|  |  |  |
| --- | --- | --- |
| RAZDA Co. | | |
| **Filename: [shipping\_integration.py]** | | |
| **Summary***:*  Handles integration with a third-party shipping API for creating shipments, tracking packages, and updating orders with shipping status. Error handling, retry logic, and email notifications for critical issues are included. | | |
| ***Processes*** | | |
| * **Shipping Creation with Retry Logic:** | **create\_shipment initiates a shipment request with the shipping provider. It uses retry logic to attempt multiple API calls in case of failure and notifies admins after exhausting retry attempts.** | |
| * **Shipping Rate Calculation:** | **Calculates estimated shipping costs based on shipping details. This supports users by providing upfront shipping cost information, enhancing order transparency and facilitating price adjustments.** | |
| * **Shipping Alerts:** | **track\_shipment retrieves and logs the latest tracking status of a shipment using the provider’s API. It can alert the admin if the tracking retrieval fails, helping address shipment issues in real time.** | |
| * **Order Update with Tracking Information:** | **Once tracking information is retrieved, update\_order\_with\_tracking updates the corresponding order in the database, ensuring customers can view the latest shipment status.** | |
| * **Daily Automated Shipment Scheduling** | **schedule\_daily\_shipments automates daily shipment requests for ready orders, making the shipping process timely and efficient. This scheduled task ensures orders are shipped promptly** | |
| * **Admin Notifications for Failures** | **Sends alert emails via SMTP if critical operations (like shipment creation or tracking) fail. Alerts contain specific error information to facilitate quick admin response.** | |
| **Files it Gets Information From:** | | **Files it Sends too:** |
| * **Orders Database**: Accesses orders awaiting shipment via fetch\_orders\_ready\_for\_shipment. | * **Logs**: Creates detailed logs for shipment requests, tracking updates, and rate calculations, providing historical data and error tracking. | |
| * E**nvironment Configuration (.env)**: Accesses API and email configurations for secure handling of sensitive data. | * **Admin Alert System**: Sends email alerts to admins via SMTP for critical failures, enhancing response time to shipment issues | |
| * **Shipping API Endpoint**: Interacts with the third-party shipping API for all shipment-related functionalities (creation, rate calculation, and tracking). |  | |
| **Expected input into file:** | | **Expected output from file:** |
| * **Order Data**: Each shipment requires detailed order information like address, weight, dimensions, and service level to be processed successfully. * **API Credentials**: API keys are required for secure interaction with the third-party shipping API. * **SMTP Details**: SMTP configuration is necessary for sending admin alerts on failures. | | * **Shipment Confirmation**: Response from create\_shipment containing tracking details and shipment status, which are logged and updated in the order database. * **Tracking Information**: Retrieves and logs tracking status for shipments, keeping order data current and providing status visibility for admins and customers. * **Error Alerts**: Sends alerts for repeated failures to the configured admin email address, specifying the nature and timing of issues for troubleshooting. |
| **Things that need to be taking place:** | | |
| |  | | --- | | **- Robust Retry Logic for API Calls: create\_shipment uses retry logic with delay intervals for resilience against temporary network or API issues.** |  |  | | --- | | **- Scheduled Shipping Automation: schedule\_daily\_shipments enables automated daily shipment processing, reducing manual intervention and ensuring timely order dispatch.** |  |  | | --- | | **- Comprehensive Error Handling and Alerts: Admin notifications on shipment failures ensure swift response to disruptions, and detailed logging aids in debugging.** |  |  | | --- | | **- Secure Email Configuration for Alerts: Sensitive information like API keys and SMTP credentials are securely managed via environment variables, protecting against exposure.** |  |  | | --- | |  | | | |
| Edit log (update each time you make changes to doc or file). | | |
| | **Possible Enhancements:** | | --- |  |  | | --- | | - **Enhanced Shipment Delay Alerts**: Notify customers as well if tracking shows delays, improving transparency in case of issues with delivery timelines. |  |  | | --- | | - **Multi-Carrier Support**: Expand API integration to support multiple carriers, allowing more flexibility in selecting shipping options based on cost and efficiency. |  |  | | --- | | - **Batch Tracking Updates**: Implement a batch tracking feature to update the status of multiple orders at once, reducing load on the shipping provider API. |  |  | | --- | | - **API Call Performance Metrics**: Track and log API call performance, identifying bottlenecks or slow responses to further optimize shipping processes. | | | |
| * Oliver Smith (Razda Admin) Nov 8, 2024: | | |