

error_handling.md

Error Handling and Logging

In the Realtime Pool Voting App, it is important to have a robust error handling and logging strategy to ensure smooth operation and effective debugging. Here are the guidelines for handling errors and exceptions and documenting the logging strategy:

Error Handling

1. Validation Errors

- Validate user inputs and request payloads to ensure they meet the required criteria.
- Return appropriate error responses with descriptive messages and relevant status codes (e.g., 400 Bad Request) for validation failures.
- Include details about the specific validation errors encountered.

2. Authentication Errors

- Verify the Firebase ID token provided in the `Authorization` header for protected endpoints.
- Return a 401 Unauthorized response if the token is missing or invalid.

3. Resource Not Found Errors

- Check if the requested pool or resource exists in the database.
- Return a 404 Not Found response if the requested resource is not found.

4. Internal Server Errors

- Handle unexpected errors and exceptions gracefully.
- Return a 500 Internal Server Error response for unhandled exceptions.
- Log the error details for further investigation.

Logging Strategy

1. Log Levels

- Use different log levels to categorize the severity of logged events. Common log levels include:
 - DEBUG: Detailed information for debugging purposes.
 - INFO: General information about the application's operation.
 - WARNING: Indication of potential issues or unexpected behavior.
 - ERROR: Critical errors that require attention.
 - CRITICAL: Severe errors that may cause the application to crash.

2. Types of Logs to Capture

- Application Logs: Capture information about the application's operation, such as successful requests, important events, and key actions.
- Error Logs: Log details of errors and exceptions encountered during the application's execution.

- Request Logs: Record information about incoming requests, including request method, URL, headers, and payload.
- Response Logs: Log details of outgoing responses, including response status codes, headers, and payload.

3. Log Storage and Retention

- Determine where logs will be stored, whether in a local file, a centralized logging system, or a cloud-based logging service.
- Define the retention period for logs to balance storage costs and the need for historical data.

4. Error Handling Guidelines

- Include relevant error details in the logs, such as error messages, stack traces, and request/response information.
- Log errors at an appropriate log level based on their severity.
- Consider implementing structured logging to facilitate log analysis and filtering.

By following these error handling and logging guidelines, you can effectively handle errors, capture important events, and facilitate debugging in the Realtime Pool Voting App.