

Lunium API · Manual de integração

Para integradores B2B · versão de **02/09/2026** · base <https://api.luniumpay.com> · autenticação: header X-API-Key · contrato OpenAPI 1.12.0

A Lunium é a camada de liquidação entre PIX e criptomoedas no Brasil. Uma única chave faz as três coisas: **cash-in** (cliente paga PIX e recebe cripto na carteira), **cash-out** (cliente entrega cripto e uma chave PIX recebe reais) e **saldo/payouts** (custódia em reais para o cliente final, com saque por PIX ou em cripto). Toda operação paga sai com comprovante e com o identificador oficial do Banco Central (E2E), verificável por qualquer pessoa sem chave. Tudo que está neste manual está **em produção e foi medido** na data acima.

R\$ 200 / dia

cash-in instantâneo por CPF/CNPJ, já na 1ª operação

até R\$ 6.000 / dia

cash-in aceito com retenção de 24h no provedor (estado `deLayed`)

2 s · 7 s

PIX pago → USDT / USDC na carteira (mediana, últimos 30 dias, Polygon)

R\$ 6 → R\$ 250 mil

cash-out por transação · R\$ 100 mil/dia por recebedor · sem teto diário por chave

49–65 s

cash-out do aceite ao PIX pago, Polygon (medido em produção)

2,5% + R\$ 1,00

taxa padrão da chave no cash-in · cash-out: cotação já líquida

Sumário

1. Fluxograma
2. Primeiros passos: sandbox, produção, grupo de operação
3. Cash-in: PIX → cripto
 - 3.1 Limites: instantâneo × retenção de 24h (como usar)
 - 3.2 Requisição, resposta e estados
 - 3.3 Ativos, prazos e taxas
4. Depósito em saldo (custódia) e saques
5. Cash-out: cripto → PIX
 - 5.1 Limites, redes e prazos
 - 5.2 Estados, devoluções e comprovante
 - 5.3 Sandbox: gatilhos determinísticos
6. Payouts (PIX direto)
7. Webhooks
8. Erros: `erro` + `acao`
9. Tabela-resumo de limites e taxas
10. Checklist de go-live · 11. Links e canais

1. Fluxograma

Versão em alta resolução: <https://docs.luniumpay.com/manual-fluxograma.png>.

Lunium API · fluxograma de integração

Produção em 02/09/2026 · base <https://api.luniumpay.com> · header X-API-Key · OpenAPI 1.12.0 · uma chave faz cash-in, cash-out, payouts e saldo

0 · Setup da chave de teste à produção

POST /keys/sandbox

1 chamada, chave lun_test_... Mesma URL, mesmos estados, nada se move. Galinhos .81-.88 no amount ensaiam os dias ruins.

Integra e testa

Quickstart
api.luniumpay.com/quickstart · docs.luniumpay.com · POST /webhooks/test · MCP /mcp

POST /keys ("name")

Chave de produção lun_... na hora. Tier Início: R\$ 5.000/dia em cash-in + payouts (cresce com o volume liquidado).

Grupo de operação (fortemente recomendado)

Grupo Telegram com @LuniumNotifyBot + @luniumB2B. Um admin manda /vincular 1k... Todo PIX/USDT avisa ali, PDF do comprovante incluso.

PATCH /keys/me

settlement_address (carteira Polygon que recebe o cash-in), webhook_url, chain. Confira tudo em GET /keys/me.

1 · Cash-in cliente paga PIX → recebe USDT/USDC (ou outro ativo)

GET /cashin/limits?

payer_tax=CPF
Pergunte antes de cobrar: instant_available_cents (na hora), max_amount_cents (máximo aceite, retenção incluída), held_gr.

POST /cashin/charge

amount_cents · payer_tax_number (CPF/CNPJ, exigência do BCB) · asset + chain · payout_address · external_id. Devolve QR + copia-e-cola + expires_at.

Valor no dia daquele CPF/CNPJ

até R\$ 200 instantâneo · QR sai held:false · vale já na 1ª operação

R\$ 200 → R\$ 6.000 aceite com retenção de 24h · QR sai held:true, held_hours:24 · pago, fica delayed até delay_until · também na 1ª operação

acima de R\$ 6.000/dia 403 limite_do_pagador (acao: corrigir) · min. R\$ 1,00 · máx. R\$ 6.000 por cobrança

Cliente paga o PIX

QR ou copia-e-cola, valor exato. Provedor confirma; o E2E e o nome do pagador entram em GET /cashin/{id}/status.

Liquidação na carteira

Instantâneo: paid → USDT/USDC Polygon enviado ao payout_address (medidor: p50 2 s USDT, 7 s USDC) → settlement_status: sent + settlement_tx_hash. Retido: delayed ~24h → paid + mesma liquidação.

pending → delayed (só quando retido) → under_review → paid → settlement sent · terminais: expired refunded failed · webhooks:

cashin.delayed · cashin.paid · cashin.settled · cashin.expired · cashin.refunded · cashin.failed

Taxa padrão da chave: 2,5% + R\$ 1,00 por cobrança, descontada antes de converter (POST /cashin/preview mostra líquido e cotação). Ativos: GET /cashin/catalog → instant (USDT/USDC Polygon, segundos) e convert (1.300+ ativos via exchange, minutos). Recusa do provedor por compliance vira 403 pagador_recusado_pelo_provedor com provider_reference — não é queda, não repita.

1b · Depósito em saldo custódia BRL para o cliente final

POST /cashin/charge

destino: "saldo" + customer_ref
Nada sai on-chain: o PIX vira saldo em reais na sub-conta do seu cliente. Depósito 0%.

Valor do depósito

até R\$ 60 credita na hora, disponível D+1 (disponivel_en)

acima de R\$ 60 QR sai retido 24h no provedor → saldo aparece congelado (delayed) → libera disponível na hora (a retenção conta como carência)

Saque

POST /payouts (PIX, 1,8% + ~R\$ 1,00 do provedor, chave do mesmo CPF) ou POST /saldo/sacar · cripto: 1º depósito da sub-conta trava saque por 24h (423). GET /saldo · /saldo/extrato.

Precisa da custódia habilitada na chave (custodia_ativa em GET /saldo) — pedimos no grupo. Janela de 24h antes de o dinheiro sair = espaço para MED/contestação.

3 · Comprovante e verificação o que a ordem paga devolve

Na resposta e no webhook

pix_e2e (BCB) · receipt_url (página) · receipt_pdf_url (PDF) · verify_url. Nunca monte URL na mão.

GET /v1/verificar/{e2e}

Aberto, sem chave: qualquer contraparte confirma que o PIX aconteceu (valor, data, iniciais, instituição). Não expõe chave, nome completo nem documento.

Webhooks

Assinatura X-Lunium-Signature (t=, v1= HMAC de "Lcorpo", rejete >5 min) · dedupe por event_id · 12 tentativas/~7h · GET /webhooks/deliveries /retry /test.

Erros: todo erro traz erro (código estável) + acao (corrigir · repetir · esperar · parar) + agent_guidance. Ramifique pela acao, não pela mensagem. 60 req/min por chave — prefira webhook a polling de 1s.

2 · Cash-out cliente entrega cripto → chave PIX recebe reais

GET /catalog · GET /pix/keys/lookup

fast: USDT/USDC Polygon (segundos). convert: 1.300+ ativos (prazo da rede). Lookup devolve titular (DICT) para mostrar antes de aceitar. Solana fora do cash-out.

POST /cash-outs (cotar)

asset · network · amount ou brt_amount · pix_key (tipo deduzido; 11 dígitos exige pix_key_type) · refund_address (sempre!) · external_id. Devolve brt_amount liquido + expires_at (15 min na Polygon).

Limites da venda

R\$ 6 → R\$ 250.000 por transação · recusa devolve limits_min/max_amount já na cripto

R\$ 100.000/dia por recebedor (CPF/CNPJ ou chave) · limits_max_brt_cents = teto efetivo

sem teto diário por chave não vale para a venda · CPF do vendedor nunca é exigido

POST /cash-outs/{id}/accept

Trava a cotação e devolve deposit_address. Ponto sem volta. 503 Liquidante indisponível · espere retry_after_seconds e re-aceite (a mesma ordem).

Cliente envia a cripto

Valor exato da cotação, na rede cotada. Depósito diferente: paga-se o que chegou. Depois de expires_at: ordem revive por até 7 dias.

PIX pago

Polygon: liquidação nossa, PIX em segundos (medido 49-65 s do aceite ao pago). COMPLETED + comprovante. Webhook cashout.completed + PDF no grupo.

QUOTE_CREATED → accept → AWAITING_DEPOSIT → DEPOSIT_DETECTED → PROCESSING → COMPLETED · desvios: MANUAL_REVIEW → REFUNDING → REFUNDED (cripto volta a refund_address, refund_tx_hash) · EXPIRED

FAILED

Devolução: chave recusada pelo provedor, PIX revertido ou falha antes do envio → cripto volta para refund_address (sem ela, volta à origem on-chain — errado se veio de exchange). Automática até 5.000 USDT/ordem. Pagar QR/copia-e-cola (br_code): desligado em produção hoje (503 br_code_indisponivel) — a venda paga chaves PIX.

chamada à API ação do cliente final decisão / limite sucesso retido / revisão recusa / terminal infra / aberto

Números de produção lidos em GET /keys/me, /etc do gateway e no banco em 02/09/2026 — podem mudar; a chave lê os dela em GET /keys/me. Docs: docs.luniumpay.com · Status: docs.luniumpay.com/status · Contato: contato@luniumpay.com

2. Primeiros passos

2.1 Chave de teste (sandbox) — uma chamada

```
curl -s -X POST https://api.luniumpay.com/keys/sandbox \
  -H 'Content-Type: application/json' -d '{"name":"meu-projeto"}'
→ {"api_key":"lun_test_...", "sandbox":true, "como_usar":{"...}}
```

A chave `lun_test_...` roda na **mesma URL base**, com as mesmas formas, estados e contrato de erro da produção. Nada se move. O que o sandbox cobre hoje:

Superfície	Sandbox	Observação
Cash-out (POST /cash-outs → accept → status)	completo	ordem caminha pelos estados reais em ~15 s; gatilhos determinísticos (seção 5.3); E2E de teste verificável em GET /v1/verificar/{e2e}.
Cash-in (POST /cashin/charge)	não gera QR	responde 400 sandbox_sem_cashin: um QR de teste seria pagável de verdade. Teste o cash-in com a chave de produção e valores pequenos (R\$ 1 a R\$ 5) ; GET /cashin/limits, POST /cashin/preview, GET /cashin/catalog e GET /keys/me funcionam no sandbox.
Webhooks	sim	POST /webhooks/test dispara um webhook.test real contra o seu endpoint e devolve status e latência.
MCP (agentes de IA)	sim	chave de sandbox funciona no https://api.luniumpay.com/mcp sem nada especial.

2.2 Chave de produção — só o nome é obrigatório

```
curl -s -X POST https://api.luniumpay.com/keys \
  -H 'Content-Type: application/json' -d '{"name":"Minha Empresa"}'
→ {"api_key":"lun_...", "dashboard_url":"...", "grupos_obrigatorios":{"..."}, "limits":{"...}}
```

A chave nasce na hora com cash-in e cash-out habilitados, no tier **Início** (R\$ 5.000/dia em cash-in + payouts; sobe para R\$ 25.000 com R\$ 50 mil liquidados e R\$ 100.000 com R\$ 500 mil). A resposta traz a chave crua **uma única vez** (só o hash fica no banco); guarde em cofre. Limite de 5 chaves de produção por IP por dia.

Nunca cole a chave `lun_...` no grupo do Telegram.
Se precisar recuperar, crie outra (é grátis) e troque no seu código; a antiga pode ser desativada por nós.

2.3 Grupo de operação no Telegram — fortemente recomendado

1. Crie um grupo chamado **Lunium <> SeuProjeto** com **@LuniumNotifyBot** (o agente) e **@luniumB2B** (o humano do outro lado). A API funciona sem grupo, mas operar sem ele é operar às cegas: é por ali que você fica sabendo de uma liquidação que falhou às 3h da manhã.
2. No privado do bot, crie ou consulte a chave: `https://t.me/LuniumNotifyBot` → `/chave` mostra o código `lk...`.

3. Um **admin do grupo** manda `/vincular lk...`. Sem isso o grupo não recebe aviso nenhum.
4. A partir daí o bot avisa cada PIX que entra, cada cripto que sai, anexa o **PDF do comprovante** de cada venda paga e responde dúvidas quando mencionado. Comandos úteis: `/configurar` (checklist), `/carteira 0x...`, `/status`, `/extrato`, `/limites`.

2.4 Configurar a chave

```
curl -s -X PATCH https://api.luniumpay.com/keys/me -H "X-API-Key: $KEY" \
-H 'Content-Type: application/json' -d '{
  "settlement_address": "0xSUA_CARTEIRA_POLYGON",
  "webhook_url": "https://api.seudominio.com/lunium/webhook"
}'
→ devolve webhook_secret (whk_...) UMA vez; rotate_webhook_secret:true gera outro
```

- `settlement_address`: carteira padrão que recebe a cripto do cash-in quando a cobrança não traz `payout_address`.
- `webhook_url` + `webhook_secret`: assinatura HMAC dos eventos (seção 7).
- `GET /keys/me` devolve **todos os limites vigentes para a sua chave** (tier, por operação, escada do pagador, teto do recebedor). É o número que nunca envelhece: leia-o em vez de fixar as tabelas deste manual no código.

2.5 Links de teste e referência

O quê	Onde
Quickstart executável (do zero a uma ordem com E2E, em sandbox)	https://api.luniumpay.com/quickstart
Documentação para humanos (curl, Node, Python, redes x prazos, webhooks)	https://docs.luniumpay.com · passo a passo: /comecar
Contrato OpenAPI 3.1 (importável no Postman/Insomnia)	https://api.luniumpay.com/openapi.json
Resumo para agentes de IA / LLMs	https://api.luniumpay.com/llms.pt.txt (PT) · /llms.txt (EN)
Status em tempo real (chamadas reais a cada 5 min)	https://docs.luniumpay.com/status · máquina: /status.json
Servidor MCP (Streamable HTTP, 10 ferramentas)	https://api.luniumpay.com/mcp
Exemplos que rodam (Node/Python) e skills para agentes de código	github.com/guilhermezanqueta-collab/lunium-examples · .../lunium-agent-skills
App para testar o fluxo sem escrever código	https://app.luniumpay.com
Canal de parceiros (incidentes e mudanças de contrato antes de irem ao ar)	https://t.me/+Fer0ZMRRP_g5MDk5

3. Cash-in: cliente paga PIX → recebe cripto

Fluxo: consultar o limite do pagador → criar a cobrança (QR + copia-e-cola) → cliente paga → provedor confirma → Lunium envia a cripto ao `payout_address` → webhook `cashin.settled` com a hash da transação. O CPF/CNPJ do pagador é **obrigatório** (exigência do Banco Central, o provedor não gera QR sem ele) e é o que alimenta a escada de limites.

3.1 Limites: instantâneo x retenção de 24h — e como usar a retenção

Existe um degrau antifraude **por pagador** (CPF/CNPJ), independente da chave. Ele tem dois patamares e vale **já na primeira operação** do documento — não é preciso histórico nem esperar 24h:

Valor no dia daquele CPF/CNPJ	O que acontece	Como aparece na API
até R\$ 200,00	Instantâneo. PIX pago → cripto enviada em segundos.	QR sai com <code>held:false</code> ; <code>status pending</code> → <code>paid</code> ; <code>settlement_status: sent</code> .
de R\$ 200,00 a R\$ 6.000,00	Aceito com retenção de 24h no provedor. O PIX é pago normalmente; o valor fica retido e libera sozinho ao fim da janela. Não é falha e não é recusa.	QR sai com <code>held:true</code> , <code>held_hours:24</code> (antes de pagar). Pago, vira <code>delayed</code> com <code>delay_until</code> ; ao liberar vira <code>paid</code> e a cripto sai. Webhooks <code>cashin.delayed</code> → <code>cashin.paid</code> → <code>cashin.settled</code> .
acima de R\$ 6.000,00/dia	Recusado na criação.	<code>403 limite_do_pagador</code> , <code>acao: corrigir</code> , com <code>payer_limits</code> dizendo quanto cabe.

Por cobrança: mínimo **R\$ 1,00** (com a taxa padrão o mínimo efetivo é R\$ 1,02), máximo **R\$ 6.000,00**. A escada conta o que já foi pago **e o que está em voo** (cobranças pendentes ainda válidas); o relógio só começa no primeiro pagamento confirmado — criar QR não amadurece o documento.

Como usar a retenção na prática (o que o seu app precisa fazer)

1. Pergunte antes de cobrar:

`GET /cashin/limits?payer_tax=CPF`. Leia `instant_available_cents` (o que entra na hora) e `max_amount_cents` (o maior valor que o `POST` aceita agora, retenção incluída). Acima do instantâneo e até o máximo, a cobrança sai retida.

2. Mostre ao cliente antes de ele pagar:

a resposta do `POST /cashin/charge` traz `held` e `held_hours`. Se `held:true`, diga "o valor fica retido por 24h e libera sozinho".

3. Não trate `delayed` como falha e não cancele:

o PIX foi pago. Guarde `delay_until` e mostre "pago, libera em ...". A ordem vira `paid` sozinho.

4. Ouça o webhook

`cashin.delayed` (pagou, retido), depois `cashin.paid` e `cashin.settled` (cripto enviada, `settlement_tx_hash`). Se fizer polling, use `GET /cashin/{id}/status` a cada 30–60 s enquanto `delayed`.

5. A retenção é a janela para contestação (MED) antes de a cripto sair — é o que permite aceitar R\$ 6.000 de um documento que nunca pagou antes.

```
GET /cashin/limits?payer_tax=52998224725 (resposta real, pagador sem histórico)
{
  "stage": "first_transaction",
  "instant_available_cents": 20000,           ← entra na hora
  "max_amount_cents": 600000,               ← maior valor aceito agora (com retenção)
  "held_qr": { "max_amount_cents": 600000, "used_cents": 0, "available_cents": 600000, "hold_ho
urs": 24 },
  "ladder": { "first_transaction_cents": 20000, "first_24h_cents": 20000,
    "mature_day_cents": 600000, "held_max_cents": 600000, "hold_hours": 24 },
  "next_stage": { "stage": "first_24h", "limit_cents": 20000, "quando": "após o primeiro pagame
nto confirmado" },
  "explicacao": "Primeira operação deste CPF/CNPJ: até R$ 200.00 entra na hora; acima disso, at
é R$ 6000.00 o QR sai com retenção de 24h do provedor..."
}
```

Duas recusas do provedor que não são queda (não repita a chamada):

- 403 pagador_recusado_pelo_provedor, acao: parar, com provider_reference: o provedor de PIX recusou aquele CPF/CNPJ por decisão de compliance própria. Só outro documento ou revisão junto ao provedor citando a referência resolve.
- 403 limite_do_pagador com stage: provider_first_qr: o provedor limita o primeiro QR de um documento que ele ainda não identificou a R\$ 200; a partir do próximo QR o documento pode usar a escada inteira.
- 502 provedor_indisponivel, acao: repetir: aí sim é instabilidade; tente de novo em instantes.

3.2 Requisição, resposta e estados

```
POST /cashin/charge (X-API-Key: lun_...)
{
  "amount_cents": 15000, // R$ 150,00 – SEMPRE em centavos, inteiro
  "payer_tax_number": "12345678901", // CPF (11) ou CNPJ (14), só dígitos – obrigat
ório
  "payer_name": "Nome do pagador", // opcional
  "asset": "usdt", "chain": "polygon", // par do GET /cashin/catalog; omitido = usdt/
polygon
  "payout_address": "0x51e3d4...bc75e0", // carteira do SEU cliente; omitido = settleme
nt_address da chave
  "external_id": "dep-0001" // idempotente: repetir devolve a mesma cobrança (2
00)
}
→ 200
{
  "cashin_id": "ci_3ca9f23b...", "status": "pending",
  "amount_cents": 15000, "asset": "usdt", "chain": "polygon", "payout_address": "0x51e3...",
  "qr_copypaste": "00020126...", "qr_image_url": "https://...png",
  "expires_at": "2026-09-02T23:25:00.000Z", // QR vale 15 minutos
  "held": false, "held_hours": 0, // true/24 quando o valor exige retenção
  "delay_until": null, "liberar_em": null, "hold_hours": 0,
  "external_id": "dep-0001", "created_at": "..."
}
```

```
GET /cashin/{cashin_id}/status
{
  "cashin_id": "ci_...", "status": "paid", // pending | delayed | under_review | paid | expired | refunded | failed
  "settlement_status": "sent", // pending | sending | sent | incerto | failed
  "amount_cents": 15000, "depix_received_cents": 14625, // líquido após taxa
  "usdt_amount": "28.620000", "settlement_tx_hash": "0x...", "settlement_tx_url": "https://polygonscan.com/tx/0x...",
  "e2e": "E18236120202609021420...", "payer_name": "NOME DO PAGADOR", "payer_tax_number": "123.**.***-01",
  "held": false, "held_hours": 0, "delay_until": null, "liberar_em": null,
  "destino": "cripto", "customer_ref": null,
  "paid_at": "...", "settled_at": "...", "expires_at": "...", "created_at": "..."
}
```

Estado	Significado	Webhook
pending	QR emitido, ninguém pagou.	—
delayed	PIX pago , provedor segurando a liberação (retenção de 24h). <code>delay_until</code> diz quando libera. Não é falha.	<code>cashin.delayed</code>
under_review	PIX recebido, liquidação em trânsito.	—
paid	Creditado. A cripto sai em seguida (<code>settlement_status</code> pending → sending → sent).	<code>cashin.paid</code> → <code>cashin.settled</code> (com <code>settlement_tx_hash</code>) · <code>cashin.settlement_failed</code> se o envio falhar (retentado)
expired	Ninguém pagou até <code>expires_at</code> .	<code>cashin.expired</code>
refunded	Devolvido ao pagador (ex.: contestação executada pelo provedor).	<code>cashin.refunded</code>
failed	Provedor cancelou/errou antes do pagamento.	<code>cashin.failed</code>

O QR tem que ser pago no valor exato.

Cobrança de valor fixo não concilia com nenhuma diferença, nem para mais. Instrua o cliente a não editar o valor no app do banco.

3.3 Ativos, prazos e taxas

GET /cashin/catalog é a fonte da verdade e muda sozinho. Formato real: `{updatedAt, instant[], convert[], dropped[], totals}`.

Trilha	Ativos	Prazo medido (PIX pago → cripto na carteira, 30 dias)
instant (liquidação nossa)	USDT e USDC na Polygon	USDT: mediana 2 s (p90 2 s) · USDC: mediana 7 s (p90 8 s)
convert (via exchange)	1.300+ ativos e redes (BTC, ETH, SOL, TON, BNB...), cada um com <code>eta</code> , mínimo e taxa de saque	minutos; depende da rede — ex.: BTC on-chain ~37 min. Leia <code>etaSeconds</code> de cada item.
liquid	<code>chain: "liquid"</code> → o pagamento é entregue em DePix (stablecoin BRL na Liquid) direto na carteira Liquid informada, sem desconto de taxa da chave	segundos após a confirmação do provedor

Taxa padrão da chave: `admin_fee_bps: 250` (2,5%) + `admin_fee_fixed_cents: 100` (R\$ 1,00) por cobrança, descontados do valor pago antes da conversão. `POST /cashin/preview` mostra a conta sem criar nada:

```
POST /cashin/preview {"amount_cents":10000,"asset":"usdt","chain":"polygon"}
→ {"amount_cents":10000,"admin_fee_bps":250,"admin_fee_fixed_cents":100,"fee_total_cents":348,"net_cents":9652,"brl_per_usdt":"5.11","usdt_amount":"18.874787"}
```

Condições comerciais diferentes (taxa, tetos por pagador, carência) são configuradas por chave pela Lunium e aparecem em `GET /keys/me`.

4. Depósito em saldo (custódia) e saques

Para apps de carteira que precisam de uma aba "depositar e sacar" em reais. A chave precisa da custódia habilitada pela Lunium (`custodia_ativa: true` em `GET /saldo`) — pedimos no grupo. Nenhuma cripto sai: o PIX pago vira saldo em reais na sub-conta do seu cliente (`customer_ref`).

Passo	Chamada	Regra
Depositar	<code>POST /cashin/charge</code> com <code>"destino":"saldo"</code> e <code>"customer_ref":"id-do-seu-cliente"</code>	Depósito 0% . Mesma escada do pagador (200 / 6.000 por dia por CPF).
Retenção acima de R\$ 60	automática	Até R\$ 60,00: credita na hora, disponível em D+1 (<code>disponivel_em</code>). Acima de R\$ 60,00 por depósito: o QR sai retido 24h no provedor (<code>held:true</code>); pago, o saldo aparece congelado (<code>delayed</code>) e, quando o provedor libera, fica disponível na hora — a retenção conta como a carência. É a janela para contestação (MED) antes de o dinheiro sair.
Consultar	<code>GET /saldo?customer_ref=...</code> · <code>GET /saldo/extrato</code>	<code>disponivel_cents</code> , <code>bloqueado_cents</code> , <code>proximas_liberacoes</code> , <code>carencia_ate</code> , <code>taxas</code> (<code>deposito_fee_bps</code> , <code>saque_fee_bps</code> , <code>retencao_acima_cents</code>). Livro-razão append-only: <code>saldo</code> = soma.
Sacar por PIX	<code>POST /payouts</code> com o mesmo <code>customer_ref</code> , <code>amount_cents</code> , <code>pix_key</code> , <code>tax_number</code>	Debita valor + 1,8% (taxa da casa) + taxa do provedor (~R\$ 1,00) antes de o PIX sair; 402 <code>saldo_insuficiente</code> se faltar. A chave PIX tem que pertencer ao <code>tax_number</code> (regra do banco). PIX cai em até 24h, em geral segundos.
Sacar em cripto	<code>POST /saldo/sacar-cripto</code> { <code>amount_cents</code> , <code>chain</code> , <code>asset</code> , <code>payout_address</code> , <code>tax_number</code> , <code>customer_ref</code> }	Reais saem do saldo, cripto vai para a carteira do cliente pelo mesmo trilho do cash-in; a resposta é uma cobrança já <code>paid</code> — acompanhe <code>settlement_status</code> .
1º depósito	—	O primeiro depósito de uma sub-conta trava qualquer saque por 24h: 423 <code>carencia_primeiro_deposito</code> com <code>libera_em</code> .

Nada é adiantado e nenhuma taxa é absorvida: o que o provedor cobra é o que o saldo paga. As cobranças trazem `fonte` (`pix` | `saldo`) e `destino` (`cripto` | `saldo`) para um saque nunca ser confundido com uma compra.

5. Cash-out: cliente entrega cripto → chave PIX recebe reais

Três passos, nesta ordem. A cotação já é **líquida**: `brl_amount` é exatamente o que cai na chave PIX.

```

1) POST /cash-outs (cotar – não compromete nada)
{
  "asset": "usdt", "network": "polygon",
  "brl_amount": "250.00", // OU "amount": "50.77" (cripto) – exatamente um dos
dois
  "pix_key": "cliente@exemplo.com", // CPF · CNPJ · +55DDDNÚMERO · e-mail · UUID; 11 dígi
tos puros exige pix_key_type
  "refund_address": "0x51e3d4...bc75e0", // carteira de RETORNO do cliente, mesma rede – mande
sempre
  "external_id": "venda-001"
}
→ 201 (resposta real de sandbox)
{ "cashout_id": "sbx_141ce3e9...", "state": "QUOTE_CREATED", "amount": "50.774309", "brl_amount":
"250.00",
  "pix_key_type": "email", "deposit_address": null, "expires_at": "2026-09-03T02:33:31Z", ... }

2) POST /cash-outs/{cashout_id}/accept (trava a cotação – ponto sem volta)
→ { "state": "AWAITING_DEPOSIT", "deposit_address": "0x2952552e...", "deposit_tag": null, ... }
O cliente envia EXATAMENTE "amount" para deposit_address, na rede cotada.

3) GET /cash-outs/{cashout_id} (acompanhar; ou espere o webhook)
→ { "state": "COMPLETED", "pix_e2e": "E0000000020260903...", "pix_paid_at": "...",
  "receipt_url": "https://app.luniumpay.com/comprovante/...", "receipt_pdf_url": "...pdf",
  "verify_url": "https://api.luniumpay.com/v1/verificar/E...", "refund_tx_hash": null, "deposit
_from": "0x..." }

```

- **Antes de aceitar, mostre quem vai receber:** GET /pix/keys/lookup?key=... devolve nome, documento mascarado e instituição do titular (DICT). 5 consultas/min por conta; chave inexistente → 404 chave_nao_encontrada.
- **Nunca peça CPF de quem vende.** A chave PIX já identifica o recebedor; o campo não existe na venda.
- expires_at : 15 min na Polygon, até 300 min nas outras redes. Leia o campo; não fixe prazo.
- Se o aceite responder 503 liquidante_indisponivel (acao: esperar, retry_after_seconds: 120): o liquidante de PIX está fora naquele instante; espere e **re-aceite a mesma ordem**. Nenhum endereço é emitido enquanto isso, de propósito — cripto nunca entra numa esteira parada.
- Pagar QR/copia-e-cola (br_code) em vez de chave: **desligado em produção hoje** (503 br_code_indisponivel). A venda paga **chaves** PIX.

5.1 Limites, redes e prazos

Limite	Valor	Na API
Por transação	R\$ 6,00 a R\$ 250.000,00	recusa devolve limits.min_amount / max_amount já convertidos na cripto da ordem
Por recebedor por dia	R\$ 100.000,00 por CPF/CNPJ (ou por chave, quando ela não é documento). Renova à meia-noite de Brasília.	limits.max_brl_cents = teto efetivo (o menor dos dois); limits.max_brl_cents_por_transacao = nominal. Estouro: limite_diario_recebedor, acao: esperar.
Por chave por dia	não existe no cash-out	o tier (R\$ 5k/25k/100k) vale só para cash-in e payouts — daily_limit_applies_to em GET /keys/me
Mínimos por ativo	ex.: USDT 10, USDC 20 no trilho convert	GET /catalog → min / max por item

Rede	Prazo do depósito confirmado ao PIX pago
Polygon (USDT/USDC) — fast	liquidação nossa: segundos ; medido 49–65 s do aceite ao PIX pago (o depósito em si leva ~10–20 s para confirmar)
Demais redes — convert (1.300+ ativos via exchange)	o número de confirmações da rede: TON/Aptos ~1 min · Avalanche/NEAR ~2 min · BSC ~3 min · Base ~5 min · Ethereum ~20 min · Arbitrum ~21 min · Optimism ~1 h · Celo ~3 h. Nunca prometa "instantâneo" fora da Polygon.
Solana	temporariamente fora do cash-out — a cotação recusa (rede_indisponivel)

5.2 Estados, devoluções e comprovante

QUOTE_CREATED → accept → AWAITING_DEPOSIT → DEPOSIT_DETECTED → PROCESSING → COMPLETED · desvios: MANUAL_REVIEW → REFUNDING → REFUNDED · EXPIRED · FAILED .

Webhooks: cashout.awaiting_deposit, deposit_detected, processing, under_review, refunding, completed, refunded, expired, failed.

Situação	O que acontece
Provedor recusa a chave depois do depósito (inválida/inexistente)	MANUAL_REVIEW → cripto volta automaticamente para refund_address → REFUNDED com refund_tx_hash. Devolve-se o líquido que o provedor devolveu.
Envio falha antes de sair (liquidante fora, chave rejeitada na submissão)	retentado a cada 5 min por até 45 min; se não liquidar, o valor cheio (taxa inclusive) volta para refund_address. O status enquanto retenta é PROCESSING, não falha.
PIX revertido depois de pago (raro)	mesmo caminho: REFUNDED + hash.
Depósito diferente do cotado	paga-se o valor realmente recebido; token diferente fica em MANUAL_REVIEW.
Depósito depois de expires_at	a ordem revive quando o depósito aparece (até 7 dias); depois disso, MANUAL_REVIEW.
Sem refund_address	a devolução vai para a origem on-chain do depósito — errado quando veio de exchange (é a hot wallet dela). Por isso: mande sempre.
Valor da devolução	automática até 5.000 USDT por ordem; acima disso uma pessoa da operação executa e você recebe o mesmo REFUNDED + hash. Sempre na rede e no token do depósito; o gas é nosso.

Comprovante — três formas do mesmo fato, na mesma resposta e no mesmo webhook (nunca monte URL na mão): pix_e2e (identificador do Banco Central), receipt_url (página), receipt_pdf_url (PDF com nome, documento e instituição do recebedor, valor, data, E2E), verify_url. E **qualquer pessoa**, sem chave, confirma o pagamento:

```
GET https://api.luniumpay.com/v1/verificar/{e2e}
→ { "verificado": true, "pago": true, "valor_br": "250.00", "pago_em": "2026-09-03T02:18:47Z",
    "recebedor_iniciais": "T. S. T.", "instituicao": "...", "liquidado_por": "Lunium", "comprovan
te_url": "" }
(não expõe chave PIX, nome completo nem documento)
```

5.3 Sandbox: gatilhos determinísticos

As **duas primeiras casas decimais do** `amount` (em cripto) escolhem o desfecho — dá para afirmar em CI:

<code>amount</code> termina em	O que acontece	O que prova
<code>.01</code>	vai para <code>delayed</code> e conclui sozinha	seu código não chama retenção de falha
<code>.02</code>	falha	seu caminho de erro roda
<code>.03</code>	cotação expira em 5 s	você recota em vez de insistir
<code>.04</code>	recusa por limite com <code>limits</code> preenchido	você lê <code>limits.min_amount / max_amount</code>
<code>.05</code>	conclui em ~2 min	seu polling tem paciência
<code>.06</code>	recusa de compliance na liquidação → <code>FAILED</code> com <code>failure_reason: pagador_recusado_pelo_provedor</code>	você mostra o motivo e não repete
<code>.07</code>	1º aceite → <code>503 provedor_indisponivel</code> ; 2º aceite da mesma ordem funciona	seu retry re-aceita em vez de recotar
<code>.08</code>	recusa na criação por <code>limite_do_pagador</code>	você lê <code>payer_limits</code>

Reusar um `external_id` com destino diferente devolve **409**, igual à produção. Ordens de sandbox somem em 2 h. Nunca envie cripto a um `deposit_address` de sandbox: ele não tem dono.

6. Payouts (PIX direto, sem perna cripto)

`POST /payouts` envia um PIX para uma chave. Precisa de `payout_enabled` na chave (nasce desligado; pedimos no grupo). Com custódia, o valor sai do saldo da sub-conta (`customer_ref`); sem custódia, sai da tesouraria da Lunium conforme acordo comercial. Limites por operação: R\$ 1,00 a R\$ 6.000,00; conta na cota diária do tier.

```
POST /payouts { "amount_cents": 5000, "pix_key": "+5548999990000", "tax_number": "12345678901",
  "external_id": "saque-001", "customer_ref": "cliente-42" }
→ { "payout_id": "po_...", "status": "processing", // processing | sent | failed | refunded
  "amount_cents": 5000, "fee_cents": 100, "pix_key": "+5548*****0000", "sent_at": null, ... }
Webhooks: payout.sent · payout.failed · payout.refunded
```

7. Webhooks

Envelope único para todos os eventos. `data` é exatamente o objeto que o `GET` de status devolve para aquele recurso.

```
POST {webhook_url}
X-Lunium-Event-Id: evt_9f2c... // id do EVENTO – estável entre tentativas: deduplique por ele
X-Lunium-Delivery-Id: dlv_... // id da TENTATIVA – cite no suporte
X-Lunium-Signature: t=1756850000,v1=<hex HMAC-SHA256(secret, "t.corpo")>
{ "event": "cashin.settled", "event_id": "evt_9f2c...", "created_at": "2026-09-02T23:10:11.000Z",
  "data": { "cashin_id": "ci_...", "status": "paid", "settlement_status": "sent", "settlement_tx_hash": "0x...", ... } }
```

- **Verifique a assinatura v2** e recuse o que tiver mais de ~5 minutos. O header antigo `X-Webhook-Signature` (só o corpo) deixa de ser enviado em **09/11/2026**; chaves criadas depois de 08/08/2026 já não o recebem.
- Responda **2xx em até 15 s**. Sem 2xx: 12 tentativas com backoff (~7 h). A que esgota aparece como `failed` em `GET /webhooks/deliveries` e é recuperável com `POST /webhooks/deliveries/{event_id}/retry`. Nada se perde em silêncio: o bot também avisa no grupo.
- Trate evento desconhecido como no-op — a lista cresce sem aviso.
- `POST /webhooks/test` dispara `webhook.test` pelo mesmo transporte e devolve o que o SEU endpoint respondeu. Rode antes da primeira cobrança real.
- Prefira webhook a polling: 1 consulta/s consome sozinha os 60 req/min da chave (429 é proteção, não falha).

Cash-in	Cash-out	Payout / outros
cashin.held (QR emitido com retenção) · cashin.delayed · cashin.paid · cashin.settled · cashin.settlement_failed · cashin.expired · cashin.refunded · cashin.failed	cashout.awaiting_deposit · cashout.deposit_detected · cashout.processing · cashout.under_review · cashout.refunding · cashout.completed · cashout.refunded · cashout.expired · cashout.failed	payout.sent · payout.failed · payout.refunded · webhook.test

```
// Node – verificação da assinatura v2
const [t, v1] = req.get('X-Lunium-Signature').split(',').map(p => p.split('=')[1]);
const esperado = crypto.createHmac('sha256', SECRET).update(`${t}.${rawBody}`).digest('hex');
const ok = crypto.timingSafeEqual(Buffer.from(esperado), Buffer.from(v1)) && Math.abs(Date.now() / 1000 - t) < 300;
```

8. Erros: o campo que diz o que fazer

Todo erro traz `erro` (código estável), `acao`, `detail`, `request_id` e `agent_guidance`. **Ramifique pela `acao`**, não pela mensagem:

acao	Significado	Exemplos de erro
corrigir	o pedido está errado; repetir igual nunca passa	<code>campos_obrigatorios</code> · <code>amount_invalido</code> (decimal em string) · <code>pix_key_invalida</code> · <code>tipo_ambiguo</code> (11 dígitos: informe <code>pix_key_type</code>) · <code>valor_abaixo_do_minimo</code> / <code>valor_acima_do_maximo</code> (leia <code>limits</code>) · <code>limite_do_pagador</code> · <code>rede_indisponivel</code> · <code>nao_encontrado</code> · <code>campos_conflitantes</code>
repetir	falha transitória nossa	<code>provedor_indisponivel</code>
esperar	uma cota renova ou um serviço volta	<code>limite_diario</code> (tier, só cash-in/payouts) · <code>limite_diario_recebedor</code> (venda, R\$ 100 mil/dia) · <code>liquidante_indisponivel</code> (<code>retry_after_seconds</code>) · <code>br_code_indisponivel</code> · <code>cashout_indisponivel</code>
parar	não insista; mostre ao usuário / fale com a gente	<code>chave_ausente</code> · <code>chave_incorreta</code> (chave cortada ao copiar) · <code>chave_desconhecida</code> · <code>pagador_recusado_pelo_provedor</code> · <code>sandbox_sem_cashin</code>

Todo carimbo de tempo é ISO 8601 com fuso explícito (`2026-08-01T22:35:49-03:00` ou `...Z`). Nunca suponha local ou UTC.

9. Tabela-resumo de limites e taxas (produção, 02/09/2026)

Produto	Por operação	Por pessoa	Por chave	Taxa padrão	Tempo medido
Cash-in (PIX → cripto)	R\$ 1,00 a R\$ 6.000,00	pagador (CPF/CNPJ): até R\$ 200/dia na hora · R\$ 200 a R\$ 6.000/dia com retenção de 24h · já na 1ª operação	tier: R\$ 5.000 → 25.000 → 100.000 por dia (cash-in + payouts)	2,5% + R\$ 1,00	USDT 2 s · USDC 7 s após o PIX cair (Polygon)
Depósito em saldo	idem cash-in	idem; acima de R\$ 60 por depósito sai retido 24h (congelado) e libera disponível; até R\$ 60 disponível D+1	idem	0% no depósito · saque 1,8% + ~R\$ 1,00	saque PIX: segundos, até 24h
Cash-out (cripto → PIX)	R\$ 6,00 a R\$ 250.000,00	recebedor: R\$ 100.000/dia (CPF/CNPJ ou chave)	sem teto diário	já dentro da cotação (<code>brl_amount</code> é líquido)	Polygon: 49–65 s do aceite ao PIX pago; outras redes: prazo da rede
Payout (PIX direto)	R\$ 1,00 a R\$ 6.000,00	—	tier	provedor ~R\$ 1,00 (+1,8% se sair do saldo)	segundos, até 24h
Requisições	60 por minuto por chave (configurável) · QR de cash-in vale 15 min · cotação de cash-out vale 15 min na Polygon				

Todo número daqui pode mudar; `GET /keys/me` devolve os que valem para a sua chave. Condições comerciais específicas (taxa, tetos por pagador, carência) são acordadas com a Lunium e refletidas na chave.

10. Checklist de go-live

- Chave de produção criada (`POST /keys`) e guardada em cofre; nunca no chat.
- Grupo de operação com `@LuniumNotifyBot` + `@luniumB2B`, `/vincular` feito por um admin; `/configurar` mostra tudo .
- `settlement_address` (Polygon) e `webhook_url` configurados via `PATCH /keys/me`; `POST /webhooks/test` respondeu 2xx.
- Webhook verifica `X-Lunium-Signature v2`, deduplica por `event_id`, ignora eventos desconhecidos, responde em < 15 s.
- Cash-in: front consulta `GET /cashin/limits` antes de cobrar, mostra `held / held_hours`, trata `delayed` como "pago, libera em ..." e credita só em `cashin.settled` (ou `paid` com `settlement_status: sent`). Testado com PIX real de R\$ 1 a R\$ 5.
- Cash-out: fluxo cotar → mostrar titular (`/pix/keys/lookup`) e valor → aceitar → mostrar `deposit_address` e `amount` exatos; sempre manda `refund_address` e `external_id`; polling a cada 10–15 s ou webhook; trata `503 liquidante_indisponivel` re-aceitando a mesma ordem. Gatilhos `.01 – .08` passaram em sandbox.

7. Erros ramificados por `acao`; `limits` lidos da resposta em vez de fixados; `expires_at` lido em vez de prazo fixo.
8. Entrou no canal de parceiros (`https://t.me/+Fer0ZMRRP_g5MDk5`) e acompanha `docs.luniumpay.com/status`.

11. Links e canais

API	<code>https://api.luniumpay.com</code> · OpenAPI <code>/openapi.json</code> · <code>/quickstart</code> · <code>/llms.pt.txt</code> · MCP <code>/mcp</code> · saúde <code>/ping</code>
Docs	<code>https://docs.luniumpay.com</code> · <code>/comecar</code> · <code>/status</code>
Código	<code>github.com/guilhermezanqueta-collab/lunium-examples</code> · <code>github.com/guilhermezanqueta-collab/lunium-agent-skills</code>
Operação	grupo do seu projeto (Lunium <> SeuProjeto) com @LuniumNotifyBot · humano: @luniumB2B · canal de parceiros <code>t.me/+Fer0ZMRRP_g5MDk5</code> · e-mail <code>contato@luniumpay.com</code>

Lunium API · manual gerado em 02/09/2026 a partir do serviço em produção (OpenAPI 1.12.0, `GET /keys/me`, catálogos vivos e medições dos últimos 30 dias). Linguagem de limites e verificação proporcional ao volume conforme política da Lunium.