центре, что и сервер и быть не более размера диска выбранного тарифного плана
datacenter	обязательно	Число, <i>ID дата-центра</i> , в который нужно произвести копирование услуги резервной копии

Возможные статусы резервной копии:

- new – резервная копия заказана, но не ещё не создана
- active – резервная копия активна
- block – резервная копия заблокирована администрацией
- notpaid – резервная копия заблокирована за неуплату
- deleted – резервная копия удалена

Дополнительно к статусу резервной копии может быть дополнительный статус в поле `status_text`. В нём содержится описание действия, которое выполняется с резервной копией в данный момент.

### Настройка резервных копий сервера по расписанию

Доступные методы:

- GET `/v1/server.schedule/ID` – список расписаний, *ID услуги сервера*
- POST `/v1/server.schedule/ID` – создание расписания, *ID услуги сервера*, доступны поля: `type`, `day`, `hour`, `left`
- DELETE `/v1/server.schedule/ID` – удаление расписания, *ID услуги сервера*, доступно поле: `type`, если не указать тип расписания, будут удалены все расписания

Поле	Признак	Описание, тип
<code>type</code>	обязательно	Строка, тип расписания, варианты: <code>day</code> , <code>week</code> , <code>month</code> , для одного сервера может быть создано не более одного расписания каждого типа
<code>day</code>	необязательно	Число, день создания резервной копии (число месяца (1-30), в случае месячной копии или номер дня недели (1-7) в случае недельной копии), поле неприменимо к ежедневной резервной копии
<code>hour</code>	необязательно	Число, час создания резервной копии (только ночные часы: 1-6)
<code>left</code>	необязательно	Число, количество сохраняемых копий (1-4)

### Подключение и отключение ISO для сервера

Доступные методы:

- PUT `/v1/server.iso/ID` – подключение ISO для сервера, *ID услуги сервера*, необходимое поле: `iso`
- DELETE `/v1/server.iso/ID` – отключение ISO для сервера, *ID услуги сервера*, без дополнительных полей

Поле	Признак	Описание, тип
iso	обязательно	Число, <i>ID услуги с ISO</i> для подключения

### Дополнительные IP-адреса для сервера

Доступные методы:

- GET /v1/server-ip – список всех услуг с дополнительными IP для серверов
- POST /v1/server-ip/ID – заказ дополнительных IP для сервера, *ID услуги сервера*, необходимые поля: type, count
- PUT /v1/server-ip/ID – удаление дополнительных IP-адресов для сервера по списку, *ID услуги дополнительных IP*, дополнительные поля: type, delete
- DELETE /v1/server-ip/ID – удаление услуги с дополнительными IP для сервера, *ID услуги дополнительного IP*
- GET /v1/server.ip/ID – список услуг с дополнительными IP для сервера, *ID услуги сервера*
- POST /v1/server.ip/ID – заказ дополнительных IP для сервера, *ID услуги сервера*, необходимые поля: type, count
- PUT /v1/server.ip/ID – удаление дополнительных IP-адресов для сервера по списку, *ID услуги сервера*, дополнительные поля: type delete
- GET /v1/ip – список всех присвоенных клиенту IP-адресов (IP-пул)
- GET /v1/ip/ID – подробная информация об адресе в IP-пуле, *ID адреса*

Поле	Признак	Описание, тип
type	необязательно	Число, 4 или 6, тип IP-адреса, по умолчанию 4
count	обязательно	Число, количество создаваемых IP-адресов, учитываются лимиты аккаунта (account.limit), больше нуля
delete	обязательно	Список чисел, список <i>ID IP-адресов</i> для удаления

Возможные статусы услуги с дополнительными IP:

- new – услуга с IP заказана, но не ещё не создана
- active – дополнительный IP активен и работает
- block – дополнительный IP отключён и заблокирован администрацией
- notpaid – дополнительный IP отключён за неуплату
- deleted – услуга с дополнительным IP удалена

Дополнительно к статусу услуги с дополнительными IP может быть дополнительный статус в поле status\_text. В нём содержится описание действия, которое выполняется с услугой в данный момент.

### Локальный IP-адрес для сервера

Доступные методы:

- GET /v1/server.ip.local/ID – информация о локальном IP для сервера, *ID услуги сервера*
- POST /v1/server.ip.local/ID – создание локального IP для сервера, *ID услуги сервера*
- DELETE /v1/server.ip.local/ID – удаление локального IP-адреса для сервера, *ID услуги сервера*

## Статистика сервера, GET /v1/server.stat/ID

Получение данных статистики для сервера по *ID услуги*. По умолчанию выводится информация за последние 30 дней. При указании параметров *from* и *to* можно получить вывод статистики за указанный период, в параметрах допустимо указывать дату/время в стандартных форматах. Пример получения информации:

```
curl -X GET -H 'Authorization:
024ccf95e8544260c0f1f78a6da38d376d8636f6baa326c0da7b0a2c207693ad'
"https://userapi.vdsina.ru/v1/server.stat/1345"
```

И пример возвращаемого ответа в формате JSON:

```
{
  "status": "ok",
  "status_msg": "Server stat information",
  "data": [
    {
      "dt": "2019-07-12 00:00:00",
      "stat": {
        "cpu": 2.0839426931666667,
        "disk_reads": 1,
        "disk_writes": 11,
        "lnet_rx": 0,
        "lnet_tx": 0,
        "vnet_rx": 9948,
        "vnet_tx": 103054
      }
    },
    ...
  ]
}
```

Статистика выводится блоками через каждый час. Значения можно расшифровать: *cpu* – средняя загрузка процессора в процентах за сегмент времени, *disk\_reads/disk\_writes* – количество операций чтения/записи за указанный сегмент времени, *lnet\_rx/lnet\_tx* – принятый и переданный трафик по локальной сети в байтах за сегмент времени, *vnet\_rx/vnet\_tx* – принятый и переданный трафик во внешней сети (интернет) в байтах.

## Операции по балансам аккаунта

Доступные методы:

- GET /v1/operation – список всех операций всех типов, доступные поля: *from*, *to*
- GET /v1/operation/ID – подробная информация об операции, *ID операции*
- POST /v1/operation – создание новой операции пополнения баланса аккаунта, доступные поля: *summ*
- DELETE /v1/operation/ID – удаление неоплаченной операции пополнения, *ID операции*

Поле	Признак	Описание, тип
<i>from</i>	необязательно	Строка, дата, с которой получать список операций
<i>to</i>	необязательно	Строка, дата по которую получать список операций
<i>summ</i>	необязательно	Число, сумма пополнения

Пример получения информации:

```
curl -X GET -H 'Authorization:
024ccf95e8544260c0f1f78a6da38d376d8636f6baa326c0da7b0a2c207693ad'
"https://userapi.vdsina.ru/v1/operation?from=2019-07-17&to=2019-07-19"
```

И пример возвращаемого ответа в формате JSON:

```
{
  "status": "ok",
  "status_msg": "Operation list",
  "data": [
    {
      "id": 2177311,
      "purse": "real",
      "type": 1,
      "status": 0,
      "summ": "1000",
      "created": "2019-07-18 14:39:29",
      "updated": "2019-07-18 14:48:28",
      "comment": "Пополнение баланса",
      "payment": {
        "type": "webmoney",
        "name": "WebMoney R"
      },
      "service": null,
      "paylink":
        "https://cp.vdsina.ru/operation/select/41cd63800a8beb1961527ddeebf530a521391f"
    },
    {
      "id": 2170929,
      "purse": "real",
      "type": -1,
      "status": 1,
      "summ": "8.3",
      "created": "2019-07-17 14:24:25",
      "updated": "2019-07-17 14:24:25",
      "comment": "Списание за услугу Сервер 1 ГБ #137841 - api test server",
      "payment": null,
      "service": {
        "id": 137841
      },
      "paylink": null
    },
    ...
  ]
}
```

Краткое описание некоторых возвращаемых полей: purse – тип баланса (real – основной баланс, bonus – бонусный баланс, partner – партнёрский баланс), type – тип операции (1 – зачисление, -1 – списание), status – статус операции (0 – не оплачено, 1 – оплачено), summ – сумма операции (приведено к строковому типу). Если это операция пополнения и её можно оплатить, у такой операции есть поле paylink, в нём содержится ссылка на процесс оплаты, процесс не требует аутентификации в панели управления и ссылка может быть

передана кому угодно, ссылка уникальная и одноразовая, если операция оплачена, повторно оплатить по такой ссылке нельзя.

## Управление DNS

Доступные методы:

- GET /v1/dns – список всех услуг с размещёнными в DNS доменами
- GET /v1/dns/ID – подробная информация услуги с доменом в DNS, *ID услуги с доменом*
- POST /v1/dns – создание нового домена в DNS, доступные поля: name, ip
- DELETE /v1/dns/ID – удаление услуги домена в DNS, *ID услуги с доменом*
- GET /v1/dns.record/ID – список DNS-записей для услуги с доменом, *ID услуги с доменом*
- POST /v1/dns.record/ID – создание новой DNS-записи, *ID услуги с доменом*, доступные поля: host, type, priority, tag, value
- PUT /v1/dns.record/ID – правка существующей DNS-записи, *ID записи DNS*, доступные поля: priority, tag, value
- DELETE /v1/dns.record/ID – удаление существующей DNS-записи, *ID записи DNS*

Поле	Признак	Описание, тип
name	обязательно	Строка, имя домена для размещения в DNS
ip	необязательно	Строка, адрес IPv4, на основе которого нужно создать DNS-записи по умолчанию
host	обязательно	Строка, имя DNS-записи, либо с постфиксом домена, либо только префикс, доступны также @ и * в имени, должно быть правильным именем домена в итоге
type	обязательно	Строка, тип записи, один из списка: A, AAAA, CNAME, MX, NS, SRV, CAA, TXT
value	обязательно	Строка, значение DNS-записи, в зависимости от типа разные проверки, например, для записи типа A значение должно быть правильным IPv4-адресом
tag	необязательно	Строка, необходимый параметр для записей типа CAA, может быть одним из списка: issue, issuewild, iodef, unknown
priority	необязательно	Положительное число, приоритет/флаг DNS-записи для поддерживаемых типов (MX, SRV, CAA)

Возможные статусы DNS-доменов:

- new – DNS-домен заказан, но не ещё не создан
- active – DNS-домен активен и работает
- block – DNS-домен заблокирован администрацией
- notpaid – DNS-домен заблокирован за неуплату
- deleted – DNS-домен удалён

Дополнительно к статусу DNS-домена может быть дополнительный статус в поле status\_text. В нём содержится описание действия, которое выполняется с DNS-доменом в данный момент.