

Hostija — VPS Reseller API

Provision & manage Linux VPS programmatically at wholesale prices.

Base URL: `https://api.offsh.nl`

Client area: `my.hostija.com`

REST + HMAC

WHMCS module

White-label

Prepaid

Contents

- | | |
|----------------------------------|---|
| 1. Overview | 10. Server lifecycle & states |
| 2. Key concepts | 11. Operating systems |
| 3. Getting started | 12. Errors |
| 4. Plans & specifications | 13. How your customers manage their VPS |
| 5. Volume pricing (recurring) | 14. WHMCS module |
| 6. Currency & billing | 15. Security best practices |
| 7. Authentication (HMAC signing) | 16. FAQ |
| 8. Idempotency & rate limits | 17. Support |
| 9. Endpoint reference | |

1. Overview

The Hostija VPS Reseller API lets you create and manage Linux VPS **under your own brand**, billed from your prepaid **account credit** at wholesale prices. You resell to your own customers at whatever price you choose and keep the margin. Your customers manage their servers through a one-time SSO link into a neutral white-label panel — they never see our brand and never get a login from you to manage.

- **Two ways to integrate:** install our **WHMCS module** for a point-and-click setup, or call the **REST API** directly from your own scripts, panel, or bots.
- **Prepaid credit:** top up your Hostija account balance; each new server and each monthly renewal debits it automatically.
- **Recurring volume tiers:** the more live servers you run, the bigger your standing discount — applied to every renewal, not just new orders.
- **Fully white-label:** customers only ever see your brand.

2. Key concepts

Term	Meaning
Account credit	Your prepaid balance in your Hostija client area. Every provision and renewal is charged against it. Top up via Add Funds .
Wholesale price	Our retail price minus your live volume-tier discount. This is what you actually pay per server per month.
Volume tier	A recurring discount band based on how many live servers you run right now (recomputed on every charge).
client_ref	Your own end-customer identifier that you attach to each server, keeping your tenants isolated in your own records.
SSO / console URL	A short-lived link that drops your customer straight into the white-label panel to manage just their server.
API key + HMAC secret	Your credentials. The key identifies you; the secret signs each request. Both are shown in your client dashboard.

3. Getting started

1. **Order the VPS Reseller API package** in your Hostija client area (my.hostija.com). This unlocks your API access.
2. **Open the VPS Reseller API dashboard** in your client area and click **Generate key**. You'll get an **API key** and an **HMAC secret** — copy the secret now; it's shown once.
3. **Top up your account credit** (Add Funds). New servers and renewals are charged from this balance.
4. **Integrate:** either install the [WHMCS module](#), or call the [REST API](#) directly. Test connectivity with [GET /v1/ping](#).
5. **Provision your first VPS** with [POST /v1/servers](#) (or an order in WHMCS).

Fast check: once you have keys, [GET /v1/account](#) returns your balance, currency, tier and live server count — a one-call health check that your credentials work.

4. Plans & specifications

Linux VPS only. All plans include **unmetered traffic**, full root access, IPv4 + IPv6, NVMe storage, and a choice of Linux OS templates. Live plans and your current wholesale price are always available from [GET /v1/products](#).

ID	Plan	vCPU	RAM	NVMe disk	Traffic	Retail EUR/mo	Retail USD/mo
1	Linux VPS V1	1	2 GB	25 GB	Unmetered	€8.50	\$9.99
2	Linux VPS V2	2	4 GB	40 GB	Unmetered	€15.50	\$16.99
3	Linux VPS V3	4	8 GB	80 GB	Unmetered	€27.50	\$29.99
4	Linux VPS V4	6	16 GB	120 GB	Unmetered	€35.50	\$38.99
5	Linux VPS V5	8	24 GB	160 GB	Unmetered	€44.50	\$47.99
6	Linux VPS V6	10	32 GB	200 GB	Unmetered	€59.50	\$63.99

Retail prices above are the reference list price; **you pay wholesale** = retail minus your tier discount (see next section). Prices are per month.

5. Volume pricing (recurring)

Your wholesale price = retail price minus your tier discount, applied to **every** renewal and recalculated automatically as your fleet grows or shrinks:

Live servers	Discount off retail
1 - 25	15%
26 - 50	20%
51+	30%

Full price list — EUR (per month)

Plan	Specs	Retail	15% (1-25)	20% (26-50)	30% (51+)
V1	1 vCPU · 2 GB · 25 GB	€8.50	€7.23	€6.80	€5.95
V2	2 vCPU · 4 GB · 40 GB	€15.50	€13.18	€12.40	€10.85
V3	4 vCPU · 8 GB · 80 GB	€27.50	€23.38	€22.00	€19.25
V4	6 vCPU · 16 GB · 120 GB	€35.50	€30.18	€28.40	€24.85
V5	8 vCPU · 24 GB · 160 GB	€44.50	€37.83	€35.60	€31.15
V6	10 vCPU · 32 GB · 200 GB	€59.50	€50.58	€47.60	€41.65

Full price list — USD (per month)

Plan	Specs	Retail	15% (1-25)	20% (26-50)	30% (51+)
V1	1 vCPU · 2 GB · 25 GB	\$9.99	\$8.49	\$7.99	\$6.99
V2	2 vCPU · 4 GB · 40 GB	\$16.99	\$14.44	\$13.59	\$11.89
V3	4 vCPU · 8 GB · 80 GB	\$29.99	\$25.49	\$23.99	\$20.99
V4	6 vCPU · 16 GB · 120 GB	\$38.99	\$33.14	\$31.19	\$27.29
V5	8 vCPU · 24 GB · 160 GB	\$47.99	\$40.79	\$38.39	\$33.59
V6	10 vCPU · 32 GB · 200 GB	\$63.99	\$54.39	\$51.19	\$44.79

Add-ons: Extra IPv4 = €2.50 / \$2.95 per IP per month (flat, no tier discount, up to 8 per server). Order via the API/WHMCS `extra_ips` field at create; extra IPs recur with the server. **OS choice** is free — pick any Linux template at create (`os_id`), and your customer can reinstall to a different OS anytime in the panel.

The tier is based on your live server count at the moment of each charge, so as you cross 26 or 51 live servers every renewal automatically moves to the better rate. Your current tier and live wholesale prices are returned by `GET /v1/products` and `GET /v1/account`.

6. Currency & billing

The API supports **EUR** and **USD**. Your default billing currency follows your Hostija account currency, and all prices from `/v1/products` and every charge are shown in it.

- **Both balances visible:** your dashboard shows your EUR and USD account credit side by side; the one used for VPS charges is marked as your billing currency.
- **Use either currency:** a charge is taken from your default currency if it has enough balance; if not, it automatically falls back to your other currency when that balance covers the price. Keep whichever you prefer topped up.
- **New server** = charged immediately at your tiered wholesale price.
- **Monthly renewal** = each server auto-charges its wholesale price on its due date. If credit runs out, the server is suspended after a short grace period and automatically restored once you top up.
- **Top up** anytime from your Hostija client area → **Add Funds**.

7. Authentication (HMAC signing)

Every request is signed with your API key + HMAC secret (both from your client dashboard). Send these three headers:

Header	Value
<code>X-API-Key</code>	your API key
<code>X-Timestamp</code>	current Unix time in seconds; must be within 5 minutes of server time
<code>X-Signature</code>	<code>base64(hmac_sha256(secret, "METHOD\nPATH\nsha256hex(body)"))</code>

The signing string is four lines joined by newlines: the HTTP method, the request path (no host, no query for signing purposes use the exact path), the timestamp, and the lowercase hex SHA-256 of the raw request body (use the SHA-256 of an empty string for GET requests).

Signing example — bash

```
APIKEY="your_api_key"
SECRET="your_hmac_secret"
TS=$(date +%s)
BODY='{"package_id":1,"os_id":14,"hostname":"web1.customer.com","client_ref":"42"}'
BH=$(printf '%s' "$BODY" | openssl dgst -sha256 -r | awk '{print $1}')
CANON=$(printf 'POST\n/v1/servers\n%s\n%s' "$TS" "$BH")
SIG=$(printf '%s' "$CANON" | openssl dgst -sha256 -hmac "$SECRET" -binary | base64)
curl -X POST https://api.offsh.nl/v1/servers \
  -H "X-API-Key: $APIKEY" -H "X-Timestamp: $TS" -H "X-Signature: $SIG" \
  -H "Content-Type: application/json" --data "$BODY"
```

Signing example — PHP

```
$apiKey = 'your_api_key';
$secret = 'your_hmac_secret';
$ts = (string) time();
$body =
json_encode(['package_id'=>1,'os_id'=>14,'hostname'=>'web1.customer.com','client_ref'=>'42']);
$bh = hash('sha256', $body);
$canon= "POST\n/v1/servers\n{$ts}\n{$bh}";
$sig = base64_encode(hash_hmac('sha256', $canon, $secret, true));
$ch = curl_init('https://api.offsh.nl/v1/servers');
curl_setopt_array($ch, [
    CURLOPT_POST => true, CURLOPT_POSTFIELDS => $body, CURLOPT_RETURNTRANSFER => true,
    CURLOPT_HTTPHEADER => ["X-Api-Key: {$apiKey}","X-Timestamp: {$ts}","X-Signature:
{$sig}","Content-Type: application/json"],
]);
echo curl_exec($ch);
```

Signing example — Python

```
import time, hashlib, hmac, base64, json, requests
api_key = "your_api_key"; secret = b"your_hmac_secret"
ts = str(int(time.time()))
body = json.dumps({"package_id":1,"os_id":14,"hostname":"web1.customer.com","client_ref":"42"})
bh = hashlib.sha256(body.encode()).hexdigest()
canon = f"POST\n/v1/servers\n{ts}\n{bh}".encode()
sig = base64.b64encode(hmac.new(secret, canon, hashlib.sha256).digest()).decode()
r = requests.post("https://api.offsh.nl/v1/servers", data=body, headers={
    "X-Api-Key": api_key, "X-Timestamp": ts, "X-Signature": sig, "Content-Type":"application/json"})
print(r.status_code, r.text)
```

8. Idempotency & rate limits

- **Idempotency:** add `Idempotency-Key: <unique>` on `POST /v1/servers`. A retried create with the same key returns the original result instead of provisioning or charging twice — essential for safe retries on network timeouts.
- **Rate limits:** requests are rate-limited per API key. If you exceed the limit you receive `429 rate_limited`; back off and retry. Read-only endpoints (ping/account/products/os) are generously limited.
- **Clock skew:** keep your server clock accurate (NTP). Requests more than 5 minutes off are rejected as `stale_timestamp`.

9. Endpoint reference

Method	Path	Purpose
GET	/v1/ping	health + auth check
GET	/v1/account	credit balance, currency, tier, server counts
GET	/v1/products	plans + your wholesale price + current tier
GET	/v1/os	available OS templates (id + name)
GET	/v1/regions	locations
POST	/v1/servers	provision (create + build)
GET	/v1/servers/{id}	status, IP, power state, specs
POST	/v1/servers/{id}/suspend	suspend
POST	/v1/servers/{id}/unsuspend	unsuspend
DEL	/v1/servers/{id}	terminate (deletes the VPS)
POST	/v1/servers/{id}/actions	boot / reboot / shutdown / poweroff / rebuild
GET	/v1/servers/{id}/console	one-time SSO/console URL for your customer

GET /v1/account

```
-> 200 {
  "reseller": { "id": 3, "currency": "EUR" },
  "balance": 192.77,
  "tier": { "servers": 12, "discount_pct": 15, "label": "1-25 servers (15% off)" }
}
```

GET /v1/products

```
-> 200 {
  "tier": { "servers_sold": 12, "discount_pct": 15, "label": "1-25 servers (15% off)" },
  "products": [
    { "id":1, "name":"Linux VPS V1 - 2GB", "memory_mb":2048, "cpu_cores":1,
      "disk_gb":25, "traffic":"unmetered", "retail":8.50, "wholesale":7.23, "currency":"EUR" },
    ...
  ]
}
```

POST /v1/servers — create a server

Provisions and builds a VPS, charges your account at the tiered wholesale price, and (optionally) emails your customer their login + server details.

Field	Req?	Description
<code>package_id</code>	yes	plan id from <code>/v1/products</code> (1-6)
<code>os_id</code>	yes	OS template id from <code>/v1/os</code>
<code>hostname</code>	yes	FQDN/hostname for the VPS
<code>client_ref</code>	yes	your end-customer id (keeps your tenants isolated)
<code>email</code>	no	customer email — if set, the panel emails them their login + server details on build
<code>extra_ips</code>	no	additional IPv4 addresses (0-8), €2.50 / \$2.95 each per month on top of the plan
<code>ssh_key</code>	no	public SSH key to inject at build

```
POST /v1/servers
{
  "package_id": 1,
  "os_id": 14,
  "hostname": "web1.customer.com",
  "client_ref": "42",
  "email": "customer@example.com",
  "extra_ips": 2,
  "ssh_key": "ssh-ed25519 AAAA..."
}
-> 201 { "server": { "id": 123, "status": "provisioning", "extra_ips": 2,
  "charged": 12.23, "currency": "EUR", "balance": 187.77 } }
```

GET /v1/servers/{id}

```
-> 200 { "server": { "id":123, "status":"active", "hostname":"web1.customer.com",
  "ip":"185.x.x.x", "power":"running", "package_id":1, "os_id":14 } }
```

POST /v1/servers/{id}/actions

```
POST /v1/servers/123/actions {"action":"reboot"}
POST /v1/servers/123/actions {"action":"rebuild","os_id":12}
-> 200 { "server": { "id":123, "action":"reboot", "accepted":true } }
```

Valid actions: `boot`, `reboot`, `shutdown`, `poweroff`, `rebuild` (rebuild requires `os_id`).

Suspend / unsuspend / terminate

```
POST /v1/servers/123/suspend -> 200 {"server":{"id":123,"status":"suspended"}}
POST /v1/servers/123/unsuspend -> 200 {"server":{"id":123,"status":"active"}}
DELETE /v1/servers/123 -> 200 {"server":{"id":123,"status":"terminated"}}
```

GET /v1/servers/{id}/console — give your customer the panel

```
-> 200 { "console_url": "https://vps-panel.com/sso/...", "expires_in": 60 }
```

Redirect your customer to `console_url` — they land in the white-label panel managing just their server (power, VNC, reinstall), no login required.

10. Server lifecycle & states

State	Meaning
provisioning	create accepted; VF is building the VM (IP appears shortly after)
active	built and running; counts toward your volume tier and renews monthly
suspended	stopped — either you suspended it, or credit ran out past the grace period. Still counts toward tier; auto-unsuspends when you top up
terminated	deleted; no longer billed or counted

Renewal flow: each active server has a monthly due date. On that date the API charges your account the current wholesale price. On success the due date advances a month. On failure you get a short grace period; if still unpaid, the server is suspended, and it auto-restores once your balance covers the charge.

11. Operating systems

Fetch the live list from `GET /v1/os`. Currently available templates:

ID	OS	ID	OS
2	CentOS Stream 9	13	Debian 13 (Trixie)
5	Rocky Linux 8	3	Ubuntu 20.04 LTS
9	Rocky Linux 9	7	Ubuntu 22.04 LTS
10	Rocky Linux 10	14	Ubuntu 24.04 LTS
8	AlmaLinux 9	4	openSUSE Leap 15
17	AlmaLinux 10	15	Fedora 42
11	Oracle Linux 9	16	Fedora 43
6	Debian 11 (Bullseye)	18	CloudLinux 9
12	Debian 12 (Bookworm)		

12. Errors

Errors are JSON: `{ "error": "code", "message": "human readable" }` with an appropriate HTTP status.

HTTP	error	meaning / fix
400	bad_package / bad_os / bad_hostname / bad_client_ref	a required field is missing or invalid
401	bad_signature / stale_timestamp / invalid_key	check your secret, system clock, and that the key is active
402	insufficient_credit	top up your balance (or your other currency)
404	not_found	unknown server, or it isn't yours
409	terminated	the server is already terminated
429	rate_limited	slow down and retry
502	billing_unreachable / upstream_error	transient backend issue — retry shortly

13. How your customers manage their VPS

Your customers never need a separate login from you. Two things happen automatically:

- **Automatic welcome email on creation:** the moment a VPS is built, **vps-panel.com emails your customer directly** with their **panel login** (username & password) and their **SSH server details** (server IP address and root password). Just include the customer's `email` when you create the server — the WHMCS module passes it automatically; via the raw API, send the `email` field on `POST /v1/servers`.
- **Self-service management via SSO:** your customer manages the VPS (boot, reboot, reinstall, VNC console) in the panel — reachable from that welcome email, or via a one-time SSO link you generate with `GET /v1/servers/{id}/console` (the WHMCS module exposes this as a **Manage in Panel** button). No password for you to hand out.

White-label: the panel is the neutral vps-panel.com — your customers never see our company name or that you resell from us.

14. WHMCS module

Prefer a UI? Install the Hostija WHMCS module (download it from your VPS Reseller API dashboard). It handles create / suspend / unsuspend / terminate automatically, emails your customer their panel login on creation, and gives them a **Manage in Panel** SSO button.

Install

1. Unzip into `whmcs/modules/servers/vpsreseller/`.
2. In WHMCS go to **Setup → Products/Services → Servers → Add New Server**.
3. Set: **Hostname** = `https://api.offsh.nl`, **Username** = your API key, **Password** = your HMAC secret. Save, then **Test Connection**.
4. Create a Product (type **Hostija VPS Reseller**), assign the server, and in the module settings pick the **Plan** and default **OS** (both live-fetched from the API).
5. Orders now provision automatically; the customer receives their welcome email and a **Manage in Panel** button in their client area.

The module signs every request for you — you only ever paste your API key + secret into the WHMCS server config. Keep the secret private.

15. Security best practices

- **Guard your HMAC secret** like a password. It's shown once at generation — store it in your secrets manager. If it leaks, click **Regenerate key** in the dashboard to rotate instantly (old key stops working).
- **Use Idempotency-Key** on creates so retries never double-charge.
- **Keep NTP running** so your timestamps stay within the 5-minute window.
- **Use HTTPS only** (the API is TLS-only; never send keys over plain HTTP).
- **Rotate periodically** and immediately if a machine holding the secret is decommissioned.

16. FAQ

Do my customers see your brand? No. Management is via a neutral white-label panel; emails and the panel carry no reference to us.

What happens if I run out of credit? The affected server is suspended after a short grace period and automatically restored when you top up. Data is retained during suspension.

Can I change a server's OS later? Yes — `POST /v1/servers/{id}/actions` with `{"action": "rebuild", "os_id": N}`. This wipes the server and reinstalls.

Windows VPS? Not offered — Linux VPS only.

How is my discount decided? Automatically, by your live server count at each charge. No need to request tier changes.

Can I bill in both EUR and USD? Yes — you can hold both balances. Charges use your default currency, falling back to the other if it covers the price.

17. Support

Open a ticket in your Hostija client area at **my.hostija.com**. Include the server id and the API response (error code + message) so we can help quickly. Keep your HMAC secret private — never paste it into a ticket.