How to Setup VSCode Frappe Debugger

Debugging
</erpnext.projects.doctype.task.task.task>
https://frappeframework.com/docs/user/en/debugging

▼ Prerequisites

1. Visual Studio Code
2. Remote - SSH extension
3. Python extension (don't forget to install in on the SSH session too)

▼ Steps without explanation

1. Open Frappe bench directory in VSCode (SSH session). This will be the **workspace folder**
2. Open the `Procfile` file in the bench directory, and comment out the line `web: bench serve --port 8000`
3. Open the `Run and Debug` panel, create a `launch.json`, choose Python. Then, it does not matter what you choose for the last prompt
4. Open the `launch.json` file, and replace the `configurations` with this:

```
{
    "name": "Bench",
    "type": "python",
    "request": "launch",
    "program": "${workspaceFolder}/apps/frappe/frappe/utils/bench_helper.py",
    "args": ["frappe", "serve", "--port", "8000", "--noreload", "--nothreading"],
    "python": "${workspaceFolder}/env/bin/python",
    "cwd": "${workspaceFolder}/sites",
    "env": {
        "DEV_SERVER": "1"
    }
}
```

5. After saving the file, you can start debugging now

▼ Steps with explanation

Debugging in VS Code for Frappe:

(Assuming you already opened the bench directory using SSH session)

1. Update Procfile:

- In your bench directory, locate the `Procfile`.
- Comment out the line that looks like this: `web: bench serve --port 8000`. This is because you'll run this process from VS Code instead of using `bench start`.

2. Update launch.json in VS Code:

▼ If you haven't created one...

- Open the `Run and Debug` panel
- Create a new `launch.json`
- Choose Python, then choose anything
- `launch.json` file should be created & opened

- Add/replace the following configurations to your `launch.json` in VS Code:

```
{
        "name": "Bench",
        "type": "python",
        "request": "launch",
        "program": "${workspaceFolder}/apps/frappe/frappe/utils/bench_helper.py",
        "args": ["frappe", "serve", "--port", "8000", "--noreload", "--nothreading"],
        "python": "${workspaceFolder}/env/bin/python",
        "cwd": "${workspaceFolder}/sites",
        "env": {
            "DEV_SERVER": "1"
        }
    }
```

This configuration assumes that your `bench` directory is set as your workspace directory in VS Code. If not, adjust the paths (`workspaceFolder`, `pythonPath`, and `cwd`) accordingly.

▼ **Understanding the Configuration**

- `"name": "Bench"`: This is just a name for the debugging configuration.
- `"program": "${workspaceFolder}/apps/frappe/frappe/utils/bench_helper.py"`: This points to the `bench_helper.py` file in your Frappe app. The `${workspaceFolder}` variable automatically points to the root of your current opened directory in VS Code.
- `"args": ["frappe", "serve", "--port", "8000", "--noreload", "--nothreading"]`: These are the arguments passed to the program, replicating the command `bench serve --port 8000 --noreload --nothreading`.
- `"pythonPath": "${workspaceFolder}/env/bin/python"`: This points to the Python executable in your bench's virtual environment.
- `"cwd": "${workspaceFolder}/sites"`: This sets the current working directory for the debugger. The command should be executed from the `sites` directory.
- `"env": {"DEV_SERVER": "1"}`: This sets an environment variable required for Frappe. It is required for the correct functioning of Socket.io in Frappe. This is standard and should be correct.

3. Start Debugging in VS Code:

- Go to the Debug Panel in VS Code (`⌘⇧D`).
- Start debugging by selecting the "Bench" configuration and pressing the green play button or using the keyboard shortcut (`⌘⇧F5`).

Explanation:

- The `program` and `args` in the configuration replicate the command `bench serve --port 8000 --noreload --nothreading`. This starts the Frappe server with specific options.
- The `cwd` (current working directory) ensures that the command is executed from the `sites` directory.
- The `env` sets an environment variable `DEV_SERVER` to `1`. This is required for the correct functioning of Socket.io in Frappe.

Caveats:

- The VS Code Debugger might not work well with the `use_reloader=True` and `threaded=True` options. This is a known issue with other frameworks like Django and Flask as well.