

A fast end-to-end test framework powered by Robotframework

 github.com/maple24/vta

Vite Test Automation Framework(VTA)

English | 简体中文






version 1.0.0

Windows

python

code style black

Features

-  Python code formatter with Black
-  Run test cases with Robot Framework
-  Interacting database with SQLAlchemy Engine
-  Run keywords asynchronously [GeventLibrary]<https://github.com/eldaduzman/robotframework-gevent>
-  An extremely fast Python linter Ruff
-  Out-of-box RQM API
-  Mail with smtp server
-  Delete logs automatically
-  Light-weighted, less than 2MB

Movitation

1. Run functional test with RF, which has sudo-code like syntax and can generate beautiful report automatically.
2. A pressure test is just running one functional test multiple times with no extra effort.
3. Easy to debug with out-of-box logger tools, and can be easily parsed or data-analysed afterwards.
4. Test cases are compatible with both RF or Python.
5. Hardwares are separated from framework which can be called with APIs.
6. Email and RQM APIs are well structured for efficient-first purpose.
7. Use built-in modules, to keep it as light-weighted as possible.
8. Remove database and data statistics from robot to let user focus on test only.

Prerequisites

python >= 3.8

Quick start

1. clone project

```
git clone https://github.com/maple24/vta.git
```



2. create python virtual environment

```
cd vta  
python -m venv .venv
```



3. activate venv and install dependencies

```
.venv/scripts/activate  
pip install -r requirements.txt
```



4. run HelloWorld.bat and check logs in /log

5. (optional) set your pythonpath for robotframework

Go to the Robot Framework Language Server extension and go to extension settings
In there you'll find: Robot > Language-server: Python



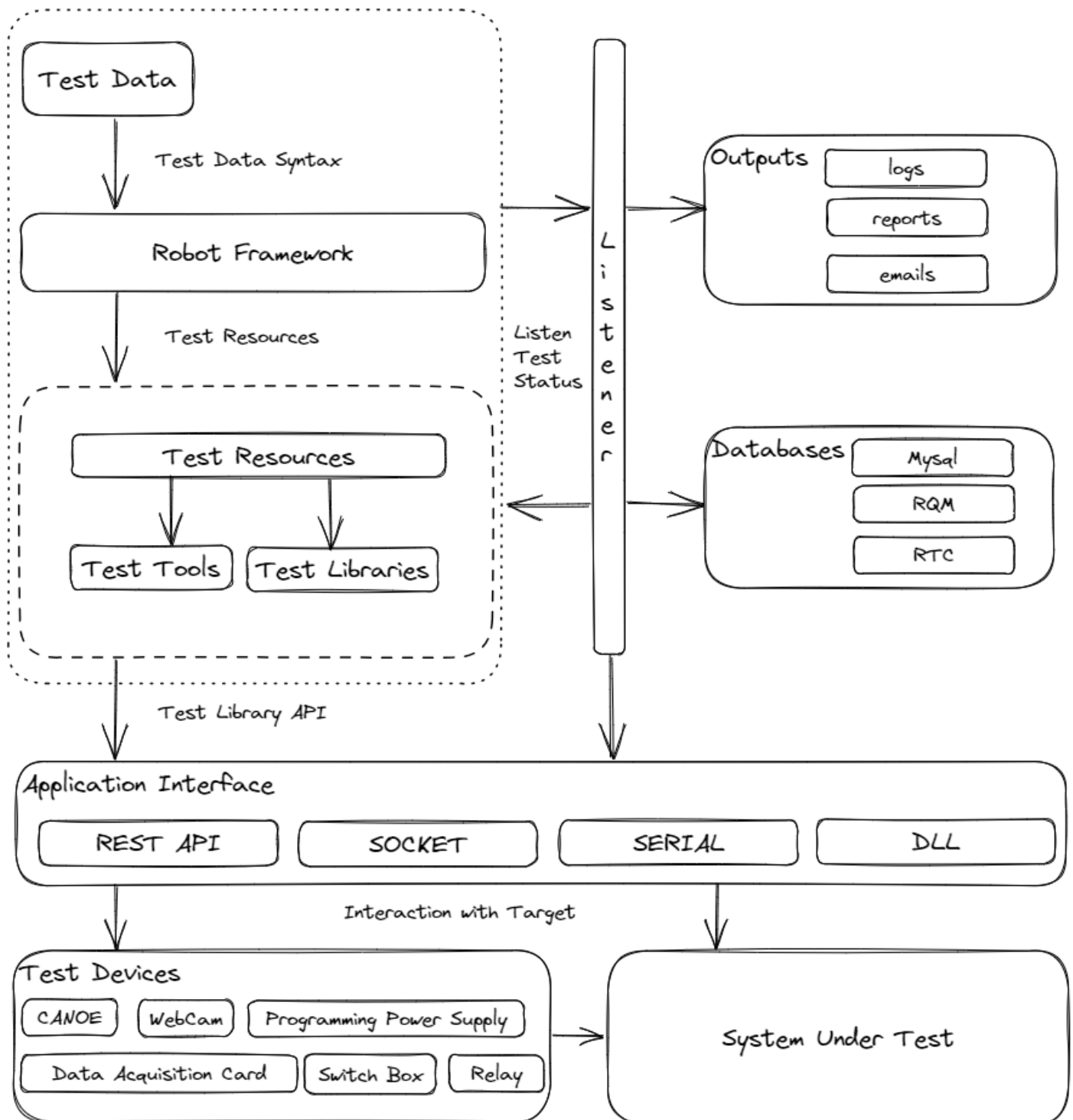
6. format your code

```
python -m black .\vta\
```

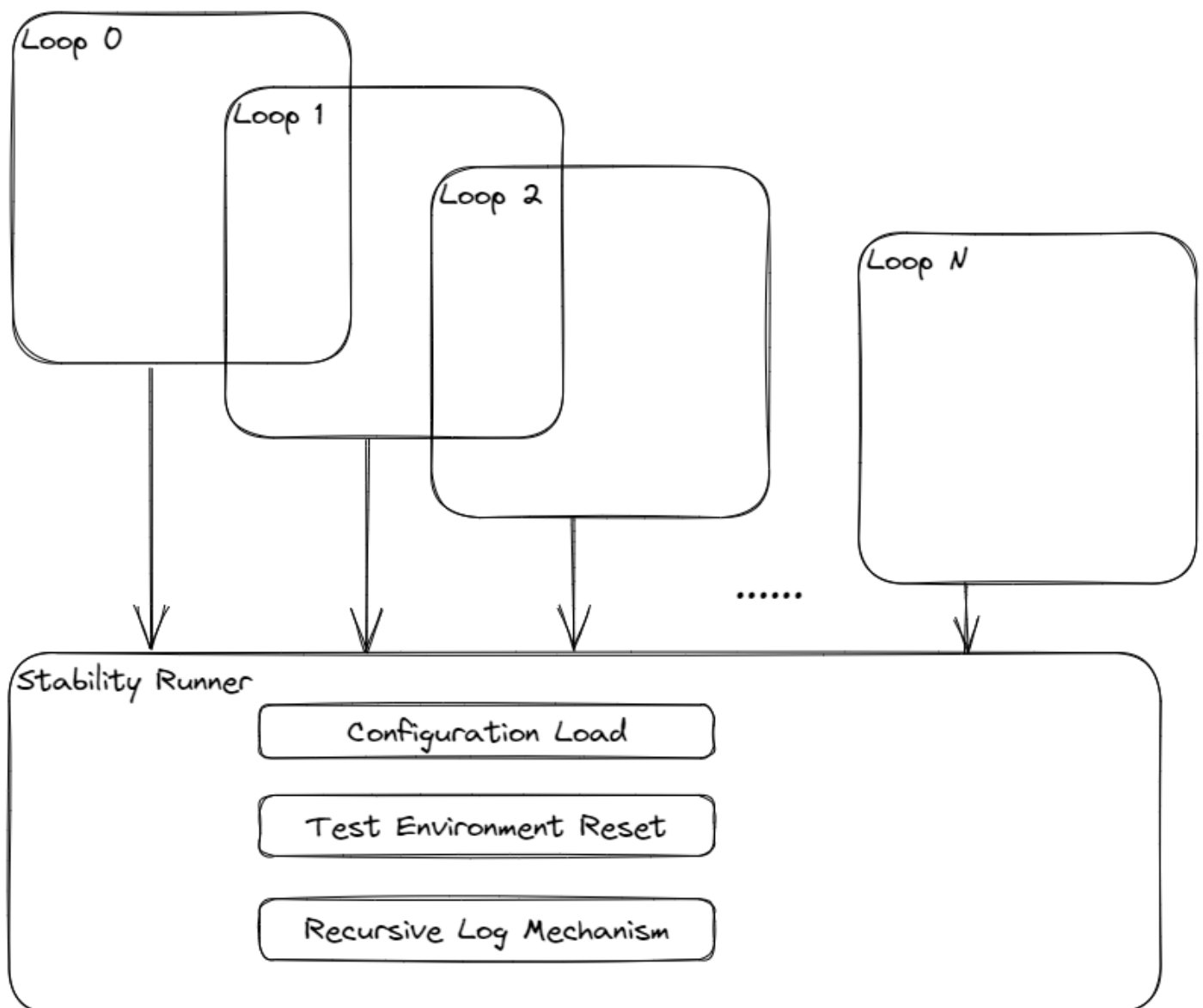


Architecture

1. Functional test building block



2. Stability test building block



- Helpers: pure functions/classes/apis in independent files, no relative import needed
- Managers: higher level functions which use Helpers to buildup

License

License [Apache 2.0](#)