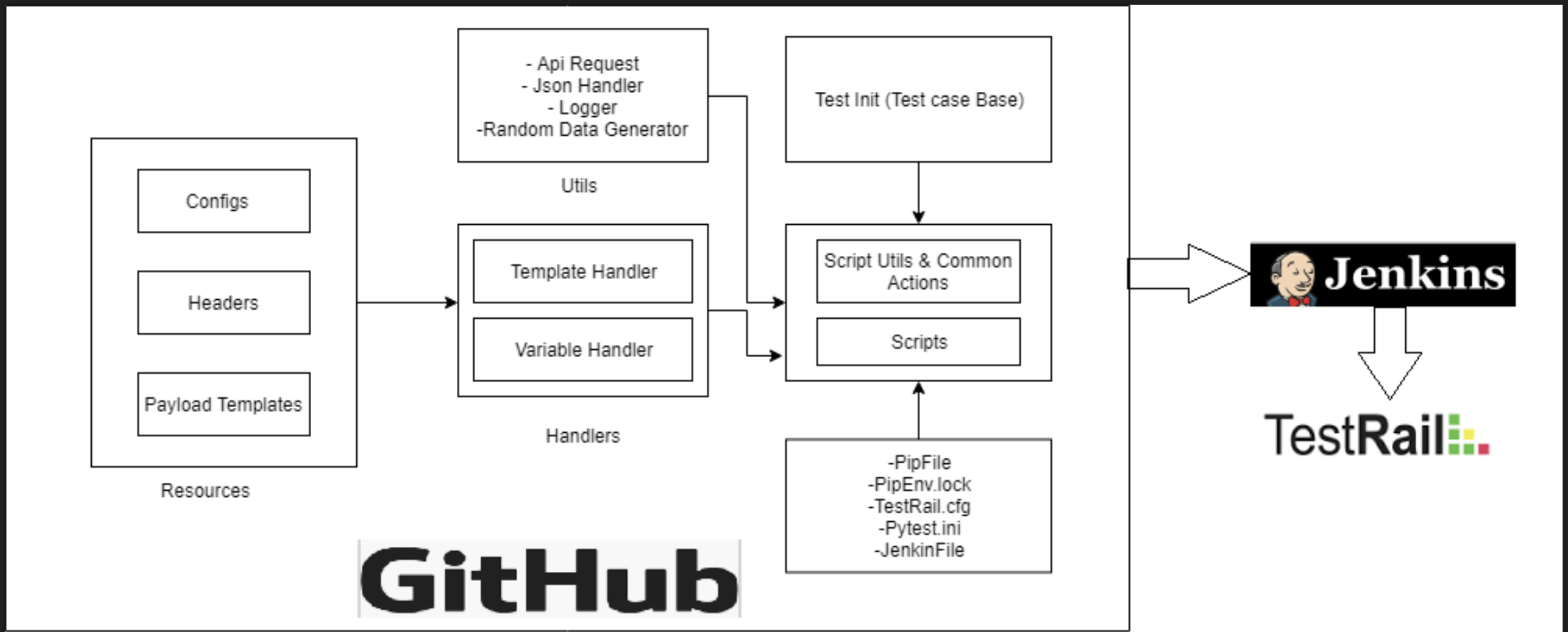


Api Automation Framework

Powered By – Pytest, Jenkins, Testrail

Framework Architecture Diagram



Resource Handler

- <Environment_name>.json file
- Contains all the variables used in the particular environment

```
{  "$tenant_id": "tenantid1",  "$tenant_name": "testtenant",  "$tenant_token": "tenanttoken1",  "$vendor_id": "vendorid1",  "$vendor_name": "testvendor",  "$vendor_token": "vendortoken1",  "$tenant_host": "summon.mocklab.io",  "$vender_host": "summon.mocklab.io",  "$device_id": "1234",  "$rental_id": 1234,  "$rfid_key": "1234",  "$rfid_val": "1234",  "$ble_key": "ble_key11",  "$baseurl": "https://eu.prod.svc.ridecell.io",
```

Resource Handler

Api_paths.json – irrespective of the environment we have all api paths

```
{  
  "si_action": "/hardware-integration/client/api/v3/asset/#asset_id/action",  
  "vi_action": "/hardware-integration/vendor/api/v3/asset/#asset_id/action",  
  "drop_tables": "/hardware-integration/api/v3/droptables",  
  "create_tenant": "/hardware-integration/api/v3/tenant/",  
  "create_vendor": "/hardware-integration/api/v3/vendor/",  
  "create_asset": "/hardware-integration/api/v3/asset/",  
  "get_asset": "/hardware-integration/api/v3/asset/#asset_id",  
  "mensa_push_synthesis": "/vendor-mensa/api/v1/synthesis",  
  "vendor_mensa": "/vendor-mensa/api/v3/asset/#asset_id/action"  
}
```

Resource Handler

- Variable with '\$' are evaluated by Variable handler.
- Variable with '\$' are searched in the environment file.

```
1 {  
2   "Authorization-Microservice": "$microservice_token_sum",  
3   "Application-Id": "$application_id_sum"  
4 }
```

```
1 {  
2   "Authorization-Microservice": "$microservice_token_sum",  
3   "Application-Id": "$application_id_sum"  
4 }
```

Resource Handler

- Payload Template . All variables with '\$' are evaluated from environment.json file

```
1  {
2      "action": "reserve",
3      "validity_in_minutes": 60,
4      "vendor_data": {
5          "$vendor_id": "$device_id"
6      },
7      "rfids": ["$rfid_key", "456"],
8      "rental_id": 1234,
9      "key_validity_start_in_epoch": 1550643333,
10     "key_validity_stop_in_epoch": 1550668696
11 }
```

TestRail integration

- To integrate with Testrail we need following .cfg file

[API]

url = https://pytestdemo.testrail.io

email = ravikadagoudar@gmail.com

password = kGnEkHfj7ZybIDB4hO.t-mYSCq8bMkJU.vrN7RYlv

[TESTRUN]

assignedto_id = 1

project_id = 1

TestRail integration

○ Pytest package - pytest-testrail

Sample Code using pytest-testrail

```
@pytest.trail.case('C14')
@pytest.mark.hwms_functional
def test_engine_started_with_valid_data(self, configureTenantVendor, client):
    baseurl = self.Templates.getFromConfig('$baseurl')

    asset_id = 'api-'+generateRandom(RandomDataType.STRING, 10) # Creating a random asset id
    HALTestbase.commonAction.createAsset(asset_id) # calling create asset from common function

    HALTestbase.commonAction.reserveAsset(asset_id)
    HALTestbase.commonAction.reservation_confirmed(asset_id)

    api_path = self.Templates.getFromApiPaths('vi_action')
    api_path = api_path.replace('#asset_id', asset_id)

    header = self.Templates.getFromHeaders('vendor_initiated')
    payload = self.Templates.getFromPayload('engine_started')

    response_status, response_payload, response_headers = client.doPut(baseurl + api_path, header, payload)
    assert response_status == 204
```


GitHub Link

- <https://github.com/ravindrab5/ApiAutomation.git>