

UiPath Coded Agent handbook

This document provides a step-by-step guide on how to initialize and deploy a UiPath Coded Agent. It uses `uv` for dependency management and aims for a reproducible environment.

The guide is organized into four main sections:

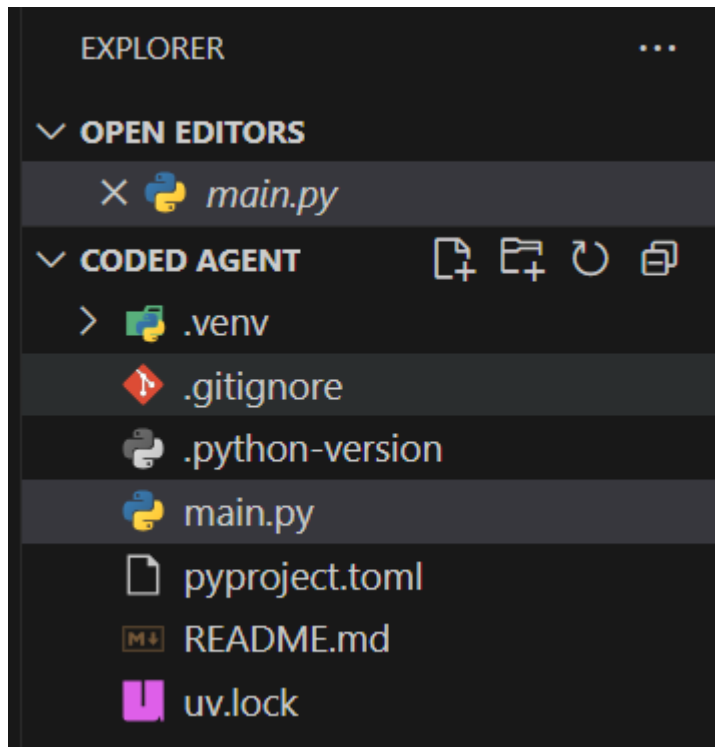
Prerequisites

Before starting the initialization process, ensure that the following tools are installed on your system.

- **Python:** A supported version of Python (e.g., Python 3.11 or newer).
- **VS Code:** IDE for developing the agent with python code

Virtual Environment Setup

- **Update pip**
`python.exe -m pip install --upgrade pip`
- **uv:** The high-performance Python package installer and resolver. You can usually install it via
`pip install uv.`
- **Create virtual environment**
`uv venv`
- **Activate virtual environment**
`.venv\Scripts\activate`
- **Initialize project**
`uv init`

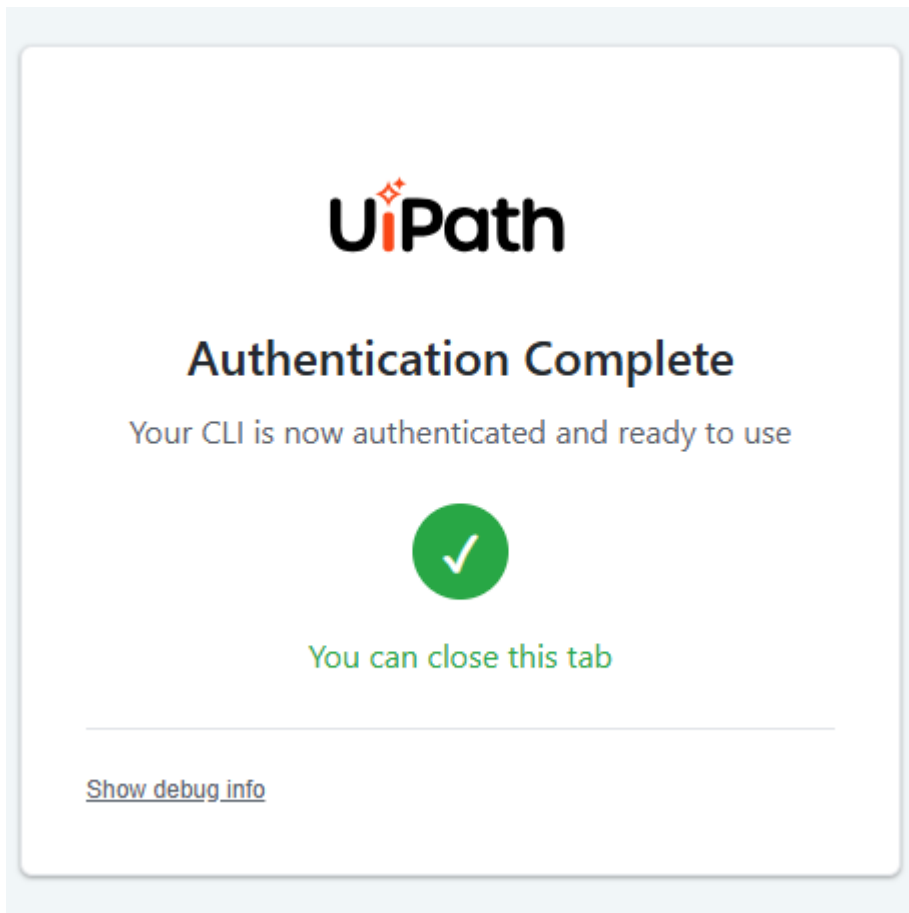


- Install UiPath LangChain package
`uv add uipath-langchain`

Building Agent

- Generate a quick-start project
`uipath new my-test-agent`
- Get authentication details
`uipath auth`

This will open the default browser on the machine and navigate to the UiPath Cloud platform url - [Cloud.UiPath.com](https://cloud.uipath.com) If already logged in, it will show the below popup.



If not logged in, the login window will be visible. Enter your login credentials and choose the Orchestrator tenant to give authorization. Upon successful authentication, you should be getting an Authentication Successful message in the terminal.

```
● (Coded Agent) PS E:\Coded Agent> uipath auth
" Authenticating with UiPath ...
🔗 If a browser window did not open, please open the foll
ea5b8-e8bb-423d-8e7b-c808df8f1c00&redirect_uri=http%3A%2F
estimatorApiUserAccess+StudioWebBackend+IdentityServerApi+
+Du.Extraction.Api+Du.Validation.Api+EnterpriseContextSer
es+TM.Requirements+TM.TestSets+AutomationSolutions&state=
jRW11czhE&code_challenge_method=S256
👉 Select tenant:
   0: DefaultTenant
Selected tenant: DefaultTenant
✓ Authentication successful.
```

- Create uipath.json with input/output schemas and bindings
`uipath init`

- (EventAgent) PS E:\EventAgent> **uipath**
• Initializing UiPath project ...
'uipath.json' already exists, skipping
'bindings.json' already exists, skipping
✓ Created 'entry-points.json' file with 1 entry point.
✓ Created 1 mermaid diagram file(s).
✓ Created: CLAUDE.md, CLI_REFERENCE.md, E.md

- Test the agent by running it locally
`uipath run agent '{"topic":"uipath"}'`

- (Coded Agent) PS E:\Coded Agent> **uipath** run agent '{"topic":"uipath"}'
▶ START
HTTP Request: GET https://cloud.uipath.com/assistmaigic/D
HTTP Request: POST https://cloud.uipath.com/assistmaigic/D
● generate_report
state
└─ report: **Report on UiPath**

 Introduction
 UiPath is a leading Robotic Process Automation (RPA) platform for automating business processes. Founded in 2005, UiPath has grown significantly and is now a global leader in the RPA market.
● END
output
└─ report: **Report on UiPath**

 Introduction
 UiPath is a leading Robotic Process Automation (RPA) platform for automating business processes. Founded in 2005, UiPath has grown significantly and is now a global leader in the RPA market.
✓ Successful execution.

Publish Agent

- Update `.toml` file

```
pyproject.toml
[project]
name = "test-agent"
version = "0.0.1"
description = "test-agent"
authors = [{ name = "Ashok Karale", email = "assis"
dependencies = [
    "uipath-langchain>=0.0.158",
]
requires-python = ">=3.11"
```

- Pack the agent to deploy it over the Orchestrator
`uipath pack`

```
(Coded Agent) PS E:\Coded Agent> uipath pack
⋮ Packaging project ...
Name       : test-agent
Version    : 0.0.1
Description: test-agent
Authors    : John Doe
✓ Project successfully packaged.
```

- Publish the agent to the Orchestrator by selecting the package feed
`uipath publish`

```

● (Coded Agent) PS E:\Coded Agent> uipath publish
: Fetching available package feeds...
👉 Select package feed:
  0: Orchestrator Tenant Processes Feed
  1: Orchestrator Personal Workspace Feed
Select feed number: 0
Selected feed: Orchestrator Tenant Processes Feed
: Publishing most recent package: test-agent.0.0.1.nupkg
✓ Package published successfully!

```

- See the published agent from the Orchestrator

The screenshot shows the UiPath Orchestrator web interface. On the left sidebar, the 'Tenant' tab is highlighted with a green box. Below it, there are sections for 'My Folders' and 'My Workspace'. The main content area has a top navigation bar with tabs: 'Monitoring', 'Robots', 'Folders', 'Manage Access', and 'Machines'. Under 'Monitoring', there are sub-tabs: 'RPA', 'Libraries', 'Agentic processes', and 'Agents'. The 'Agents' sub-tab is highlighted with a green box. Below the sub-tabs, there is a search bar and a table of agents. The table has columns for 'Name' and 'Runtime'. The first agent listed is 'test-agent', which is highlighted with a green box. Other agents listed include 'AssistmAlgic', 'ReviewEmailResponseSolution.Agent.TestAgent', and 'Solution.Agent.TestAgent'. The table shows 4 agents in total, with the first one being 'test-agent'.

- The agent can then be seen and run from the Orchestrator or other processes (like Agentic Process)

Documentation to refer: [Getting Started - UiPath SDK](#)