Technical Report: Onboarding Optimization for Contributors (Dev Branch)
Project: PyModbus |**Target Version:** v4.0.0 (dev)
Author: Federico Musa | **Date:** January 28, 2026

1 Executive Summary

Objective: To evaluate and optimize the initial setup process for new developers aiming to contribute to the dev branch (v4.0.0).

Key Finding An opportunity to reduce friction during installation was identified. Currently, the standard documented setup does not enable the immediate execution of example scripts due to the modular package structure in the development version. This results in a blocking import error for users setting up the environment for the first time.

Proposed Solution Update the documentation to recommend **"Editable Mode" (-e)** installation as the standard for development, and refine the error messages within the examples to better guide the user.

2. Test Environment

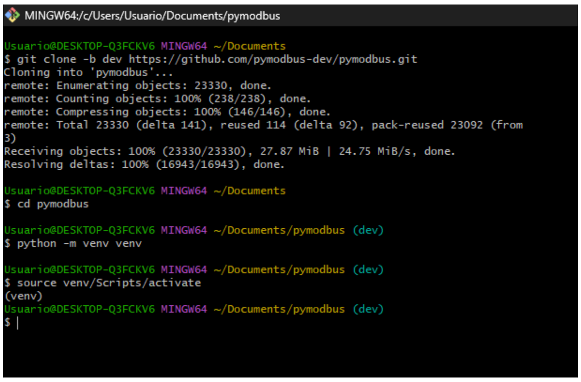
Tests were conducted simulating a "Clean Install" scenario for a new contributor.

- **Operating System:** Windows / Standard Environment.
- **Hardware:** Workstation (Intel i5 13th Gen).
- **Language:** Python 3.13.5.
- **Repository Branch:** dev.
- **Methodology:** Repository cloning, virtual environment (venv) creation, and direct execution attempt of example scripts (e.g., examples/server_async.py).

3. Discovery Methodology

Step 1: Cloning and Environment Setup

A clean clone of the repository was performed, and a fresh virtual environment (venv) was created to simulate a first-time contributor scenario.



```
Usuario@DESKTOP-Q3FCKV6 MINGW64 ~/Documents
$ git clone -b dev https://github.com/pymodbus-dev/pymodbus.git
Cloning into 'pymodbus'...
remote: Enumerating objects: 23330, done.
remote: Counting objects: 100% (238/238), done.
remote: Compressing objects: 100% (146/146), done.
remote: Total 23330 (delta 141), reused 114 (delta 92), pack-reused 23092 (from
3)
Receiving objects: 100% (23330/23330), 27.87 MiB | 24.75 MiB/s, done.
Resolving deltas: 100% (16943/16943), done.

Usuario@DESKTOP-Q3FCKV6 MINGW64 ~/Documents
$ cd pymodbus

Usuario@DESKTOP-Q3FCKV6 MINGW64 ~/Documents/pymodbus (dev)
$ python -m venv venv

Usuario@DESKTOP-Q3FCKV6 MINGW64 ~/Documents/pymodbus (dev)
$ source venv/Scripts/activate
(venv)
Usuario@DESKTOP-Q3FCKV6 MINGW64 ~/Documents/pymodbus (dev)
$ |
```

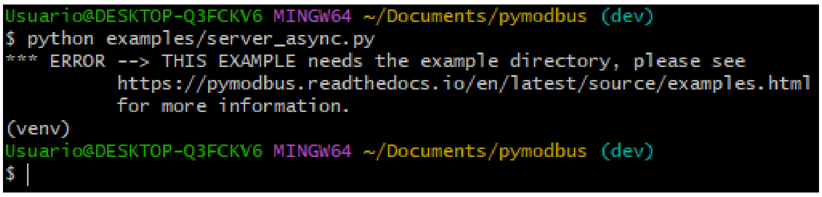
Step 2: Standard Execution Attempt

Attempting to run an example script immediately after setup, following standard Python developer intuition

Command:

python examples/server_async.py

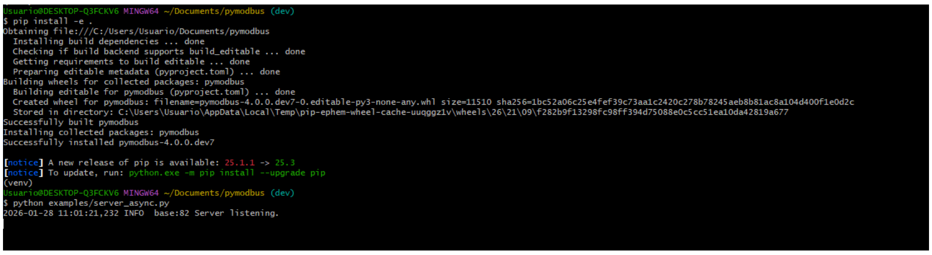
Observation: The script fails immediately. The internal validation logic triggers an error indicating the package is missing, despite the code being present in the folder.



```
Usuario@DESKTOP-Q3FCKV6 MINGW64 ~/Documents/pymodbus (dev)
$ python examples/server_async.py
*** ERROR --> THIS EXAMPLE needs the example directory, please see
          https://pymodbus.readthedocs.io/en/latest/source/examples.html
          for more information.
(venv)
Usuario@DESKTOP-Q3FCKV6 MINGW64 ~/Documents/pymodbus (dev)
$ |
```

Step 3: Validation of the Fix (Editable Install)

pip install -e .



```
Usuario@DESKTOP-Q3FCKV6 MINGW64 ~/Documents/pymodbus (dev)
$ pip install -e .
Obtaining file:///C:/Users/Usuario/Documents/pymodbus
  Installing build dependencies ... done
  Checking if build backend supports build_editable ... done
  Getting requirements to build editable ... done
  Preparing editable metadata (pyproject.toml) ... done
Building wheels for collected packages: pymodbus
  Building editable for pymodbus (pyproject.toml) ... done
  Created wheel for pymodbus: filename=pymodbus-4.0.0.dev7-0.editable-py3-none-any.whl size=11510 sha256=1bc52a06c25e4fef39c73aa1c2420c278b78245aeb8b81ac8a104d400f1e0d2c
  Stored in directory: C:\Users\Usuario\AppData\Local\Temp\pip-ephem-wheel-cache-uuqggz1v\wheels\26\21\09\f282b9f13298fc98ff394d75088e0c5cc51ea10da42819a677
Successfully built pymodbus
Installing collected packages: pymodbus
Successfully installed pymodbus-4.0.0.dev7

[notice] A new release of pip is available: 25.1.1 -> 25.3
[notice] To update, run: python.exe -m pip install --upgrade pip
(venv)
Usuario@DESKTOP-Q3FCKV6 MINGW64 ~/Documents/pymodbus (dev)
$ python examples/server_async.py
2026-01-28 11:01:21,232 INFO  base:82 Server listening.
```

4. Root Cause Analysis (RCA)

A. Internal Validation Logic ("The Gatekeeper")

The example scripts in the dev branch include a safety check. They attempt to import pymodbus to ensure dependencies are met. If the Python interpreter cannot find the package registered in sys.path, it aborts with the error message observed in Figure 2.

B. Pathing in Modular Architectures

In the dev branch, the project uses a modern modular structure. Python does not automatically add subdirectories to the search path just because the user is in the root folder. Without an installation step, import pymodbus fails because the namespace is not linked to the venv.

C. Documentation Gap (Implicit Context)

The root cause is a documentation gap regarding the "Install with github" section.

- Existing contributors likely have the environment pre-configured.
- The README instructions (pip install ".[development]") perform a static install if the -e flag is omitted.
- The documentation states that the install creates "pointers directly to the pymodbus directory", but technically, this only happens if installed with -e.

5. Resolution & Recommendations

5.1 Recommended Fix: Update README.rst

The documentation should be updated to explicitly include the -e (editable) flag. This ensures the "pointers" mentioned in the text are actually created.

- **Current:** pip install ".[development]"
- **Proposed:** pip install -e ".[development]" (or pip install -e . as a preliminary step).

5.2 Developer Experience (DX) Improvement

To prevent confusion for future contributors, the error message in the example scripts should be more actionable.

Current Message:

```
*** ERROR --> THIS EXAMPLE needs the example directory...
```

Proposed Message:

```
*** ERROR --> PyModbus package not detected. If you are working on the 'dev' branch, please ensure you installed the package in editable mode: pip install -e .
```