

CashKavach API Documentation

1. Introduction

Welcome to the **CashKavach API**! This document provides a comprehensive guide to integrating with our payment gateway. You can use our API to accept payments from customers, manage transactions, and handle payouts to your bank account.

This documentation covers the following key areas:

- **Authentication:** How to authenticate your requests to our API.
- **Payment Flow:** A step-by-step guide on how to create and manage payments.
- **API Reference:** Detailed information about each API endpoint.
- **Data Models:** An explanation of the data structures used in our API.

2. Authentication

CashKavach uses two methods of authentication:

- **JWT (JSON Web Token):** For authenticating users on the merchant dashboard.
- **API Key:** For authenticating API requests for payment processing.

2.1. JWT Authentication

To get a JWT, you need to sign up and log in to the merchant dashboard.

Endpoint: POST /api/auth/signup

This endpoint allows you to create a new merchant account.

Request Body:

```
{
  "name": "Your Business Name",
  "email": "you@example.com",
  "password": "your_secure_password"
}
```

Endpoint: POST /api/auth/login

This endpoint allows you to log in to your merchant account and receive a JWT.

Request Body:

```
{
  "email": "you@example.com",
```

```
"password": "your_secure_password"
}
```

The JWT you receive should be included in the x-auth-token header for all subsequent requests to the merchant dashboard APIs.

2.2. API Key Authentication

To process payments, you need to generate an API key.

Endpoint: POST /api/create

This endpoint generates a new API key for your account.

Headers:

- x-auth-token: Your JWT obtained from logging in.

Response:

```
{
  "apiKey": "cashcavash_XXXXXXXXXXXXXXXXXXXXXXXXX"
}
```

This API key must be included in the **x-api-key** header for all API requests related to payment processing.

3. Payment Flow

The payment flow in CashKavach is designed to be simple and flexible. Here's a step-by-step overview:

1. **Onboarding:** Sign up and generate your API key.
2. **Create a Payment:** You can either create a simple payment link or build a custom checkout experience.
3. **Customer Pays:** The customer completes the payment using the provided link or your custom checkout.
4. **Check Status:** You can poll our API to check the status of the payment.
5. **Payouts:** Request a payout to your bank account.

3.1. Option A: Simple Payment Link

This is the easiest way to start accepting payments.

Endpoint: POST /api/payments/create-link

This endpoint creates a payment link that you can share with your customers.

Request Body:

```
{
  "amount": 100,
  "customerName": "John Doe",
  "customerEmail": "john.doe@example.com",
  "customerPhone": "9876543210",
  "description": "Payment for order #123"
}
```

The response will contain a `paymentUrl` that you can redirect your customer to.

3.2. Option B: Custom Checkout

For a more integrated experience, you can build your own checkout form.

Step 1: Create a Payment Session

Endpoint: POST /api/payments/create

This endpoint creates a payment session and returns a `paymentSessionId`.

Request Body:

```
{
  "amount": 100,
  "customerName": "John Doe",
  "customerEmail": "john.doe@example.com",
  "customerPhone": "9876543210"
}
```

Step 2: Initiate the Payment

Endpoint: POST /api/payments/pay

Use the `paymentSessionId` from the previous step to initiate the payment with a specific method.

Request Body (for UPI):

```
{
  "paymentSessionId": "session_xxxxxxxxxxxx",

```

```
"paymentMethod": "upi",  
"channel": "collect",  
"upild": "customer@upi"  
}
```

Request Body (for Card):

```
{  
  "paymentSessionId": "session_xxxxxxxxxxxx",  
  "paymentMethod": "card",  
  "cardDetails": {  
    "cardNumber": "4111111111111111",  
    "cardHolderName": "John Doe",  
    "cardExpiryMM": "12",  
    "cardExpiryYY": "25",  
    "cardCvv": "123"  
  }  
}
```

4. API Reference

4.1. Auth Endpoints

- **POST /api/auth/signup:** Register a new user.
- **POST /api/auth/login:** Login and get a JWT.
- **GET /api/auth/profile:** Get user profile details.
- **PUT /api/auth/profile:** Update user profile details.

4.2. API Key Endpoints

- **POST /api/create:** Create an API key.
- **DELETE /api/delete:** Delete an API key.

4.3. Payment Endpoints

- **POST /api/payments/create:** Create a payment session.
- **POST /api/payments/create-link:** Create a payment link.
- **POST /api/payments/pay:** Initiate a payment with a specific method.
- **GET /api/payments/methods:** Get available payment methods for an order.
- **GET /api/payments/status/:orderId:** Get the status of a payment.
- **GET /api/payments/transactions:** Get a list of all transactions.
- **POST /api/payments/refund/:orderId:** Refund a payment.

4.4. Merchant Dashboard Endpoints

- **GET /api/payments/merchant/payouts:** Get all payouts for the merchant.
- **POST /api/payments/merchant/payout/request:** Request a new payout.
- **GET /api/payments/merchant/balance:** Get the merchant's account balance.
- **POST /api/payments/merchant/payout/:payoutId/cancel:** Cancel a payout request.

4.5. Super Admin Endpoints

- **GET /api/payments/admin/transactions:** Get all transactions for all merchants.
- **POST /api/payments/admin/payout:** Create a payout for a merchant.
- **GET /api/payments/admin/payouts:** Get all payouts for all merchants.

5. Data Models

5.1. User

Field	Type	Description
name	String	Name of the user.
email	String	Email of the user (unique).
businessName	String	Business name of the user.
businessDetails	Object	Details about the user's business.
password	String	Hashed password of the user.
role	String	Role of the user (admin or superAdmin).
apiKey	String	API key for the user (unique).

5.2. Transaction

Field	Type	Description
transactionId	String	Unique ID for the transaction.

orderId	String	Unique ID for the order.
merchantId	ObjectId	ID of the merchant.
merchantName	String	Name of the merchant.
customerId	String	ID of the customer.
customerName	String	Name of the customer.
customerEmail	String	Email of the customer.
customerPhone	String	Phone number of the customer.
amount	Number	Transaction amount.
currency	String	Currency of the transaction (e.g., INR).
status	String	Status of the transaction (created, pending, paid, failed, cancelled, refunded, partial_refund).
cashfreeOrderToken	String	Cashfree order token.
cashfreePaymentId	String	Cashfree payment ID.
cashfreeOrderId	String	Cashfree order ID.
paymentMethod	String	Payment method used (e.g., upi, card).
paidAt	Date	Timestamp when the transaction was paid.
refundAmount	Number	Amount that has been refunded.
refundReason	String	Reason for the refund.
refundedAt	Date	Timestamp when the refund was processed.

5.3. Payout

Field	Type	Description
payoutId	String	Unique ID for the payout.
merchantId	ObjectId	ID of the merchant.
merchantName	String	Name of the merchant.
amount	Number	Payout amount.
commission	Number	Commission deducted from the payout.
netAmount	Number	Net amount transferred to the merchant.
transferMode	String	Mode of transfer (bank_transfer, upi, wallet).
beneficiaryDetails	Object	Details of the beneficiary.
status	String	Status of the payout (pending, processing, completed, failed, cancelled).
cashfreeTransferId	String	Cashfree transfer ID.