API

Naše API běží na adrese https://api.vpsfree.cz. Pomocí API lze vykonat vše co jde naklikat ve webovém rozhraní, a více. Ve skutečnosti webové rozhraní běžící na https://vpsadmin.vpsfree.cz pro každý úkon volá API.

Dokumentace API

Dokumentace API, tzn. seznam zdrojů, možných akcí, vstupních a výstupních parametrů je k vidění na https://api.vpsfree.cz/v6.0/.

Bez přihlášení se zobrazuje seznam všech zdrojů, tj. i těch, se kterými můžou pracovat pouze administrátoři. Vpravo nahoře se lze přihlásit stejnými údaji jako do vpsAdminu a poté se zobrazí pouze objekty, akce a parametry, se kterými může aktuálně přihlášený uživatel pracovat.

Práce s API

API je postaveno na námi vyvinutém frameworku HaveAPI, které vytváří samodokumentující se API, což znamená, že lze využít předpřipravené generické klienty:

- Ruby - https://github.com/vpsfreecz/vpsfree-client
- PHP - https://github.com/vpsfreecz/haveapi/tree/master/clients/php
- JavaScript - https://github.com/vpsfreecz/haveapi/tree/master/clients/js
- Go - https://github.com/vpsfreecz/haveapi/tree/master/clients/go
- Webové rozhraní z HaveAPI - https://github.com/vpsfreecz/haveapi-webui
- Souborový systém založený na FUSE - https://github.com/vpsfreecz/haveapi-fs

Ukázka použití je vždy v `README.md` každého klienta. Obecně se klientovi předá URL k API, poté si sám stáhne dokumentaci a podle ní se nastaví.

API je RESTful, takže pro jednoduché úkony lze využít i libovolný REST klient. Popis vlastního protokolu pro přenos dat, který HaveAPI klienti abstrahují, je k nahlédnutí v dokumentaci.

Autentizace

Lze využívat dvě autentizační metody. Tou první a jednodušší z nich je HTTP basic. S každým požadavkem na API se musí zaslat jméno a heslo. Je to dobrá volba pro jednorázové akce, pokud je ale potřeba API volat vícekrát nebo automatizovaně, ukládání hesla na disk či neustálé opisování není dobrý nápad.

Druhou metodou je autentizace přes tokeny. Funguje to tak, že klient nejprve požádá o vytvoření tokenu, k tomu potřebuje jméno, heslo a případně i TOTP. Jakmile klient dostane token, může jméno a heslo zapomenout a dále se autentizuje získaným tokenem.

Tokeny mohou být několika typů s různě dlouhou životností:

- fixed - platnost tokenu je pevně dána
- renewable_manual - platnost tokenu lze manuálně prodloužit
- renewable_auto - platnost tokenu je prodloužena při každém požadavku
- permament - token je platný napořád, resp. dokud není smazán

Typ tokenu a časový interval, o který se prodlužuje, si volí klient.

CLI

Klient pro Ruby obsahuje i CLI. Pro správnou funkci vyžaduje Ruby >= 2.0 a nainstalované hlavičkové soubory Ruby, OpenSSL a ncurses (většinou balíčky s příponou `-dev` či `-devel`).

Linux

Ubuntu 20.04

Instalace závislostí

```
sudo apt-get install ruby ruby-dev make g++ libssl-dev libncurses-dev
```

Instalace vpsfree-client

```
sudo gem install vpsfree-client
```

Instalace na Centos 7

Zdrojový kod klienta lze najít zde

vpsfree-client vyžaduje ruby >–2.30

Ruby ve verzi nižší než 2.5 jsou EOL

Aktuální Ruby je možné nainstalovat pomocí tohoto návodu.

Například:

```
yum install -y openssl-devel readline-devel zlib-devel
/build/vpsfree-client:curl -sL
https://github.com/rbenv/rbenv-installer/raw/master/bin/rbenv-installer |
bash -
# následně je zapotřebí přidat rbenv do PATH a provést trochu magie(snippet
z .bashrc):
export PATH="${PATH}:/root/.rbenv/bin"
```

```
if [ -d '/root/.rbenv' ]; then
  eval "$(rbenv init -)"
fi
/build/vpsfree-client:rbenv install 2.5.7
/build/vpsfree-client:rbenv global 2.5.7
#projistotu spustte nový bash který uvidí novou verzi ruby
/build/vpsfree-client:bash
#proveďte instalaci
/build/vpsfree-client:gem install vpsfree-client
#otestujeme
/build/vpsfree-client:which vpsfreectl
/root/.rbenv/shims/vpsfreectl
/build/vpsfree-client:vpsfreectl -h
Usage: /root/.rbenv/versions/2.5.7/bin/vpsfreectl [options] <resource>
<action> [objects ids] [-- [parameters]]
...
```

Instalace CLI na Linux distribucích založených na Arch Linux

Testováno na Manjaro 18.04 Xfce:

```
sudo pacman -Syu #aktualizujeme systém
sudo pacman -S rubygems #nainstaluje rubygems a ruby
gem install vpsfree-client #nainstalujeme vpsfreeclient (nepoužívejte tady
SUDO install!)
```

Přidáme PATH k nainstalovaným gems do `~./bashrc`.

```
sudo nano ~/.bashrc
```

Na konec souboru přidáme:

```
if which ruby >/dev/null && which gem >/dev/null; then
    PATH="$(ruby -r rubygems -e 'puts Gem.user_dir')/bin:$PATH"
fi
```

Restartujeme shell `exec  $SHELL` nebo počítač. Otestujeme funkčnost CLI, třeba příkazem:

```
vpsfreectl ip_traffic top
```

Výše uvedené příkazy nefungují na distribucích založených na Debian/Ubuntu a jiných.

macOS

Na OS X je nutné nainstalovat OpenSSL přes Homebrew a až poté se dá nainstalovat EventMachine (gem, jenž klient vyžaduje).

```
$ brew install openssl
```

```
$ sudo gem install eventmachine -- --with-opt-
include="/usr/local/opt/openssl/"
```

Nainstalovat jej lze pomocí ruby gems:

```
$ sudo gem install vpsfree-client
```

gem vypíše PATH pod kterým lze najít nainstalovaný balíček

```
WARNING:  You don\'t have /home/user/.gem/ruby/2.5.0/bin in your PATH,
      gem executables will not run.
```

Windows

Instalace ve Windows 10 využitím Ubuntu Linux subsystem:

1. ve Windows 10 povolit developer mode, nechat nainstalovat
2. přes Programy a funkce otevřít přidání součástí Windows, úplně dole zvolit Linux subsystem, nechat nainstalovat a restartovat počítač
3. po restartu jako admin spustit v nabídce Start bash
4. vytvořit Unix username a heslo
5. stisknout y a nechat nainstalovat základ Ubuntu

Instalace závislostí

```
sudo apt-get install ruby2.1 ruby2.1-dev libssl-dev make g++
```

Quick & Dirty fix pro nastavení ruby2.1 jako výchozího namísto 1.9

```
sudo rm /usr/bin/ruby /usr/bin/gem /usr/bin/irb /usr/bin/rdoc /usr/bin/erb
sudo ln -s /usr/bin/ruby2.1 /usr/bin/ruby
sudo ln -s /usr/bin/gem2.1 /usr/bin/gem
sudo ln -s /usr/bin/irb2.1 /usr/bin/irb
sudo ln -s /usr/bin/rdoc2.1 /usr/bin/rdoc
sudo ln -s /usr/bin/erb2.1 /usr/bin/erb
sudo gem update --system
sudo gem pristine --all
```

Zdroj: http://blog.costan.us/2014/04/restoring-ruby-20-on-ubuntu-1404.html

Instalace vpsfree-client

```
sudo gem install vpsfree-client
```

Po instalaci by v $PATH měl být k dispozici `vpsfreectl`. Pokud tomu tak není, lze to jednoduše napravit:

```
$ gem env | grep "EXECUTABLE DIRECTORY"
```

Tento příkaz vypíše, kde jsou uloženy spustitelné soubory instalované přes gem. Stačí tedy onen adresář přidat do $PATH, např.:

```
$ PATH="$PATH:/home/user/.gem/ruby/2.0.0/bin"
```

Použití CLI

```
$ vpsfreectl --help
Usage: vpsfreectl [options] <resource> <action> [objects ids] [--
[parameters]]
    -u, --api URL                    API URL
    -a, --auth METHOD                Authentication method
        --list-versions              List all available API versions
        --list-auth-methods [VERSION]
                                     List available authentication methods
        --list-resources [VERSION]   List all resource in API version
        --list-actions [VERSION]     List all resources and actions in API
version
        --version VERSION            Use specified API version
    -c, --columns                    Print output in columns
    -H, --no-header                  Hide header row
    -L, --list-parameters            List output parameters
    -o, --output PARAMETERS          Parameters to display, separated by a
comma
    -r, --rows                       Print output in rows
    -s, --sort PARAMETER             Sort output by parameter
        --save                       Save credentials to config file for
later use
        --raw                        Print raw response as is
        --timestamp                  Display Datetime parameters as
timestamp
        --utc                        Display Datetime parameters in UTC
        --localtime                  Display Datetime parameters in local
timezone
        --date-format FORMAT         Display Datetime in custom format
        --[no-]block                 Toggle action blocking mode
        --timeout SEC                Fail when the action does not finish
within the timeout
    -v, --[no-]verbose               Run verbosely
        --client-version             Show client version
        --protocol-version           Show protocol version
        --check-compatibility        Check compatibility with API server
```

```
    -h, --help                          Show this message

Commands:
action_state wait <STATE ID>        Block until the action is finished
vps remote_console VPS_ID           Open VPS remote console
vps migrate_many VPS_ID...          Migrate multiple VPSes using a
migration plan
snapshot download [SNAPSHOT_ID]     Download a snapshot as an archive or a
stream
snapshot send SNAPSHOT_ID           Download a snapshot stream and write it
on stdout
backup dataset [DATASET_ID] FILESYSTEM Backup dataset locally
backup vps [VPS_ID] FILESYSTEM      Backup VPS locally
ip_traffic top                      Live IP traffic monitor

Available resources:
cluster
cluster_resource
cluster_resource_package
cluster_resource_package.item
dataset
dataset.snapshot
dataset.plan
dataset.property_history
dataset_plan
dns_resolver
environment
environment.config_chain
environment.dataset_plan
host_ip_address
integrity_check
integrity_fact
integrity_object
ip_address
ip_traffic
ip_traffic_monitor
language
location
mail_log
mail_recipient
mail_template
mail_template.recipient
mail_template.translation
migration_plan
migration_plan.vps_migration
network
network_interface
node
node.status
object_history
os_template
```

```
pool
session_token
snapshot_download
system_config
transaction
transaction_chain
user
user.environment_config
user.cluster_resource
user.public_key
user.mail_role_recipient
user.mail_template_recipient
user.state_log
user_cluster_resource_package
user_cluster_resource_package.item
user_namespace
user_namespace_map
user_namespace_map.entry
user_session
vps
vps.state_log
vps.config
vps.feature
vps.mount
vps.outage_window
vps.console_token
vps.status
vps_config
monitored_event
monitored_event.log
outage
outage.entity
outage.handler
outage_update
user_outage
vps_outage
vps_outage_mount
help_box
news_log
incoming_payment
payment_stats
user_account
user_payment
user_request
user_request.registration
user_request.change
action_state
```

Výběrem autentizační metody nebo objektu či akce se v nápovědě přibydou další možné přepínače:

```
$ vpsfreectl --auth basic --help
```

```
$ vpsfreectl vps --help
$ vpsfreectl vps list --help
```

Autentizace

HTTP basic

Jméno a heslo buď předáme parametry `--username` a `--password`, nebo parametry vynecháme a program k zadání údajů uživatele vybídne.

```
$ vpsfreectl --auth basic user current
```

Tokeny

Autentizace přes tokeny zavádí několik nových přepínačů:

```
$ vpsfreectl --auth token --help
    ...
    -a, --auth METHOD                Authentication method
    -s, --save                       Save credentials to config file for
later
        --user USER                  User name
        --password PASSWORD          Password
        --token TOKEN                Token
        --token-lifetime LIFETIME    Token lifetime, defaults to
renewable_auto
        --token-interval SECONDS     How long will token be valid in seconds
        --new-token                  Request new token
        --token-via VIA              Send token as a query parameter or in
HTTP header (query_param, header)
    ...
```

Jméno a heslo opět nemusíme předávat jako parametry, program se na ně zeptá.

Jednorázový token:

```
$ vpsfreectl --auth token user current
```

Po ukončení je token zapomenut a příště je nutné vyžádat další. Token pro takové použití postráda smysl.

Uložení tokenu:

```
$ vpsfreectl --auth token --save user current
```

Token se uložil do souboru `~/.haveapi-client.yml` a při dalším spuštění je automaticky načten:

```
$ vpsfreectl user current
```

Takový token ale do 20 minut expiruje a je nutné vyžádat si nový. Je možné si rovnou udělat token s delší platností:

```
$ vpsfreectl --auth token --save --token-interval $((24*60*60)) user current
# 1 den
```

Nebo rovnou token s platností neomezenou:

```
$ vpsfreectl --auth token --save --token-lifetime permanent user current
```

Na token s neomezenou platností je třeba dávat pozor, protože kdokoli s přístupem k `~/.haveapi-client.yml` může token získat a použít.

Akce a parametry

Objekty a akce mají názvy stejné jako v dokumentaci, jen malými písmeny a místo mezer se píší podtržítka. Každá akce má svoje vstupní a výstupní parametry. Zobrazí se přepínačem `--help`, pokud je zadán název objektu i akce:

```
$ vpsfreectl vps list --help
...
Action description:
List VPS

Input parameters:
        --offset OFFSET             The offset of the first object
        --limit LIMIT               The number of objects to retrieve
        --node NODE                 Filter by node
        --location LOCATION         Filter by location
        --environment ENVIRONMENT   Filter by environment
        --os-template OS_TEMPLATE   OS template
        --object-state OBJECT_STATE Object state
    -h, --help                      Show this message

Output parameters:
    id
    user                            VPS owner
    hostname                        VPS hostname
    os_template
    dns_resolver                    DNS resolver the VPS will use
    node                            Node VPS will run on
    dataset                         Dataset the VPS resides in
    created_at
    memory                          Minimally 1024, maximally 12288, step
size is 128
    swap                            Minimally 0, maximally 12288, step size
```

```
is 128
    cpu                             Minimally 1, maximally 8, step size is
1
    maintenance_lock
    maintenance_lock_reason
    object_state
    expiration_date                 A date after which the state will
progress
    running
    process_count
    used_memory                     in MB
    used_disk                       in MB
...
```

Můžeme tedy VPS filtrovat dle serveru, lokace, prostředí, distribuce, stavu a omezit počet zobrazených položek či stránkovat.

Parametry akcí jsou od parametrů klienta odděleny dvěmi pomlčkami `--`. Následující příkaz vypíše první tři VPS:

```
$ vpsfreectl vps list -- --limit 3
```

Pokud bychom chtěli filtrovat např. dle lokace, nejdříve si je vypíšeme:

```
$ vpsfreectl location list
 ID  Label
  3  Praha
  4  Brno
  5  Playground
```

Nyní zobrazíme VPS nacházející se v lokaci Praha:

```
$ vpsfreectl vps list -- --location 3
```

Formátování výstupu

Výstup lze formátovat buď to sloupců, nebo řádků. Ve výchozím stavu se pro zobrazení vícero objektů použijí sloupce a pro jeden objekt řádky. Formát se vybírá přepínači `-c`, `--columns` pro sloupce a `-r`, `--rows` pro řádky.

Ukázka výpisu do sloupců (v tomto případě je přepínač `--columns` nadbytečný – jelikož výstup obsahuje více objektů, zvolí se automaticky):

```
$ vpsfreectl vps list --columns
 ID  Label
  3  Praha
  4  Brno
  5  Playground
```

Ukázka výpisu do řádků:

```
$ vpsfreectl location list --rows
    ID:  3
 Label:  Praha

    ID:  4
 Label:  Brno

    ID:  5
 Label:  Playground
```

Při použití ve skriptech by při formátování do sloupců mohla vadit hlavička s názvy parametrů. Její výpis můžeme potlačit přepínačem `-H`, `--no-header`.

Výběr parametrů k zobrazení

Přepínačem `-o`, `--output` se nastavuje, které výstupní parametry akce a v jakém pořadí budou vypsány. Názvy parametrů se oddělují čárkou.

```
$ vpsfreectl vps list -o id,hostname,node,os_template
      VPS id:  4710
    Hostname:  vps
        Node:  node7.prg (#108)
 OS template:  CentOS 7 (#43)
```

Řazení

Přepínačem `-s`, `--sort` lze výstup vzestupně seřadit podle určitého parametru (na straně klienta).

```
$ vpsfreectl os_template list --sort label
 ID  Label                  Info   Supported
 42  Arch Linux [TEST]      -      0
 24  CentOS 6               -      1
 43  CentOS 7               -      1
 20  Debian 6               -      1
 31  Debian 7               -      1
 38  Debian 7 [TEST]        -      0
 46  Debian 8               -      1
 33  Fedora 20              -      1
 40  Fedora 22              -      1
 14  Gentoo 13.0            -      1
 37  Gentoo [TEST]          -      0
 32  OpenSUSE 12.3          -      1
 26  Scientific Linux 6.6   -      1
 45  Scientific Linux 7     -      0
 30  Ubuntu 12.04           -      1
 35  Ubuntu 14.04           -      1
```

```
 39  Ubuntu 14.04 [TEST]    -      0
 47  openSUSE 13.2 [TEST]   -      0
```

Formát času a data

Klient obsahuje několik přepínačů, kterými se volí formát data a času, vrací-li akce parametr typu `Datetime`.

- `--utc`
- `--localtime`
- `--timestamp`
- `--date-format FORMAT` - formát dle Time#strftime

Blokující mód

Volání akcí jako vytvoření VPS, reinstalace, start/stop/restart, apod., mohou trvat delší dobu. `vpsfreectl` může buď po dobu vykonávání akce blokovat, nebo se hned ukončit a nečekat. Informace o dokončení akce je však důležitá, neboť nad jedním objektem (tedy třeba VPS) lze vykonávat v jednom čase jen jednu akci, ostatní skončí s chybou „Resource is locked“.

Ve výchozím stavu `vpsfreectl` blokuje:

```
$ vpsfreectl vps start 3438

Waiting for the action to complete (hit Ctrl+C to skip)...
Executing:
[=========================================================================]
1/1 transactions
```

Pokud nechceme čekat, můžeme použít přepínač –`no-block`:

```
$ vpsfreectl vps start 3438 --no-block

Run
  vpsfreectl action_state show 653411
or
  vpsfreectl action_state wait 653411
to check the action's progress.
```

Pokud bychom chtěli zkontrolovat stav akce, můžeme využít příkazy, které nám `vpsfreectl` napověděl. Příkaz `action_state show` vypíše aktuální stav a ukončí se, `action_state wait` naopak blokuje, dokud není akce dokončena.

```
$ vpsfreectl action_state show 653411
               Id:  653411
            Label:  Start
         Finished:  true
           Status:  true
```

```
 Current progress:  1
           Total:  1
            Unit:  transactions
      Can cancel:  -
      Created at:  2016-11-26 16:14:05 +0100
      Updated at:  2016-11-26 16:14:05 +0100
```

Aktualizace

Pro využití nových vlastností je potřeba klienta občas aktualizovat, což se provede příkazem

```
$ gem update vpsfree-client
```

Následně můžeme odstranit starou verzi:

```
$ gem cleanup vpsfree-client
```

Tento příkaz odstraní pouze staré verze gemu `vpsfree-client`, ale jeho závislosti zůstanou. Pro odstranění všech starých gemů použijte příkaz

```
$ gem cleanup -n  # vypíše, které gemy by smazal
$ gem cleanup  # smaže gemy
```

Souborový systém

`haveapi-fs` je souborový systém založený na FUSE, tzn. umožňuje připojit API jako souborový systém z userspace. Objekty, akce a jejich parametry můžeme procházet jako adresáře a soubory v libovolném programu. Instaluje se stejným způsobem jako CLI, má jen jiný název:

```
$ gem install haveapi-fs
```

Použití

Použití `haveapi-fs` je podrobně popsáno v README.md, zde jen ve zkratce.

```
$ haveapi-fs https://api.vpsfree.cz /mnt/api.vpsfree.cz
Username: mujlogin
Password:

$ cd /mnt/api.vpsfree.cz
$ ls -1
auth_token
cluster
cluster_resource
dataset
```

```
dataset_plan
dns_resolver
environment
help.html
help.man
help.md
help.txt
ip_address
language
location
mail_template
migration_plan
node
object_history
os_template
snapshot_download
transaction
transaction_chain
user
user_session
vps
vps_config
```

V každém adresáři se nachází soubory `help.{html,txt,md,man}`, které popisují co aktuální adresář obsahuje.

```
$ cat vps/5685/hostname
moje-vps

$ echo lepsi-hostname > vps/5685/hostname
$ echo 1 > vps/5685/save
$ cat vps/5685/actions/update/status
1
$ cat vps/5685/hostname
lepsi-hostname
```

Z ukázky výše lze vidět, že jednotlivé atributy objektů lze měnit jednoduše zapsáním do souboru. Akce uložení se vyvolá zapsáním jedničky, případně je to i spustitelný soubor, takže jej můžeme spustit.

Pro přehlednější vytváření/úpravu objektů lze použít YAML soubory `create.yml`, resp. `update.yml`. Tyto soubory obsahují parametry jako hash a po uložení a uzavření souboru se příslušná akce provede.

Tímto způsobem lze provést jakoukoli operaci v API.

Ukázky použití

Následují skripty demonstrují možnosti využití API. Tyto skripty lze spouštět odkudkoli – manuálně, z

cronu, apod.

Ač jsou tyto ukázky stále validní, klient v CLI už obsahuje příkazy, díky kterým je stahování záloh ještě jednodušší.

Každodenní stahování záloh VPS

```ruby
#!/usr/bin/env ruby
# Stažení poslední (nejnovější) zálohy všech VPS.
require 'vpsfree/client'

api = VpsFree::Client.new
api.authenticate(:token, token: 'PUT YOUR TOKEN HERE')

api.dataset.list(role: :hypervisor).each do |ds|
  last_snapshot = ds.snapshot.list.last

  # Volání této akce může trvat i několik hodin
  dl = api.snapshot_download.create(snapshot: last_snapshot.id)

  unless dl.api_response.ok?
    warn "#{ds.name}@#{last_snapshot.created_at}:
#{dl.api_response.message}"
    next
  end

  # Soubor je připraven ke stažení
  puts "Downloading #{id} to #{dl.file_name} (#{dl.size / 1024} GB)"
  `wget #{dl.url}`

  # Nakonec smažeme soubor ze serveru
  dl.delete
end

puts "Done"
```

Denní zálohování NASu s týdenní historií

```ruby
#!/usr/bin/env ruby
# Denní snapshotovaní NASu se 7 denní historií
require 'vpsfree/client'

api = VpsFree::Client.new
api.authenticate(:token, token: 'PUT YOUR TOKEN HERE')

api.dataset.list(role: :primary).each do |ds|
  # Vytvoření nového snapshotu
  ds.snapshot.create
```

```
  # Smazání starších snapshotů
  t = Time.now - 7 * 24 * 60 * 60
  snapshots = ds.snapshot.list(meta: {count: true})
  deleted =

  snapshots.each do |snap|
    # Necháme si nejméně 7 snapshotů
    break if snapshots.total_count - deleted < 7

    # Smazaní přebytečných snapshotů starších než 7 dní
    if snap.created_at < t
      puts "Delete #{ds.name}@#{snap.created_at}"
      snap.delete
      deleted += 1
    end
  end
end
```

- aither