

Integracao Bling API v3

Base de Conhecimento

Vullos Company

12 de Julho de 2026 | Versao 1.0.0

<https://bling.vullos.com.br>

Sumario

- 1 Visao Geral
- 2 Infraestrutura
- 3 Arquitetura
- 4 Servidor de Callback
- 5 Passo a Passo para Configuracao
- 6 Uso da API
- 7 Manutencao
- 8 Seguranca
- 9 Troubleshooting
- 10 Referencia Rapida

Base de Conhecimento — Integração Bling API v3

Projeto: Vullos Company

Subdomínio: bling.vullos.com.br

Versão: 1.0.0 — 12/07/2026

Visão Geral

Sistema de integração com a **API v3 do Bling** utilizando OAuth2.

O fluxo é:

1. `curl -X POST https://api.bling.com.br/oauth2/authorize --header 'Authorization: Bearer `refresh_token`'`

Infraestrutura

DNS

Registro	Tipo	Valor
bling.vullos.com.br	A	2.24.126.57

Certificado SSL

- `openssl req -x509 -newkey rsa:2048 -keyout key.pem -out cert.pem -days 365` (recorre a cada 12h)

Nginx (Docker)

Container: ``auditly-nginx-1``

Arquivo de configuração: ``/opt/auditly/nginx/default.conf``

Montado como read-only no container em ``/etc/nginx/conf.d/default.conf``

Server block criado:

```

server {
    listen 443 ssl;
    http2 on;
    server_name bling.vullos.com.br;
    ssl_certificate /etc/letsencrypt/live/bling.vullos.com.br/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/bling.vullos.com.br/privkey.pem;
    ssl_protocols TLSv1.2 TLSv1.3;
    ssl_prefer_server_ciphers off;
    ssl_session_cache shared:SSL:10m;
    ssl_session_timeout 1d;
    client_max_body_size 10M;

    location / {
        proxy_pass http://172.18.0.1:8765;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection "upgrade";
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
        proxy_set_header X-Forwarded-Host $host;
        proxy_set_header X-Forwarded-Port 443;
        proxy_cache_bypass $http_upgrade;
        proxy_read_timeout 60s;
    }
}

```

Firewall (UFW)

Status: Ativo

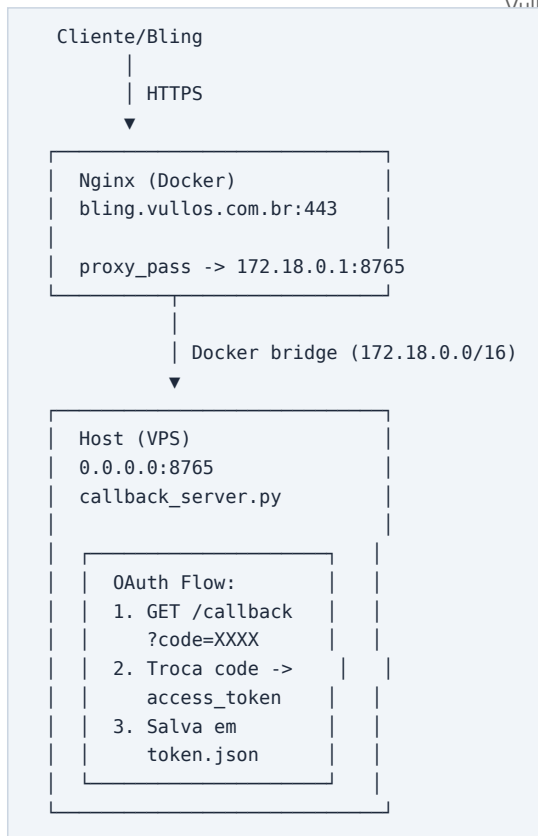
Política padrão: Deny incoming, Allow outgoing

Porta	Serviço	Acesso
22/tcp	SSH	Público
80/tcp	HTTP	Público
443/tcp	HTTPS	Público
8443/tcp	Vaultwarden	Público
8642/tcp	Hermes API	Localhost apenas
8765/tcp	Bling Callback	Rede Docker (172.18.0.0/16)
8025/tcp	Mailpit UI	Localhost apenas
1025/tcp	Mailpit SMTP	Localhost apenas

Portas do Servidor

Porta	Processo	Binding
8765	callback_server.py	0.0.0.0 (protegido por UFW)
8642	Hermes Agent API	0.0.0.0 (UFW: só localhost)
80	Nginx (HTTP)	0.0.0.0
443	Nginx (HTTPS)	0.0.0.0

Arquitetura



Fluxo de Autorização OAuth2

1. Bling retorna uma URL de autorização: `https://bling.com.br/autorizacao?code=XXXX&redirect_uri=https://bling.vullos.com.br/callback`

Renovação de Token

Quando o `access\_token` expira, usa-se `refresh\_token`:

- `curl -X POST https://api.bling.com.br/v3/oauth/token -H 'Content-Type: application/json' -d '{ "refresh_token": "XXXX", "grant_type": "refresh_token" }'`

Servidor de Callback

Localização

```

/home/hermes/bling-integration/
├─ callback_server.py    -> Servidor HTTP (porta 8765)
├─ config.yaml          -> Credenciais e configurações
├─ token.json           -> Token OAuth salvo automaticamente
└─ server.log           -> Log do servidor
  
```

Configuração (config.yaml)

```

client_id: ""           # Client ID da app Bling (preencher)
client_secret: ""       # Client Secret da app Bling (preencher)
redirect_uri: "https://bling.vullos.com.br/callback"

scopes:
  - anuncios_marketplaces
  - campos_customizados

callback_port: 8765
  
```

Endpoints

Rota	Método	Descrição
`/`	GET	Página de status da integração
`/callback`	GET	Recebe `?code=XXXX` do Bling
`/callback`	POST	Recebe JSON `{ "code": "XXXX" }`
`/status`	GET	JSON com status do servidor e token

Serviço Systemd

Nome: `bling-callback.service`

Arquivo: `/etc/systemd/system/bling-callback.service`

```
[Unit]
Description=Bling API v3 Callback Server - Vullos Integration
After=network.target docker.service
Requires=docker.service

[Service]
Type=simple
User=hermes
WorkingDirectory=/home/hermes/bling-integration
ExecStart=/usr/bin/python3 /home/hermes/bling-integration/callback_server.py
Restart=always
RestartSec=5

[Install]
WantedBy=multi-user.target
```

Comandos:

```
sudo systemctl status bling-callback    # Ver status
sudo systemctl start bling-callback    # Iniciar
sudo systemctl stop bling-callback     # Parar
sudo systemctl restart bling-callback  # Reiniciar
sudo journalctl -u bling-callback -f   # Ver logs
```

Passo a Passo para Configuração

No Site do Bling

2. **Exemplo de Configuração de um Servidor de Integração com a API de Integração do Bling**

Na VPS

1. **Edite o arquivo:**

```
nano /home/hermes/bling-integration/config.yaml
```

Preencha `client\_id` e `client\_secret`

2. **Reinicie o servidor:**

```
sudo systemctl restart bling-callback
```

3. **Autorize o acesso Bling - você será redirecionado para:**

```
https://bling.vullos.com.br/callback?code=XXXX
```

4. **Verifique se o token foi salvo:**

```
cat /home/hermes/bling-integration/token.json
```

Verificação

```
# Testar se o servidor está rodando
curl http://127.0.0.1:8765/status

# Testar via nginx
curl https://bling.vullos.com.br/status

# Ver logs
journalctl -u bling-callback -f --no-pager
```

Uso da API

Endpoints da API Bling v3

Com o `access\_token` em mãos, use como header:

```
Authorization: Bearer <access_token>
```

Exemplos:

```
# Listar produtos
curl -H "Authorization: Bearer TOKEN" \
  "https://www.bling.com.br/Api/v3/produtos"

# Criar produto
curl -X POST \
  -H "Authorization: Bearer TOKEN" \
  -H "Content-Type: application/json" \
  -d '{"nome": "Produto Teste", "preco": 99.90}' \
  "https://www.bling.com.br/Api/v3/produtos"

# Buscar produto por ID
curl -H "Authorization: Bearer TOKEN" \
  "https://www.bling.com.br/Api/v3/produtos/123"
```

Script de Exemplo (Python)

```
import json
import urllib.request

TOKEN_FILE = "/home/hermes/bling-integration/token.json"

# Carrega token
with open(TOKEN_FILE) as f:
    token = json.load(f)
    access_token = token["access_token"]

# Lista produtos
req = urllib.request.Request(
    "https://www.bling.com.br/Api/v3/produtos",
    headers={"Authorization": f"Bearer {access_token}"})
)
with urllib.request.urlopen(req) as resp:
    produtos = json.loads(resp.read())
    print(f"Produtos encontrados: {len(produtos)}")
```

Manutenção

Renovação do Certificado SSL

Automática (certbot container recorre a cada 12h).
Para renovar manualmente:

```
sudo docker exec auditly-certbot-1 certbot renew
```

Logs

```
# Log do callback server
tail -f /home/hermes/bling-integration/server.log

# Log do nginx
sudo docker logs auditly-nginx-1 --tail 50

# Log do sistema (serviço)
sudo journalctl -u bling-callback -f
```

Backup

```
# Backup da integração completa
tar -czf bling-integration-backup-$(date +%Y%m%d).tar.gz \
/home/hermes/bling-integration/

# Backup apenas do token (essencial)
cp /home/hermes/bling-integration/token.json \
/home/hermes/bling-integration/token.json.bak
```

Reinicialização Completa

```
# Se o servidor de callback parar
sudo systemctl restart bling-callback

# Se o nginx precisar ser reiniciado
sudo docker restart auditly-nginx-1

# Após modificar config.yaml
sudo systemctl restart bling-callback
```

Segurança

Firewall

UFW ativo com política `deny incoming`.
 Portas liberadas apenas as essenciais.
 Porta 8765 só acessível pela rede Docker (172.18.0.0/16).

Lições Aprendidas

- ⚠ **Chrome remote debugging:** Sempre usar `--remote-debugging-address=127.0.0.1`
- ⚠ **Docker containers** com portas internas devem bindar em `127.0.0.1`, não `0.0.0.0`
- ⚠ **UFW não protege containers Docker** automaticamente — é preciso bind em localhost

Arquivos Sensíveis

Arquivo	Risco	Proteção
`config.yaml`	Client ID e Secret	Apenas usuário hermes
`token.json`	Access Token	Apenas usuário hermes
`/etc/letsencrypt/live/`	Chaves SSL	Root apenas

Troubleshooting

"Empty reply from server" no nginx

Causa: Config desatualizada no container (bind mount read-only)

Solução: `sudo docker restart auditly-nginx-1`

Certificado SSL não emite

Causa: DNS não propagou ou nginx não está servindo ACME challenge

Solução: Verificar DNS, adicionar domínio no server block HTTP, reiniciar nginx

Servidor de callback não inicia

Causa: Porta 8765 ocupada ou erro de sintaxe no Python

Solução:

```
sudo lsof -i :8765
python3 -c "import py_compile; py_compile.compile('callback_server.py', doraise=True)"
```

Token não é salvo

Causa: `client\_id` ou `client\_secret` incorretos no config.yaml

Solução: Verificar credenciais, reiniciar servidor, reautorizar

Acesso negado ao editar nginx config

Causa: Arquivo pertence ao root

Solução: Usar `sudo` para editar `/opt/auditly/nginx/default.conf`

Referência Rápida

Comandos Essenciais

```
# Status do servidor
curl https://bling.vullos.com.br/status

# Ver servidor rodando
ps aux | grep callback_server

# Log do servidor
tail -f /home/hermes/bling-integration/server.log

# Reiniciar servidor
sudo systemctl restart bling-callback

# Status do nginx
sudo docker exec auditly-nginx-1 nginx -t

# Log do nginx
sudo docker logs auditly-nginx-1 --tail 20

# Status do firewall
sudo ufw status verbose
```

Caminhos Importantes

Caminho	Descrição
`/home/hermes/bling-integration/`	Projeto de integração
`/home/hermes/bling-integration/callback_server.py`	Servidor de callback
`/home/hermes/bling-integration/config.yaml`	Configuração
`/home/hermes/bling-integration/token.json`	Token OAuth
`/opt/auditly/nginx/default.conf`	Configuração do nginx
`/etc/systemd/system/bling-callback.service`	Serviço systemd
`/etc/letsencrypt/live/bling.vullos.com.br/`	Certificado SSL

Documentação gerada em 12/07/2026 — Vullos Company © 2026