Scope of Work: Cin7 Integration

Overview

Nimble is a robotic 3pl business. Our customers are e-commerce merchants (B-to-B) who run storefronts to sell their products to their end consumers (B-to-C). Our merchant team is responsible for building integrations between these merchant storefronts (i.e. shopify, amazon marketplace, wish, skubana, bigcommerce, apparel magic, netsuite, etc.) and our robotic fulfillment system. Due to crazy demand and limited resources on the merchant team, we are looking for help from contractors to help us scale our development efforts here.

This document outlines the core deliverables we are expecting out of such a contracting group with the features we hope to build as part of these deliverables. These features will help empower our merchants and their experience with our robotic fulfillment system. Customer (consumer and merchant) experience is very important here.

For this particular integration, you will be building software to integrate with Cin7 which is a shipment records processing tool for one of our merchants (Maison De Sabre, we call them MDS for short). At a high level, when our robotic 3pl solution ships orders (packs into parcels and slaps on shipping labels so the parcel is ready to ship to the end customer's door step), we will want to forward this shipping information to Cin7. The scope of this project we anticipate to be small.

Deliverables

- Zip folder with ***GoLang*** code written to build web server (HTTP REST and/or GraphQL)
- 70% unit test coverage minimum
- Sequence diagrams to showcase how data will flow throughout the system from your web server to our internal system for each feature
- After all the above is reviewed and complete, we do rounds of integration testing between our softwares to make sure the deliverable number 1 works (for each feature)
- There are **0 infrastructure and deployment tooling requirements** here, since Nimble will run deliverable #1 within our own cloud infrastructure (kubernetes and AWS)
- There are **0 frontend development requirements** here since we will be purely integrating the storefront api to our existing backend service.

General Software Requirements

- Please benchmark the web service and provide average CPU and memory utilization metrics for an average workload (let's say 5 orders placed per minute for 24 hours total).
- Should be able to run on amd64 platform architecture (we use AWS EC2 instance type of m7g.large in our k8s cluster).
- Multi tenancy should be considered as part of building this web service. Each tenant here is the merchant which Nimble does business with. This means each tenant should have different app credentials on each of their Cin7 integrations.
- Rate limiting from the api should be considered as well (exponential backoff retry can be a solution here).
- Detailed and easy to understand logging/tracing should be included for debugging problems when they arise
    a. We prefer using opentelemtry for collecting logs (through HTTP) and traces (through HTTP/json), but this is not required (any observability solution, even stdout will suffice)
- We expect close to realtime processing for each feature (no cronjobs or polling needed).

Features

1. **Nimble:** After registering a "Orders Shipment" webhook on Nimble Cloud Logistics, when Nimble ships orders we will send a POST request to this webhook url (which would be your web service).
2. **Contractor:** The web service would then listen to HTTP POST requests on this endpoint and transform + forward this information to Cin7 via API
    a. There are finer details with how to transform the order details and shipment details from Nimble into a format which Cin7 can understand

Time Estimate

Would like all deliverables above to be completable in 2-3 weeks since our existing customers on our sales pipeline need this functionality in order for our business to go live with them