

■ OITH Payment Setup Guide

This guide explains how to set up payment processing for OITH using Stripe.

■ Overview

OITH supports three payment integration methods:

1. **Stripe Checkout** (Recommended) - Hosted payment page
 2. **Stripe Payment Links** - No-code option
 3. **Test Mode** - For development without Stripe
-

■ Quick Start (5 minutes)

Option 1: Test Mode (Development)

No setup required! The app automatically runs in test mode if Stripe isn't configured.

- Users can complete the payment flow
- No real charges are made
- Data is stored locally

Option 2: Stripe Payment Links (No Code)

1. Go to [Stripe Dashboard](#)
2. Create a new payment link
3. Set price: \$9.99/month or \$79.99/year
4. Copy the link
5. Replace the checkout redirect URL in `app.js`

Option 3: Full Stripe Integration (Production)

Follow the detailed setup below.

■ Prerequisites

1. **Node.js** (v18 or higher) - [Download](#)
 2. **Stripe Account** - [Sign up free](#)
 3. **Bank Account** - For receiving payouts
-

■ Step-by-Step Setup

Step 1: Install Dependencies



This installs:

- `express` - Web server
- `stripe` - Stripe SDK
- `cors` - Cross-origin requests
- `dotenv` - Environment variables

Step 2: Create Stripe Account

1. Go to [stripe.com](#) and click "Start now"
2. Verify your email
3. Complete business verification:
 - Business type (Individual/Company)
 - Business details
 - Personal information (for identity verification)
 - Tax ID (EIN or SSN)

Step 3: Link Bank Account

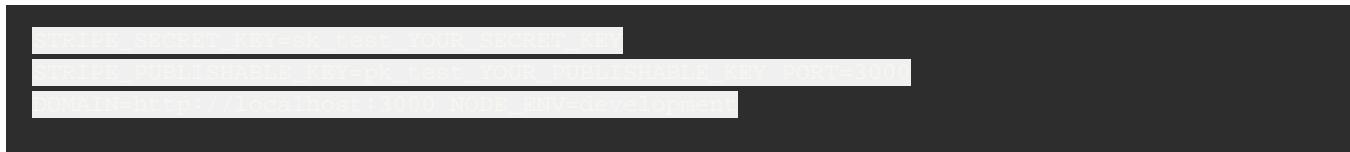
1. In Stripe Dashboard, go to **Settings → Bank accounts and scheduling**
2. Click **Add bank account**
3. Enter:
 - Bank name
 - Routing number (9 digits)
 - Account number
4. Verify with micro-deposits (2 small amounts sent to your account)
5. Set payout schedule (daily, weekly, monthly)

Step 4: Get API Keys

1. In Stripe Dashboard, go to **Developers → API Keys**
2. Copy your keys:
 - **Publishable key:** `pk_test_xxx` (safe for frontend)
 - **Secret key:** `sk_test_xxx` (keep private!)

Step 5: Configure Environment

1. Rename `env-example.txt` to `.env`
2. Add your keys:



Step 6: Update Frontend Config

In `app.js`, update the payment config:



Step 7: Start the Server



You should see:



Step 8: Test Payment Flow

1. Open `http://localhost:3000/index.html`
2. Go through registration
3. On payment screen, click "Subscribe"
4. Complete payment on Stripe Checkout

-
- Verify subscription is active

■ Testing

Test Cards (Stripe Test Mode)

Card Number	Description
4242 4242 4242 4242	Successful payment
4000 0000 0000 0002	Card declined
4000 0000 0000 3220	3D Secure required

Use any future expiry date and any 3-digit CVV.

Test Webhooks Locally

- Install Stripe CLI: <https://stripe.com/docs/stripe-cli>
 - Run: `stripe listen --forward-to localhost:3000/webhook`
 - Copy the webhook signing secret
 - Add to `.env`: `STRIPE_WEBHOOK_SECRET=whsec_xxx`
-

■ Bank Account Configuration

Payout Settings

In Stripe Dashboard → Settings → Payouts:

Setting	Recommended
Payout schedule	Daily (automatic)
Minimum payout	\$0 (pay out everything)
Statement descriptor	OITH DATING

Bank Requirements

- Must be a US bank account (for USD)
 - Business checking account recommended
 - Personal accounts work for sole proprietors
-

■ Pricing Configuration

Current Plans

Plan	Price	Interval
Monthly	\$9.99	Per month
Yearly	\$79.99	Per year (save 33%)

Changing Prices

1. Edit `server.js`:



2. Update `app.js`:



■ Security Checklist

- ■ Never expose secret key (`sk_xxx`) in frontend code
- ■ Use HTTPS in production
- ■ Verify webhook signatures
- ■ Enable 3D Secure for additional authentication
- ■ Review Stripe's security best practices

■ Going Live Checklist

1. **Complete Stripe verification** - Business details, bank account
 2. **Switch to live keys** - Replace `test` keys with `live` keys
 3. **Update domain** - Set `DOMAIN` to your production URL
 4. **Configure webhooks** - Add production webhook endpoint
 5. **Test with real card** - Make a small real payment
 6. **Set up monitoring** - Enable Stripe email notifications
-

■ Webhook Events

The server handles these Stripe events:

Event	Action
<code>checkout.session.completed</code>	Activate subscription
<code>customer.subscription.updated</code>	Update subscription status
<code>customer.subscription.deleted</code>	Revoke premium access
<code>invoice.payment_failed</code>	Notify user, retry payment

■ Troubleshooting

Payment fails immediately

- Check API keys are correct
- Verify server is running
- Check browser console for errors

Webhooks not received

- Run `stripe listen` for local testing

- Check webhook URL is correct
- Verify webhook secret

Bank payout pending

- Allow 2-7 days for first payout
- Verify bank account is verified
- Check for any Stripe dashboard alerts

■ Support

- **Stripe Documentation:** <https://stripe.com/docs>
- **Stripe Support:** <https://support.stripe.com>
- **OITH Issues:** Create a GitHub issue

■ Files Reference

File	Purpose
<code>server.js</code>	Payment server (backend)
<code>payment-service.js</code>	Frontend payment handling
<code>package.json</code>	Dependencies
<code>env-example.txt</code>	Environment template
<code>PAYMENT_SETUP.md</code>	This guide