Shiprocket Checkout Integration Guide for Custom Websites

Shiprocket Checkout—a seamless, high-converting checkout solution designed for Indian D2C brands.

How It Helps You
✅ One-click Checkout – Reduce drop-offs with a lightning-fast experience
✅ Address Autofill & Validation – Minimize delivery errors and RTOs
✅ Smart Payment Options – UPI, BNPL, Cards, COD (with fraud protection)
✅ Post-purchase Engagement – Keep customers informed and reduce cancellations

Introduction to this guide

Welcome to Shiprocket Checkout API Guide. This guide is designed to help developers and sellers seamlessly integrate Shiprocket Checkout into their custom websites. By integrating our APIs, you can sync your product catalog, initiate checkout, manage orders, and integrate loyalty points or wallets.

API Documentation: API Documentation

https://documenter.getpostman.com/view/25617008/2sB34bL3ig

This guide provides detailed API references, required parameters, example requests, and responses to help you understand the integration process efficiently.

Our APIs follow RESTful architecture, utilizing standard HTTP request methods (GET, POST) and JSON as the request/response format.

Getting Started

To begin integrating Shiprocket Checkout, follow these steps:

1️⃣ **Set Up the Catalog Sync APIs:**

- Develop the required catalog APIs (`Fetch Products`, `Fetch Products By Collection`, `Fetch Collections`).
- Configure these APIs & webhooks with Shiprocket for automatic product sync.

2️⃣ **Register for Shiprocket Checkout API Access:**

- Contact Shiprocket to onboard your custom website for API-based checkout.
- Shiprocket will provide API credentials (API Key & Secret Key) for authentication.

3️⃣ **Integrate Checkout Button & Generate Access Token:**

- Implement the checkout button script on your website.
- Call the Access Token API to get an authentication token for checkout initiation.

4️⃣ **Configure Webhooks for Order Updates:**

- Implement order creation and catalog update webhooks to receive real-time updates.

API Usage Guidelines

🚀 Base URL:
All API calls should be made to:

```
https://checkout-api.shiprocket.com/
```

🚀 Request & Response Format:

- API requests should be sent in JSON format.
- The API response will also be in JSON format.

🚀 HTTP Methods Used:

Method	Description
GET	Retrieve data (e.g., Fetch product details, order details)
POST	Create new data (e.g., Initiate checkout, redeem loyalty points)

🚀 Authentication:
All API requests must be authenticated using API Key & HMAC in the headers.

```
X-Api-Key: Bearer <your-api-key>
X-Api-HMAC-SHA256: <calculated-hmac-using-secret-key-and-request-body>
```

HMAC SHA256 in Base64. See - https://www.devglan.com/online-tools/hmac-sha256-online

---

Integration Steps

Now that you're familiar with the Shiprocket Checkout API, proceed to the detailed API references for step-by-step integration.

💡 Start with Catalog Sync APIs → Checkout Initiation → Order Processing → Webhooks → Loyalty Integration.

---

1. Catalog Sync – Seller API Requirements

To sync the entire website catalog with Shiprocket Checkout, merchants must develop and host the following APIs on their backend.

Seller APIs to be Implemented:

API Name	Description	Example Endpoint
Fetch Products	Returns all products in the seller's catalog.	`https://seller-url-for-product-list?page=1&limit=100`
Fetch Products By Collection	Fetches products belonging to a specific collection.	`https://seller-url-for-collection-product-list?collection_id=1234&page=1&limit=100`
Fetch Collections	Retrieves all the collections / categories.	`https://seller-url-for-collections-list?page=1&limit=100`

Implementation Steps:

1) Merchant's development team needs to create the above APIs.
2) Response of these APIs should be the same as given in the API documentation in the Seller API section.

Documentation: API Documentation

3) Each and every field is mandatory. If no value exists, then a blank string needs to be passed against that key. Pagination is required to be inherited in these APIs.
4) Once developed, these APIs and website domain URL need to be shared with Shiprocket's team. They will configure the same at Shiprocket Checkout's backend.
5) Upon successful integration, Shiprocket will sync the catalog from the merchant's website to the Shiprocket ecosystem.
6) After syncing, API credentials (API Key & Secret Key) will be generated and shared with the merchant.

---

2. Real-time Catalog Sync Using Webhooks

If any update (e.g., price, title, weight or any product details) is made in the merchant's catalog, the merchant must configure webhooks to sync the changes automatically.

Product Update Webhook

This webhook should be triggered whenever a new product added or existing product is updated in the merchant's system.

```
curl --location 'https://checkout-api.shiprocket.com/wh/v1/custom/product' \
--header 'X-Api-Key: YOUR-API-KEY' \
--header 'X-Api-HMAC-SHA256: <calculated-hmac>' \
--data '{
    "id": 632910392,
    "title": "IPod Nano - 8GB",
    "body_html": "<p>It'\''s the small</p>",
    "vendor": "Apple",
    "product_type": "Cult Products",
    "updated_at": "2023-11-07T09:50:12-05:00",
    "status": "active",
    "variants": [
        {
            "id": 808950810,
            "title": "Pink",
            "price": "199.00",
            "quantity": 10,
            "sku": "IPOD2008PINK",
            "updated_at": "2023-11-07T09:50:12-05:00",
            "image": {
```

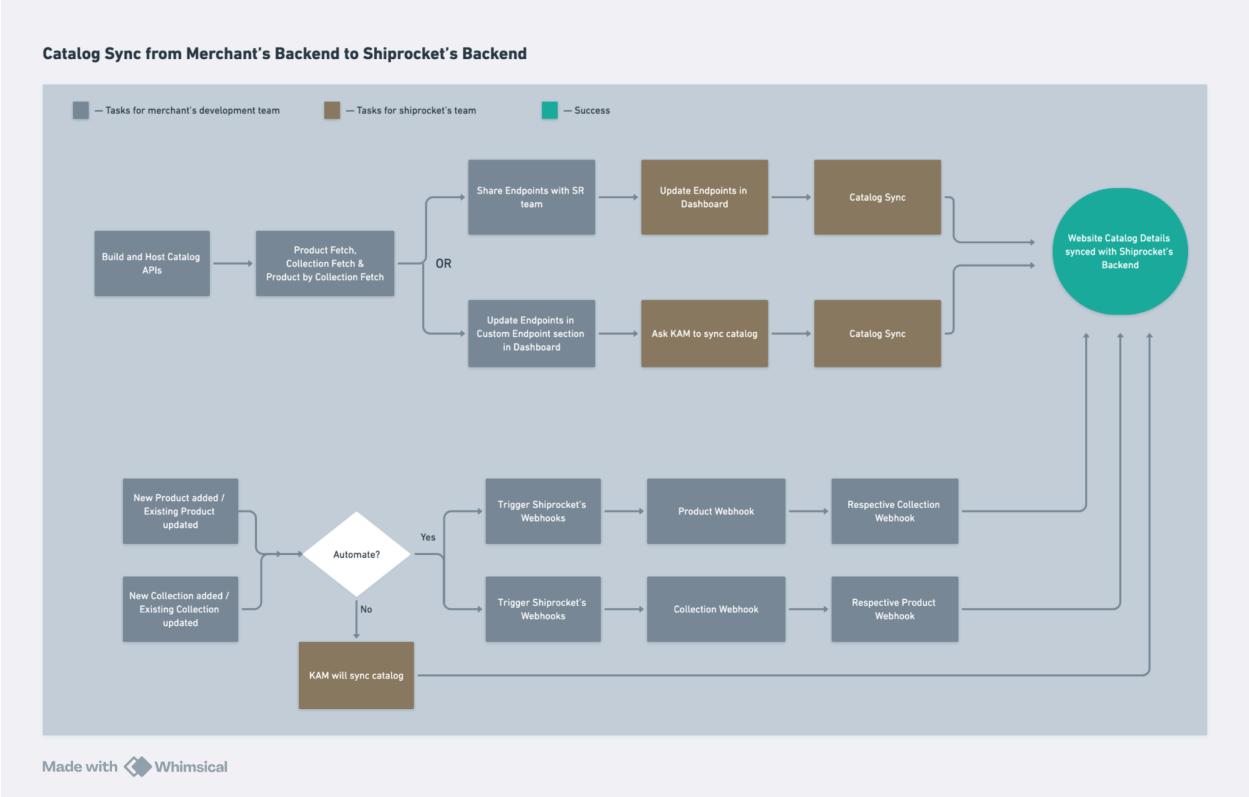
```
                "src":
"https://cdn.shopify.com/s/files/1/0005/4838/0009/products/ipod-nano.png"
            },
            "weight": 1.25
        }
    ],
           "image": {
                    "src":
"https://cdn.shopify.com/s/files/1/0005/4838/0009/products/ipod-nano.png"
                  }
}'
```

Collection Update Webhook

This webhook should be triggered whenever a **collection** is updated.

```
curl --location 'https://checkout-api.shiprocket.com/wh/v1/custom/collection'
\
--header 'X-Api-Key: YOUR-API-KEY' \
--header 'X-Api-HMAC-SHA256: <calculated-hmac>' \
--data '{
    "id": 482865238,
    "updated_at": "2023-10-03T13:19:52-04:00",
    "title": "Smart iPods",
    "body_html": "<p>The best selling iPod ever</p>",
    "image": {
        "src":
"https://cdn.shopify.com/s/files/1/0005/4838/0009/collections/ipod_nano_8gb.j
pg"
    }
}'
```

✅ **These webhooks will ensure real-time catalog sync between the merchant's website and Shiprocket Checkout.**



---

3. Checkout Initiation – Generating Access Token

At the time of checkout initiation, merchants need to generate an **access token** by calling the following API.

API: Generate Access Token for Checkout

```
curl --location
'https://checkout-api.shiprocket.com/api/v1/access-token/checkout' \
--header 'X-Api-Key: H3E8hebrr7oZFnVV' \
--header 'X-Api-HMAC-SHA256: C3TMxIORicQUmJ70OYFCSqlXxTO1tADvFItwGp0kE60=' \
--data '{
    "cart_data": {
        "items": [
            {
                "variant_id": "1244539923890450",
                "quantity": 1
            }
        ]
```

```
    },
    "redirect_url": "https://test-checkout.requestcatcher.com/test?key=val",
    "timestamp": "2023-12-29T12:06:33.085563Z"
}'
```

✅ The generated token (`result.token`) needs to be passed in the checkout button function for initiating checkout.

---

4. Embedding Checkout Button in Website

Once the token is generated, it must be embedded in the checkout button using the following script:

```
<html>
<head>
    <title>Test Checkout</title>
    <link rel="stylesheet"
href="https://checkout-ui.shiprocket.com/assets/styles/shopify.css">
</head>
<body>
    <button id="buyNow">Checkout</button>
    <script>
        const button = document.getElementById('buyNow');
        button.addEventListener('click', async (e) => {
            const token = "GENERATED_TOKEN"; // Pass generated token here
            HeadlessCheckout.addToCart(event, token, {fallbackUrl:
"https://your.fallback.com?product=123"})
        })
    </script>
    <input type="hidden" value="www.example.com" id="sellerDomain"/>
    <script
src="https://checkout-ui.shiprocket.com/assets/js/channels/shopify.js">
    </script>
</body>
</html>
```

✅ **On clicking the "Checkout" button, the Shiprocket Checkout iframe will open.**

---

5. Receiving Order Details – Webhook Integration

Once an order is successfully placed through Shiprocket Checkout, sellers need to set up a webhook to receive order details and create the order accordingly in their backend system.

```
curl --location '<SELLER_REGISTERED_WEBHOOK_URL>' \
--header 'Content-Type: application/json' \
--data-raw '{
    "order_id": "659fc40044f41a36bf1c556c",
    "cart_data": {
        "items": [
            {
                "variant_id": "1244539923890450",
                "quantity": 1
            }
        ]
    },
    "status": "SUCCESS",
    "phone": "9999999999",
    "email": "kaushal.test@pickrr.com",
    "payment_type": "CASH_ON_DELIVERY",
    "total_amount_payable": 224.0
}'
```

Note: Full payload is given in the API documentation.

✅ **Merchants must ensure that order details are stored and processed in their backend.**

---

6. Fetching Order Details via API

If the merchant wants to retrieve order details after checkout, they can use the following API by passing the order ID received in the Access Token API response.

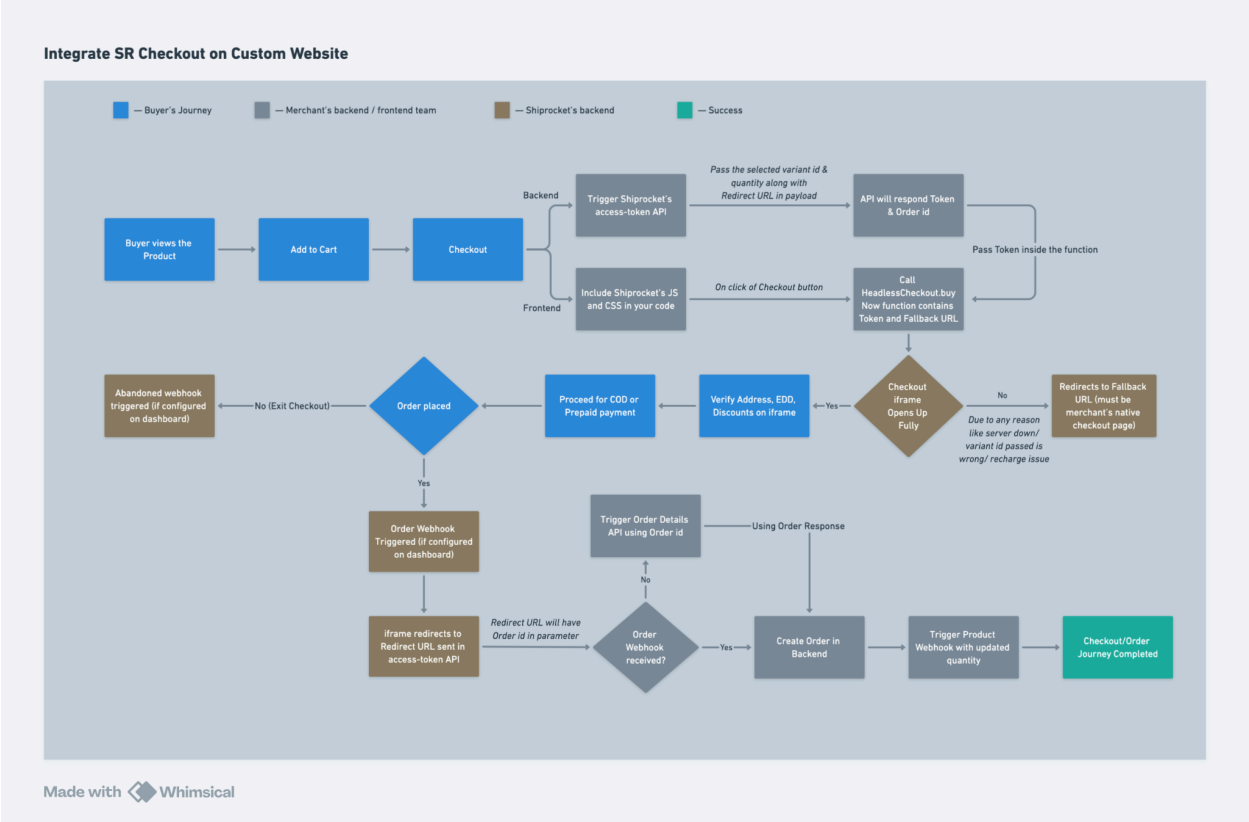
API: Fetch Order Details

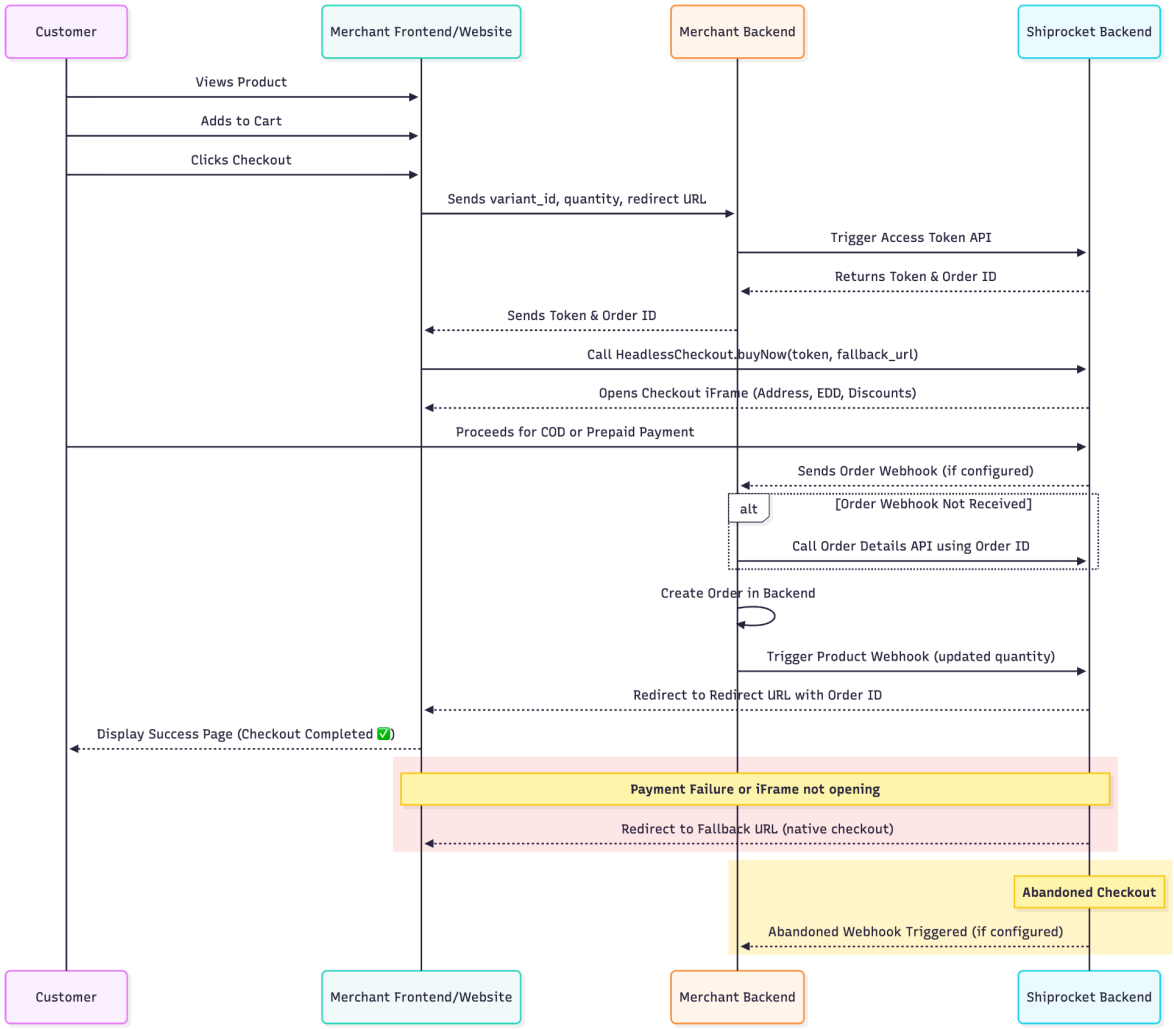
```
curl --location
'https://fastrr-api-dev.pickrr.com/api/v1/custom-platform-order/details' \
--header 'X-Api-Key: H3E8hebrr7oZFnVV' \
--header 'X-Api-HMAC-SHA256: w/D90XTBaptDjN4vU0HeaYiS+1NHflLHZ7hOi2jvGmE=' \
--header 'Content-Type: application/json' \
```

```
--data '{
      "order_id": "659fc40044f41a36bf1c556c",
      "timestamp": "2024-01-11T10:43:01.561Z"
}'
```

Note: Response for this Order Details API is given in the documentation.

✅ Sellers can call this API to fetch real-time order details for processing in their backend.
✅ This API helps in order verification, shipping coordination, and record-keeping.





---

7. Loyalty & Wallet System Integration

If a merchant has a **wallet system or loyalty points**, they can configure the following APIs.

1️⃣ Fetch Available Points

```
curl --location 'https://merchant-url-for-get-points' \
--header 'Content-Type: application/json' \
--data '{
    "mobile_number":"9999999999"
}'
```

2️⃣ Block Points at Checkout

```
curl --location 'https://seller-url-for-block-points' \
--header 'Content-Type: application/json' \
--data '{
    "mobile_number":"9451018768",
    "transactional_points":50,
    "order_id":12345
}'
```

3️⃣ Unblock Points on Removing Points or Cart Abandonment

```
curl --location 'https://seller-url-for-unblock-points' \
--header 'Content-Type: application/json' \
--data '{
    "order_id":12345,
    "transactional_points":30
}'
```

✅ **This integration ensures seamless reward point redemption as discount and refund management at merchant's end through wallets.**

---

8. Final Steps for Integration

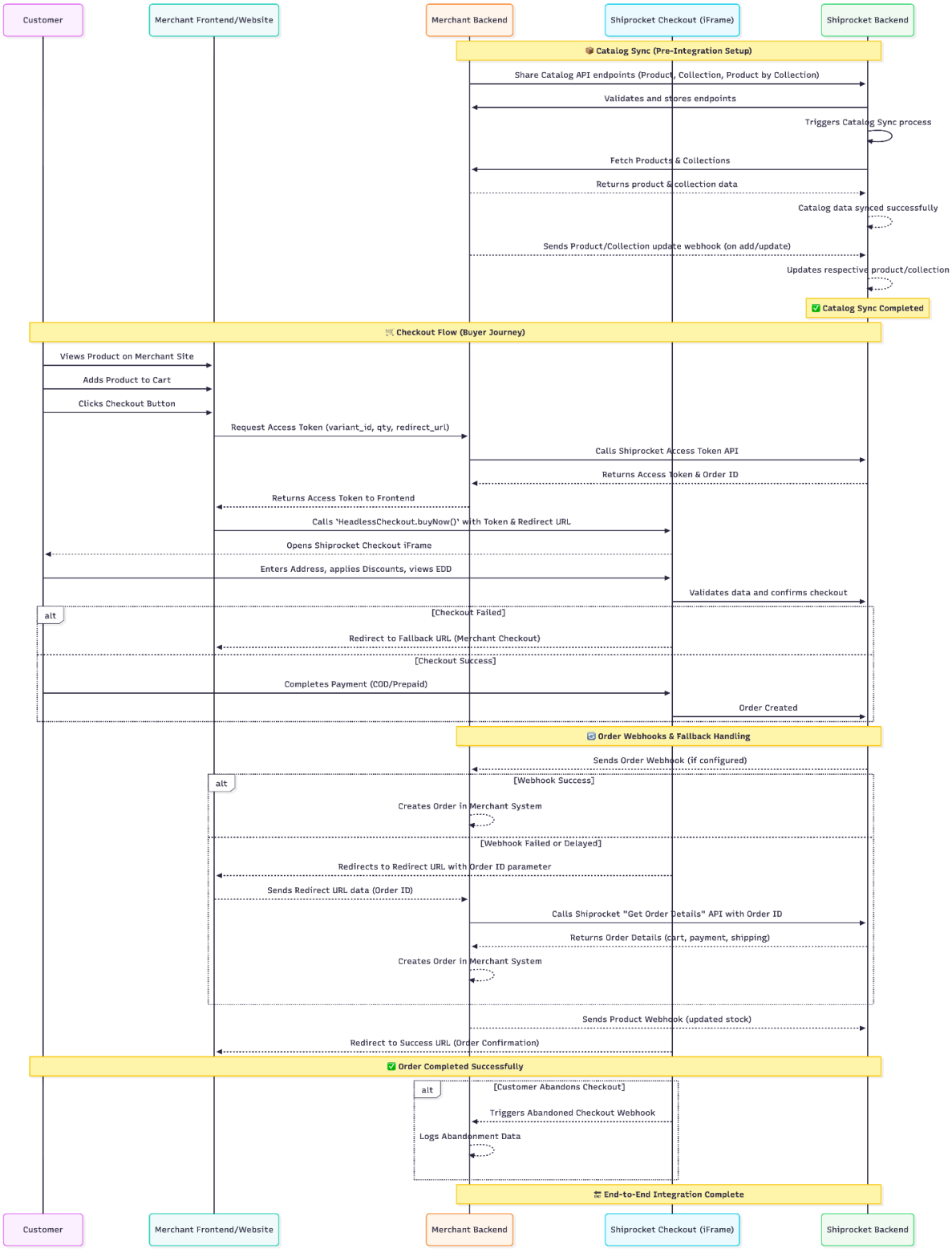
Merchant's Development Team Responsibilities:

- 🔹 Develop & deploy catalog sync APIs and configure webhooks.
- 🔹 Implement checkout button integration using the provided script.
- 🔹 Develop Order Webhooks.
- 🔹 Set up loyalty point APIs if applicable.

Shiprocket Responsibilities:

- 🔹 Configure merchant's catalog sync APIs in Shiprocket's backend.
- 🔹 Provide API Key & Secret Key.
- 🔹 Configure order webhooks endpoint in shiprocket's backend.
- 🔹 Ensure checkout iframe functions correctly.

Integration Architecture



Conclusion

Once all steps are completed, **Shiprocket Checkout will be fully integrated** with the merchant's custom website, enabling a seamless checkout experience.