External API Integrations

Configured External APIs

1. Agent Management System

Purpose: Manage agent locations and configurations

Configuration:

- Base URL: `API_URL_AGENT_ONAGENT`
- Authentication: JWT token
- Usage: Agent CRUD operations

2. User State Management

Purpose: Track user session and state

Configuration:

- Base URL: `API_URL_USERSTATE`
- Authentication: JWT token
- Usage: User activity tracking

3. User OnDelivery System

Purpose: User profile management

Configuration:

- Base URL: `API_URL_USERONDELIV`
- Authentication: JWT token
- Usage: User profile operations

4. Sunshine Location Service

Purpose: Location and area data

Configuration:

- Base URL: `API_URL_SUNSHINE`
- Authentication: API key
- Usage: Location lookup, province/city data

5. APISAT

Purpose: Satellite tracking (purpose unclear from code)

Configuration:

- Base URL: `API_URL_APISAT` and `APISAT_LINK`
- Authentication: API key
- Usage: External tracking integration

6. EatSOK

Purpose: Food delivery partner integration

Configuration:

- Base URL: `API_EATSOK`
- Authentication: Basic Auth
  - API Key: `EATSOK_API_KEY`
  - Secret: `EATSOK_SECRET_KEY_ONDEL`
- Special Features:
  - Separate image directories
  - Custom scanning workflow

7. Xendit Payment Gateway

Purpose: Payment processing and payouts

Features:

- Payment collection
- Payout disbursement
- Transaction tracking

Models:

- `xendit_payment` - Payment records
- `xendit_payout` - Payout records

8. Blibli Marketplace

Purpose: E-commerce integration

Configuration:

- Webhook Key: `BLIBLI_WEBHOOK_KEY`
- Authentication: Webhook signature validation
- Features: Order notifications, fulfillment

9. OnMarket Platform

Purpose: Marketplace platform integration

Configuration:

- Webhook Key: `ONMARKET_WEBHOOK_KEY`
- Authentication: Webhook signature validation
- Features: Multi-marketplace aggregation

API Utility Modules

Location: `app/config/api.js`

Purpose: Centralized API endpoint configuration

Usage:

```
const api = require('./config/api');


// Make external API call
const response = await axios.get(api.AGENT_SYSTEM + '/agents');
```

Webhook Handling

Incoming Webhooks:

1. **Courier Partner Webhooks**
   - Endpoint: `/api/3p/{partner}/webhook`
   - Authentication: Basic Auth or API Key
   - Purpose: Status updates from couriers
2. **Marketplace Webhooks**
   - Endpoint: `/api/marketplace/webhook`
   - Authentication: Webhook signature
   - Purpose: New order notifications

Outgoing Webhooks:

Utility: `app/utility/web-hook.js`

Purpose: Notify external systems of status changes

Configuration:

- Configured per partner/customer
- Triggered on status updates
- Includes retry logic

Example:

```
// Trigger webhook on status change
await triggerWebhook({
  awb: "ONX1234567890",
  status: "DELIVERED",
  timestamp: new Date(),
  location: "Jakarta"
});
```

Revision #1
Created 24 February 2026 06:56:00 by ondeliveloper
Updated 24 February 2026 06:57:19 by ondeliveloper