

Online Integration
Partner Certification

Introduction

In order to facilitate a payment workflow on the Zip NZ platform, merchant websites must integrate with the Zip NZ API in order to initiate the workflow. This document describes a typical integration workflow for any merchant website.

All requests to the Zip NZ API are to be made from the Merchant Server (i.e. server to server communication).

# API Documentation

Documentation for the Zip NZ API can be found at <https://docs-nz.zip.co/>.

# Authentication

All calls to the Zip NZ API need to be authenticated. Zip NZ uses OAuth (Client Credentials Flow) to authorise requests to the API.

# Payment Flow

The below diagram shows the flow between the merchant site and Zip NZ for a typical payment process.



Each of these steps is described in more detail below:

1. **User selects Zip NZ**
This is typically offered as a payment option for users when they are at the stage of checkout on the merchant site.
2. **Merchant Server POSTs to Zip NZ API**
The merchant server creates an API call to the Zip NZ ‘Create Order’ endpoint. This contains:
	* Customer details (email, name, phone number)
	* Order details (what the customer is purchasing from the merchant site)
	* Billing and Shipping addresses
	* Amount (plus optional information regarding discounts / shipping / tax)
	* RedirectUrl - this is a merchant hosted URL, to return the customer back to the merchant once they have completed the Zip payment process.
3. **Zip NZ API returns a token and a Redirect URL**The Zip NZ API returns a unique token for the order and the URL to redirect the user to.
4. **Redirect User to Zip payment page**
Redirect the user to the URL returned in step 2 from the API
5. **User Completes Payment Steps at Zip**
User completes their payment information on Zip site. This will result in a user being able to pay via Zip NZ or not.
6. **User is returned to the Merchant page when complete**
Once the user has completed the Zip process, they are redirected to the Merchant page. When the user is returned to the merchant page, the URL will have the following parameters appended to the URL:
	* paymentStatus indicates the status of the payment. Possible values are:
		+ success: Payment via Zip completed.
		+ cancelled: User cancelled out of Zip process and returned to merchant site.
	* orderId used to uniquely identify the order
7. **(Optional) Merchant Server queries Zip NZ API for order information**This can be used to reconcile any orders that have started, but users have not returned to the merchant site, for whatever reason.
8. **Merchant site reconciles payment**On receipt of a payment status (successful or otherwise), it is expected the merchant server will reconcile the order status.

#

# Refund Flow

In the event of a refund (partial or full), Zip NZ provides an endpoint to initiate the refund of monies to the customer.

The refund API endpoint can be called with a (Zip returned) orderId or the unique merchant reference. It is also designed to be idempotent, in the event of any transient failure. For this reason, a requestId must be supplied from the merchant server for each refund that the merchant is trying to perform.

# Acceptance Criteria

### Payment Method at Checkout

* User will be able to select Zip NZ as a Payment Method at checkout

### Order Creation

* Upon proceeding with an order with Zip NZ as the Payment Method, an order will be created
* Once the user proceeds with Zip NZ, the user will be redirected to the Zip checkout site whereby the user will be prompted to enter their e-mail address
* The merchantReference for the created order should match the Order Number/Reference on the eCommerce platform

### Order Confirmation

* When a user has successfully completed the Zip checkout, they should be redirected to the Merchant’s Order Confirmed page

### Order Abandoned

* When a user manually exits from the Zip checkout page without completing the transaction, either by clicking “Return to Merchant” or “Exit” on the top right corner - they should be redirected back to the Merchant’s checkout page

### Refund

* The admin user should be able to refund any successful Zip order - either for the full amount or a partial amount of the order

# Checklist

## Overview

Each of the items in the below document should be completed for a successful integration. If any items remain unclear, please reach out to our team and refer to the relevant sections of the API docs where all these items are outlined:

## Test Results

### Onboarding

|  |  |
| --- | --- |
| **Use Case** | **Completed** |
| Zip Client ID & Client Secret is configurable |   |
| Zip API Endpoints are configurable (To allow for Sandbox & Production Environment) |  |

Notes

### Miscellaneous

Once the user is redirected to the Zip payment page, there are several possible outcomes for the user, and each must be handled appropriately. The below are the non-transactional outcomes at the checkout stage:

|  |  |
| --- | --- |
| **Use Case** | **Completed** |
| Closed Zip checkout page |   |
| Timed out/Expired Zip order (60 mins) |   |
| Pressed “X” button |   |

Notes

### Transactions

There are a number of possibilities when transacting with Zip at checkout, from existing customers to new customers, as well as those applications that do not provide an instant result. There are also several scenarios that require different items or values to be passed in the API calls.

These will need to match our spec for their respective scenarios to achieve certification. Each of these are listed below and must be handled for a complete integration.

|  |  |
| --- | --- |
| **Use Case** | **Completed** |
| Checkout with existing Zip Customer |   |
| Checkout with New Approved Zip Customer |   |
| Checkout with Declined Zip Customer |   |
| Exception: POST with a Duplicate Merchant Reference |  |
| Customer is redirected to a success page on order completion |  |
| Customer is redirected back to the payment page on order cancellation/failure |  |
| Exception: Handling of customer force quit/network/system failure during payment |  |
| **Refund (Recommended)** |
| Operator can select a historical order and choose to refund it |  |
| Exception: Operator attempts to refund more than the order value |  |
| Operator can select the amount they wish to refund (partial/full refund) |  |
| Operator can process multiple partial refunds on an order |  |
| **Transaction Data** |
| Transaction data appears correctly on Zip portal side - API call contains as much data as possible (shipping/billing address, name, amount, unique order id, etc) |   |
| Ensure the right item type (name, price, sku, quantity) is used for item details |   |
| Discount items - If gift card, voucher, store credit or coupons are accepted at checkout please make sure you can use it in conjunction with Zip |   |
| **Order/Payment Management** |
| Full capture is supported (Auth payment flow) |   |
| Acknowledge authorised orders if a delayed capture is required (Auth payment flow) |  |
| Handling authorised orders that are voided after 15min expiry period (Auth payment flow) |  |
| Partial and full refunds are supported |   |
| Ability to cancel/void authorised payments (Auth payment flow) |   |
| Reconcile orders that may not have reached an end state using /GET order |  |
| **Other** |
| Exception: Platform is configured with invalid credentials |  |
| (Optional) Zip Order Number should be presented in the customers receipt/invoice |  |
| Log all API calls (request, response, etc.) |  |
| (Recommended) Idempotency-Key is present in the header for transactional API requests (/order, /capture, /refund)  |  |
| **Lightbox/Embedded** |
| Listen to the PostMessage and handle iFrame accordingly |  |
| Mobile version should redirect to a full Zip payment page |  |

###

### Product awareness and conversion optimisation

As a part of all Zip integrations, key elements have been identified as vital to increased checkout and conversion rates. As such, as a part of our partnerships, we include this as a requirement when integrating Zip as a payment method. There can be a number of pieces that are specifically called out in Zip agreements that are listed as below:

|  |  |
| --- | --- |
| **Use Case** | **Completed** |
| Permanent Implementation of Zip landing page |   |
| Permanent placement on the footer (link) |   |
| Permanent placement on the footer (Zip payment icon) |   |
| Permanent placement on the product detail page |   |
| Permanent placement on the cart page |   |
| Homepage banner (any asset on the homepage) |   |
| Payment method is presented in parity with other payment methods |   |

**ASSETS**

Widget Script

 <script async src="https://widgets.partpay.co.nz/zip-widget-classic-v3.0.0.js?&min=1&max=2000&amount=50.00&locale=nz"></script>

Landing Page Script

<zip-information instore="false" online="true" faqs="true"></zip-information>

<script src="https://zip.co/nz/assets/landing.js?v=1.0"></script>

Payment Icon HTML

<li><img src="https://static.zipmoney.com.au/assets/default/footer-tile/footer-tile-new.png" style="height:25px; display:inline-block; vertical-align:middle;"></li>

*Extra css may be required for best alignment/positioning*

 **Notes**