



مستندات فنی پذیرندگان – نسخه ۲.۰

راهنمای اتصال به درگاه پرداخت اعتباری راتک لند

فروش بیشتر با «الان بخر، بعداً پرداخت کن»
مستند رسمی یکپارچه سازی REST API برای فروشگاه های آنلاین

فهرست مطالب

3	۱. مقدمه
4	۲. شروع به کار و فرآیند آنبوردینگ
5	۳. قرارداد کلی API
7	۴. احراز هویت و امنیت دسترسی
8	۵. بررسی امکان پرداخت اعتباری (Eligibility)
9	۶. ایجاد جلسه پرداخت (Initiate)
12	۷. صفحه پرداخت و بازگشت کاربر
14	۸. تأیید و نهایی سازی سفارش (Finalize و Confirm)
15	۹. عملیات پس از خرید (Amend و Reverse، Void)
17	۱۰. استعلام وضعیت و پروتکل ضد مغایرت
19	۱۱. ماشین وضعیت تراکنش
20	۱۲. جدول کدهای خطا
21	۱۳. راهنمای نمایش روش پرداخت در فروشگاه
22	۱۴. ملاحظات امنیتی
23	۱۵. ضمیمه: کالکشن Postman و سمپل کدها
25	۱۶. پرسش های متداول

۱. مقدمه

درگاه پرداخت اعتباری **راتکلند** به مشتریان فروشگاه شما امکان می‌دهد بدون پرداخت یکجای مبلغ سفارش، خرید خود را تکمیل کنند. این سرویس در دو روش اصلی ارائه می‌شود:

- **پرداخت اقساطی (INSTALLMENT):** خریدار فقط قسط اول را همان لحظه از طریق درگاه بانکی می‌پردازد و مابقی مبلغ در ماه‌های بعد به صورت خودکار قسط‌بندی می‌شود.
- **اعتبار باز (POSTPAID):** خریدار همان لحظه هزینه‌ای نمی‌پردازد و کل مبلغ خریدهای ماه در پایان دوره تسویه می‌شود.
- **فاینانس (FINANCING):** تسهیلات بلندمدت برای سفارش‌های با مبلغ بالا (در صورت فعال‌سازی برای پذیرنده).

تسویه مبلغ سفارش با فروشگاه طبق چرخه تسویه راتکلند انجام می‌شود و پس از نهایی‌شدن سفارش (Finalize) در صف تسویه قرار می‌گیرد. اگر مبلغ سفارش از سقف اعتبار خریدار بیشتر باشد، سامانه به صورت خودکار امکان **پرداخت ترکیبی** را فراهم می‌کند: سهم اعتباری از اعتبار کاربر و مابه‌التفاوت به صورت نقدی همان لحظه از درگاه بانکی دریافت می‌شود.

این سند مرجع رسمی پیاده‌سازی برای تیم‌های فنی پذیرندگان است و تمام سرویس‌ها، پارامترها، قواعد کسب‌وکار، کدهای خطا و پروتکل‌های الزامی را پوشش می‌دهد.

چشم‌انداز جریان

پذیرنده ابتدا توکن دسترسی می‌گیرد، امکان پرداخت اعتباری را برای مبلغ سفارش بررسی می‌کند، جلسه پرداخت می‌سازد و خریدار را به صفحه پرداخت راتکلند هدایت می‌کند. پس از بازگشت خریدار، پذیرنده سفارش را تأیید (Confirm) و نهایی (Finalize) می‌کند. در صورت عدم امکان ارائه خدمت، تراکنش برگشت می‌خورد (Reverse/Void).

۲. شروع به کار و فرآیند آنبوردینگ

اتصال به درگاه در چهار گام انجام می‌شود:

گام	عنوان	توضیح
۱	صدور کلید تست	از پنل پذیرنده، بخش «کلیدهای API»، اعتبارنامه محیط تست (Staging) بسازید: شامل <code>username</code> ، <code>client_secret</code> ، <code>client_id</code> و <code>password</code> .
۲	پیاده‌سازی	با کالکشن Postman و سمپل‌کدهای ضمیمه (بخش ۱۵) چرخه کامل را در محیط تست پیاده‌سازی کنید.
۳	دموی فنی	جلسه pre-demo و demo با کارشناسان فنی راتکلند برگزار و سناریوهای موفق/ناموفق مرور می‌شود.
۴	صدور کلید پروداکشن	اعتبارنامه پروداکشن صادر می‌شود. ثبت آی پی وب سرور در وایت لیست برای پروداکشن الزامی است.

۲.۱ اطلاعاتی که هنگام صدور کلید نیاز است

- **دامنه بازگشت (callback domain):** دامنه پایهای که آدرس بازگشت کاربر (`callbackUrl`) باید با آن منطبق باشد؛ مثلاً `shop.example.com`.
- **آی پی خروجی وب سرور:** برای وایت لیست (در پروداکشن اجباری). روی سرور خود `curl` `https://api.ipify.org` را اجرا کنید.
- **بازه مبلغ سفارش:** حداقل و حداکثر مبلغ قابل پرداخت اعتباری (ریال) که هنگام صدور کلید تنظیم می‌شود.

نگهداری از کلیدها

`password` و `client_secret` فقط یک بار هنگام صدور نمایش داده می‌شوند. آن‌ها را در مکان امن (Vault/ENV) نگه دارید و هرگز در کد فرانت‌اند یا مخزن عمومی قرار ندهید.

۳. قرارداد کلی API

۳.۱ آدرس پایه و نسخه‌بندی

تمام سرویس‌ها تحت آدرس پایه زیر و نسخه v1 ارائه می‌شوند:

```
https://{base-url}/api/v1/...
```

آدرس‌های محیط تست و پروداکشن هنگام صدور کلید جداگانه تحویل داده می‌شود.

۳.۲ قالب پاسخ‌ها

همه سرویس‌ها از یک قالب یکپارچه استفاده می‌کنند:

```
// موفق - HTTP 2xx
{ "ok": true, "data": { /* ... */ } }

// ناموفق - HTTP 4xx/5xx
{ "ok": false, "error": { "code": "INVALID_INPUT", "message": "...", "details": null } }
```

- `ok`: بولین؛ معیار اصلی موفقیت.
- `data`: بدنه پاسخ در صورت موفقیت.
- `error.code`: رشته ثابت و machine-readable – برنامه شما باید روی این مقدار سوئیچ کند، نه روی `message`.
- `error.message`: پیام فارسی قابل نمایش برای اپراتور.
- `error.details`: داده تکمیلی خطا (مثلاً وضعیت فعلی تراکنش در `currentState`).

۳.۳ واحد مبالغ

تمام مبالغ در سرویس‌ها به **ریال** هستند. در متن‌های نمایشی (title/description) سامانه به تومان تبدیل و فارسی‌سازی می‌کند.

۳.۴ روش‌های پرداخت

معنی	مقدار
پرداخت اقساطی – قسط اول آنلاین، باقی اقساط ماهانه	INSTALLMENT
اعتبار باز – تسویه یکجای خریدهای ماه در پایان دوره	POSTPAID
فاینانس بلندمدت (در صورت فعال‌بودن برای پذیرنده)	FINANCING

۴. احراز هویت و امنیت دسترسی

احراز هویت دو لایه است: (۱) توکن JWT کوتاه مدت که از سرویس توکن دریافت می‌کنید و (۲) **وایت لیست آی پی** – درخواست از آی پی ثبت نشده حتی با توکن معتبر هم رد می‌شود (403 / IP_NOT_ALLOWED).

۴.۱ دریافت توکن دسترسی

```
POST /api/v1/auth/token
Authorization: Basic base64(client_id + ":" + client_secret)
Content-Type: application/x-www-form-urlencoded

grant_type=password&scope=credit-gateway&username={username}&password={password}
```

پارامتر	مقدار
grant_type	همیشه password
scope	همیشه credit-gateway
username / password	همان مقادیری که هنگام صدور کلید دریافت کرده‌اید

پاسخ:

```
{
  "ok": true,
  "data": {
    "access_token": "eyJhbGciOiJIUzI1NiIs... ",
    "token_type": "bearer",
    "expires_in": 3600,
    "scope": "credit-gateway",
    "iat": 1750000000,
    "jti": "..."
  }
}
```

توکن با الگوریتم RS256 امضا می‌شود و ۳۶۰۰ ثانیه اعتبار دارد. در تمام سرویس‌های دیگر، توکن را در هدر Authorization: Bearer {access_token} ارسال کنید.

قواعد استفاده از توکن

توکن را کش نکنید و قبل از انقضا توکن جدید بگیرید. با انقضا، سرویس‌ها 401 / UNAUTHORIZED برمی‌گردانند؛ در این حالت یکبار توکن را تازه و درخواست را تکرار کنید.

۵. بررسی امکان پرداخت اعتباری (Eligibility)

قبل از نمایش روش پرداخت راتکلند در صفحه تسویه حساب (Checkout) یا صفحه محصول (PDP)، این سرویس را فراخوانی کنید. پاسخ **داینامیک** است و بر اساس مبلغ سفارش و طرح‌های اعتباری فعال محاسبه می‌شود.

```
GET /api/v1/credit-offers/eligibility?amount=120000000&methods=INSTALLMENT,FINANCING
Authorization: Bearer {access_token}
```

پارامتر	اجباری	توضیح
amount	بله	مبلغ سفارش به ریال
methods	خیر	فیلتر روش‌ها با جداکننده ویرگول: POSTPAID, INSTALLMENT, FINANCING – در صورت عدم ارسال، همه روش‌های فعال بررسی می‌شوند

پاسخ:

```
{
  "ok": true,
  "data": {
    "eligible": true,
    "title": "خرید اعتباری با راتکلند",
    "description": "فقط ۴,۲۳۰,۰۰۰ تومان الان بپردازید و مابقی را در ۲ قسط ماهانه بپردازید"
  }
}
```

قواعد نمایش – الزامی قراردادی

1. اگر `eligible=false` بود، روش پرداخت راتکلند **اصلاً نمایش داده نشود**.
2. مقادیر `title` و `description` را **بدون هیچ تغییری** نمایش دهید؛ هاردکد کردن متن ثابت تخلف قراردادی است.
3. با **هر تغییر مبلغ سفارش** (افزودن/حذف آیتم، کد تخفیف، هزینه ارسال) این سرویس را دوباره فراخوانی کنید.

۶. ایجاد جلسه پرداخت (Initiate)

به ازای هر خرید، یک جلسه پرداخت جداگانه بسازید. خروجی این سرویس، شناسه مرجع پرداخت (paymentReference) و آدرس صفحه پرداخت (checkoutUrl) است که خریدار را به آن هدایت می‌کنید.

```
POST /api/v1/credit-payments/initiate
Authorization: Bearer {access_token}
Content-Type: application/json
```

```
{
  "amount": 120000000,
  "discountAmount": 0,
  "otherSourceAmount": 0,
  "mobile": "+989123456789",
  "callbackUrl": "https://shop.example.com/ratek/callback",
  "merchantOrderId": "order-58291",
  "allowedPaymentMethods": ["INSTALLMENT"],
  "orderLines": [
    {
      "lineId": 1,
      "shippingIncluded": true,
      "taxIncluded": true,
      "shippingFee": 0,
      "tax": 0,
      "lineTotal": 120000000,
      "items": [
        {
          "id": 101,
          "name": "گوشی موبایل",
          "category": "کالای دیجیتال",
          "count": 1,
          "amount": 120000000,
          "feeCategory": 100
        }
      ]
    }
  ]
}
```

جدول ۱ – پارامترهای سطح سفارش

پارامتر	اجباری	توضیح
amount	بله	جمع کل سفارش به ریال – باید با فرمول بخش ۶.۱ مطابقت کند
discountAmount	بله	تخفیف سفارش (ریال)؛ در صورت نبودن 0

mobile	بله	موبایل خریدار با الگوی +98xxxxxxxxx – باید متعلق به کاربر فعال راتکلند باشد
callbackUrl	بله	آدرس بازگشت (HTTPS)؛ دامنه باید با دامنه ثبت شده در پنل منطبق باشد
merchantOrderId	بله	شناسه یکتای سفارش در سیستم شما (حداقل ۵ کاراکتر)؛ تکراری → 409 DUPLICATE_ORDER
allowedPaymentMethods	خیر	محدودسازی روش های قابل انتخاب برای خریدار؛ نیازمند فعال سازی سمت راتکلند
orderLines	بله	آرایه سطرهاى سفارش (حداقل یک سطر)

جدول ۲ – پارامترهای هر سطر سفارش (orderLine) و آیتمها

پارامتر	اجباری	توضیح
lineId	بله	شناسه عددی سطر (سبد) در سیستم شما
items	بله	اقلام سطر: amount، count (≥1)، category، name، id (ریال)، feeCategory (پیش فرض ۱۰۰)
shippingIncluded	بله	اگر false باشد، shippingFee به مبلغ سطر اضافه می شود
taxIncluded	بله	اگر false باشد، tax به مبلغ سطر اضافه می شود
shippingFee / tax	بله	ریال
lineTotal	بله	جمع مبلغ سطر طبق فرمول بخش ۶.۱

۶.۱ اعتبارسنجی مبالغ (سمت سرور کنترل می شود)

```
// هر سطر:
lineTotal = Σ(count × amount) + shippingFee (اگر shippingIncluded=false) + tax (اگر taxIncluded=false)

// کل سفارش:
amount = Σ lineTotal - discountAmount - otherSourceAmount
```

هرگونه ناهماهنگی با INVALID_INPUT / 400 رد می شود.

پاسخ موفق:

```
{
  "ok": true,
  "data": {
    "paymentReference": "rlcp_51939678ffed43d8b01e1371406efce1",
    "checkoutUrl": "https://{base-url}/checkout/rlcp_51939678ffed43d8b01e1371406efce1"
  }
}
```

نگهداشت **paymentReference**

مقدار **paymentReference** را در دیتابیس خود و در کنار سفارش ذخیره کنید؛ در تمام مراحل بعدی (**confirm**، **finalize**، **reverse**، **void**، **amend**، **status**) همین شناسه استفاده می‌شود.

۷. صفحه پرداخت و بازگشت کاربر

خریدار را به `checkoutUrl` هدایت کنید. در صفحه پرداخت راتکلند این مراحل طی می‌شود:

1. احراز هویت خریدار با موبایل (رمز یک بار مصرف).
 2. نمایش طرح‌های اعتباری فعال خریدار و انتخاب روش پرداخت.
 3. پرداخت قسط اول (و مابه‌التفاوت نقدی در صورت پرداخت ترکیبی) از طریق درگاه بانکی.
- پس از پایان، خریدار با **POST یک فرم** به `callbackUrl` شما برمی‌گردد:

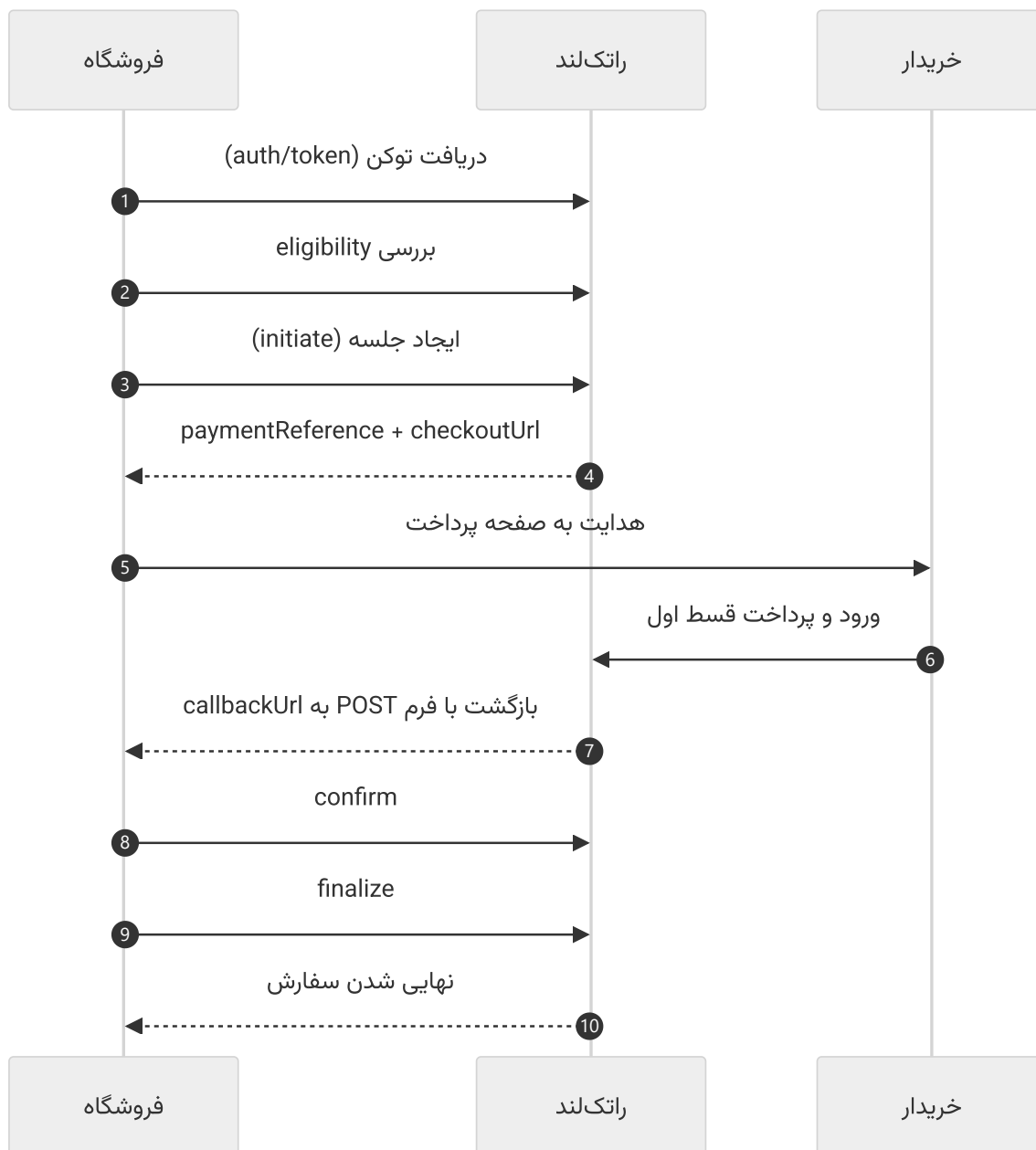
جدول ۳ – پارامترهای فرم بازگشت (`application/x-www-form-urlencoded`)

پارامتر	مقادیر	توضیح
<code>merchantOrderId</code>	همان مقدار ارسالی شما	برای یافتن سفارش در سیستم خود
<code>state</code>	<code>SUCCESS</code> / <code>FAILED</code>	<code>SUCCESS</code> → فراخوانی <code>confirm</code> ؛ <code>FAILED</code> → فراخوانی <code>reverse</code> و باطل کردن سفارش
<code>amount</code>	ریال	مبلغ تراکنش – با مبلغ سفارش خود تطبیق دهید

نکته امنیتی مهم

به محتوای فرم بازگشت به‌تنهایی اعتماد نکنید؛ پس از دریافت `callback` حتماً از طریق API (`confirm` یا `status`) نتیجه واقعی تراکنش را از سرور راتکلند استعلام کنید. صفحه `callback` شما باید درخواست‌های POST را بدون نیاز به نشست کاربر بپذیرد (کاربر ممکن است کوکی فروشگاه را نداشته باشد).

۷.۱ تصویر کلی جریان



۸. تأیید و نهایی‌سازی سفارش (Finalize و Confirm)

۸.۱ Confirm – تأیید اولیه

بلافاصله پس از بازگشت خریدار با `state=SUCCESS` و اطمینان از امکان ارائه کالا/خدمت، این سرویس را فراخوانی کنید:

```
POST /api/v1/credit-payments/confirm
{ "paymentReference": "rlcp_..." }
```

- فقط **یک بار** فراخوانی شود – حتی اگر callback چند بار برسد (idempotent نیست؛ تکرار آن / 400 `INVALID_STATE` می‌دهد).
- مهلت فراخوانی محدود است (پیش‌فرض ۱۵ دقیقه). در صورت انقضا، مبلغ خریدار **خودکار** برمی‌گردد و تراکنش `REVERSED` می‌شود.
- اگر امکان ارائه خدمت نیست، به جای `confirm` سرویس **reverse** را بنزید.

پاسخ موفق:

```
{ "ok": true, "data": { "merchantOrderId": "order-58291" } }
```

۸.۲ Finalize – نهایی‌سازی قطعی (الزامی)

پس از `confirm` موفق، **حتماً** `finalize` را فراخوانی کنید:

```
POST /api/v1/credit-payments/finalize
{ "paymentReference": "rlcp_..." }
```

- در این نقطه طرح اقساطی/اعتباری خریدار ساخته می‌شود و سفارش وارد صف تسویه پذیرنده می‌شود.
- پس از `finalize`، سفارش فقط با **void** قابل لغو است.
- اگر `finalize` را فراموش کنید، سامانه پس از حدود **۲.۵ ساعت** خودکار نهایی می‌کند تا تسویه شما معطل نماند – با این حال اتکا به مکانیزم خودکار مجاز نیست.

۹. عملیات پس از خرید

۹.۱ Reverse – برگشت قبل از نهایی‌سازی

```
POST /api/v1/credit-payments/reverse
{ "paymentReference": "rlcp_..." }
```

- کاربرد: پرداخت موفق بوده اما خدمت قابل ارائه نیست (مثلاً اتمام موجودی)، یا callback با state=FAILED رسیده است.
- فقط روی وضعیت‌های PENDING و CONFIRMED ممکن است؛ روی سفارش نهایی‌شده باید از void استفاده شود.
- اثر: برگشت کامل مبلغ پرداخت‌شده به کیف پول خریدار و آزادسازی اعتبار او.
- این سرویس idempotent است؛ تکرار آن خطا نمی‌دهد.

۹.۲ Void – ابطال سفارش نهایی‌شده (الزامی پیاده‌سازی)

```
POST /api/v1/credit-payments/void
{ "paymentReference": "rlcp_..." }
```

برگشت ناپذیر

void فقط روی سفارش‌های FINALIZED قابل اجراست و **غیرقابل بازگشت** است. قبل از فراخوانی، تأییدیه مدیر سیستم (admin confirmation) را در فرآیند داخلی خود الزامی کنید.

- اثر: ابطال طرح اقساطی، برگشت اقساط پرداخت‌شده به کیف پول خریدار، و اصلاح/ابطال تسویه پذیرنده.

۹.۳ Amend – اصلاح مبلغ سفارش (فقط کاهشی)

```
POST /api/v1/credit-payments/amend
Content-Type: application/json
```

بدنه مانند initiate است به‌علاوه paymentReference (بدون callbackUrl، mobile و merchantOrderId):

```
{
  "paymentReference": "rlcp...",
  "amount": 100000000,
  "discountAmount": 0,
  "otherSourceAmount": 0,
  "orderLines": [ /* ساختار کامل جدید */ ]
}
```

- فقط روی سفارش‌های FINALIZED .
- مبلغ جدید باید **کمتر یا مساوی** مبلغ قبلی باشد؛ افزایش مبلغ → 409 / AMOUNT_NOT_REDUCIBLE .
- مبلغ جدید نمی‌تواند کمتر از مبلغ نقدی پرداخت‌شده توسط خریدار باشد.
- آیتمی که کاملاً حذف می‌شود باید از items هم حذف شود.
- اثر: بازمحاسبه اقساط باقی‌مانده و اصلاح مبلغ تسویه پذیرنده. این عملیات نیز برگشت‌ناپذیر است.

۱۰. استعلام وضعیت و پروتکل ضد مغایرت

```
GET /api/v1/credit-payments/status?paymentReference=rlcp_...
Authorization: Bearer {access_token}
```

پاسخ:

```
{
  "ok": true,
  "data": {
    "merchantOrderId": "order-58291",
    "state": "FINALIZED",
    "amount": 120000000
  }
}
```

جدول ۴ – وضعیت‌های چرخه عمر تراکنش

وضعیت	معنی	عملیات مجاز بعدی
PENDING	جلسه ساخته شده؛ در انتظار پرداخت/تأیید	confirm یا reverse
CONFIRMED	توسط پذیرنده تأیید شده	finalize یا reverse
FINALIZED	نهایی و ثبت‌شده در صف تسویه	void یا amend
REVERSED	برگشت‌خورده (نهایی)	—
VOIDED	ابطال‌شده پس از نهایی (نهایی)	—

۱۰.۱ پروتکل مدیریت خطا (الزامی)

مدیریت Confirm:

- confirm را با تایم‌اوت ۳۰ ثانیه فراخوانی کنید.
- در صورت timeout یا خطای ۵۰۰، یک‌بار با فاصله چند ثانیه retry کنید.
- اگر نتیجه همچنان نامشخص بود، status را بزنید و بر اساس وضعیت تصمیم بگیرید:
 - CONFIRMED → finalize کنید.
 - PENDING → مجدد confirm کنید.
 - FINALIZED → سفارش را نهایی بدانید (اگر سفارش سمت شما ناموفق بود، void کنید).
 - REVERSED → تراکنش نهایی شده؛ سفارش را لغو کنید.

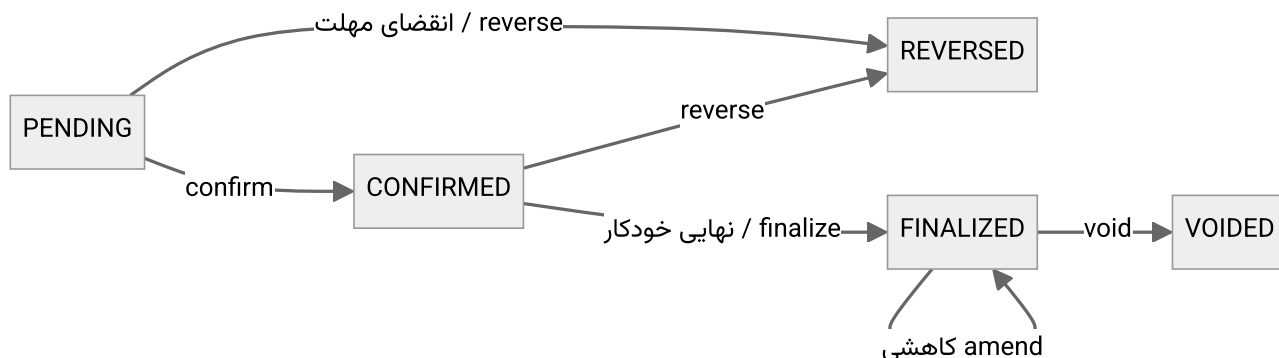
4. اگر status هم خطا داد: ۴۰ دقیقه صبر، سپس مجدد status؛ در نهایت تراکنش را «نامشخص» علامت بزنید تا کارشناس از پنل پیگیری کند.

مدیریت Finalize: در صورت خطا یا عدم پاسخ، status را بزنید: retry finalize → CONFIRMED ؛ FINALIZED → نهایی.

دستیار تصمیم

در پاسخ خطای INVALID_STATE ، وضعیت فعلی تراکنش در error.details.currentState برگردانده می‌شود تا بدون فراخوانی اضافه status تصمیم بگیرید.

۱۱. ماشین وضعیت تراکنش



- **VOIDED** و **REVERSED** وضعیت‌های پایانی هستند.
- انقضای مهلت **confirm** (پیش‌فرض ۱۵ دقیقه) تراکنش را خودکار **REVERSED** می‌کند و مبلغ به خریدار برمی‌گردد.
- تراکنش **CONFIRMED** ای که **finalize** نشود، پس از حدود ۲.۵ ساعت خودکار **FINALIZED** می‌شود.
- درخواست هم‌زمان روی یک تراکنش در حال پردازش → **REQUEST_IN_PROGRESS** / 429 ؛ چند ثانیه بعد **retry** کنید.

۱۲. جدول کدهای خطا

جدول ۵ - کدهای خطا و اقدام متناظر

سرویس	HTTP	error.code	دلیل	اقدام پذیرنده
initiate	500	INTERNAL_ERROR	خطای داخلی/شبکه	retry با فاصله زمانی
initiate	400	INVALID_MOBILE	فرمت موبایل اشتباه	اصلاح موبایل، بدون retry
initiate	409	DUPLICATE_ORDER	merchantOrderId تکراری	شناسه جدید، بدون retry
initiate	403	CALLBACK_URL_NOT_ALLOWED	دامنه callbackUrl ثبت نشده	اطلاع به پشتیبانی فنی
confirm / finalize / reverse / void / amend	400	INVALID_STATE	وضعیت تراکنش نامناسب	بر اساس details.currentState تصمیم بگیرید
confirm / reverse / void / amend	429	REQUEST_IN_PROGRESS	درخواست همزمان در حال پردازش	retry با چند ثانیه تأخیر
همه	400	MERCHANT_MISMATCH	paymentReference متعلق به پذیرنده دیگر	وضعیت «نامشخص» + پیگیری دستی
همه	404	PAYMENT_NOT_FOUND	paymentReference اشتباه	بررسی دستی
amend	404	ORDER_NOT_FOUND	سفارشی برای تراکنش یافت نشد	بررسی دستی
amend	409	AMOUNT_NOT_REDUCIBLE	مبلغ جدید باید $\geq$ مبلغ قبلی باشد	اصلاح مبلغ
همه	400	INVALID_INPUT	پارامتر نامعتبر/خالی	اصلاح بدنه و retry
همه	401	UNAUTHORIZED	توکن نامعتبر/منقضی	توکن جدید بگیرید
همه	403	IP_NOT_ALLOWED	آی پی در وایت لیست نیست	آی پی صحیح را در پنل ثبت کنید

۱۳. راهنمای نمایش روش پرداخت در فروشگاه

- در صفحه تسویه حساب، روش پرداخت راتکلند را با `title` و `description` دریافتی از `eligibility` نمایش دهید.
- لوگوی راتکلند را در کنار عنوان قرار دهید (فایل‌های برند در پکیج ضمیمه موجود است؛ در ابعاد موبایل و دسکتاپ از نسخه متناسب استفاده کنید).
- در صورت `eligible=false` گزینه را مخفی کنید – نمایش غیرفعال (`disabled`) مجاز نیست.
- متن داینامیک را هاردکد نکنید؛ مبالغ اقساط با تغییر مبلغ سفارش تغییر می‌کند.
- پس از انتخاب روش توسط کاربر و قبل از هدایت به درگاه، `initiate` را بزنید و بلافاصله کاربر را به `checkoutUrl` ببرید؛ از باز کردن آن در `iframe` جداً خودداری کنید.

۱۴. ملاحظات امنیتی

1. **وایت لیست آی پی:** همه فراخوانی‌های سرور به سرور فقط از آی پی‌های ثبت شده پذیرفته می‌شود.
2. **دامنه بازگشت:** `callbackUrl` فقط روی دامنه ثبت شده و زیردامنه‌های آن پذیرفته می‌شود و HTTPS الزامی است.
3. **JWT با RS256:** توکن‌ها با کلید خصوصی RSA امضا و با انقضای ۳۶۰۰ ثانیه صادر می‌شوند.
4. **عدم اعتماد به callback مرورگر:** نتیجه نهایی را همیشه سرور به سرور (confirm/status) استعلام کنید.
5. **Idempotency:** یکتایی `merchantOrderId` سمت سرور تضمین می‌شود؛ از ارسال دوباره initiate با همان شناسه خودداری کنید.
6. **قفل هم‌زمانی:** عملیات حساس (confirm/finalize/void/amend) روی یک تراکنش به صورت اتمیک قفل می‌شوند.
7. **لاگ کامل:** تمام فراخوانی‌های API شما (زمان، آی پی، وضعیت پاسخ) برای حسابرسی ثبت می‌شود.

۱۵. ضمیمه: کالکشن Postman و سمپل‌کدها

پکیج اتصال شامل این سند به‌علاوه فایل‌های زیر است:

فایل	توضیح
RatekLend-CreditGateway.postman_collection.json	کالکشن آماده Postman شامل هر ۹ سرویس؛ اسکریپت pre-request خودکار ساخت هدر Basic و ذخیره jwt و paymentReference در متغیرهای کالکشن
sample-code/nodejs.js	کلاينت کامل ۱۸ Node.js+ شامل چرخه کامل و مدیریت امن confirm با پروتکل ضد مغایرت
sample-code/php.php	نمونه (cURL) PHP
sample-code/csharp.cs	کلاينت 6 .NET / C#
sample-code/curl.sh	اسکریپت cURL برای تست سریع همه سرویس‌ها

۱۵.۱ متغیرهای محیط کالکشن Postman

متغیر	مقدار
url	آدرس پایه محیط (تست/پروداکشن) – بدون اسلش پایانی
client-id / client-secret	از پنل پذیرنده، بخش کلیدهای API
username / password	همان‌طور که هنگام صدور کلید نمایش داده شد
amount	مبلغ تست به ریال
mobile	موبایل کاربر تست با پیشوند +98
callback-url	آدرس بازگشت تستی روی دامنه ثبت‌شده

ترتیب اجرای پیشنهادی در Postman:

1. Authentication → توکن در متغیر jwt ذخیره می‌شود.
2. Eligibility → بررسی امکان پرداخت.
3. Initiate Payment → ساخت جلسه؛ paymentReference خودکار ذخیره و checkoutUrl در کنسول چاپ می‌شود.

4. باز کردن checkoutUrl در مرورگر و تکمیل پرداخت تستی.

5. Confirm و سپس Finalize.

۱۵.۲ نمونهٔ حداقلی cURL

```
# ۱) توکن
BASIC=$(printf '%s:%s' "$CLIENT_ID" "$CLIENT_SECRET" | base64)
curl -X POST "$BASE_URL/api/v1/auth/token" \
  -H "Authorization: Basic $BASIC" \
  -H 'Content-Type: application/x-www-form-urlencoded' \
  --data-urlencode 'grant_type=password' \
  --data-urlencode 'scope=credit-gateway' \
  --data-urlencode "username=$USERNAME" \
  --data-urlencode "password=$PASSWORD"

# ۲) ایجاد جلسهٔ پرداخت
curl -X POST "$BASE_URL/api/v1/credit-payments/initiate" \
  -H "Authorization: Bearer $JWT" \
  -H 'Content-Type: application/json' \
  -d '{"amount":120000000,"discountAmount":0,"otherSourceAmount":0,
    "mobile":"+989123456789",
    "callbackUrl":"https://shop.example.com/ratek/callback",
    "merchantOrderId":"order-58291",
    "orderLines":[{"lineId":1,"shippingIncluded":true,"taxIncluded":true,
    "shippingFee":0,"tax":0,"lineTotal":120000000,
    "items":[{"id":101,"name":"گوشی موبایل","category":"digital",
    "count":1,"amount":120000000,"feeCategory":100}]}]}'
```

۱۶. پرسش‌های متداول

خطای IP_NOT_ALLOWED می‌گیرم. آی‌پی خروجی وب‌سرور خود را با `curl https://api.ipify.org` از روی همان سرور پیدا و در پنل پذیرنده ثبت کنید. توجه: آی‌پی سرور توسعه با آی‌پی پروداکشن متفاوت است.

خطای UNAUTHORIZED می‌گیرم. انکد `base64(client_id:client_secret)` را بررسی کنید (بدون newline اضافه)، صحت `username/password` را چک کنید، و اگر توکن بیش از یک ساعت پیش گرفته‌اید، توکن جدید بگیرید.

پاسخ eligible=false می‌گیرم. مبلغ سفارش خارج از بازه مجاز کلید شماسست یا طرح اعتباری فعالی برای آن مبلغ وجود ندارد. با پشتیبانی فنی بازه مبلغ را بازبینی کنید.

خطای DUPLICATE_ORDER می‌گیرم. `merchantOrderId` قبلاً استفاده شده است. برای هر تلاش پرداخت شناسه جدید بسازید (مثلاً پسوند عدد تلاش: `order-58291-2`).

خریدار پرداخت کرد ولی callback به سرورم نرسید. وضعیت را با سرویس `status` استعلام کنید؛ اگر `PENDING` بود و مهلت `confirm` در حال گذر است، سناریوی «نامشخص» پروتکل بخش ۱۰ را اجرا کنید.

آیا می‌توان چند سبد (cart) در یک سفارش داشت؟ بله؛ هر سطر `orderLines` معادل یک سبد است و `amount` باید مجموع همه سطرها منهای تخفیف و سایر منابع باشد.

پرداخت ترکیبی چیست؟ اگر اعتبار خریدار کمتر از مبلغ سفارش باشد، سامانه سهم اعتباری را از اعتبار او و مابه‌التفاوت را نقدی همان لحظه از درگاه می‌گیرد؛ برای شما به‌عنوان پذیرنده فرآیند تغییری نمی‌کند و تسویه کامل انجام می‌شود.

پشتیبانی فنی

تیم اتصال پذیرندگان راتک‌لند – از طریق پنل پذیرنده، بخش پشتیبانی، یا ایمیلی که هنگام صدور کلید دریافت کرده‌اید. در گزارش خطا حتماً `paymentReference` و زمان تقریبی فراخوانی را ذکر کنید.