🌟🌟🌟 Day 122 🌟🌟🌟
of Our Daily UiPath Q&A Challenge!

Welcome to Day 122 of our exciting UiPath Q&A Challenge! Each day, we'll be answering one key question to help you master UiPath and revolutionize your automation journey. 🚀

⛔ Question for Today

How does RE-Framework handle system exceptions?

⚠️ What is a System Exception?

A system exception is an unexpected error that occurs due to issues like:

- Application crashes
- Selector not found
- Timeout errors
- Network failures
- File access issues

These are different from business exceptions, which are expected and related to business rules (e.g., invalid data).

🔄 How REFramework Handles System Exceptions

1. Try-Catch in Process.xaml

- The main business logic is wrapped in a Try-Catch block.
- If a system exception occurs, it is caught and passed to the SetTransactionStatus.xaml.

2. SetTransactionStatus.xaml

This is where the framework decides what to do next:

- Logs the exception using Log Message and optionally a custom logger.
- Screenshots can be taken for debugging (if implemented).
- Retries the transaction if the RetryNumber is less than the max retry count.
- Marks the transaction as Failed in Orchestrator if retries are exhausted.

3. Retry Mechanism

- Controlled by the MaxRetryNumber in Config.xlsx.
- If the transaction is from a **queue**, Orchestrator handles retries.
- If it's from a **data table**, REFramework handles retries manually.

4. Global Exception Handler (optional)

- You can implement a Global Exception Handler.xaml to catch unhandled exceptions across the entire project.
- Useful for logging, cleanup, or sending alerts before the process crashes.

5. TransactionItem Status

- For queue-based processes : SetTransactionStatus.xaml sets the status to Failed with the exception details.
- For non-queue processes : The framework logs the error and moves to the next item.

✅ Best Practices

- Use specific exception types in Try-Catch blocks when possible.
- Log detailed error messages including activity name, exception type, and message.
- Use screenshots and error codes for better diagnostics.
- Implement custom alerts (e.g., email, Teams, Slack) for critical failures.