|  |  |  |
| --- | --- | --- |
| RAZDA Co. | | |
| **Filename: [error\_page\_handler.py]** | | |
| **Summary***:*  The error\_page\_handler.py file manages custom error pages, logs critical issues, and triggers alerts for specific error types (e.g., 404, 403, 500). It customizes error messages based on user roles (e.g., admin, market admin, guest) and integrates with error\_alerts.py and alert\_automation.py to enhance user experience by providing informative and role-specific error pages. This file ensures that issues are logged and critical alerts are raised for immediate attention. | | |
| ***Processes*** | | |
| * **Error Page Rendering** | **Renders custom error pages for common HTTP errors (400, 401, 403, 404, 500), tailored to user roles (admin, market admin, guest).** | |
| * **Trigger Alerts for Errors** | **Uses trigger\_alert function to log errors and trigger critical alerts for serious issues, leveraging error\_alerts and alert\_automation.** | |
| * **Role-Based Error Messages** | **Customizes error messages based on the user's role (e.g., guest vs. admin), enhancing the user experience for different types of users.f** | |
| * **Error Simulation** | **Provides routes to simulate errors (e.g., /simulate\_error/404) for testing custom error handling.** | |
| * **Log User Specific Errors** | **Logs detailed errors for specific users, aiding admins in tracking issues by user ID.** | |
| * **Dynamic Error Template Selection** | **Generates error templates that vary based on role and error type, ensuring relevant messaging for different user levels.** | |
| **Files it Gets Information From:** | | **Files it Sends too:** |
| * **Session Data** (session): Retrieves user role and ID to customize error messages. | * **Logging File** (error\_page\_handler.log): Logs all error-related activities, warnings, and alerts. | |
| * **Templates** (/Templates/): Retrieves HTML templates for different error pages (e.g., 404.html, 403.html). | * **Error Alerts** (error\_alerts.py): Sends error messages for display or alerts for critical issues. | |
| * **Alert Automation** (alert\_automation.py): Utilizes automation functions for critical alerts. |  | |
| **Expected input into file:** | | **Expected output from file:** |
| **Session and Request Data**: Uses session and request data to determine user role and request path.  ‘’’python   |  | | --- | | session['role'], request.path |  |  | | --- | |  |   ‘’’  **User Roles and Error Types**: Determines type of error and user role to select appropriate templates and messages.  **Simulated Error Codes**: Tests various error pages for development and QA. | | **Error Pages**: Renders error pages (e.g., 404, 403, 500) based on user role and error type.  ‘’’python  render\_template("generic\_error.html", code=code, message=message)  ‘’’  **Alert Logs**: Logs errors and triggers alerts via alert\_automation for critical issues.  **Critical Error Alerts**: Notifies critical issues to the alert\_automation.py module. |
| **Things that need to be taking place:** | | |
| |  | | --- | | **● Role-Based Error Customization: Ensures error pages are dynamically tailored to reflect user roles, enhancing clarity and relevance of messages.** |  |  | | --- | | **● Error Logging and Alerting: Logs each error occurrence and raises alerts for critical conditions, improving admin oversight and incident response.** |  |  | | --- | | **● Template Handling for Error Pages: Retrieves and displays role-specific templates (e.g., admin\_error.html, market\_admin\_error.html) to ensure informative and suitable error pages.** |  |  | | --- | | **● Error Testing Routes: Simulates various error scenarios during development to ensure proper handling and alerting mechanisms are in place for production.** |  |  | | --- | | **● Integration with Error and Alert Modules: Collaborates with error\_alerts.py for logging/displaying errors and alert\_automation.py for sending critical alerts, forming a cohesive alerting infrastructure.** |  |  | | --- | | **● User Error Management: Provides a mechanism to log and alert user-specific errors, aiding in tracking user-triggered issues.** |  |  | | --- | |  | | | |
| Edit log (update each time you make changes to doc or file). | | |
| Contribution to the Entire Project:  The **error\_page\_handler.py** file is essential for improving the user experience by delivering customized error pages based on user roles and providing detailed logs for administrative review. By linking with error\_alerts.py and alert\_automation.py, it ensures that critical errors are addressed promptly and that the platform maintains a high standard of user feedback and issue tracking. The file is also highly valuable for testing error-handling workflows, ensuring that all possible error scenarios are covered and appropriately managed.  Please let me know if you'd like to proceed with the next file or if there are further details to add. | | |
| * - Oliver Smith (Razda Admin) Nov 8, 2024: Implemented role-based customization for error pages; added integration with error\_alerts.py and alert\_automation.py for comprehensive alerting. | | |