

LUNIUM PAY

API 接入手册

简体中文

Pix · BRL 资金托管 · 加密资产 · USDT/USDC 钱包

为您的团队提供完整的流程、接口示例、状态说明及运营指南。

欢迎中国企业及全球中文合作伙伴。

中文版本：2026年10月17日

在线手册：docs.luniumpay.com/zh/manual

接口字段、枚举、签名输入及示例响应中的原始值按 API 规范保留。费率、限额、可用性及最终金额以您的密钥和具体交易返回值为准。

目录

当前费率	4
1. 选择交易流程	4
2. 密钥与配置	5
测试环境	5
无需固定钱包的生产环境接入	5
恢复监控访问并管理凭证	6
现有监控与控制台演进	6
合作伙伴新控制台	6
3. Pix 收款至托管余额	7
创建收款单	7
确认入账	8
充值费用与资金可用性	8
付款人限额	9
4. 余额、子账户与流水	9
按交易查看流水：固定费用与合并费用	9
5. 以加密资产形式提取余额	10
6. 通过 Pix 提取余额	11
7. Pix 直接兑换加密资产	12
目录与报价	12
8. 外部加密资产兑换 Pix	12
状态与凭证	14
退款钱包：仅在卖出失败时要求	14
Pix 官方付款凭证	15
收款人查询与二维码付款	16
USDT/USDC 钱包：加密资产余额与 Pix 付款	16

9. 通过 Telegram 操作	18
Lunium Partners - 官方接入者社区	18
有人工跟进的接入群组	19
机器人绑定与命令	19
10. 可靠的 webhook	20
事件	21
合作伙伴实时监控	21
构建您自己的控制台事件流	22
Webhook + 状态查询 + 界面更新	22
各事件在界面中的含义	23
配置、测试与恢复通知	24
实时跟踪接入验收	24
11. 错误、幂等性与对账	24
操作指南：查询内容与客户提示	26
12. 测试环境与接入测试	26
13. 查询当前规则	27
14. 佣金、智能体与其他操作	28
配置合作伙伴佣金	28
区分费用、佣金、汇率与交付数量	28
佣金入账、流水与提现	29
15. 上线检查清单	32
16. 本次修订内容	32

合作伙伴文档

Pix、BRL 资金托管及加密资产，为每类交易选择适合的流程。

修订版本 2026-10-09.1 • API 1.44.7 • [OpenAPI](#) • [服务运行状态](#)

简体中文版：2026年10月17日 • 当前 API：1.44.7。历史修订日期按原记录保留。

当前费率

2026年10月15日费率政策：余额充值每笔 R\$ 1.00，无比例费用及佣金；直接购买为 3.3% + R\$ 1.00；BRL 余额兑换加密资产为 3.3%，不再收取固定费用；BRL 余额通过 Pix 提现为 1.5%。合作伙伴佣金另计于购买及转出费用。参考汇率采用 Binance 卖一价（ask），不另加溢价。适用的网络和处理成本将在预览中列明。

saque_cripto_conversao_fee_bps：兑换费率； saque_cripto_fee_bps=0；
saque_cripto_conversao_fee_fixed_cents=0。

01 / 收款Pix 充值至托管余额以 BRL 收款，之后再决定提现用途。02 / 提现余额兑换加密资产每次提现均可选择币种、网络及钱包。03 / 业务操作通过 Telegram 使用 API在项目群组中管理收款单、余额与提现。

本手册已对照服务接口及校验规则进行核查。示例仅供说明；限额、费用、可用性及处理期限须以当前密钥的 API 响应为准。文档修订不代表所有接口均已完成生产环境结算验证。

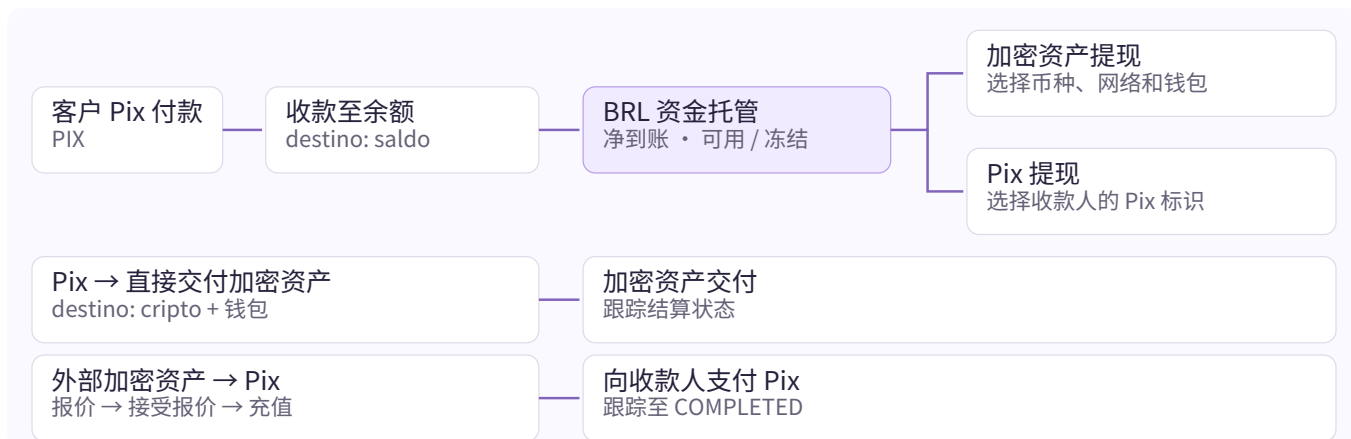
各币种及网络的最低与最高金额查询 cash-in、cash-out 及资金托管限额，使用主目录币种搜索表，并了解计算示例以及付款人 / 收款人规则。

1. 选择交易流程

您的目标	资金来源	操作	处理结果
收取 Pix，之后再决定提现用途	付款人的 Pix 付款	POST /cashin/charge 并传入 destino: "saldo"	余额币种为 BRL，记入主账户或子账户
将余额以加密资产形式提取	可用 BRL 余额	POST /saldo/sacar-cripto	兑换并交付所选币种及网络的资产
通过 Pix 提取余额	可用 BRL 余额	POST /payouts	向指定收款标识支付 Pix
收取 Pix 并直接交付加密资产	付款人的 Pix 付款	POST /cashin/charge 并传入 destino: "cripto"	交付至该收款单指定的钱包
卖出外部钱包中的加密资产	加密资产充值	POST /cash-outs → 接受报价 → 充值	向收款人支付 Pix
使用 USDT 余额支付 Pix	向钱包充值 USDT 或 USDC	POST /carteira-usdt/deposits → POST /carteira-usdt/payments	向指定收款标识或 Copia e Cola 支付 Pix

BRL 托管余额以巴西雷亚尔计价。合作伙伴收到 Pix 净额，提现时再选择币种及网络，无需登记固定钱包。一次托管充值不会同时生成 BTC、ETH 和 USDT 余额。

请注意各渠道的默认行为：调用 API 时，省略 destino 仍表示 crypto。若要收款至托管余额，请传入 destino: "saldo"。在 Telegram 中，/cobrar valor cpf 则默认使用托管余额。



2. 密钥与配置

欢迎加入 [Lunium Partners](#)，即官方接入者社区。强烈建议从接入初期就加入社区。我们在此分享运行通告、维护通知及 API 变更，合作伙伴也可互相交流。 [了解如何结合社区、专属群组及监控工具开展工作。](#)

基础地址：https://api.luniumpay.com。需要身份验证的接口须传入 X-API-Key；JSON 请求体还须设置 Content-Type: application/json。密钥可授权账户交易，请仅在您的服务器端使用。

测试环境

```
curl --fail-with-body https://api.luniumpay.com/keys/sandbox \
-H 'Content-Type: application/json' \
-d '{"email": "voce@empresa.com", "name": "Minha integração de teste"}'
```

请将 voce@empresa.com 替换为测试人员的邮箱：email 为必填项。约一分钟后，LuniumPay 支持团队会向该邮箱发送一次介绍邮件，包含 LuniumPay 简介、三步接入指南、手册及 PDF、流程图，以及 VendaCripto（无需接入 API，即可使用 Pix 买卖加密资产的现成应用）。每个邮箱仅发送一次，并附退订链接。若未提供有效的 email，响应将为 400，错误为 erro: "email_obrigatorio"。整个流程仍只需一次调用，无需审批或 API 密钥；name 为可选项。

保存 api_key 作为 LUNIUM_API_KEY 保存在服务器环境中。新建测试密钥的前缀为 lun_test_，生产密钥前缀为 lun_live_。旧版 lun_ 密钥仍然有效。

测试环境可模拟 Pix 入金、BRL 托管入账、加密资产交付与提现、Pix 对外付款、内部转账及 cash-out。二维码不可用于真实付款，请通过 POST /sandbox/cashin/{cashin_id}/pay 确认。不会调用任何银行、交易所或区块链。有关场景、数据持久化及与生产环境的差异，请参阅 [测试环境指南](#) 及 GET /sandbox。使用托管余额无需固定钱包。

无需固定钱包的生产环境接入

```
curl --fail-with-body https://api.luniumpay.com/keys \
-H 'Content-Type: application/json' \
-d '{"name": "Meu projeto", "email": "operacao@example.com"}'
```

请换成您自己的姓名及邮箱。响应会返回 api_key、key、limits、monitor_url 及 Telegram 使用说明。完整密钥仅在创建时显示一次。monitor_url 是项目监控面板；该链接也属于项目私密信息，请妥善保管。

新建生产密钥默认启用托管余额及 Pix 对外付款。实际应以 您的 密钥状态为准，请查询 `payout.enabled` → GET `/keys/me` 及 `custodia_ativa` → GET `/saldo`。

```
curl --fail-with-body https://api.luniumpay.com/keys/me \
-H "X-API-Key: $LUNIUM_API_KEY"

curl --fail-with-body https://api.luniumpay.com/saldo \
-H "X-API-Key: $LUNIUM_API_KEY"
```

通过以下接口配置 HTTPS webhook 接收端：PATCH `/keys/me` 并传入 `webhook_url`；请保存响应返回的 `webhook_secret`。
`settlement_address` 用于可选的 Polygon 直接交付，不是托管余额的必要条件，也不决定之后的提现地址。

请重复使用同一测试密钥；创建密钥受 IP 限流约束。遇到 429 时，须按指定时间等待。不要为每笔交易新建密钥。

恢复监控访问并管理凭证

私密监控 以只读方式展示项目情况，而 API 密钥 可授权执行交易。监控链接、API 密钥及 webhook 签名密钥是不同的凭证；恢复其中一项，不会恢复其他项。

若丢失监控链接，可申请将其发送至生产密钥登记的邮箱：

```
curl --fail-with-body https://api.luniumpay.com/keys/painel-link \
-H 'Content-Type: application/json' \
-d '{"email":"operacao@example.com"}'
```

该接口无需 API 密钥。出于账户隐私保护，响应 200 `{"ok":true,...}` 为统一格式，不会确认账户是否存在或邮件是否送达。请检查收件箱及垃圾邮件。遇到 429 时须等待，不要自动连续提交。如登记邮箱有误或无法访问，请联系项目的人工支持人员核验并恢复访问。

若仍持有密钥，GET `/keys/me` 会返回 `monitor_url`。若链接泄露，请使用 PATCH `/keys/me` 并传入 `{"rotate_monitor_token":true}` 使旧链接失效，并仅向授权团队分发新链接。旧链接立即失效，但 API 密钥及 webhook 签名密钥不会改变。

完整 API 密钥仅在创建时显示一次，此流程不会重新显示或恢复密钥。若密钥丢失或泄露，请联系支持团队处理替换 / 撤销及账户连续性。不要假设新建密钥后，余额、收款单或绑定关系会自动转移。更换 webhook 签名密钥请参阅 [专门流程](#)。

现有监控与控制台演进

本次修订所列的跟踪方式包括私密监控、事件流及接口契约中的公开接口。额外控制台功能及 webhook 多接收端能力，须在正式发布并经生产验证后再接入。请按文档在以下接口配置接收端：PATCH `/keys/me`；不要根据开发预告推测存在公开的多接收端 API，或登录 / 2FA 接口。

Fable 及其他 AI 接入工具请核对 [经核查的版本及接口清单](#)，并结合 [OpenAPI](#) 使用。如有差异，请记录涉及的资料来源及操作，暂停该能力的实现直至澄清，不要自行编造接口补齐流程。

合作伙伴新控制台

合作伙伴控制台位于 parceiros.luniumpay.com，使用同一邮箱及密码登录；也可使用邮件验证码或 Google 账户。尚未开通访问权限的用户可使用 创建账户。控制台供团队操作及跟踪项目，系统接入仍通过项目密钥调用 API。控制台页面使用登录用户的会话，不属于 API 公开接口契约。

- 首页：按时间范围查看业务概览（Pix 收款、加密资产卖出、Pix 付款、加密资产提现及佣金）、收款单 / 卖出订单 / 付款状态，以及日历。
- 凭证：查看卖出订单、Pix 付款及收款单的 PDF 凭证。

- 使用加密资产付款：可搜索并选择稳定币（所有支持网络上的 USDT、USDC，以及 DePix）和 30 种主要加密资产，并显示图标。仅在卖出交易失败时才要求填写退款钱包（退款钱包）。
- USDT/USDC 钱包：充值时展示地址二维码及 1 小时倒计时，可使用余额支付 Pix。详情参阅 USDT/USDC 钱包。
- 佣金：提现最低 US\$ 10。
- 通知：通过通知铃铛查看项目动态，并通过 Telegram 及邮件接收各笔交易通知。
- 团队：角色包括所有者、财务、操作员、开发者及只读成员；双重验证位于 我的访问权限。
- 我的访问权限：点击 退出其他会话 可退出其他设备上的控制台会话，并保留当前会话。无密码账户（通过验证码或 Google 登录）可用验证器代码或恢复代码关闭双重验证。更改密码时，当前密码每小时最多允许 10 次错误尝试；密码被更改、创建或重置时，系统均会发送邮件通知。
- Webhook：立即重发 可在控制台中重新投递该事件。
- 可读性：提高对比度、增大字号，统一各页面的 Pix 拼写，保持页面标题与菜单一致，并按所选界面语言提供清晰的错误提示。

新账户激活。若邮箱尚未注册账户，通过机器人或邮件链接、或项目密钥激活控制台时，该邮箱会收到 6 位数验证码。验证码有效期为 30 分钟，最多允许 5 次尝试；重新发送验证码可在 1 分钟后生成新验证码，最多 5 次，仅最新验证码有效。已有账户的邮箱仍使用账户密码确认。

通过验证码或 Google 登录的受邀用户。无密码账户可这样接受邀请：先用受邀邮箱登录控制台，再次打开邀请链接；保持登录后，即可一键接受。

控制台安全规则：

- 仅所有者 可查看 Telegram 绑定链接及代码、断开会话或生成新代码。当成员失去访问权限、不再是所有者，或会话被断开时，绑定代码会自动更新。
- 某位成员不再是所有者时，其他所有者会收到邮件，提醒其前往 通知检查已连接的 Telegram 会话；这些会话会保持连接，直至所有者主动断开。
- 已连接的 Telegram 会话也可通过机器人查询余额及执行资金操作：私聊由绑定者操作，群组由管理员操作。仅连接可信的会话。
- 在控制台中，仅所有者 可修改结算钱包（settlement_address）及子佣金（partner_fee_bps、partner_cashout_fee_bps、partner_saque_fee_bps）。开发者 仍可调整 webhook、名称及监控。
- 会话有效期为 12 小时，使用控制台时会续期，但自登录起最长不超过 7 天。会话 cookie 受到保护，页面启用了内容安全策略。

3. Pix 收款至托管余额

创建收款单

请先查询 GET /cashin/limits?payer_tax=CPF_OU_CNPJ，并使用项目密钥验证身份。证件必须属于 实际支付 Pix 的付款人；customer_ref 则用于标识您系统中的子账户，两者含义不同。

示例金额为 R\$ 50.00。请替换证件及引用标识，并选择该付款人额度内的金额。在生产环境调用会创建真实收款单。

```
curl --fail-with-body https://api.luniumpay.com/cashin/charge \
-H "X-API-Key: $LUNIUM_API_KEY" \
```

```
-H 'Content-Type: application/json' \
-d '{
  "amount_cents":5000,
  "payer_tax_number":"00000000191",
  "destino":"saldo",
  "customer_ref":"cliente-42",
  "external_id":"deposito-cliente42-001"
}'
```

此流程不要传入 `payout_address`。若未提供 `customer_ref`，入账将归属于该密钥自身的余额，称为主账户；提供引用标识后，则仅归属于对应子账户。

保存 `cashin_id` 及 `external_id`。请向客户展示返回的 `qr_copypaste`、`qr_image_url`（如有）、金额及 `expires_at`。生成二维码仅表示已创建收款单，不代表资金已经到账。

确认入账

使用 `GET /cashin/{cashin_id}/status` 或 `webhook` 事件，并分别跟踪 两种状态：

字段	响应所表示的内容	处理方式
<code>status</code>	Pix 付款状态	<code>pending</code> 、 <code>delayed</code> 、 <code>under_review</code> 、 <code>paid</code> 、 <code>expired</code> 、 <code>refunded</code> 或 <code>failed</code>
<code>settlement_status</code>	入账 / 交付状态	<code>pending</code> 、 <code>sending</code> 、 <code>entregando</code> 、 <code>sent</code> 、 <code>incerto</code> 或 <code>failed</code>

仅在以下状态下视为入账 / 交付完成：`cashin.settled`，或通过查询 `status: "paid"` 且 `settlement_status: "sent"` 确认。使用托管余额时，还须通过 `GET /saldo?customer_ref=cliente-42` 及流水核对金额与可用性。已登记的入账仍可能被冻结，不能将 `total_cents` 当作 `disponivel_cents`。

`depix_received_cents` 表示从结算服务商收到、尚未扣除密钥充值费用的金额。该字段沿用旧名称，不代表客户最终净入账金额。对账请使用余额流水。

充值费用与资金可用性

`GET /saldo` 会列明 `deposito_fee_bps`、`deposito_fee_fixed_cents`、`carencia_horas`、`carencia_ate`、`retencao_acima_cents` 及 `proximas_liberacoes`。100 bps = 1%。新充值适用 `deposito_fee_bps=0` 及 `deposito_fee_fixed_cents=100`，不收取合作伙伴佣金。

以下三个概念相互独立：

规则	查看位置	含义
服务商资金冻结	<code>held</code> 、 <code>held_hours</code> 、 <code>delay_until</code> ，状态为 <code>delayed</code>	Pix 已支付，等待服务商释放资金
收款单本地冻结	<code>hold_hours</code> （创建时）、 <code>liberar_em</code> （状态查询），事件为 <code>cashin.held</code>	Pix 可能已处于 <code>paid</code> 状态，但入账 / 交付仍需等待
余额等待期	<code>carencia_horas</code> 、 <code>carencia_ate</code> ，被冻结的入账列于 <code>/saldo</code>	密钥 / 子账户的资金可用及提现规则

`held_hours` 及 `hold_hours` 并非同义字段。即使收款单显示 `held: false`，仍可能处于本地冻结期。不能仅凭该布尔值承诺资金立即可用。

并非所有合作伙伴都适用统一的 24 小时等待期。即时释放余额的密钥有独立规则，但仍受付款人冻结、收款单冻结及其他限制影响。须结合创建响应、交易状态及余额判断；因等待期拒绝提现时会返回 423 及资金释放时间。

付款人限额

GET /cashin/limits 会列明即时到账额度、是否允许冻结收款、已用额度及该证件下一次释放时间。密钥条件可能改变分级限额。查询结果仅供参考，创建时仍会重新校验交易，包括即时释放的托管收款。

allow_hold: false 会拒绝需要按付款人限额规则冻结的收款单。它不会关闭其他合同规则或之后发生的资金冻结。省略此字段即允许适用的冻结。付款前须向客户展示相关条件。

实际最低金额还取决于费用及服务商。不要将 R\$ 1.00 用作所有接口通用的测试金额，也不要将全局最低金额误认为某个接口实际接受的最低金额。

4. 余额、子账户与流水

查询	查询范围与分页
GET /saldo	主账户余额，不包含客户子账户
GET /saldo?customer_ref=cliente-42	仅指定子账户
GET /saldo/consolidado	主账户与客户子账户，包含总托管金额
GET /saldo/clientes?limit=50&com_saldo=1	客户列表，继续分页使用 cursor=next_cursor
GET /saldo/extrato?customer_ref=cliente-42&limit=50	流水列表，继续分页使用 before=proxima_pagina

按交易查看流水：固定费用与合并费用

在 Pix 对外付款响应及 data 类型 webhook 的 payout.* 中，使用 fee_service_total_cents 作为客户服务费，其中已包含 Lunium 及合作伙伴费用。fee_provider_cents 是向客户收取的 Pix 服务商费用。fee_cents = fee_service_total_cents + fee_provider_cents；完整扣款金额为 amount_cents + fee_cents。不要再次加上 fee_partner_cents：该字段与 fee_service_cents 均为已包含的会计组成项。金额来自已保存的交易，并保留分位舍入结果；null 表示旧记录中的组成项未知。新交易显示的费率来自 saque_fee_total_bps；佣金调整仅影响新交易。示例：Pix 金额 R\$ 50.00，Lunium 1.5% + 合作伙伴 1%，Pix 固定费用 R\$ 1.00：服务费 R\$ 1.25，总费用 R\$ 2.25，完整扣款 R\$ 52.25。合作伙伴应用须读取这些字段，才能更新界面与通知。

展示用户历史记录时，请查询 GET /saldo/extrato?customer_ref=cliente-42&visao=operacoes。响应每笔交易仅一行。在详情中，taxas 仅有一个 servico 类别，说明为“费用”，合并 Pix 固定费用及全部其他费用；不向客户单独展示合作伙伴佣金行、标签或比例。测试环境同样适用。

新提现确认前，当前适用费用可通过 GET /saldo 查询，其中提供 saque_fee_total_bps（平台费率 + 合作伙伴附加费率；250 bps = 2.5%）。固定费用以分为单位，位于 saque_taxa_estimada_cents，条件为 saque_taxa_provedor_tipo=fixo。界面仅显示固定费用 + 总比例费用。各费用组成字段及默认会计流水供后端和对账使用，不应直接展示给客户。请使用报价或已保存的金额，因为各组成项在内部按分舍入；不要用当前费率重算历史费用。

valor_cents 是对余额产生的带符号总影响：负数为扣款，正数为入账。valor_operacao_cents 是带符号的本金；taxas_cents 是已计入费用的绝对值总额。每项费用均提供 tipo、descricao、valor_cents 的绝对值及

efeito_saldo_cents 的带符号值。恒等式为 $\text{valor_cents} = \text{valor_operacao_cents} + \text{soma}(\text{taxas[]}.\text{efeito_saldo_cents})$ 。不要再次扣除费用。

示例：Pix 金额 R\$ 2.00，固定费用 R\$ 1.00，总比例费用 R\$ 0.05 → 一笔 R\$ 3.05 的支出，本金为 -200，费用为 105，总额为 -305。详情仅显示 费用：R\$ 1.05。每笔交易只生成一条通知，以 `origem_id` 及 `tipo` 识别，不按费用组成项生成通知。不要将费用显示为独立卡片、收款通知或绿色正数。退款另列一行；借记形式的退款保留负号。缺少本金信息的费用调整仍须展示，以免遗漏资金变动。

`status/status_label` 反映查询时已持久化保存的交易状态；`settlement_status` 表示加密资产交付状态。未知状态显示为 `null`，不应默认已完成。页面打开期间持续刷新查询，并通过状态 `webhook` 触发重新读取；通知按交易及状态变化去重，不按费用流水去重。分组视图不改变会计流水，也不会自动更新第三方应用。

分页先分组，再应用 `limit`（1-200）。继续请求时传回 `before=proxima_pagina`，直至 `has_more=true`；`proxima_pagina=null` 表示结束。游标为十进制字符串，以首笔流水为锚点；不要转换为 `Number`。所有分页请求须使用同一 `customer_ref`。`movimentos_ids` 引用原始流水以便对账，不要将这些 ID 渲染为新动态。刷新第一页以获取后续到账的费用及状态。该视图的游标或条数错误返回 HTTP 400。

充值、查询及提现须使用完全相同的 `customer_ref`，最长 80 个字符。该引用用于隔离余额，但 API 密钥仍可操作其所有子账户；您的系统还须自行实现用户权限控制。

内部转账不生成 Pix，使用 `POST /saldo/transferir`：

```
{
  "amount_cents":1000,
  "from_customer_ref":"cliente-42",
  "to_customer_ref":"casa",
  "external_id":"transferencia-interna-001",
  "descricao":"Transferência autorizada pelo cliente"
}
```

`casa` 表示不属于任何子账户的主账户余额。仅可转移可用余额，来源子账户的等待期仍然有效。该接口余额不足时返回 409 `saldo_insuficiente`；提现接口通常返回 402。

5. 以加密资产形式提取余额

1. 查询 `GET /saldo`，并须使用正确的账户范围。
2. 查询 `GET /cashin/catalog`：这是 交付目录，也用于托管余额提现。
3. 请提供以下字段组成的有效组合：`instant` 及 `convert` 并传入 `entregavel: true`。余额提现须排除 Liquid/DePix，此流程不支持这两项。
4. 选择币种、网络、收款钱包，以及需要时的 `memo/tag`。每次提现可使用不同钱包。
5. 发送 `POST /saldo/sacar-cripto`，并使用以下接口跟踪返回的 `cashin_id`：`GET /cashin/{cashin_id}/status`。

以下请求体仅为示例，发送前请替换钱包及证件：

```
{
  "amount_cents":10000,
  "asset":"usdt",
  "chain":"polygon",
  "payout_address":"0x1111111111111111111111111111111111111111",
  "tax_number":"00000000191",
  "customer_ref":"cliente-42",
  "external_id":"saque-cripto-cliente42-001"
}
```

amount_cents 表示待兑换的 BRL 金额，而非加密资产数量。证件字段 tax_number 属于提现客户。需要 memo 的网络须根据收款钱包要求传入 payout_tag，并遵守目录中的 memoRequired。也支持以下别名：memo、tag 及 destination_tag。

响应会返回 fonte: "saldo"、destino: "cripto"、fee_cents 及 debited_cents。扣款包含本金及平台提现费用；最终加密资产数量还考虑兑换与交付成本，不能仅用扣款金额除以市场汇率计算。

GET /saldo 会列明 saque_cripto_fee_bps。POST /cashin/preview 可辅助查询兑换情况及该路由的最低金额，但不是提现完整扣款的预览。Telegram 使用的提现预览为内部功能，不存在公开的 /saldo/sacar-cripto/preview。

可选字段 max_debited_cents 可限制 BRL 扣款总额。若总额超过上限，API 将返回 409 taxa_alterada。该字段不会锁定汇率，也不保证收到最低数量的代币。

不要笼统宣称“支持所有加密资产”。实际选项为当前目录中可交付的币种 / 网络组合，仍受最低金额、维护及地址校验约束。stale: true 表示备用目录，确认前须重新校验。

6. 通过 Pix 提取余额

POST /payouts 会在发送 Pix 前预留并扣除主账户或子账户的可用余额。须具备 payout.enabled、已启用托管功能，以及足够覆盖本金和费用的余额。

```
{
  "amount_cents": 5000,
  "pix_key": "recebedor@example.com",
  "pix_key_type": "email",
  "tax_number": "00000000191",
  "customer_ref": "cliente-42",
  "external_id": "saque-pix-cliente42-001"
}
```

请将 Pix 收款标识及证件替换为实际收款人的资料。tax_number 须包含有效的 CPF 或 CNPJ，适用于邮箱、手机号及随机类型的 Pix 收款标识。若收款标识本身为有效 CPF/CNPJ，API 将直接提取证件；如另行提供，必须一致。

beneficiary_name 仍为可选项，且不能证明账户归属。11 位数字的收款标识须指定 pix_key_type: "cpf" 或 "phone"；手机号须包含国际区号，例如 +55。

此流程仅使用通过 Pix 收到的 BRL 余额。funding_source 为可选项，仅接受 pix_brl。每笔 Pix 最高为 600000 分 (R\$ 6,000.00)，还须遵守账户较低的限额，另计适用费用。

遇到 409 external_id_divergente 或 external_id_nao_verificavel 时，请查询 payout_id/recovery_endpoint 所指向的订单。请保留原标识，不要新建标识绕过冲突。遇到 503 idempotencia_indisponivel 时，应等待并查询同一 external_id，因为之前的订单可能已存在。

USDT 钱包与 BRL 余额相互独立，接收外部 USDT 充值，不使用 /payouts。对应钱包付款流程使用 funding_source=crypto_usdt，不要求收款人证件；限额以报价及 API 条件为准，不适用 BRL 余额的 R\$ 6,000 上限。测试密钥使用独立模拟钱包，不涉及真实资金。

费用构成：amount_cents + fee_cents，其中 fee_cents = fee_service_total_cents + fee_provider_cents。通过 GET /saldo 读取 saque_fee_total_bps、saque_taxa_provedor_tipo、saque_taxa_provedor_piso_cents、saque_taxa_estimada_cents 和 saque_taxa_provedor_formula。请使用 API 当前费用，不要在代码中写死历史费率或固定费用。付款记录会保存实际费用。

查询 GET /payouts/{payout_id}。processing 不代表付款完成；sent 表示发送已确认。failed 及 refunded 须结合状态及流水对账。超时后继续查询原交易，不要假定已退款，也不要换用新标识再次付款。

确认后，e2e、paid_at、receiver_name、receiver_tax_number 及 verify_url 等字段可能提供 Pix 付款证明。尚未取得的字段可能为 null。

7. Pix 直接兑换加密资产

如需每笔 Pix 收款后直接交付加密资产，请使用 destino: "cripto"、交付目录中的有效组合，以及 payout_address。若请求体未指定地址，服务可能使用密钥配置的 Polygon 钱包；不要把它当作适用于其他网络的通用钱包。

```
{
  "amount_cents":10000,
  "payer_tax_number":"00000000191",
  "destino":"cripto",
  "asset":"usdt",
  "chain":"polygon",
  "payout_address":"0x1111111111111111111111111111111111111111",
  "external_id":"compra-cripto-001",
  "allow_hold":false
}
```

目录与报价

GET /cashin/catalog 列出 API 可交付的资产。使用目录中的 asset 及 network；在 cash-in 及提现请求体中，网络字段名为 chain。instant 可能为空。出现在 convert 中，并不免除 entregavel、最低金额或 memo 要求。当前各币种及网络的最低金额表见 [最低金额与网络](#)。

```
curl --fail-with-body https://api.luniumpay.com/cashin/preview \
-H "X-API-Key: $LUNIUM_API_KEY" \
-H 'Content-Type: application/json' \
-d '{"amount_cents":10000,"asset":"usdt","chain":"polygon"}
```

预览不会创建收款单，也不会锁定付款时的汇率。请读取响应中的 min_brl_cents、费用及预估数量。客户钱包实际预计收到的数量列于 asset_amount，适用于所有路由及测试环境，已扣除网络提现费用（withdraw_fee，以资产单位表示）。该费用按每次发送固定收取，因此小额交易受影响更大，例如 Bitcoin 网络上的 BTC；若要以 BRL 展示，请使用 $withdraw_fee \times brl_per_asset$ 。需要兑换的路由中，usdt_amount 是用于购买的 USDT，不是客户最终收到的数量；amount 是以下字段的旧名称：asset_amount。eta/etaSeconds 是路由时效预估，不是合同约定的最长处理时间。

实际收到的金额及结算时适用汇率决定最终交付数量。若存在以下字段，请展示：settlement_tx_hash 及 settlement_tx_url；不要将所有链接都固定指向 Polygon。内部交付标识不一定有公开区块浏览器链接。

所填 CPF/CNPJ 必须属于付款人。资料不符可能导致退款（PAYER_MISMATCH），若退款未完成则可能进入审核。不要仅凭客户发送的凭证或 under_review。

8. 外部加密资产兑换 Pix

该流程接收外部钱包的加密资产。若要使用托管余额支付 Pix，请使用 /payouts。

加密资产兑换 Pix：交易与退款流程

客户发送加密资产，收款人接收 Pix。可通过 API 或合作伙伴控制台操作。

A · 兑换与付款

1. 获取报价

QUOTE_CREATED

POST /cash-outs

填写加密资产数量，或 BRL 金额与 Pix 收款标识。



2. 接受报价

AWAITING_DEPOSIT

POST /cash-outs/{cashout_id}/accept

报价锁定，并生成充值地址。



3. 按要求充值

EXPIRED（未按时发送）

在有效期内，按指定网络、地址和准确金额发送。地址二维码不包含金额。



4. 等待网络确认

cashout.deposit_detected · cashout.processing

网络确认后才执行 Pix。各路由的具体阶段可能不同；失败时进入右侧退款流程。



5. Pix 已付款

COMPLETED

cashout.completed · cashout.receipt_ready

提供交易凭证及 E2E: receipt_url、receipt_pdf_url、pix_e2e。

B · 仅在兑换失败时退款

1. 需要客户退款钱包

AWAITING_REFUND_ADDRESS

cashout.awaiting_refund_address

创建订单时可不提供退款钱包。如已提供，可直接进入退款阶段。



2. 填写客户退款地址

POST /cash-outs/{cashout_id}/refund-address

必须使用退款所需网络。不得自动退回原充值来源地址。



3. 正在退款

REFUNDING_CRYPTO · DePix: REFUNDING

cashout.refunding

跟踪同一笔订单的退款状态。



4. 退款已完成

REFUNDED

cashout.refunded

链上退款凭证: refund_tx_hash。



5. 如资金已发送给服务商

MANUAL_REVIEW

需等服务商退回资金并确认；退款金额不超过实际退回金额。

币种、网络及最低金额：GET /catalog。仅查询预览、不创建订单：POST /cash-outs/preview。

支持范围以实时资产目录为准。完整说明：docs.luniumpay.com/zh/manual#cashout

1. 查询 GET /catalog，即 加密资产入金目录；它与 /cashin/catalog。
2. 可选用 POST /cash-outs/preview 进行不创建订单的预估。
3. 通过以下接口创建报价：POST /cash-outs。
4. 向客户展示数量、网络、BRL 净额、收款人及有效期。通过以下接口接受报价：POST /cash-outs/{cashout_id}/accept。
5. 之后才向客户展示 deposit_address，该 deposit_tag（如有）及返回的精确金额。
6. 持续查询 GET /cash-outs/{cashout_id} 或跟踪 webhook，直至最终状态。

以下示例使用 测试环境，以及 LUNIUM_API_KEY 测试数据：

```
curl --fail-with-body https://api.luniumpay.com/cash-outs \
-H "X-API-Key: $LUNIUM_API_KEY" \
-H 'Content-Type: application/json' \
-d '{
```

```
{
  "asset": "USDT",
  "network": "polygon",
  "amount": "10.00",
  "pix_key": "teste@example.com",
  "pix_key_type": "email",
  "external_id": "sandbox-cashout-001"
}
```

发送 amount（加密资产数量）或 brl_amount（期望的 BRL 金额），须为以点作小数分隔符的字符串。两者不能同时传入。创建时可能返回 200（生产环境）或 201（测试环境）；请按接口契约接受相应的 2xx 响应。cashout_id 是接口返回的标识，不是由接入方自行生成的 ID。

refund_address 在创建时为可选项，DePix 卖出也一样：仅在卖出失败时才要求退款钱包（详见 [退款钱包](#)）。若已持有客户钱包，也可提前传入，但必须符合相应网络格式，并由退款收款人控制。交易所提币的来源可能是交易所共用钱包，不要假定它可用作退款钱包。

状态与凭证

阶段	可能出现的状态
报价与充值	QUOTE_CREATED、AWAITING_DEPOSIT、DEPOSIT_DETECTED
网络确认与兑换	DEPOSIT_CONFIRMED、SELLING、SOLD、FORWARDING、PAYING_OUT
Pix 付款完成	COMPLETED
异常情况	MANUAL_REVIEW、AWAITING_REFUND_ADDRESS、REFUNDING_CRYPTO、REFUNDING（DePix 卖出）、REFUNDED、EXPIRED、FAILED

并非所有接口都会经历全部阶段。timeline、confirmations 及 confirmations_required 等字段（如有）可辅助跟踪。充值确认不代表 Pix 已付款完成。

遇到 COMPLETED，请核对 pix_e2e、pix_paid_at、receipt_url、receipt_pdf_url 及 verify_url（如有），并使用接口返回的链接。公开查询接口 GET /v1/verificar/{e2e} 用于核验已知的 Pix 付款，不是对所有加密资产交易的通用证明。

发生退款时，请跟踪 REFUNDED 及 refund_tx_hash。退款金额及时间取决于交易已到达的阶段，以及服务商的资金返还情况。不要承诺任何失败均可全额、自动或在固定时间内退款。逾期或金额不符的充值须继续跟踪同一订单，不得将其地址用于其他交易。

退款钱包：仅在卖出失败时要求

仅当卖出失败、资金需要退还客户时，才要求填写退款钱包。如订单尚无 refund_address，将停留于

AWAITING_REFUND_ADDRESS (refund_address_required: true)，Lunium 将会发送 webhook：

cashout.awaiting_refund_address。请客户提供符合以下网络的钱包：refund_network（DePix 卖出须使用 Liquid 地址 lq1...），然后提交：

```
curl --fail-with-body -X POST \
  "https://api.luniumpay.com/cash-outs/$CASHOUT_ID/refund-address" \
  -H "X-API-Key: $LUNIUM_API_KEY" \
  -H 'Content-Type: application/json' \
  -d '{"refund_address": "CARTEIRA_DO_CLIENTE"}'
```

退款绝不会自动发送至充值来源地址，只会发送至客户提供的钱包。登记后，退款依次经过（REFUNDING_CRYPT0 用于加密资产卖出， REFUNDING 用于 DePix 卖出），直至 REFUNDED，并提供 refund_tx_hash （对应 webhook 为 cashout.refunding 及 cashout.refunded）；已向服务商发送 DePix 的情况见下文。若待退款项已由 Lunium 持有，refund_amount、refund_asset 及 refund_network 会说明退款内容。退款完成前不要重新发起付款。

DePix 卖出时 (asset: "DEPIX"、 network: "liquid"),POST /cash-outs/{cashout_id}/refund-address ，响应情况如下：

响应	当
200	钱包已登记，订单进入退款流程（REFUNDING），或在 DePix 仍由服务商持有时进入 MANUAL_REVIEW。重复提交相同钱包将返回 200，不产生新的操作
400 refund_address_invalida	地址为空、不符合 Liquid 格式，或属于 Lunium 自身，例如本订单或其他订单的充值地址
409 refund_address_imutavel	订单已登记其他退款钱包，不会替换
409 devolucao_indisponivel	订单当前未在等待钱包：仅在卖出失败且客户 DePix 已由 Lunium 持有时才要求填写。若处于 MANUAL_REVIEW，请等待服务商返还资金

DePix 已发送至服务商的情况。若 DePix 已发送至服务商后 Pix 付款失败，须先在链上确认服务商返还资金，才可向客户退款。在此之前，订单保持 MANUAL_REVIEW（webhook: cashout.under_review），且 refund_amount 为空，即使已登记退款钱包也是如此。确认返还后，refund_amount 才会显示退款金额，最高为实际返还金额加上尚未发送的部分，且不超过订单金额。之后订单进入 REFUNDING；若无钱包，则进入 AWAITING_REFUND_ADDRESS。若 DePix 始终全部由 Lunium 持有，则按订单金额退款。

Pix 官方付款凭证

每笔达到 COMPLETED 状态的卖出订单均有 官方付款凭证 ，由 Lunium 出具。您可将其交给最终客户并留存用于会计记录。凭证包含金额、日期与时间（巴西利亚时间）、E2E（巴西央行 Pix 系统中的唯一端到端标识）、收款机构与收款标识持有人、付款人（Lunium 的 Pix 结算服务商，即收款人流水中显示的付款主体），以及公开核验链接。

同一付款证明有以下四种形式，均包含在订单响应中：

字段	内容	用途
pix_e2e	巴西央行 Pix 系统的官方交易标识	与收款人流水对账，可用于银行核验
receipt_url	可在手机打开的凭证页面	Pix 完成后立即向客户展示
receipt_pdf_url	PDF 凭证文件	附加至支持工单、发送给客户或存档
verify_url	公开核验，无需 API 密钥（GET /v1/verificar/{e2e}）	任何人都可独立核对金额、日期及收款人姓名首字母，无需由您或 Lunium 人工确认

凭证的获取渠道：

- API： 上述字段位于 GET /cash-outs/{cashout_id}，订单进入以下状态后即可获取： COMPLETED。

- Webhook: `cashout.completed` 会在链接已生成时填入相应字段。如返回 `null`、`cashout.receipt_ready`，后续事件会提供完整订单及可用链接。两个事件均须按 `cashout_id`。
- 合作伙伴控制台：在**卖出订单**中打开对应订单，选择“打开凭证”“下载 PDF”或“打开核验页面”。具备金额查看权限的用户可见这些链接。
- Telegram：在密钥绑定的业务群组中，“☒ Pix 已付款”通知包含 E2E 及“查看凭证”链接，随后发送 PDF 附件（“ 官方付款凭证”）。若链接晚于通知生成，则稍后单独发送 PDF。

收款标识持有人。收款人姓名及机构来自 Lunium 付款时进行的一次 Pix 收款标识目录查询。该查询有配额，专用于凭证，不是通用查询服务（GET `/pix/keys/lookup` 即使返回 `verification_status: "not_performed"` 也不影响凭证）。若该次查询未找到姓名，凭证仍然有效，金额、收款标识、日期及机构可从 E2E 等付款信息取得，姓名会稍后补充。

建议做法。将 `pix_e2e` 和 `receipt_pdf_url` 与订单一并保存。向客户提供 PDF，并将 `verify_url` 提供给需要核验的人。客户发来的凭证不能替代到账核验，应以原订单的 `COMPLETED` 状态及 `pix_e2e` 为准。通过 POST `/payouts` 提取余额时，GET `/payouts/{payout_id}` 同样返回 E2E、收款人姓名及 `verify_url`，也可在控制台“对外付款”中打开凭证。买入 Pix 入金的 E2E 与付款人姓名位于 GET `/cashin/{cashin_id}/status` 的 `e2e` 和 `payer_name`。

收款人查询与二维码付款

GET `/pix/keys/lookup?key=...&type=...` 仅为辅助查询，可能返回 200 并传入 `available: false`、`verified: false`、`verification_status: "not_performed"`，且姓名为 `null`。此时应显示“收款人身份未核验”，不得编造姓名；查询不可用也不等于付款不可用。测试环境的收款人资料为虚构数据，并带有 `source: "sandbox"`。

`br_code` 的可用性独立判断。在生产环境创建时，可能因 503 `br_code_indisponivel` 被拒绝，即使预览或测试环境接受该二维码，也不能承诺 Copia e Cola 付款始终可用。启用后，该字段替代 `pix_key`、`amount` 及 `brl_amount`，且须指定金额并使用兼容资产。若客户粘贴或扫描了带金额的静态或动态二维码，请提交完整的 `br_code`，绝不要仅提交从中提取的 `pix_key + brl_amount`：只有这样才能按精确金额、对应收款单及 `txid` 支付。若改为按收款标识付款，金额可能相差几分，导致收款方无法匹配原收款单。

新建的加密资产兑换 Pix 订单适用运营预留款：每笔 R\$ 1.00，适用于所有合作伙伴，已计入报价，并列于 `reserve_brl`。其 USDT 等值金额按当时报价计算，创建后固定在订单中：`pix_key_reserve_usd` 用于 Pix 收款标识付款，或 `copia_e_cola_buffer_usd` 用于 Copia e Cola。这些字段表示换算后的金额，不是另收 US\$ 1；不要将预留款再次加到总额中。比例费用及服务商成本仍包含在报价中，二维码仅按其精确金额支付。付款完成后，运营预留款不予退还；临时余额预留则是另一项机制。已有订单保留原约定金额，DePix/Liquid 按独立的 BRL 费率执行。

仅使用该交易目录列出的网络。即使某些需要 `memo` 的网络出现在交付目录中，也可能不支持 cash-out。不要将某一产品的网络列表直接用于另一产品。

USDT/USDC 钱包：加密资产余额与 Pix 付款

钱包保存您项目的 USDT 余额。您可通过多条网络充值 USDT 或 USDC，再使用该余额通过收款标识或 Copia e Cola 支付 Pix。所有生产密钥均可使用；`lun_test_` 密钥使用模拟器。

USDT/USDC 钱包：充值与 Pix 付款

充值 USDT 或 USDC 后，使用余额通过 API 或控制台支付 Pix。

A · 充值与入账

1. 创建充值请求

POST /carteira-usdt/deposits
填写金额与网络。



2. 核对充值要求

waiting
获取准确金额、地址及 1 小时有效期。地址二维码不包含金额。



3. 通过所选网络发送

expired (1 小时内未发送)
在有效期内发送准确金额的 USDT 或 USDC。



4. 等待网络确认

confirming
达到要求的网络确认后，资金自动入账。



5. 余额可用

available
wallet.deposit_credited
USDC 按 1:1 计入钱包余额。

B · 使用余额支付 Pix

1. 预览所需 USDT (可选)

POST /carteira-usdt/quote
仅查询报价，不影响余额。



2. 请求付款，预留 USDT

reservado
POST /carteira-usdt/payments
可填写 Pix 收款标识或完整 Pix 支付码。使用 external_id 保证幂等。预留的 USDT 可覆盖最多 1% 的变动，多余部分在结算后释放。



3. 跟踪原付款

GET /carteira-usdt/payments/{payment_id}
查询同一 payment_id 的付款进度。



4. 付款完成

concluido
wallet.payment_completed
提供交易凭证及 E2E: receipt_url、pix_e2e。



5. 付款失败

falhou
wallet.payment_failed
Pix 未执行时，预留资金返回可用余额。

revisao 表示核查中；falhou_apos_envio 表示发送后结果待查，均不能直接认定付款失败。

图示：9 条网络；最低充值 5 USDT/USDC，Pix 最低 R\$ 25。以 API 当前目录与报价为准；充值请求有效期为 1 小时。EVM 地址相同，选对网络。

完整说明：docs.luniumpay.com/zh/manual#carteira-usdt

1. GET /carteira-usdt: 余额 (available、reserved、confirming)、当前支持的网络 (networks[])、含币种及预估时间)、min_deposit 及 min_payment_brl。

2. POST /carteira-usdt/deposits 并传入 {"amount": "100", "network": "tron"}: 返回 amount_to_send (金额必须精确一致，末尾小数用于识别该笔充值)、address、network_name 及 expires_at。请向用户展示这四项信息。若发送平台从转账金额中扣除手续费，须额外补足；识别依据是实际到账的金额。充值请求有效期为 1 小时 (deposit_validity_hours = 1; expires_at，即创建时间 + 1 小时)，请在期限内发送。

3. 系统会自动入账，请跟踪 GET /carteira-usdt/deposits (waiting → confirming → available) 或 webhook: wallet.deposit_credited。USDC 按 1:1 计入余额。

4. POST /carteira-usdt/quote 并传入 {"brl_amount": "50.00"} 或 {"br_code": "000201..."}: 以 USDT 计价的预览 (usdt_total、usdt_reserved)，不执行交易。

5. 调用 POST /carteira-usdt/payments, 提交 {"brl_amount": "50.00", "pix_key": "cliente@exemplo.com", "pix_key_type": "email", "external_id": "pedido-123"}, 或提交 br_code。创建成功返回 201, 状态为 reservado。持续查询 GET /carteira-usdt/payments/{payment_id} 直至 final: true, 或跟踪 wallet.payment_completed 与 wallet.payment_failed Webhook。

6. 完成后会提供 pix_e2e、recipient_name、recipient_document (部分遮盖的 CPF) 及 receipt_url (官方付款凭证)。GET /carteira-usdt/statement 用于查询流水: 金额始终为正数, type 表示方向 (deposit、reserve、debit、release、reversal)。

规则: Pix 最低金额为 min_payment_brl (目前 R\$ 25.00)。临时预留可覆盖最高 1% 的波动, 未使用的部分会释放回余额。falhou = 未执行, 预留款已释放; falhou_apos_envio = 结果核查中, 预留款不会自动释放; revisao 表示正在核查, 不等于失败。同一 external_id 配合相同资料会返回同一笔付款 (200); 资料不同时返回 409 external_id_divergente。409 saldo_insuficiente 表示可用余额不足以覆盖预留金额。

共支持 9 条网络: USDT 和 USDC 支持 Polygon、BNB Smart Chain (BEP20)、Ethereum (ERC20)、Arbitrum One、Optimism、Avalanche C-Chain 及 Solana; Tron (TRC20) 仅支持 USDT, Base 仅支持 USDC。当前密钥实际可用网络以 networks[], 详见 GET /carteira-usdt 为准。最低充值为 5 USDT 或 USDC (min_deposit)。未在期限内发送时, 请求将进入 expired。Lunium MCP 提供 6 项钱包工具: 查询余额、创建及列出充值请求、预览付款、执行付款及查询付款。

在 [合作伙伴控制台](#) 中, 充值请求只展示地址二维码, 并显示精确到秒的倒计时, 直至 expires_at。二维码不包含金额, 须在发送应用中手动输入精确金额 (amount_to_send)。EVM 网络的 0x 地址相同, 务必在发送应用中选择正确网络。

测试环境中 (lun_test_): 地址为虚构地址, 切勿转入真实资金; 充值约 15 秒后自动入账, Pix 约 15 秒后完成, 并提供测试 E2E。金额末尾小数用于选择场景: 充值 .02 会过期; Pix .01 会经过 revisao、.02 会失败并释放预留款, .05 约需 2 分钟完成。

9. 通过 Telegram 操作

Lunium Partners - 官方接入者社区

加入 Lunium Partners

Lunium Partners 是 API 接入方建议持续关注的渠道。建议从项目开始就让合作伙伴的技术及运营负责人加入, 并在生产运行期间持续关注通知。参与社区 受到强烈推荐, 但不是创建密钥或使用 API 的技术前提。

在社区中可关注 运行通告、维护通知及 API / 接口契约变更, 也可与其他接入方交流。不要等到故障发生后才加入; 系统上线后仍须持续跟踪变更。

渠道	用途	参与人员
Lunium Partners	官方社区、通知及合作伙伴交流	合作伙伴的技术及运营负责人
项目专属群组	跟进接入进展, 调查具体问题	您的团队、人工联系人 @luniumb2b 及 @luniumb2c 及 @LuniumNotifyBot; 群组可由 Lunium 或合作伙伴创建
私密监控与 webhook	跟踪交易并更新您的系统	合作伙伴授权团队及后端; 参阅 实时跟踪指南

开始步骤: 加入 Lunium Partners, 与两位人工联系人安排项目专属群组, 绑定机器人, 并在正确环境核验监控与 webhook。社区不替代项目专属支持, 也不替代 API 资金对账。

不要在社区公开 API 密钥、webhook 签名密钥、monitor_url、付款人证件或客户交易资料。具体问题请在专属群组处理，仅提供必要的 ID 及最少资料。

使用 Fable 或其他 AI 接入：交付给人工负责人的方案及检查清单中，应包含 Lunium Partners 链接。AI 应解释这一建议，并区分上述三个渠道。未经确认，不应声称某人已加入群组；不要将入群作为执行测试环境测试的前提，也不要将 Telegram 消息视为资金操作授权。

有人工跟进的接入群组

强烈建议为项目建立专属 Telegram 群组，并加入人工联系人 @luniumb2b 及 @luniumb2c。。Lunium 可创建群组并邀请合作伙伴团队，合作伙伴也可自行创建并添加两位联系人。群组用于跟进接入、验收及生产运行，但不是使用 API 的技术前提。

- 1. 自行创建群组，或联系上述人员申请创建，并使用能识别项目的群名。
- 2. 加入两位人工联系人，以及合作伙伴的技术和运营负责人，明确各方职责及升级处理渠道。
- 3. 同时添加 @LuniumNotifyBot，并按下方步骤绑定。机器人发送通知及执行命令，人工联系人跟进疑问和故障，两者各有职责。
- 4. 在群内置顶手册、公开状态页及操作说明。私密监控链接仅可分享给有权查看项目资料的成员。
- 5. 验收时，请核对绑定项目、监控、已签名 webhook，以及所选流程的失败与恢复场景。

请求支持时，需提供环境（测试或生产）、交易 ID、external_id、含时区的时间、预期与实际行为，以及 request_id（如有）。通知问题还须提供 event_id 及 X-Lunium-Delivery-Id。不要发送 API 密钥、webhook 签名密钥、完整付款人证件或其他客户资料。群组成员身份不代表 24 小时人工支持或结算时效承诺，请与团队约定服务安排。

以下资料列出的合作伙伴 / 社区通用渠道： /keys/me，可补充维护与 API 变更通知，但不替代项目专属群组。

机器人绑定与命令

将 @LuniumNotifyBot 添加至项目群组。身份明确的管理员须使用 /vincular 及创建 / 查询密钥时提供的绑定流程。若机器人展示确认按钮，须点击按钮；发送文字“是”不能代替点击。不要在群组中公开 lun_... 密钥。

命令	功能
/configurar	检查绑定与配置
/cobrar <valor BRL> <cpf/cnpj> [cliente=ref]	创建 Pix 托管收款单，无需固定钱包
/saldo [customer_ref]	查询指定账户范围的余额
/extratosaldo [customer_ref]	查询托管余额流水
/moedas [moeda]	查看可交付的资产路由
/sacarcripto <valor BRL> <moeda> <rede> <endereço> <cpf/cnpj> [memo] [cliente=ref]	预估并确认托管余额的加密资产提现
/pagar <valor BRL> <chave PIX> [tipo] [cpf/cnpj]	从项目余额扣款并支付 Pix
/status <id>	跟踪交易

命令	功能
/limites	查询项目限额及付款人分级规则
/monitor	打开项目监控
/comissao [taxa%]	查询佣金，或提交变更供管理员确认； 指南

Telegram 中的金额单位为 巴西雷亚尔，例如 50 或 50,50；API 中的 amount_cents 以分为单位。对子账户操作前，须核对显示的账户范围。/pagar 不提供加密资产提现中的 cliente=ref 参数。子账户 Pix 对外付款请通过 API 传入 customer_ref。

加密资产提现会在确认前展示目的地、预估数量及扣款金额。确认有效期为 60 秒，仅对发起请求的用户、群组及项目有效。群组提现要求身份明确的管理员操作，匿名发送的消息不能授权提现。确认时会重新校验条件，必要时要求重新预览，不会锁定汇率。

/sacar 是外部加密资产兑换 Pix 的流程。将余额以加密资产形式提取，请使用 /sacarcripto。/qr、/qrwhitelist 及 /qrdelay 等命令在未明确提供钱包时，也会使用托管余额；处理时间及冻结条件须以响应为准。

10. 可靠的 webhook

配置 webhook_url，保存 webhook_secret，并执行 POST /webhooks/test 测试接收端。此调用会向已配置的地址发送测试通知。

事件封装示例：

```
{
  "event": "cashin.settled",
  "event_id": "evt_exemplo",
  "created_at": "2026-09-17T20:00:00.000Z",
  "data": {
    "cashin_id": "ci_exemplo",
    "destino": "saldo",
    "status": "paid",
    "settlement_status": "sent"
  }
}
```

示例省略了部分 data。收款单、cash-out 及 Pix 对外付款事件使用各自资源的基础状态对象；GET 查询补充的信息，例如 timeline，不一定出现在事件中。saldo.* 事件则有独立的流水数据。不要要求每个 webhook 的所有字段均与补充信息后的 GET 响应完全一致。

请求头	用途
X-Lunium-Signature	t=<unix>,v1=<HMAC-SHA256 em hex>，签名内容为 timestamp.corpo_bruto
X-Lunium-Event-Id	稳定的事件标识，用于去重
X-Lunium-Delivery-Id	投递尝试标识，用于诊断
X-Lunium-Attempt	尝试次数

验证 原始请求体，不要重新序列化 JSON；还须检查时间戳有效期。比较长度相同的缓冲区时，应使用恒定时间比较。可使用 [完整 Node.js 验签示例](#)，它也会安全拒绝格式错误的签名，不会因缓冲区长度不同而抛错。

持久化保存或入队事件后，尽快返回 2xx。投递请求的 超时时间为 10 秒，耗时处理应放在之后执行。不要按投递尝试去重：同一事件的所有尝试仅应产生一次业务效果。

未成功时，最多尝试 12 次，并逐步延长重试间隔。整体约 7 小时，但受队列影响，不构成投递 SLA。请查询 `GET /webhooks/deliveries`，并在修复接收端后通过 `POST /webhooks/deliveries/{event_id}/retry` 重发。重试耗尽的投递将标记为 failed，仍可依据记录恢复。

新接入应使用 v2 签名。X-Webhook-Signature 为旧版字段，新密钥不再提供，已公告于2026年11月9日移除，请勿依赖。

事件

分组	事件
cashin	cashin.paid • cashin.settled • cashin.settlement_failed • cashin.expired • cashin.refunded • cashin.failed • cashin.delayed • cashin.held
cashout	cashout.awaiting_deposit • cashout.deposit_detected • cashout.processing • cashout.confirming • cashout.under_review • cashout.awaiting_refund_address • cashout.refunding • cashout.completed • cashout.refunded • cashout.expired • cashout.failed • cashout.receipt_ready
payout	payout.sent • payout.failed • payout.refunded
saldo	saldo.liberado • saldo.congelado • saldo.transferido
webhook	webhook.test

请兼容新增事件与字段，记录未知内容，并在必要时通过 GET 核对资源。不要假定事件按顺序到达；更新已完成的资源前，先查询当前状态，避免回退。

合作伙伴实时监控

合作伙伴可通过 Lunium 私密监控查看自己的项目。查询 `GET /keys/me`，在后端获取响应中的 monitor_url 并打开，或在已绑定群组中使用 /monitor。创建密钥时也会返回该链接。链接提供项目只读访问，仅可分享给授权人员，不得放入公开文档或交给最终客户。

监控每隔 3 秒 查询事件流，每隔 15 秒查询托管余额。它展示收款单、cash-out、Pix 对外付款及 webhook 投递；启用托管后还展示可用 / 冻结余额、子账户及近期流水。这些间隔仅表示界面查询频率，不保证数据库、服务商或区块链在该期限内更新。

私密链接泄露时，请使用 `PATCH /keys/me` 并传入 {"rotate_monitor_token":true}，并保存新的 monitor_url，旧 token 随即失效。监控 token 虽为只读，但可查看整个项目，而非单个客户；不要将其用作公开交易详情页的身份验证。

构建您自己的控制台事件流

```
# 在后端执行；从密钥管理器读取 LUNIUM_API_KEY。
curl --fail-with-body 'https://api.luniumpay.com/cashin/events?limit=80' \
-H "X-API-Key: $LUNIUM_API_KEY"
```

GET /cashin/events 接受在请求头中传入 X-API-Key 或 X-Dashboard-Token。托管监控页面使用 ?token=。响应包含 projeto、agora 及 eventos。每个条目提供 ts、fluxo (cash-in、cash-out、payout 或 webhook)、provedor、ref、valor、estado、detalhe 及 tx，部分字段可能为空。limit 默认为 60，允许范围为 1 至 200。

这是近期摘要，不是完整的状态变化历史。列表按 ts 降序排列。交易可能仅显示当前状态；事件流没有重放游标，也没有 webhook 封装中的 event_id。不要根据 valor/estado 等说明文本计算余额、费用或进行对账，也不要将 ts 当作全局变更序列。金额、网络、哈希及凭证须查询原始资源；尤其应使用 cash-in 状态返回的 settlement_tx_url 打开对应网络的区块浏览器，不要根据事件流摘要默认使用 Polygon。

Webhook + 状态查询 + 界面更新

Webhook 是 Lunium 向合作伙伴后端发送的 HTTPS POST 请求。若要在您的应用中提供实时体验，请实现以下流程：

1. 接收：读取原始请求体并限制大小，在解析 JSON 前验证签名及时间戳。核对请求体中的 event_id 是否与请求头一致。签名密钥对应指定 API 密钥及环境，不要依赖 payload 自行选择项目。
2. 保存：先将事件写入持久化收件箱，并设置以下唯一约束：(projeto, ambiente, event_id)，再返回 2xx。重复的已保存事件直接返回 2xx，不重复执行。写入失败时返回错误，以便重试。不要静默丢弃已通过身份验证的未知事件。
3. 对账：后台任务处理收件箱，必要时查询权威资源。按交易串行更新，避免旧响应覆盖新查询。分别保存收到的事件历史、查询状态及最后查询时间。
4. 更新界面：您的后端通过 SSE、WebSocket 或轮询，将结果发布给应用中已验证身份的会话。每次读取及连接均须校验项目和客户权限。这些通道由合作伙伴自行实现；本接口契约未提供 Lunium 公开 SSE/WebSocket 接口。
5. 恢复：定期核对未完成或交付结果不确定的交易，采用可调整间隔、退避机制及密钥限额；不要为每个打开的浏览器单独向 API 发起查询。客户端重连后，加载已保存状态，并提示更新是否延迟。

Lunium 监控提供现成的运行视图，合作伙伴控制台可补充自己的订单、客户及持久历史。Webhook 不替代流水，也不记录交易所、银行或其他服务商的每个内部步骤。

交易 / 关联	状态核对接口	如何展示完成
Pix 收款单或托管余额的加密资产提现：cashin_id	GET /cashin/{cashin_id}/status	status=paid 及 settlement_status=sent；通过 destino/fonte 区分 BRL 入账与加密资产交付
外部加密资产兑换 Pix：cashout_id	GET /cash-outs/{cashout_id}	state=COMPLETED；返回凭证 / E2E 时予以展示
Pix 对外付款：payout_id	GET /payouts/{payout_id}	status=sent；不要与 webhook 投递状态混淆 sent
流水 saldo.*：customer_ref、origem_id、charge_id、movimento_id	GET /saldo 及 GET /saldo/extrato，并使用正确的 customer_ref（如有）	可用余额来自账本，已登记入账与资金已释放是不同概念

还须保存创建交易时自行设置的 `external_id`。Cash-in 状态可能提供 `timeline` 并传入 `etapa`、`em`、`detalhe`；cash-out 状态可能提供 `timeline` 并传入 `state`、`at`。仅展示接口实际返回的阶段及时间。确认数、哈希或凭证可能在生成前缺失或为 `null`。不要编造进度百分比、完成时间、链上交易或失败原因。

各事件在界面中的含义

事件	解释与处理方式
<code>cashin.paid</code>	Pix 已确认，仍须核对交付 / 入账。不能仅凭此事件显示“加密资产已交付”
<code>cashin.settled</code>	已报告结算，须核对资金目的地与状态。托管余额还须查询可提现金额
<code>cashin.delayed</code> 、 <code>cashin.held</code>	延迟 / 冻结；如有返回，请展示 <code>delay_until</code> / <code>liberar_em</code> ，不要视为最终失败
<code>cashin.settlement_failed</code> 、 <code>cashin.failed</code>	结算或交易异常；查询当前状态，记录错误并升级处理，不要自动新建付款
<code>cashin.expired</code> 、 <code>cashin.refunded</code>	已过期或退款，须核对原收款单
<code>cashout.awaiting_deposit</code> 、 <code>cashout.deposit_detected</code> 、 <code>cashout.confirming</code>	等待充值、已检测到或确认中；仅在接口提供数据时显示倒计时 / 确认计数
<code>cashout.processing</code>	综合处理状态，多个内部阶段共用此事件，不会为每一步单独发送
<code>cashout.under_review</code> 、 <code>cashout.refunding</code>	审核或退款进行中，须继续跟踪
<code>cashout.awaiting_refund_address</code>	卖出失败，须退款，但订单尚无退款钱包。请向客户索取钱包并提交至 <code>POST /cash-outs/{cashout_id}/refund-address</code> ；退款完成前不要再次付款
<code>cashout.completed</code> 、 <code>cashout.refunded</code> 、 <code>cashout.expired</code> 、 <code>cashout.failed</code>	Cash-out 结果已报告，须核对资源及可用凭证
<code>cashout.receipt_ready</code>	Pix 凭证已生成，晚于 <code>cashout.completed</code> 到达（付款后会查询收款人，链接可能数秒后才生成），并提供完整订单，填入 <code>receipt_pdf_url</code> 、 <code>receipt_url</code> 及 <code>verify_url</code> 等字段。可用此事件向客户交付凭证，无需持续重复查询
<code>payout.sent</code> 、 <code>payout.failed</code> 、 <code>payout.refunded</code>	余额支付 Pix 的结果，须核对交易及流水
<code>saldo.liberado</code> 、 <code>saldo.congelado</code> 、 <code>saldo.transferido</code>	余额变化，须查询正确账户范围；不要把事件直接相加作为流水
<code>webhook.test</code>	投递 / 验签测试，不代表资金或金融交易

`settlement_status=incerto` 须先对账，再考虑任何新的交付。旧 `webhook` 可能晚于已显示完成的查询到达；将其保存到历史，但不要回退当前状态。已完成交易也可能有合法后续更新，例如退款；应查询权威来源，而不是首次成功后忽略全部更新。

配置、测试与恢复通知

在正确环境的密钥中配置公开 HTTPS 接收端。系统会拒绝私网 / 本地地址。签名密钥仅在生成时返回，请保存在后端。更新 URL 不一定生成或返回新签名密钥；如需轮换，请使用 rotate_webhook_secret 协调切换，后续尝试使用届时有效的签名密钥。

```
curl --fail-with-body -X PATCH 'https://api.luniumpay.com/keys/me' \
  -H "X-API-Key: $LUNIUM_API_KEY" -H 'Content-Type: application/json' \
  -d '{"webhook_url":"https://seu-dominio.example/webhooks/lunium"}'

# 仅测试投递与签名，不创建付款。
curl --fail-with-body -X POST 'https://api.luniumpay.com/webhooks/test' \
  -H "X-API-Key: $LUNIUM_API_KEY"

curl --fail-with-body 'https://api.luniumpay.com/webhooks/deliveries?status=failed&limit=100' \
  -H "X-API-Key: $LUNIUM_API_KEY"

# 修复接收端后再重发，使用属于当前密钥的 event_id。
curl --fail-with-body -X POST \
  "https://api.luniumpay.com/webhooks/deliveries/$EVENT_ID/retry" \
  -H "X-API-Key: $LUNIUM_API_KEY"
```

请将示例域名替换为真实接收端。POST /webhooks/test 在生产环境每个密钥每分钟最多调用 6 次，返回 ok、status_code、ms、event_id 及 delivery_id。此类单次测试不会保存至生产投递队列，只验证通信，不验证真实结算或应用的所有业务路径。

GET /webhooks/deliveries 接受在请求头中传入 status (pending、sent、failed)、event 及 limit (默认 50，最多 200)。每行提供 event_id、operation_id、attempts、last_error、url、next_attempt_at、created_at 及 updated_at，以及事件 / 状态。failed_count 仅统计本次返回记录中的失败数量。该接口不支持按 operation_id 筛选，也没有分页游标；请维护自己的持久历史，并从返回记录中查找交易。

重试响应：200 表示投递成功；202 表示失败且已重新入队；409 表示之前已送达；404 表示该密钥下不存在对应记录。pending/sent/failed 在此表示 webhook 的投递结果，不是付款结果。重发通知不会重新付款，也不会修复未完成的结算。

实时跟踪接入验收

- 在测试密钥中配置接收端，并执行以下指南中的适用场景：[测试环境](#)。GET /sandbox/events 用于查看模拟事件；不要用生产运行事件流证明测试环境覆盖范围。
- 验证签名错误 / 过期、重复事件、乱序、接收端不可用、重连后的恢复，以及项目 / 客户之间的隔离。
- 确认 cashin.paid 不会标记为加密资产已交付，冻结资金不会变成可用余额，且 webhook.test 不会改变余额。
- 将同一交易的创建响应、webhook 及 GET 状态按 ID 关联核对，超时后不创建新交易。
- 验证结算错误、结果不确定、退款及通知恢复；需要调查时，将 ID 提供给人工支持群组。
- 控制台应展示交易、外部订单、获授权客户、币种 / 网络、实际返回金额、资金状态、webhook 投递状态、最后更新时间，以及可用的凭证 / 哈希。资料过期时予以提示，不要误导用户认为仍在实时更新。

11. 错误、幂等性与对账

结合 HTTP 状态码与 erro、acao、detail、request_id 及 agent_guidance 等字段（如有）判断。有些路径仅返回部分字段，不要仅依据以下字段的自由文本内容编写业务分支：detail。

推荐关系	处理方式
corrigir	调整字段或金额，并读取 limits、payer_limits 及 min_brl_cents
repetir	使用退避机制，以同一标识及请求内容重试
esperar	遵守 Retry-After、资金释放时间或服务状态
parar / falar_com_suporte	停止自动重试，先解决限制条件
401 / 403	核对身份验证、密钥权限、限额及交易规则
402 saldo_insuficiente	核对可用余额、子账户及提现费用
409 external_id_divergente	该 ID 已属于另一交易，不可用它修改原请求
423 carencia_primeiro_deposito	展示返回的期限，不要换用其他子账户绕过
429	等待额度恢复，减少并发查询
5xx 或发送后超时	重试前先按 ID / 引用标识查询原交易

发送 external_id，长度不超过 64 个字符，适用于支持该字段的金融交易。每个业务意图使用一个持久标识，重试时保持所有参数完全相同。收款单及加密资产提现共用密钥的引用命名空间，应使用不同前缀。不要利用差异校验修改已有交易，各接口的比较规则可能不同。

超时恢复时，根据交易类型查询 /cashin/charges?external_id=...、/cash-outs?external_id=... 或 /payouts?external_id=...。确需重试时，保持相同请求体及标识；不要仅因响应丢失就生成新 ID。

保存 ID、金额、币种 / 网络、customer_ref、时间戳、已处理事件及流水引用。BRL 使用整数分，加密资产使用十进制字符串。日期包含明确时区，舍入及兑换须以 API 返回值为准。

操作指南：查询内容与客户提示

情况	核查与处理	给客户的提示
Pix 已支付，加密资产待交付	查询 GET /cashin/{id}/status。paid 及其结算状态 pending、sending 或 incerto 不等于交付已确认。保留 ID 并持续跟踪，不要创建新付款。如失败、结果不确定或持续没有进展，请将 ID 提供给支持团队。	“付款已收到，正在跟踪交付。”展示最后更新时间，不编造完成期限。
托管入账暂不可用	在与收款单相同的账户范围查询状态及 GET /saldo?customer_ref=...。核对冻结 / 可用金额及返回日期。服务商冻结、本地冻结与余额等待期是不同条件。	“款项已收到，可用时间以告知的条件及期限为准。”不要把总余额展示为可提现金额。
未收到 webhook	查询权威交易状态及 GET /webhooks/deliveries。核对 URL、环境、签名、超时及接收端 HTTP 响应。修复接收端后，再重发原有的 event_id。队列中没有记录不代表付款失败。	以交易查询确认的状态为准，通知延迟不改变资金状态。
POST 请求后响应丢失	使用已保存的 external_id 在对应流程列表中查找交易。找到后继续跟踪；暂未找到时，以退避方式继续核对，保持原请求体及 ID。立即查不到不代表请求未获接受。	“请求正在核查中。”不要要求客户再次支付 Pix，也不要自动创建新的付款意图。
路由 / 币种不可用或未达到最低金额	更新正确目录，读取 erro、acao 及返回的最低金额。付款前可请客户重新选择；付款后须继续跟踪原交易并升级处理，不要静默更换网络、币种或地址。	说明具体限制及已确认可用的选项，不承诺未经核实的路由。
佣金未显示	核对收款单保存的费率、结算完成情况及佣金流水。区分核算延迟、零费率与冲回。发现差异时，携原交易 ID 联系支持。	“佣金正在核查中。”不要承诺修改配置即可追溯入账。

任何交易流程中，若控制台资料过期，都应显示“最后更新时间”及连接状态。不要将超时、空响应或未收到 webhook 视为资金成功或失败。结果不确定的交易必须先对账，不要仅为消除告警就手动另发转账。

联系人工支持群组时，需提供环境、交易 ID、external_id、含时区的时间、HTTP 状态码、erro/acao、request_id（如有）及最近查询状态。Webhook 问题还须提供 event_id 及尝试的 HTTP 响应 / 时间，并对日志脱敏。不要在公开社区发送密钥、签名密钥、私密监控链接、完整 CPF 或含客户资料的 payload。本手册不保证统一的支持或结算期限。

Fable 可参考 [结构化故障处理指南](#)。它提供查询及对账步骤，不授予提现、修改费用或再次转账的权限。

12. 测试环境与接入测试

运行 `node sandbox.mjs all` 或 `python3 sandbox.py all`，工具包：[工具包](#)。示例会核验测试密钥、模拟 Pix 付款，并覆盖托管余额、USDC/Base 提现、Pix 对外付款、BTC、USDC/Polygon 及 cash-out 流程。Cash-in / Pix 对外付款使用 `sandbox_scenario`；资金冻结、退款、交付失败及结果不确定的场景，详见 [指南](#)。GET /sandbox/events 可用于检查事件；已配置的回调使用产品自带的签名队列。价格与时间均为模拟值，不代表合作交易所的实际条件。

USDT/USDC 钱包：测试密钥使用钱包模拟器，接口及数据结构相同。充值 .02 会过期；Pix .01 会进入审核，.02 会失败并释放预留款，.05 会延迟完成。详情参阅 [USDT/USDC 钱包](#)。

测试环境 cash-out 场景由 amount 的前两位小数决定：

金额尾数	场景
.00	模拟成功流程
.01	模拟资金冻结 / 延迟，稍后完成
.02	失败
.03	报价在 5 秒后过期
.04	因限额被拒绝，并返回 limits
.05	延迟完成，约 2 分钟
.06	服务商拒绝，返回 pagador_recusado_pelo_provedor
.07	首次接受报价返回 503；再次接受同一订单即可成功
.08	因付款人限额被拒绝

测试订单仅为临时数据，不代表真实充值。测试环境不能证明生产环境的实际时效、价格、服务商可用性或托管入账结果。Cash-in 与余额功能须覆盖文档列出的模拟接口和场景。之后开展真实试点时，应使用该接口及付款人允许的最低金额，并事先确认金额及收款人已获授权。

13. 查询当前规则

信息	API 来源
密钥权限、单笔限额及请求频率限制	GET /keys/me
付款人限额及资金冻结分级	GET /cashin/limits?payer_tax=...
可用 / 冻结余额、等待期及托管费用	GET /saldo，须使用正确账户范围
交付加密资产的币种 / 网络	GET /cashin/catalog 并传入 entregavel: true
接收外部加密资产的币种 / 网络	GET /catalog，以及适用时的 /catalog/dex
Pix 兑换加密资产的预估及路由最低金额	POST /cashin/preview
外部加密资产兑换 Pix	POST /cash-outs/preview 及已创建的报价
收款单 / 报价的实际有效期	expires_at，以该交易本身为准
资金释放、完成状态及凭证	GET 状态查询、webhook 及流水

GET /keys/me 本身不包含所有条件。每日限额为 null 仅表示密钥没有该项总额上限；付款人、收款人、单笔、余额及可能的 mandate 限制仍有效。不要根据手册示例将目录、费用、期限或限额固定写入代码。

14. 佣金、智能体与其他操作

配置合作伙伴佣金

Sub fee 是在 Lunium 费用之外另计的合作伙佣金。100 bps = 1%；200 bps = 2%。配置前，请通过 GET /comissao 查询规则及允许的上限。下方示例会改变该密钥新建收款单的价格，仅应用项目负责人同意的费率。

```
curl --fail-with-body -X PATCH https://api.luniumpay.com/keys/me \
  -H "X-API-Key: $LUNIUM_API_KEY" \
  -H 'Content-Type: application/json' \
  -d '{"partner_fee_bps":200}'
```

核对已保存的配置值：GET /keys/me。费率会记录在每张新收款单中；修改配置不会重新计算已有收款单，也不会生成追溯佣金。Telegram 中，/comissao 用于查询；/comissao 2 用于提交变更并请求确认。管理员须核对密钥身份，使用机器人返回的确认命令（含密钥 ID），不得替其他项目确认。

区分费用、佣金、汇率与交付数量

Cash-in 的计费基数为收到金额减去 Lunium 固定费用，再对该基数计算 Lunium 及合作伙伴比例费用。余额充值仅扣 R\$ 1.00，无比例费用、分档费率或佣金。收款单保存的参数及交易返回金额优先于示例。

以下示例以分计算，仅作教学，不代表当前报价：收到 10000（R\$ 100），固定费用 100（R\$ 1），Lunium 费率 200 bps（2%），合作伙伴费率 100 bps（1%）。

组成项	示例金额
扣除固定费用后的基数	R\$ 99.00
未扣合作伙伴佣金的净额	R\$ 97.02
扣佣金后、兑换前的净额	R\$ 96.03
Lunium 费用，含固定部分	R\$ 2.98
合作伙伴佣金	R\$ 0.99
核对	R\$ 96.03 + R\$ 2.98 + R\$ 0.99 = R\$ 100.00

计算使用 $\text{floor}(\text{base_cents} * (10000 - \text{lunium_bps} - \text{partner_bps}) / 10000)$ 。佣金为扣除 sub fee 前后净额的差值，并保留产品的舍入规则。不要对已报价的加密资产数量再次扣除佣金。

POST /cashin/preview 可能返回 lunium_fee_cents、partner_fee_cents、fee_total_cents、net_cents，加密资产预估数量列于 asset_amount，已扣网络提现费用（withdraw_fee）。提供此项拆分的响应中， $\text{fee_total_cents} + \text{net_cents} = \text{amount_cents}$ 。旧版字段 brl_per_usdt 及 usdt_amount 须结合响应中的资产判断，USDC 也可能沿用相同字段名。

汇率并非另一笔合作伙伴佣金。兑换采用 Binance 卖一价（ask），不再使用旧溢价。Lunium 费用及佣金与汇率分别计算。Cash-in 预览仅为预估，不锁定付款时汇率。路由 / 网络成本及最低金额也可能影响交付，请读取目录及响应中的适用字段。不要对已计入该费用的数量，再次加上 withdrawFee。实际结算数量须在交易状态及凭证中核对。

托管充值的结果为 BRL，不会立即购买加密资产。后续提现另有费用及兑换条件，cash-in 预览不是托管提现的完整预览。Pix 对外付款响应分别列出 fee_service_total_cents（Lunium + 合作伙伴）及 fee_provider_cents；fee_cents 为费用总额。合作伙伴佣金、客户余额及 Lunium 利润属于不同账目，单凭预估与实际交付之间的差额，不能证明利润。

佣金入账、流水与提现

1. 查询 GET /comissao: disponivel_cents、creditado_cents、estornado_cents、sacado_cents、min_saques_cents、pode_sacar_agora、proximo_saques_em 及 politica。
2. 查询 GET /comissao/extrato?limit=50。下一页使用 next_before 作为 before。请将 origem_id 关联到收款单或提现，并读取 kind、valor_cents 及 partner_fee_bps。
3. 佣金在 status=paid 及 settlement_status=sent 之后核算，晚于结算本身。Pix 已付款不代表佣金已可用。新余额充值不产生佣金；购买及转出分别使用对应类别的佣金，须查询交易保存值。测试环境不累积真实佣金。
4. 以下流水类型 credito、estorno、saque 及 saque_estorno 用于解释余额。请核对正负号，避免重复计数。符合条件的交易在核算后仍未显示时，请将 ID 提供给人工支持群组对账，不要通过修改费率尝试补记历史。
5. 提现前，查询当前规则、余额及可交付路由。以 BRL 换算的最低金额（目前按实时汇率折合 US\$ 10）及提现间隔以 API 为准；若 min_saques_cents 为 null，则无有效报价用于验证最低金额。pode_sacar_agora 不能替代网络、地址、memo、流动性及路由最低金额校验。

POST /comissao/sacar 接受 amount_cents、asset、chain、payout_address、必要时的 payout_tag 及 external_id。提现不再收取 sub fee 或 Lunium 管理费，但仍须考虑兑换及路由成本。响应包含 saque 及 comissao；请跟踪以下公共交易标识：saque → GET /cashin/{cashin_id}/status，直至确认交付。HTTP 成功不代表加密资产已交付。

使用独立前缀，例如 comissao-...，用于 external_id，因为引用命名空间与收款单共用。超时后，先查找 /cashin/charges?external_id=... 并跟踪已有 ID。提现会先校验规则及间隔，再查找引用，因此重试可能返回 429，即使之前的请求已经获接受。不要生成新 ID 或并发重复提现来尝试解除限制。

PATCH /keys/me 接受在请求头中传入 partner_fee_bps 用于 sub fee。当前规则、可用佣金及提现条件见 GET /comissao；流水见 GET /comissao/extrato；提现使用 POST /comissao/sacar。佣金余额与客户托管余额分开，不要相加，也不要将佣金当作 Mesh 推荐奖励。

面向智能体的 Mesh 提供发现、注册、验证、mandate 授权及激活能力。请使用 GET /mesh 及其返回的后续操作。MCP 地址为 https://api.luniumpay.com/mcp；通过 tools/list 发现工具，因为 MCP 工具范围并不等同于全部 HTTP 接口。没有专用工具的操作，仍可通过已验证身份的 API 调用。

下方清单由本次修订所依据的 OpenAPI 生成，亦包含 webhook 投递查询、监控访问恢复、DEX 目录、佣金及 Mesh 生命周期。管理及内部接口不属于合作伙伴公开 API。

方法	接口路径	操作
GET	/cash-outs	列出订单
POST	/cash-outs	创建报价（第 1/3 步）
POST	/cash-outs/preview	报价预览（不创建订单）
GET	/cash-outs/{cashout_id}	跟踪订单（第 3/3 步）
POST	/cash-outs/{cashout_id}/accept	接受报价并取得地址（第 2/3 步）
POST	/cash-outs/{cashout_id}/refund-address	为待退款订单提交客户确认的钱包
GET	/cashin/catalog	Cash-in 目录：Pix 收款后可交付的资产

方法	接口路径	操作
POST	/cashin/charge	创建 Pix 收款单，用于 BRL 托管或加密资产交付
GET	/cashin/charges	列出 / 恢复当前密钥的 cash-in 收款单
GET	/cashin/events	当前密钥的运行事件流
GET	/cashin/limits	查询该付款人当前可支付金额
POST	/cashin/preview	Cash-in 预估，不创建交易
GET	/cashin/{cashin_id}/status	Pix 收款单状态
GET	/catalog	当前可结算的资产、网络及限额
GET	/catalog/dex	可通过 DEX 交易的长尾代币
GET	/comissao	Sub fee 余额，即合作伙伴佣金
GET	/comissao/extrato	佣金流水
POST	/comissao/sacar	以加密资产形式提取佣金
POST	/keys	自助创建密钥
GET	/keys/me	当前限额及当日使用量
PATCH	/keys/me	更新密钥配置：名称、结算、webhook 及凭证轮换
POST	/keys/painel-link	通过邮件获取监控链接，无需 API 密钥
POST	/keys/sandbox	一次调用获取测试密钥，无需已有 API 密钥
POST	/leads	将商务或技术问题转交人工
GET	/mesh	智能体接入前须知
GET	/mesh/agents	发现网络成员
POST	/mesh/agents	注册智能体并获取测试环境凭证
GET	/mesh/agents/{agent_id}	智能体状态、mandate 授权、指标及推荐关系
POST	/mesh/agents/{agent_id}/mandate	设置自身限额策略
GET	/mesh/agents/{agent_id}/network	推荐关系、层级树及网络交易量
POST	/mesh/agents/{agent_id}/production-request	自动激活生产环境，无需人工审批
GET	/mesh/agents/{agent_id}/rewards	奖励余额及账本流水

方法	接口路径	操作
POST	/mesh/agents/{agent_id}/settlement	设置资金目的地
POST	/mesh/agents/{agent_id}/validate	提交测试环境使用证明
GET	/mesh/economy	网络经济的公开规则
GET	/mesh/graph	Mesh 智能体关系图
POST	/mesh/referrals	签发推荐证明（attestation）
GET	/payouts	列出当前密钥的 Pix 对外付款
POST	/payouts	直接支付 Pix，不经过加密资产环节
GET	/payouts/{payout_id}	Pix 对外付款状态
GET	/ping	API 健康状态
GET	/pix/keys/lookup	在核验服务可用时查询 Pix 收款人
GET	/saldo	BRL 托管余额：可用、冻结及后续资金释放
GET	/saldo/clientes	分页列出有余额的客户子账户
GET	/saldo/consolidado	密钥汇总：主账户 + 客户子账户，总托管金额
GET	/saldo/extrato	余额流水，最新记录在前
POST	/saldo/sacar-cripto	将 BRL 余额兑换并提取至自托管加密资产钱包
POST	/saldo/transferir	子账户与主账户之间内部转账，不生成 Pix
GET	/sandbox	查询测试环境功能
POST	/sandbox/cashin/{cashin_id}/pay	模拟测试 Pix 收款单的付款
GET	/sandbox/events	查看测试环境生成的事件
POST	/sandbox/operations/{operation_id}/resolve	核实测试环境中结果不确定的交易
GET	/sandbox/qr/{cashin_id}.svg	不可用于真实付款的测试二维码图片
POST	/sandbox/reset	重置当前密钥的模拟数据
GET	/v1/verificar/{e2e}	公开核验 Pix 付款，无需密钥
GET	/webhooks/deliveries	查询当前密钥的 webhook 投递历史

方法	接口路径	操作
POST	/webhooks/deliveries/{event_id}/retry	重发事件
POST	/webhooks/test	立即发送测试事件

15. 上线检查清单

- 已在正确密钥中配置佣金，费用拆分与流水已核对，无重复扣费。
- 已核对恢复邮箱，分别保管监控链接、API 密钥及 webhook 签名密钥。
- 已演练故障处理流程，不将开发中的能力视为已上线。
- 建议技术及运营负责人加入 [Lunium Partners](#)，并明确持续跟踪通知及变更的安排。
- 强烈建议与 @luniumb2b 及 @luniumb2c 建立项目专属群组，由 Lunium 或合作伙伴创建，并明确负责人。
- 已核验私密监控；自建控制台按项目 / 客户授权，并显示最后更新时间。
- 已测试持久化收件箱、重复事件、乱序、接收端不可用、重试及 webhook 对账。
- 密钥保存在服务器端，测试环境与生产环境分离。
- /keys/me 及 /saldo 已按密钥及各客户范围核对。
- 托管收款使用 destino: "saldo"；只有在提取加密资产时才选择钱包。
- 界面读取当前目录及限额，展示资金冻结条件，并在需要时要求填写 memo。
- 已持久化保存 ID；重试保持相同请求体及 external_id；已测试超时后的恢复流程。
- Webhook 验证 HMAC 与时间戳，按事件去重，并在超时前响应。
- 对账明确区分 Pix 已支付、入账已登记、资金可提现及加密资产已交付。
- 处理余额、等待期、接口、费用、服务商及密钥错误时，不重复执行交易。
- Telegram 由身份明确的管理员绑定，提现须在正确的会话及项目中确认。
- 接入验收覆盖成功、资金冻结、过期、失败、重发及响应丢失后的查询。

技术支持：[@luniumb2b](#) · [@luniumb2c](#) · contato@luniumpay.com。请关注 [服务状态](#) 及 [Lunium Partners](#)。请求支持时，请提供交易 ID 及 request_id（如有），不要分享完整 API 密钥。

16. 本次修订内容

- 2026年10月9日 · OpenAPI 1.44.1 · 修订 2026-10-09.1：测试环境：post /sandbox/operations/{operation_id}/resolve 会返回 400 outcome_invalido，条件为 outcome 不是 sent 或 failed 时（此前，请求体缺少 outcome 时，可能返回 200 却未解决任何操作）。校验发生在查找交易之前，模拟器不会因此改变。余额：在 GET /saldo、proximas_liberacoes 中，可返回所查询账户范围最多 200 项资金释放记录（原先最多 20 项）；格式及范围（主账户或 customer_ref）保持不变。

• 2026年10月8日 • OpenAPI 1.44.0 • 修订 2026-10-08.2: DePix 卖出: refund_address 在创建时不再必填, 仅在卖出失败时才要求退款钱包 ([退款钱包](#))。未提供钱包时, 订单停留于 AWAITING_REFUND_ADDRESS (webhook: cashout.awaiting_refund_address), 之后通过以下接口填写 Liquid 地址: POST /cash-outs/{cashout_id}/refund-address; 订单随后进入 REFUNDING 及 REFUNDED, 并发送对应 webhook。错误包括: 400 refund_address_invalida (含 Lunium 自身地址)、409 refund_address_imutavel 及 409 devolucao_indisponivel。已发送至服务商的 DePix, 须在链上确认返还后才可向客户退款; 此前订单保持 MANUAL_REVIEW, 退款金额最高为实际返还金额。有关付款人限额拒绝的 detail 及 allow_hold: false 说明已修订, 金额采用巴西格式 (R\$ 1.234,56), 相关字段说明亦已更新: GET /cashin/limits; erro 及 code 保持不变。佣金提现最低 US\$ 10。合作伙伴控制台 (第 2 节): 新账户通过邮件 6 位验证码激活; 通过验证码或 Google 登录的用户可一键接受邀请; 我的访问权限 并传入 退出其他会话、无密码账户可用代码关闭双重验证, 密码变更时发送邮件通知; 成员失去所有者角色时通知其他所有者; 使用加密资产付款 支持稳定币及 30 种主要加密资产, 提高可读性, 会话有效期为 12 小时并随使用续期。第 8 节。

• 2026年10月8日 • OpenAPI 1.43.1: USDT/USDC 钱包 充值请求有效期改为 1 小时 (deposit_validity_hours = 1; expires_at, 即创建时间 + 1 小时, 此前为 3 小时。新增 合作伙伴新控制台 (第 2 节): parceiros.luniumpay.com 提供业务概览、PDF 凭证、带地址二维码及倒计时的钱包充值、通知, 以及支持角色和双重验证的团队管理。仅所有者可在控制台绑定 Telegram、修改结算钱包及 sub fee。钱包充值与余额支付 Pix 的流程图见 [USDT/USDC 钱包](#)。

• 2026年10月7日 • OpenAPI 1.43.0: USDT/USDC 钱包 (/carteira-usdt): 支持多网络充值 USDT 或 USDC、使用余额通过收款标识或 Cópia e Cola 支付 Pix、查询流水, 以及 webhook: wallet.*, 并提供测试环境模拟器。

• 2026年10月7日: cash-in 预览 (POST /cashin/preview) 在生产环境开始使用与测试环境相同的字段名: asset_amount (钱包预计收到的数量)、withdraw_fee (网络提现费用, 以资产单位表示)、withdraw_min 及 brl_per_asset (用于换算 BRL)。amount 及 network_min 仍作为旧名称保留。

• 2026年10月3日: 第 Pix 官方付款凭证 (8) 节说明凭证内容、四种形式 (pix_e2e、receipt_url、receipt_pdf_url、verify_url), 以及 API、webhook、合作伙伴控制台和 Telegram 等获取渠道。

- 通过示例核对费用与佣金, 补充访问恢复及故障场景。
- 面向 AI 的结构化资料与 OpenAPI 对齐; 未来能力须待发布并验证后使用。
- 在开头、Telegram 操作及检查清单中突出 Lunium Partners 官方社区。
- 恢复人工支持群组建议: 加入 @luniumb2b 及 @luniumb2c, 由 Lunium 或合作伙伴创建。
- 详述私密监控、事件流、自建实时控制台、各流程状态、验签、恢复与验收。
- 本次更新重点为支持与交易跟踪, 并更新引用的 OpenAPI 版本。
- 记录无需固定钱包的托管流程及 Telegram 提现。
- 区分 API 的 destino 默认值与机器人的默认值。
- 将费用及限额关联到正确查询接口, 移除固定价格与时效承诺。
- 区分本地冻结、服务商冻结及余额等待期。
- 区分入金与交付目录, 包括 entregavel 筛选, 以及提现中不支持 Liquid/DePix 的限制。
- 说明 Pix 收款人查询可能为可选或不可用, 以及 br_code 须受可用性限制。
- 将 webhook 超时修正为 10 秒, 完善稳健验签及故障恢复。

- 结合接口核对状态、字段、示例及 OpenAPI 清单。
- HTML、Markdown 及 PDF 使用同一内容生成，发布前校验接口契约。

[下载 PDF](#) • [Markdown](#) • [结构化示例](#) • [接口契约核查](#)

LuniumPay支持2026-09-19.3