|  |  |  |
| --- | --- | --- |
| RAZDA Co. | | |
| **Filename: [retry\_manager.py]** | | |
| **Summary***:*  Manages retry logic for failed operations, supporting conditions for specific retry cases, exponential backoff, and admin alerts if attempts are unsuccessful. | | |
| ***Processes*** | | |
| * **Retry Execution:** | **execute\_with\_retry initiates the retry logic, calling the target function and handling errors. Supports configurable retry conditions, delay intervals, and maximum attempts. Logs attempts and success/failure status for auditing.** | |
| * **Retry Conditions Check:** | **should\_retry examines retry conditions, allowing retries for specific exception types or for all errors if retry\_on\_all\_errors is set to True.** | |
| * **Exponential Backoff** | **Exponential backoff is enabled by setting backoff to True. Delay intervals double with each retry, reducing the frequency of retries and minimizing system load.** | |
| * **Admin Alerting on Failure:** | **alert\_admin sends an email notification when all retry attempts fail, alerting administrators for immediate attention. Includes detailed error information and traceback for diagnosis.** | |
| * **Logging and Traceback Support:** | **Each retry attempt, error type, and stack trace is logged, aiding troubleshooting and tracking function reliability. Tracebacks are logged in debug mode to reduce verbosity in standard logs.** | |
| * **SMTP Email Configuration:** | **SMTP settings are configurable via environment variables for secure email alerts. The system ensures critical errors are promptly reported to admins if retries are insufficient.** | |
| **Files it Gets Information From:** | | **Files it Sends too:** |
| * **Function Integrations**: Integrates with functions across modules needing retry mechanisms, such as API calls or database operations. | * **Alerts Module**: Can be linked with an alerting system for wider notifications beyond email. | |
| **Expected input into file:** | | **Expected output from file:** |
| * **Function and Conditions**: Receives a function to retry with optional retry conditions dictating which errors to retry on. * **Delay and Backoff Settings**: Custom delay and exponential backoff configurations to handle the time between retries. | | * **Retry Logs**: Logs each attempt, delay, and retry condition, providing visibility into function resilience. * **Alerts on Critical Errors**: Sends email alerts for functions failing all retry attempts, logging failure details and sending an admin alert if configured. |
| **Things that need to be taking place:** | | |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | |  | | --- | | **- Robust Retry Execution: execute\_with\_retry allows retrying critical functions under configurable conditions, reducing system downtime and ensuring reliability.** |  |  | | --- | | **- Flexible Retry Criteria: should\_retry supports specific error types or conditions, making it adaptable for diverse retry scenarios across Razda’s operations.** |  |  | | --- | | **- Efficient Delay Management: Exponential backoff in execute\_with\_retry ensures retries don’t overwhelm the system, allowing sensible handling of repeated failures.** |  |  | | --- | | **- Automated Failure Alerts: alert\_admin notifies administrators of consistent failures, supporting rapid response to issues affecting critical functions.** | | | | |
| Edit log (update each time you make changes to doc or file). | | |
| | **Possible Enhancements:** | | --- |  |  | | --- | | - **Granular Retry Thresholds**: Support for variable retry thresholds per error type, allowing more retries for recoverable errors and fewer for critical issues. |  |  | | --- | | - **Dynamic Delay Adjustments**: Intelligent adjustments to delay intervals based on system load, reducing retry frequency under high load and increasing it during idle times. |  |  | | --- | | - **Additional Alert Channels**: Adding support for SMS or chat notifications (e.g., Slack) for failures, providing multi-channel alerting for critical incidents. |  |  | | --- | | - **Error Categorization for Retry Analysis**: Aggregate retry data to categorize errors by type and frequency, allowing Razda to identify and address persistent issues. | | | |
| * Oliver Smith (Razda Admin) Nov 8, 2024: | | |