

# Postman Automation

---

Automating the postman using the newman

## 🔗 How to Install

---

### Pre-requisite

---

- Just install the following binaries, and you are good to go
  - Python3
    - Follow the link to download and install:  
(<https://www.python.org/downloads/>)
  - NodeJs
    - Follow the link to download and install:  
(<https://nodejs.org/en/download/>)
  - newman
    - Just run the following command after installing NodeJS on your system

```
npm install --location=global newman
```

## Configuring the collection and environment files

---

### Files placement (Collections + Env File)

- Place your collection files and its environment in a specific category under suites dir; suites / <category> /  
[place\_your\_collection\_file\_here]
- After placing your files into specific category, update this info under the core / config / config\_api\_collections\_and\_data.py
  - Here you simply provide the collection file name under the api\_collections
  - and environment file into api\_environeemt

## Modifying the config.py

- After placing the files, you just need to tell the `core / config / config.py` :
  - Which suite you want to target for the execution
  - Which environment to use for the execution
  - What are the parent locations for suites and its files
  - Which report you are going to use and generate
  - And few extra things if you want to customize something else [Not necessary to set]

## Making sure, that cli run without option is True

- For this make sure that under `core / config / config.py` file
  - `"RUN_WITHOUT_OPTIONS"` is `True`
    - like: `framework = { "RUN_WITHOUT_OPTIONS": True ...`

## How to run the framework

---

- Open up the terminal/command prompt in your system and execute following command:

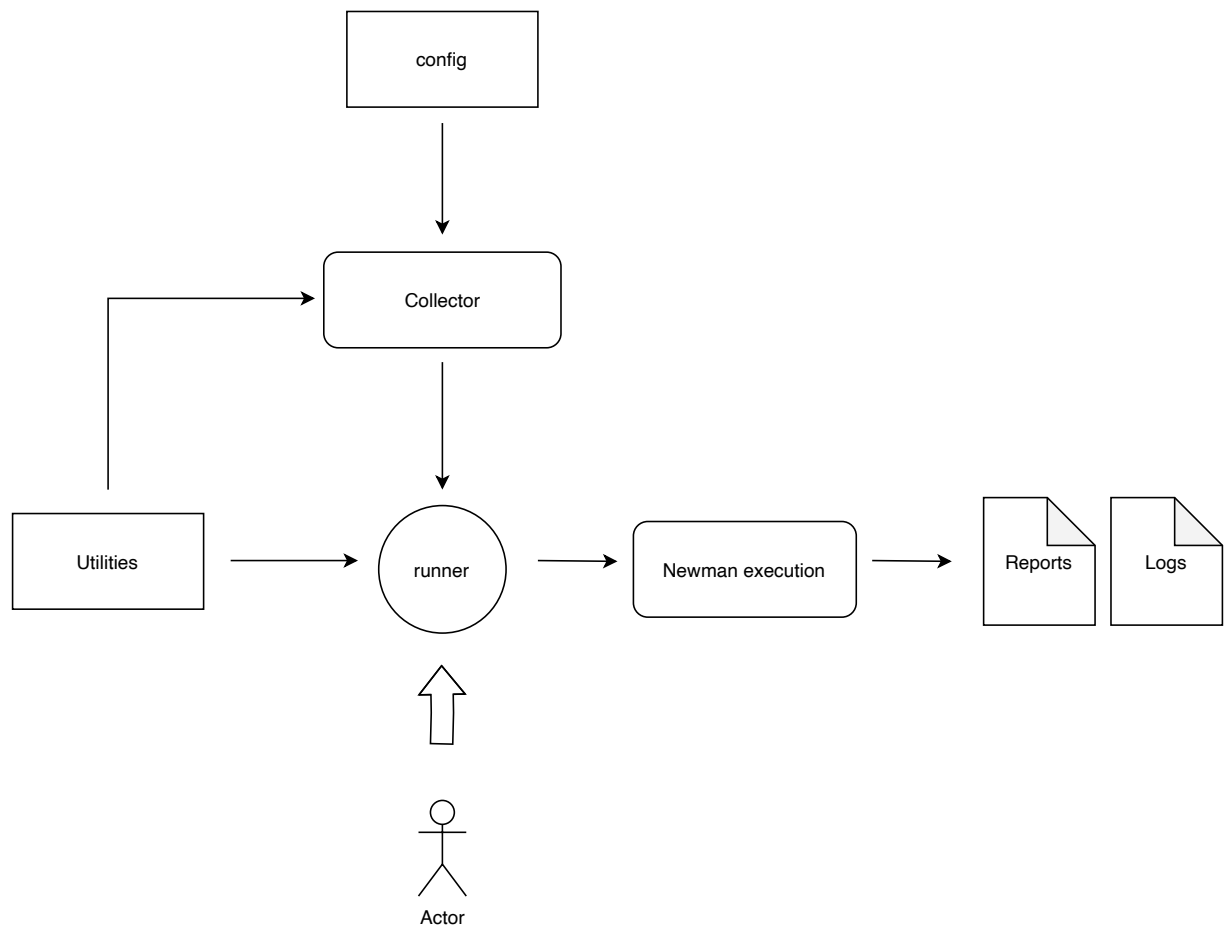
```
python runner.py
```

- In order to get to see the available option, use the following command

```
python runner.py --help
```

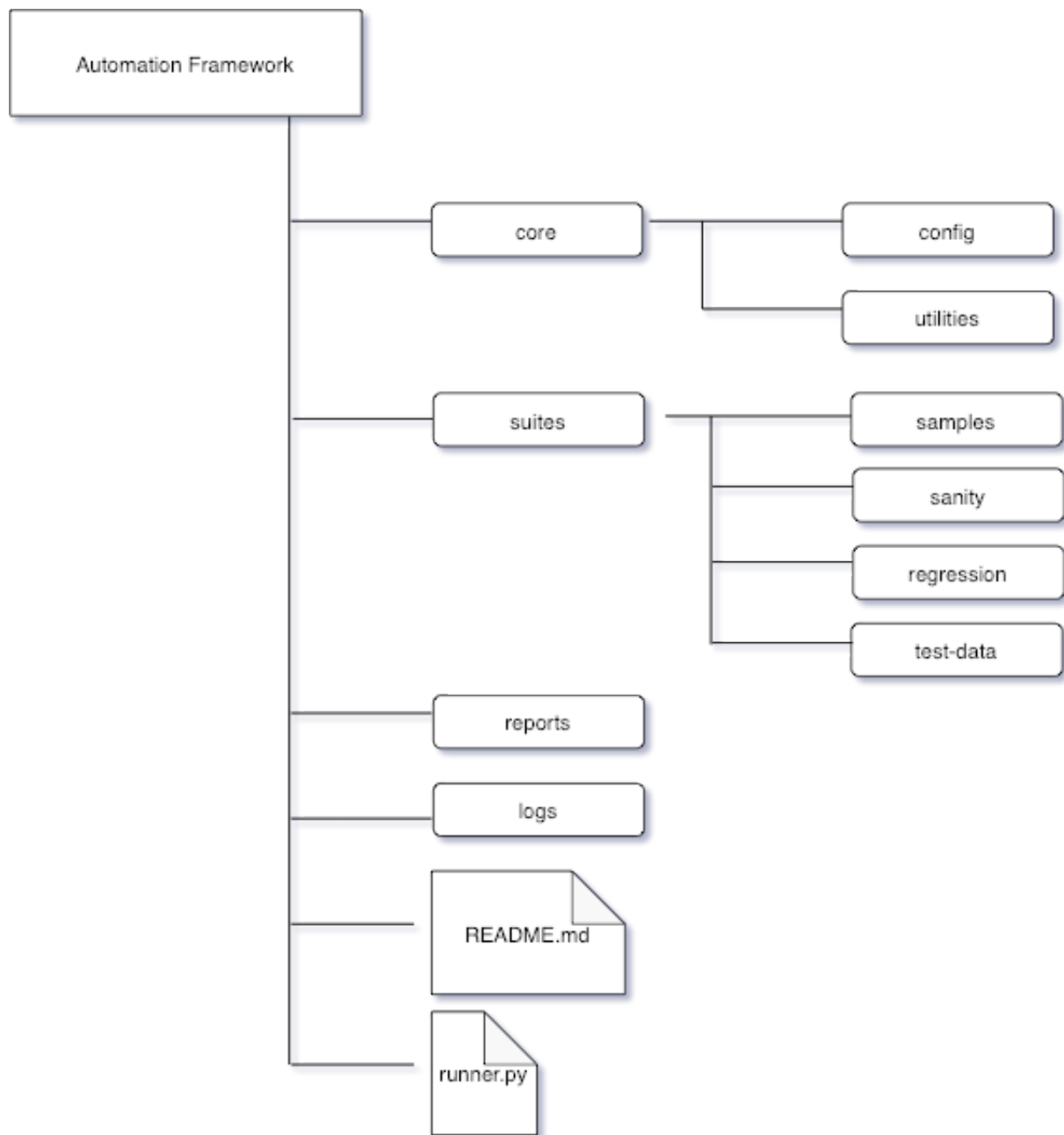
## Architecture

---



## Folder Structure

---



## Example:

`samples` has been added as sample api collection under `suites / samples / <some_sample.json> + <some_env.json>`

Now consider `samples` as a suite; and hence following changes has been made in the config

- Changes inside the `config_api_collections_and_data.py`
  - Path for the `samples` suite has been added inside

```
collections_suite_path = { "samples": "./suites/samples/",  
...
```

- Environment file has been added under `api_environment` key.
- Collections file has been added under `api_collections =`  
`{"samples": ["<Collection_file_name_1>", "`  
`<Collection_file_name_k>", ...], ...`
- Changes inside the `config.py`
  - `samples` has been added into `"TEST_SUITES_TO_EXECUTE":`  
`["samples"]`, under the `framework`

### And then execute

```
python3 runner.py
```

### Look out the generated reports

- Under the `reports / dir`
  - You will see two files
  - `<*.html>` is HTML report
  - `<*.xml>` is the JUnit report

### Look out the generated logs

- Under the `logs / dir`
  - You will see various log files based on the time and date

### End