

Partner Integration Guideline

[CONFIDENTIAL]

Table of Contents

[README](#)

[Table of Contents](#)

[1. Handling API Payment: Client Timeout \(not RC 23\)](#)

[Problem Statement Summary](#)

[Issue Handling Summary](#)

[State Machine Diagram](#)

[Scenario Listing](#)

[2. Handle API Payment: Duplicate Order ID](#)

[Problem Statement Summary](#)

[Issue Handling Summary](#)

[State Machine Diagram](#)

[3. Handle API Payment: Temporarily Closed](#)

[Problem Statement Summary](#)

[Issue Handling Summary](#)

[4. Handle API Payment: Invalid](#)

[5. Handle API Payment: Insufficient Deposit](#)

[Problem Statement Summary](#)

[Issue Handling Summary](#)

[State Machine Diagram](#)

[6. Handle API Payment: 406](#)

[Problem Statement Summary](#)

[Issue Handling Summary](#)

[7. Handle any API: unknown error \(5xx, network timeout, or any\)](#)

[Problem Statement Summary](#)

[Issue Handling Summary](#)

1. Handling API Payment: Client Timeout (not RC 23)

Problem Statement Summary

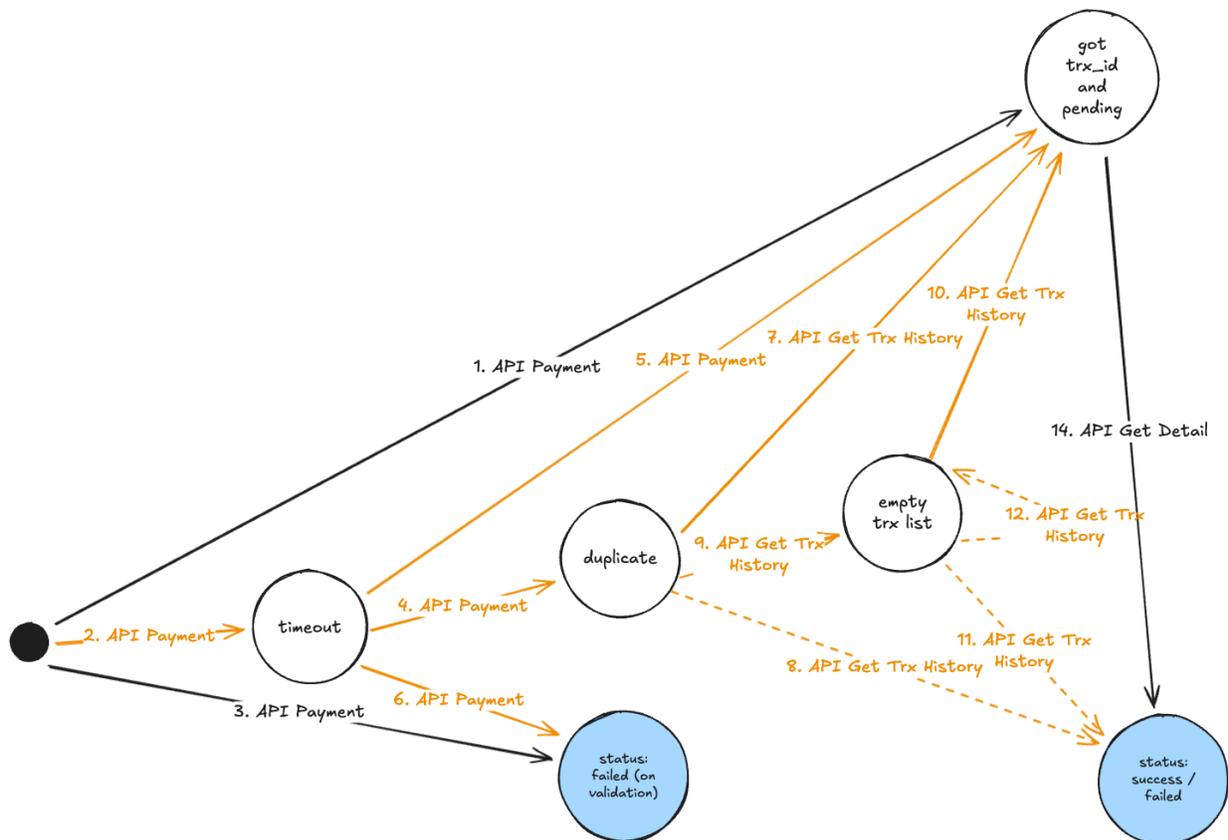
Terdapat proses yang harus dijalankan secara sekuensial antara satu transaksi dengan transaksi yang lain untuk satu partner. Jika partner tertentu melakukan request pembuatan transaksi yang tinggi dari biasanya dan adanya anomali aktivitas di database, hal ini sangat mungkin terjadi.

Issue Handling Summary

Berlaku juga secara umum, jika hit API mendapatkan timeout atau no response agar melakukan hit ulang API tersebut.

Setelah API Payment dihit ulang, partner kemungkinan tidak mendapatkan transaction_id. Partner perlu menembak API Get Trx History untuk mendapatkan transaction_id

State Machine Diagram



Scenario Listing

No	Precondition	Action	Possible Result	Suggested Status for Partner
1		Hit API Payment	Got transaction_id	pending
2		Hit API Payment	Timeout / client timeout / no response	pending
3		Hit API Payment	Status failed on validation	failed
4	API Payment got timeout / no response	Hit API Payment	Duplicate 422 Duplicate Order ID : ORDER-001	pending
5	API Payment got timeout / no response	Hit API Payment	Got transaction_id	pending
6	API Payment got timeout / no response	Hit API Payment	Status failed on validation	failed
7	API Payment got duplicate	Hit API Get Trx History by partner's order_id	Got transaction_id	pending
8	API Payment got duplicate	Hit API Get Trx History by partner's order_id	Status success / failed	Success / failed ~ not preferred path
9	API Payment got duplicate	Hit API Get Trx History by partner's order_id	Empty trx list	pending
10	API Get Trx History got empty trx list	Hit API Get Trx History by partner's order_id	Got transaction_id	pending
11	API Get Trx History got empty trx list	Hit API Get Trx History by partner's order_id	Status success / failed	Success / failed ~ not preferred path
12	API Get Trx History got empty trx list	Hit API Get Trx History by partner's order_id	Empty trx list	pending
13	Status pending	Hit API Get Detail	Status pending	pending
14	Status pending	Hit API Get Detail	Status success / failed	Success / failed

2. Handle API Payment: Duplicate Order ID

Problem Statement Summary

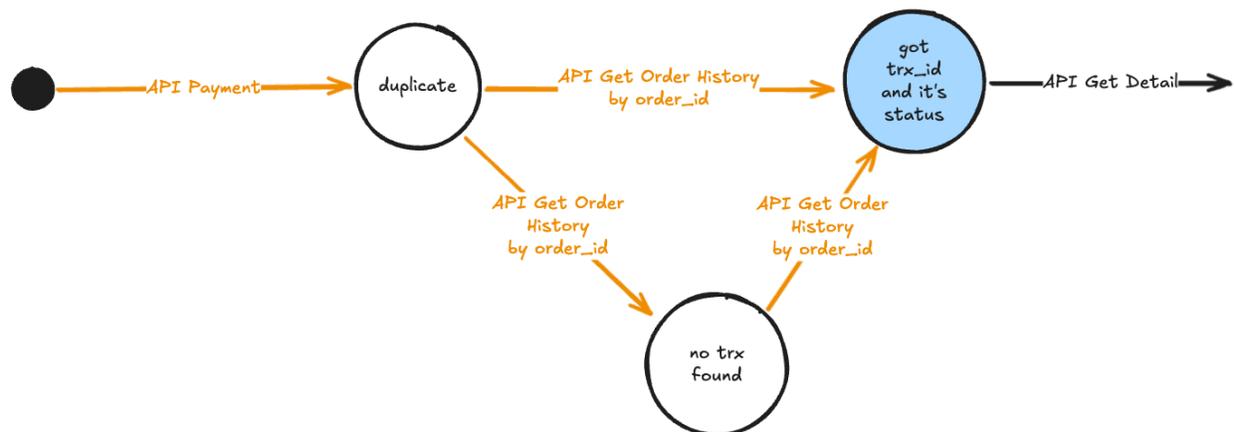
Secara umum, partner bisa mengirimkan order id / transaction id milik partner sebagai referensi ketika pembuatan transaksi ke Alterra. Order ID yang dikirim harus unik 1 x 24 jam, untuk setiap transaksi yang berhasil terbuat.

Duplicate Order ID adalah respon dari API Payment ketika partner membuat transaksi dengan order ID yang sama untuk transaksi yang sudah berhasil dibuat atau sedang dibuat kurang dari 24 jam sejak transaksi terakhir dibuat.

Issue Handling Summary

Partner yang mendapati balikan duplicate order ID tidak boleh membatalkan transaksi di sisi partner. Partner agar menembak API Get Trx History menggunakan order_id secara berkala hingga mendapatkan balikan transaction_id (Proses menembak dapat menggunakan proses exponential backoff atau menggunakan interval konstan, jika issue masih terjadi dapat menghubungi Alterra dan menandai status transaksi menjadi pending). Transaction_id tersebut yang akan digunakan untuk menembak API Get Transaction Detail yang membutuhkan transaction_id Alterra.

State Machine Diagram



3. Handle API Payment: Temporarily Closed

Problem Statement Summary

Related to product closing on partner package and master product. *From tech, there is potential indication of a relationship with smartbiller, subject for further checking.*

When API Payment returns insufficient deposit, no transaction is yet to be created and order_id is yet to be reserved.

Issue Handling Summary

Subject for further checking

4. Handle API Payment: Invalid

There are two types invalid on API Payment: Invalid Authentication and Invalid Product ID.

Invalid Authentication is related to authentication. Partner is suggested to engage with customer care immediately to prevent unexpected security issues.

Invalid Product ID is related to product mapping that is mismatched with the registered product on Alterra. Partner is suggested to engage with customer care immediately.

When API Payment returns invalid, no transaction is yet to be created and order_id is yet to be reserved.

5. Handle API Payment: Insufficient Deposit

Problem Statement Summary

Alterra will hold an amount of partner deposit up front transaction creation.

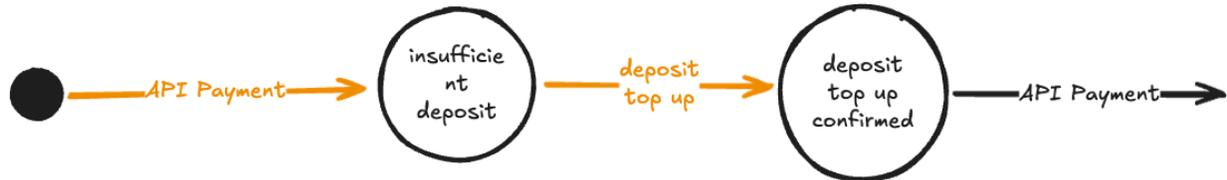
Insufficient deposit occurs when a partner is attempting to create a transaction with insufficient partner's deposit.

When API Payment returns insufficient deposit, no transaction is yet to be created and order_id is yet to be reserved.

Issue Handling Summary

Partner is safe to retry hitting API Payment using the same order_id after deposit top up has been confirmed.

State Machine Diagram



6. Handle API Payment: 406

Problem Statement Summary

HTTP Error Code 406 typically occurs when a partner sends any request that doesn't comply with our request schema. It is not related to transaction validation on the biller side, the 406 is returned only if the request does not comply with Alterra's request schema.

This error is not expected to happen in production! It means, there is a possibility that the partner modified the existing implementation after UAT with Alterra.

When the error occurs on API Payment, no transaction is yet to be created and order_id is yet to be reserved. Thus, the partner can cancel the transaction.

Issue Handling Summary

Partner needs to refer to the latest UAT and revert the modified code into the running one.

7. Handle any API: unknown error (5xx, network timeout, or any)

Notes: This is different from RC 23 (connection timeout)

Problem Statement Summary

Even though Alterra has identified all possible edge cases, there will be a time when our system goes unexpectedly. Once, a partner reporting HTTP Code 5xx returned.

Issue Handling Summary

Use either an exponential backoff strategy (e.g., 2s, 4s, 6s, ...) or fixed intervals for retry attempts at History API. If the issue remains unresolved, mark the transaction as pending and notify Alterra concurrently.

8. Handle API Get Detail: 404 not found

~ need further development and checking, consider remapping to 406 instead and **put the transaction to pending (subject for manual update on partner side, subject for integration improvement)**

Problem Summary

Partner passes invalid transaction id (possibly null, nil, 0, etc).

Issue Handling Summary

Use either an exponential backoff strategy (e.g., 2s, 4s, 6s, ...) or fixed intervals for retry attempts at History API. If the issue remains unresolved, mark the transaction as pending and notify Alterra concurrently.