🌟🌟🌟 Day 95 🌟🌟🌟
of Our Daily UiPath Q&A Challenge!

Welcome to Day 95 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

Explain Coded Workflow ?

Coded Workflow

- Coded workflows are the same as low-code workflows, the only difference being that you build them using separate interfaces:
- Workflows have a visual design interface.
- Coded workflows have a code-based interface.
- Additionally, you can integrate coded workflows with low-code activities and workflows, and use a hybrid automation approach. This enables you to combine the benefits of code-based automation with the visual design of low-code components.
- You can use Coded automations only in **Windows** and **Cross-platform** projects.
- Both **coded workflow** and **coded test case** automations use the CodedWorkflow partial class from the **UiPath.CodedWorkflows** package. This class gives the automation access to necessary interfaces for services (equal to activity packages), based on the installed activity packages in your project.

Visual Example of Coded Workflow

```
namespace GenerateTestData
{
    public partial class CodedWorkflow : CodedWorkflowBase
    {
        public CodedWorkflow()
        {
            _ = new System.Type[]{typeof(UiPath.Core.Activities.API.ISystemService),
        }

        protected UiPath.Core.Activities.API.ISystemService system { get => serviceCo

        protected UiPath.Testing.API.ITestingService testing { get => serviceContaine

        protected UiPath.UIAutomationNext.API.Contracts.IRuntimeTargetAppSyncFactory
    }
```