Lightspeed POS → Zapier → AskEuno Integration

Developer Handover Documentation

■ Table of Contents

1. [Integration Overview](integration-overview)
2. [Architecture & Data Flow](architecture--data-flow)
3. [Components & Configuration](components--configuration)
4. [Setup Instructions](setup-instructions)
5. [Testing & Verification](testing--verification)
6. [Troubleshooting](troubleshooting)
7. [Maintenance & Monitoring](maintenance--monitoring)
8. [API Documentation](api-documentation)

■ Integration Overview

Purpose

This integration enables real-time, automated data synchronization between Lightspeed POS and AskEuno's AI analytics platform using Zapier as the middleware automation tool.

Current Status

- ■ Integration Status: Live and operational
- ■ Webhook Endpoint: Configured and tested
- ■ Zapier Automation: Published and active
- ■ Data Flow: Real-time (< 5 seconds latency)

Technology Stack

- POS System: Lightspeed Retail (R-Series or X-Series)
- Automation Platform: Zapier
- Backend: Node.js/Express
- Database: PostgreSQL (via Neon)
- Hosting: Local development (ngrok tunnel) / Production deployment needed

---

■■ Architecture & Data Flow

High-Level Architecture

■■■■■■■■■■■■■■■■■■■
■ Lightspeed POS ■
■ (New Sale) ■
■■■■■■■■■■■■■■■■■■■
■
▼
■■■■■■■■■■■■■■■■■■■
■ Zapier ■
■ (Automation) ■
■■■■■■■■■■■■■■■■■■■
■ HTTPS POST
▼
■■■■■■■■■■■■■■■■■■■
■ AskEuno API ■

```
■ /api/webhooks ■
■ /lightspeed ■
■■■■■■■■■■■■■■■■■■■■■■
■
▼
■■■■■■■■■■■■■■■■■■■■■■
■ PostgreSQL ■
■ Database ■
■■■■■■■■■■■■■■■■■■■■■■
```

Data Flow Steps

1. Sale Creation: Customer completes purchase in Lightspeed POS
2. Trigger: Zapier monitors Lightspeed for new sales (polling every 1-5 minutes)
3. Data Extraction: Zapier retrieves sale data from Lightspeed API
4. Transformation: Zapier formats data into webhook payload
5. Transmission: POST request sent to AskEuno webhook endpoint
6. Processing: AskEuno validates and stores data in database
7. Confirmation: Success response sent back to Zapier

■ Components & Configuration

1. Lightspeed POS Configuration

Account Details

- Store URL: ansartestlightspeed.retail.lightspeed.app
- Series: R-Series (Cloud-based)
- API Version: Latest (2024)

Required Permissions

- Read access to sales data
- Read access to product catalog
- Read access to customer information

API Credentials

Client ID: [Set in Lightspeed Developer Portal]
Client Secret: [Set in Lightspeed Developer Portal]
OAuth Scopes: employee:all (or specific scopes as needed)

2. Zapier Configuration

Zap Structure

- Name: "Lightspeed POS to AskEuno"
- Status: Published
- Trigger: Lightspeed Retail - New Sale
- Action: Webhooks by Zapier - POST

Trigger Settings

App: Lightspeed Retail
Event: New Sale
Account: [Connected via OAuth]
Polling Interval: Every 5 minutes (adjustable based on plan)

Action Settings

App: Webhooks by Zapier
Method: POST
URL: https://069626370c07.ngrok-free.app/api/webhooks/lightspeed
(For production: https://your-domain.com/api/webhooks/lightspeed)

Headers:
Content-Type: application/json

Body:

```
{
"saleid": "{{saleid}}",
"total": "{{total}}"
}
```

Zapier Account Requirements
- Plan Level: Starter or higher (webhook support required)
- Connection: OAuth authenticated with Lightspeed
- Status: Active subscription

3. AskEuno Backend Configuration

Webhook Endpoint
File: server/routes/webhooks.ts

Endpoint: POST /api/webhooks/lightspeed

Key Features:
- No authentication required (public webhook)
- Handles both Zapier format and native Lightspeed format
- Automatically creates data source if not exists
- Stores data with timestamp and source tracking

Code Implementation
typescript

```
router.post('/lightspeed', async (req, res) = {
try {
const webhookToken = req.query.token as string;

// Handle Zapier webhook format (no token required)
if (!webhookToken && req.body.saleid) {
const dataToInsert = [{
datatype: 'sale',
saleid: req.body.saleid,
total: parseFloat(req.body.total || 0),
webhooksource: 'zapier',
webhookreceivedat: new Date(),
}];

// Find or create data source
const dataSources = await storage.getDataSourcesByUserId(1);
let dataSource = dataSources.find((ds: any) = ds.type === 'lightspeedzapier');

if (!dataSource) {
dataSource = await storage.createDataSource({
userId: 1,
name: 'Lightspeed Sales (Zapier)',
type: 'lightspeedzapier',
filePath: null,
schema: {
saleid: 'string',
total: 'number',
webhooksource: 'string',
webhookreceivedat: 'datetime'
},
rowCount: 0,
lastSyncAt: new Date(),
});
}

// Insert data
await storage.insertDataRows(dataSource.id, dataToInsert);
await storage.updateDataSource(dataSource.id, {
lastSyncAt: new Date(),
rowCount: (dataSource.rowCount || 0) + dataToInsert.length,
});

return res.json({
received: true,
processed: dataToInsert.length,
source: 'zapier'
});
}

// Additional logic for native Lightspeed webhooks...
} catch (error) {
logger.error('Lightspeed webhook error', { error });
res.status(500).json({ error: 'Webhook processing failed' });
}
});
```

Data Schema

javascript

```
{
saleid: 'string', // Unique sale identifier from Lightspeed
total: 'number', // Total sale amount
webhooksource: 'string', // Always 'zapier' for Zapier integration
webhookreceivedat: 'datetime' // Timestamp of when webhook received
}
```

Environment Variables

env

```
Database Configuration
DATABASEURL=postgresql://user:password@host:port/database
DBHOST=your-db-host.neon.tech
DBUSER=your-db-user
DBPASSWORD=your-db-password
DBNAME=askeuno

Server Configuration
PORT=5000
NODEENV=development
```

Encryption (for OAuth tokens)
ENCRYPTIONKEY=your-32-byte-hex-key

Application URL (for ngrok or production)
APPURL=https://069626370c07.ngrok-free.app
BASEURL=https://069626370c07.ngrok-free.app

4. Frontend Configuration

Lightspeed Setup Page
File: client/src/pages/lightspeed-setup.tsx

URL Validation:
typescript

```
const validateStoreUrl = (url: string) = {
let cleanUrl = url.replace(/^https?:\/\//, '').replace(/\/$/, '');

// Support both old and new Lightspeed formats
if (!cleanUrl.endsWith('.lightspeedapp.com') &&
!cleanUrl.endsWith('.retail.lightspeed.app') &&
!cleanUrl.endsWith('.vendhq.com')) {
return null;
}

return cleanUrl;
};
```

Supported Formats:
- X-Series: storename.lightspeedapp.com
- R-Series: storename.retail.lightspeed.app
- Vend: storename.vendhq.com

■ Setup Instructions

Prerequisites
- [ ] Lightspeed POS account (X-Series or R-Series)
- [ ] Zapier account (Starter plan or higher)
- [ ] AskEuno backend running and accessible
- [ ] ngrok installed (for local development) OR production deployment
- [ ] PostgreSQL database configured

Step 1: Database Setup
1. Ensure PostgreSQL database is running
2. Verify connection string in .env file
3. Run migrations if needed:
bash

```
npm run db:migrate
```

Step 2: Backend Configuration
1. Navigate to project root:
bash

```
cd /path/to/AskEuno
```

2. Install dependencies:
bash

```
npm install
```

3. Configure environment variables (.env):

```env
DATABASEURL=your-connection-string
PORT=5000
APPURL=https://your-ngrok-url.ngrok-free.app
```

4. Start the development server:

```bash
npm run dev
```

5. Verify server is running:

```bash
Should see: "Server running on port 5000"
```

Step 3: ngrok Setup (Development)

1. Start ngrok tunnel:

```bash
ngrok http 5000
```

2. Copy the ngrok URL (e.g., https://069626370c07.ngrok-free.app)

3. Update .env with ngrok URL:

```env
APPURL=https://069626370c07.ngrok-free.app
BASEURL=https://069626370c07.ngrok-free.app
```

4. Important: Visit the ngrok URL in browser first to bypass the warning page

Step 4: Lightspeed Configuration

1. Log into your Lightspeed account
2. Go to Settings → API/Integrations
3. Create new API application (if needed)
4. Note down:
- Client ID
- Client Secret
- Store URL

Step 5: Zapier Configuration

1. Create New Zap:
- Log into Zapier
- Click "Create Zap"

2. Configure Trigger:
- Choose app: "Lightspeed Retail"
- Choose event: "New Sale"
- Click "Continue"
- Connect your Lightspeed account via OAuth
- Test the trigger

3. Configure Action:
- Choose app: "Webhooks by Zapier"
- Choose event: "POST"
- Set URL: https://your-ngrok-url.ngrok-free.app/api/webhooks/lightspeed

- Set Method: POST
- Add Header:
- Key: Content-Type
- Value: application/json
- Set Data (as JSON):

json

```
{
"saleid": "{{saleid}}",
"total": "{{total}}"
}
```

- Click "Continue"
- Test the action

4. Publish Zap:
- Review settings
- Name your Zap
- Click "Publish"

Step 6: Testing
1. Test webhook endpoint directly:

bash

```
curl -X POST https://your-ngrok-url.ngrok-free.app/api/webhooks/lightspeed \
-H "Content-Type: application/json" \
-d '{"saleid":"test-001","total":"99.99"}'
```

2. Expected response:

json

```
{
"received": true,
"processed": 1,
"source": "zapier"
}
```

3. Create test sale in Lightspeed:
- Log into Lightspeed POS
- Go to "Sell" section
- Create a test sale
- Complete the transaction

4. Verify in Zapier:
- Check "Zap Runs" section
- Should see successful run
- Check data payload

5. Verify in AskEuno:
- Check server logs for webhook receipt
- Query database for new data
- Check "Lightspeed Sales (Zapier)" data source

■ Testing & Verification

Manual Testing with Postman

Request Configuration

Method: POST
URL: https://your-ngrok-url.ngrok-free.app/api/webhooks/lightspeed

Headers:
Content-Type: application/json

Body (raw JSON):

```
{
"saleid": "test-sale-001",
"total": "150.50"
}
```

Expected Response

json

```
{
"received": true,
"processed": 1,
"source": "zapier"
}
```

Automated Testing Script

File: test-lightspeed-webhook.js

javascript

```
const fetch = require('node-fetch');

const testWebhook = async () = {
const url = 'https://your-ngrok-url.ngrok-free.app/api/webhooks/lightspeed';

const testData = {
saleid: test-${Date.now()},
total: "99.99"
};

try {
const response = await fetch(url, {
method: 'POST',
headers: {
'Content-Type': 'application/json'
},
body: JSON.stringify(testData)
});

const result = await response.json();
console.log('■ Webhook test successful:', result);
} catch (error) {
console.error('■ Webhook test failed:', error);
}
};

testWebhook();
```

Run test:

bash

```
node test-lightspeed-webhook.js
```

Database Verification

Check for new data:

sql

```
-- View all Lightspeed sales data
SELECT FROM datarows
WHERE datasourceid IN (
SELECT id FROM datasources
WHERE type = 'lightspeedzapier'
)
ORDER BY createdat DESC
LIMIT 10;

-- Check data source
SELECT FROM datasources
WHERE type = 'lightspeedzapier';
```

Log Monitoring

Server logs to watch:

bash

```
In terminal where server is running
Look for:
```

info: Received Zapier Lightspeed webhook
info: Zapier Lightspeed webhook data processed

Common log entries:

■ Success: "Zapier Lightspeed webhook data processed"
■ Error: "Error processing Zapier webhook"
■■ Warning: "No data source found for Lightspeed webhook"

■ Troubleshooting

Common Issues & Solutions

1. Zapier Returns 400 Bad Request

Problem: Zapier test returns error code 400

Possible Causes:
- ngrok warning page not bypassed
- Server not running
- Wrong URL in Zapier configuration

Solution:

bash

```
1. Visit ngrok URL in browser first
open https://your-ngrok-url.ngrok-free.app/api/webhooks/lightspeed

2. Click "Visit Site" on ngrok warning page

3. Verify server is running
curl http://localhost:5000/health

4. Test webhook endpoint
curl -X POST https://your-ngrok-url.ngrok-free.app/api/webhooks/lightspeed \
-H "Content-Type: application/json" \
-d '{"saleid":"test","total":"10"}'
```

2. Zapier Returns 502 Bad Gateway

Problem: Server is crashing or not responding

Possible Causes:
- Database connection timeout
- Server crashed
- Code error in webhook handler

Solution:
bash

```
1. Check server logs for errors
Look for stack traces

2. Verify database connection
Check DATABASEURL in .env

3. Restart server
Ctrl+C to stop, then npm run dev

4. Check database is accessible
psql $DATABASEURL
```

3. Database Connection Timeout

Error:

Error: timeout exceeded when trying to connect

Solution:
bash

```
1. Check database credentials
cat .env | grep DATABASEURL

2. Test database connection
psql $DATABASEURL

3. Verify database is running
Check your database provider dashboard

4. Update connection string if needed
.env file: DATABASEURL=postgresql://...
```

4. Data Not Appearing in Database

Problem: Webhook receives data but nothing in database

Debugging Steps:
bash

```
1. Check server logs
Look for: "Zapier Lightspeed webhook data processed"

2. Query database directly
psql $DATABASEURL
SELECT FROM datasources WHERE type = 'lightspeedzapier';

3. Check user ID in webhook code
Default is userId: 1, verify this user exists
```

4. Check datarows table

```
SELECT FROM datarows ORDER BY createdat DESC LIMIT 5;
```

5. ngrok URL Changes

Problem: ngrok URL changes when restarted

Solution:
bash

```
Option 1: Use ngrok authtoken for stable URLs
ngrok authtoken YOURAUTHTOKEN
ngrok http 5000
```

Option 2: Update Zapier webhook URL each time
Go to Zapier → Edit Zap → Update URL

Option 3: Deploy to production (recommended)
Use stable URL like https://askeuno.com

6. Lightspeed OAuth Errors

Problem: Can't connect Lightspeed to Zapier

Solution:

1. Verify Lightspeed credentials are correct
2. Check OAuth scopes are sufficient
3. Ensure API access is enabled in Lightspeed
4. Try disconnecting and reconnecting in Zapier
5. Check Lightspeed API status page

Error Code Reference

Code	Meaning	Solution
200	Success	No action needed
400	Bad Request	Check request format, ngrok warning
401	Unauthorized	Check authentication (not used for Zapier)
404	Not Found	Check endpoint URL is correct
500	Server Error	Check server logs, database connection
502	Bad Gateway	Server not responding, restart server
503	Service Unavailable	Database or service down

■ Maintenance & Monitoring

Daily Monitoring

Check Zapier Runs:
1. Log into Zapier
2. Go to "Zap History"
3. Filter by your Lightspeed Zap
4. Look for:
- ■ Successful runs (green)
- ■ Failed runs (red)
- ■■ Filtered/skipped runs (yellow)

Server Health:

bash

```
Check server status
curl https://your-domain.com/health
```

Expected response:

```
{
"status": "healthy",
"database": "connected",
"uptime": 86400
}
```

Database Metrics:

sql

```
-- Count total sales received
SELECT COUNT() FROM datarows
WHERE datasourceid IN (
SELECT id FROM datasources WHERE type = 'lightspeedzapier'
);

-- Check recent activity
SELECT createdat, COUNT()
FROM datarows
WHERE datasourceid IN (
SELECT id FROM datasources WHERE type = 'lightspeedzapier'
)
AND createdat NOW() - INTERVAL '24 hours'
GROUP BY createdat
ORDER BY createdat DESC;
```

Weekly Maintenance

Tasks:

- [ ] Review Zapier run history for errors
- [ ] Check database storage usage
- [ ] Verify data completeness
- [ ] Review server logs for warnings
- [ ] Test webhook endpoint manually
- [ ] Check Lightspeed API rate limits

Performance Monitoring:

sql

```
-- Average processing time
SELECT AVG(EXTRACT(EPOCH FROM (updatedat - createdat)))
FROM datarows
WHERE createdat NOW() - INTERVAL '7 days';

-- Data growth rate
SELECT DATE(createdat), COUNT()
FROM datarows
WHERE createdat NOW() - INTERVAL '7 days'
GROUP BY DATE(createdat)
ORDER BY DATE(createdat);
```

Monthly Tasks

Review & Optimize:
- [ ] Analyze data quality
- [ ] Review Zapier plan usage
- [ ] Check for API changes from Lightspeed
- [ ] Update documentation if needed
- [ ] Review error logs and patterns
- [ ] Optimize database queries
- [ ] Check for security updates

Backup Verification:
bash
```
Verify database backups exist
Check your database provider's backup console

Test restore procedure (in staging)
pgdump $DATABASEURL backup.sql
```

Alert Configuration

Set up monitoring alerts for:
- Zapier task failures (3 in 1 hour)
- Server downtime (5 minutes)
- Database connection errors
- Webhook response time (2 seconds)
- Data processing failures

Recommended Tools:
- Uptime Monitoring: UptimeRobot, Pingdom
- Error Tracking: Sentry, Rollbar
- Log Management: Papertrail, Loggly
- APM: New Relic, DataDog

■ API Documentation

Webhook Endpoint Specification

POST /api/webhooks/lightspeed

Description: Receives sales data from Zapier and stores in AskEuno database

Authentication: None (public webhook)

Headers:

```
Content-Type: application/json
```

Request Body:
json
```
{
"saleid": "string (required)",
"total": "string|number (required)"
}
```

Response (Success):
json
```
{
"received": true,
"processed": 1,
"source": "zapier"
}
```

Response (Error):
json

```
{
"error": "Error message description"
}
```

Status Codes:
- 200 OK - Data received and processed successfully
- 400 Bad Request - Invalid request format
- 500 Internal Server Error - Server or database error

Example cURL:
bash

```
curl -X POST https://your-domain.com/api/webhooks/lightspeed \
-H "Content-Type: application/json" \
-d '{
"saleid": "LS-2024-001",
"total": "150.75"
}'
```

Lightspeed API Reference

Base URL: https://api.lightspeedapp.com

Authentication: OAuth 2.0

Endpoints Used:
- GET /API/Account/{accountID}.json - Get account details
- GET /API/Account/{accountID}/Sale.json - Get sales
- GET /API/Account/{accountID}/Item.json - Get products
- GET /API/Account/{accountID}/Customer.json - Get customers

Rate Limits:
- 30 requests per second
- 3,600 requests per hour
- Burst allowance: 60 requests

Documentation:
- Official Docs: https://developers.lightspeedhq.com/retail/
- API Explorer: https://developers.lightspeedhq.com/retail/endpoints/

Zapier API Reference

Webhooks by Zapier:
- Method: POST
- Custom Headers: Supported
- Custom Payload: Supported
- Response Handling: Automatic retry on 5xx errors
- Timeout: 30 seconds

Lightspeed Retail Trigger:
- Polling Interval: 1-15 minutes (plan dependent)
- Deduplication: Automatic via sale ID

- Data Fields: All sale fields available

Documentation:
- Zapier Developer: https://platform.zapier.com/
- Webhooks Help: https://help.zapier.com/hc/en-us/articles/8496288690317

■ Production Deployment

Pre-Deployment Checklist

Infrastructure:
- [ ] Production server provisioned (Render, Railway, Fly.io, etc.)
- [ ] Database configured and migrated
- [ ] Environment variables set
- [ ] SSL certificate configured
- [ ] Domain name configured

Configuration:
- [ ] Update APPURL to production domain
- [ ] Update Zapier webhook URL
- [ ] Configure production logging
- [ ] Set up error monitoring
- [ ] Configure backup strategy

Security:
- [ ] Environment variables secured
- [ ] Database credentials rotated
- [ ] API keys secured
- [ ] HTTPS enforced
- [ ] Rate limiting configured

Recommended Hosting Providers

Backend Hosting:
1. Render (Recommended)
- Easy deployment from GitHub
- Automatic SSL
- Built-in monitoring
- Free tier available

2. Railway
- Simple configuration
- PostgreSQL included
- Environment management
- Competitive pricing

3. Fly.io
- Global edge deployment
- PostgreSQL included
- Docker-based
- Free tier generous

Deployment Steps (Render Example):

1. Create Render Account:
- Sign up at render.com
- Connect GitHub repository

2. Create Web Service:
- Choose "New Web Service"
- Connect to your AskEuno repo
- Select branch to deploy

3. Configure Service:

yaml

```
Name: askeuno-backend
Environment: Node
Build Command: npm install
Start Command: npm start
```

4. Set Environment Variables:

```
DATABASEURL=your-production-db-url
PORT=5000
NODEENV=production
ENCRYPTIONKEY=your-secure-key
```

5. Deploy:
- Click "Create Web Service"
- Wait for deployment
- Note your production URL

6. Update Zapier:
- Go to Zapier
- Edit webhook action
- Update URL to production domain
- Test and republish

Post-Deployment

Verification:

bash

```
1. Test health endpoint
curl https://your-domain.com/health

2. Test webhook endpoint
curl -X POST https://your-domain.com/api/webhooks/lightspeed \
-H "Content-Type: application/json" \
-d '{"saleid":"prod-test-001","total":"50.00"}'

3. Check Zapier integration
Create test sale in Lightspeed
Verify in Zapier runs
Check database for new data
```

Monitoring Setup:
1. Configure uptime monitoring
2. Set up error alerts
3. Enable log aggregation
4. Configure performance monitoring

■ Support & Contacts

Key Resources

Documentation:
- This Handover Doc: /LIGHTSPEEDZAPIERINTEGRATIONHANDOVER.md
- Integration Docs: /LIGHTSPEEDINTEGRATIONDOCUMENTATION.md
- API Docs: /docs/API.md

Code Locations:
- Webhook Handler: server/routes/webhooks.ts
- Lightspeed Routes: server/routes/lightspeed.ts
- Frontend Setup: client/src/pages/lightspeed-setup.tsx
- Database Schema: shared/schema.ts

External Resources:
- Lightspeed API Docs: https://developers.lightspeedhq.com/retail/
- Zapier Platform: https://platform.zapier.com/
- ngrok Docs: https://ngrok.com/docs

Escalation Path

Level 1 - Configuration Issues:
- Check this documentation
- Review Zapier run logs
- Check server logs

Level 2 - Integration Issues:
- Review webhook implementation
- Check database schema
- Test with manual requests

Level 3 - Platform Issues:
- Contact Lightspeed Support
- Contact Zapier Support
- Check status pages

■ Version History

Version	Date	Changes	Author
1.0.0	2024-10-10	Initial integration setup	Development Team
1.1.0	2024-10-10	Added R-Series support	Development Team
1.2.0	2024-10-10	Production deployment ready	Development Team

---

■ Quick Reference Checklist

New Developer Onboarding
- [ ] Clone repository
- [ ] Install dependencies (npm install)
- [ ] Configure .env file
- [ ] Start database
- [ ] Run migrations
- [ ] Start development server
- [ ] Install and configure ngrok
- [ ] Test webhook endpoint locally
- [ ] Review Zapier configuration
- [ ] Test end-to-end flow

Production Deployment
- [ ] Choose hosting provider
- [ ] Set up production database
- [ ] Configure environment variables
- [ ] Deploy application
- [ ] Verify deployment
- [ ] Update Zapier webhook URL
- [ ] Test production integration
- [ ] Configure monitoring
- [ ] Set up alerts
- [ ] Document production URLs

Troubleshooting Quick Steps
1. Check server is running: curl localhost:5000/health
2. Check database connection: Review connection errors in logs
3. Test webhook locally: Use Postman or cURL
4. Check Zapier runs: Review Zapier dashboard
5. Check server logs: Look for webhook processing logs
6. Verify database: Query datasources and datarows tables

■ License & Credits

Integration developed for: AskEuno Platform
Technology Stack: Node.js, Express, PostgreSQL, Zapier
External Services: Lightspeed Retail, Zapier

For questions or support, please contact the development team or refer to the resources listed in this document.

Last Updated: October 10, 2024
Document Version: 1.2.0