|  |  |  |
| --- | --- | --- |
| RAZDA Co. | | |
| **Filename: [error\_alerts.py]** | | |
| **Summary***:*  The error\_alerts.py file is responsible for monitoring error logs, tracking recurring errors, and alerting the admin team when certain error conditions are met. It checks logs for critical issues, parses errors, triggers alerts for severe or repeated errors, and provides a command-line interface to display or report recent errors. This module ensures that potential issues are addressed promptly, contributing to the stability and reliability of the Razda Market platform. | | |
| ***Processes*** | | |
| * **Initialize Error Monitoring** | * **Starts error log monitoring, watches for new error entries, and processes each line for issues.** | |
| * **Error Line Processing** | * **Reads each error line, parses the timestamp, severity level, and message, and categorizes the error appropriately.** | |
| * **Error Alert Triggering** | * **Triggers alerts for high-severity errors or critical system errors by communicating with the AlertManager.** | |
| * **Recurring Error Tracking** | * **Monitors error frequency and triggers a "Recurring Error" alert if the error message reaches a specified threshold of occurrences.** | |
| * **Error Display** | * **Provides a method to display the last 50 entries from the error log for quick admin access to recent issues.** | |
| * **Error Report Generation** | * **Generates a detailed report of errors within a specified date range, facilitating a comprehensive review of past issues.** | |
| **Files it Gets Information From:** | | **Files it Sends too:** |
| * **Environment Variables** (.env): Loads configuration such as ERROR\_LOG\_FILE path, RECURRING\_THRESHOLD. | * **Logging File** (error\_alerts.log): Records actions, successes, and failures related to error monitoring and alerting. | |
| * **Error Log File** (e.g., error.log): Monitors for new errors to process and analyze. | * **AlertManager** (alert\_manager.py): Triggers alerts for severe or recurring issues. | |
| * **CLI Input**: Accepts commands for monitoring, displaying, or reporting errors. |  | |
| **Expected input into file:** | | **Expected output from file:** |
| **Environment Variables**: Defines configuration options.  ‘’’python  ERROR\_LOG\_FILE, RECURRING\_THRESHOLD  ‘’’  **Error Log Entries**: Error messages with timestamps and severity levels.  **CLI Commands**: Executes specific monitoring, display, or reporting functions based on CLI input. | | **Log Records**: Logs error monitoring and alerts in error\_alerts.log.  ‘’’python  logging.info("Error '{message}' occurred {self.error\_counter[message]} times.")  ‘’’  **Alerts**: Triggers alerts via AlertManager for high-severity or recurring errors.  **Error Reports**: Generates console output of filtered errors for given date ranges. |
| **Things that need to be taking place:** | | |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  | | --- | | **● Real-Time Monitoring: Continuously monitors the error log for new entries, enabling prompt detection of potential issues.** |  |  | | --- | | **● Error Categorization: Differentiates between normal, severe, and critical errors, ensuring that only high-priority issues trigger alerts.** |  |  | | --- | | **● Alert Integration: Communicates with AlertManager to ensure admins are informed about critical issues or recurring errors, aiding in rapid response.** |  |  | | --- | | **● Recurring Error Thresholds: Tracks error message frequency and triggers alerts when specific thresholds are exceeded, helping identify persistent problems.** |  |  | | --- | | **● Command-Line Accessibility: Allows administrators to quickly view recent errors or generate reports, supporting ease of access and review.** |  |  | | --- | | **● Detailed Logging: Logs actions, alerts, and any issues encountered during monitoring to provide an audit trail and assist in troubleshooting.** | | | | |
| Edit log (update each time you make changes to doc or file). | | |
| Contribution to the Entire Project:  The **error\_alerts.py** file plays a critical role in Razda Market's error management strategy by providing continuous monitoring of system logs and notifying the admin team of severe or recurring issues. By setting up automated alerts and tracking error trends, this module enables proactive maintenance and swift resolution of potential problems. The error reporting functionality and recent error display further enhance the module’s utility, making it an essential part of the platform's stability and reliability infrastructure.  Please let me know if you would like to proceed with another file or have any additional questions. | | |
| * - Oliver Smith (Razda Admin) Nov 8, 2024: Enhanced recurring error tracking to notify after five occurrences; adjusted log file path settings. | | |