

Roost on Windows

Steps to install

Use the Windows installer provided by Roost team.

Upon a successful installation, you will see the files copied on your system

File Name	Type	Location
roostgpt-win.exe	Windows executable	C:\Program Files (x86)\RoostGPT [Default]
RoostJavaASTParser.jar	Java archive	\$(USER_PROFILE)\var\tmp\Roost\bin
RoostUITestGenerator-win.exe	Windows executable	\$(USER_PROFILE)\var\tmp\Roost\bin
license.ral	Encrypted data file	\$(USER_PROFILE)/.roost
roost-workspace (folder)	RoostGPT Pre-canned Samples	\$(USERPROFILE)/roost-workspace

Pre-requisites to run Roost Generated Tests

To run the Roost generated tests successfully on the local system, you will need pre-requisites installed on the system

Pre-requisites for Java Unit Test:

- JDK/JRE
- Maven/Gradle

Pre-requisites for API Spec Test:

- Postman/Newman

Pre-requisites for UI Test:

- Playwright

Configure AWS Credentials

```
# Add AWS keys to your $HOME/.profile and save the file OR
setx AWS_ACCESS_KEY_ID "xxx"
setx AWS_SECRET_ACCESS_KEY "xxx"
setx AWS_DEFAULT_REGION "eu-west-1"
setx AWS_BEDROCK_MODEL_TYPE "cross-region" # Supports [foundation, cross-region]
setx AWS_BEDROCK_MODEL "global.anthropic.claude-sonnet-4-20250514-v1:0"
```

Roost Samples

Structure - Roost Sample Workspace

The workspace contains example projects to help you get started:

```
~/workspace/
├─ functional-test/
│   └─ functional-test.env          # Configuration for POC 1
│   └─ prepaid-card-requirement.docx # Sample user requirements doc for POC 1
│   └─ prepaid-card-swagger.yaml    # Required yaml for the POC 1
├─ postman-api-test/
│   └─ postman-api-test.env        # Configuration for POC 2
│   └─ prepaid-card-swagger.yaml   # Sample Swagger Spec for POC 2
└─ weather.yaml                   # Sample Swagger Spec for POC 2
```

Using RoostGPT

Running RoostGPT

You can now access RoostGPT from anywhere on your system, using the command `roostgpt`

Quick Start Example

```
# 1. Navigate to workspace
cd $env:USERPROFILE\roost-workspace`
```

```
# **Ensure the env is updated with latest/required creds.**

# you can run using the full command
**run the below the command to generate the functional-test.**
`roostgpt.exe test create -c .\functional-test\functional-test.env`

**run the below the command to generate the Postman-api-test.**
`roostgpt.exe test create -c .\postman-api-test\postman-api-test.env`
```

Viewing Roost Test Results

```
# Roostgpt test results can be viewed using the below command
roostgpt test result

# A sample Postman Collection with tests from a swagger spec will be generated under a folder
postman_collections\
  |_ roost-postman\
    |_ prepaid-card-swagger\
      |_ roost_postman_prepaid-card-swagger_<epoch>
        |_ datasets\
          |_ User_login.json
```

After running the test result command successfully.

Below is the example of Functional test.

Loading result dashboard...

```
┌────────── Summary ─────────┐
|                               |
| Test Name: functional-test   |
| Test Type: FUNCTIONAL       |
```

| Status: COMPLETED |

| |

|

Git Details

Git Information	Value
Git Type	LOCAL
Project Path	.
Branch	roost-1763108488
Repo Path	.

AI Usage Statistics

AI Usage	Value
AI Type	bedrock_ai
Model	global.anthropic.claude-sonnet-4-20250514-v1:0
API Calls	4
Prompt Tokens	35,698
Completion Tokens	21,596

Generated Files

roost_test_1763108488

File Type / Method	Path
Gherkin Feature	functional_tests/roost_test_1763108488/roost_test_1763108488.feature
Functional Test	functional_tests/roost_test_1763108488/roost_test_1763108488.json
API Specification	functional_tests/roost_test_1763108488/prepaid-card-swagger_enhance.yaml
CSV Report	functional_tests/roost_test_1763108488/roost_test_1763108488.csv
Excel Report	functional_tests/roost_test_1763108488/roost_test_1763108488.xlsx

Configuration

Configuration	Value
User Input File	prepaid-card-requirement.docx

```

| Create API Spec      | true                                     |
|-----|-----|
| Create Excel        | true                                     |
|-----|-----|
| API Spec Type       | swagger                                 |
|-----|-----|
| API Spec Source     | file                                     |
|-----|-----|
| API Spec File Path  | prepaid-card-swagger.yaml              |
|-----|-----|

```

[14/11/2025, 1:57:14 PM] - [INFO] -

```

=====
=====

```

After running the Postman test result command successfully.

Below is the example of Postman-api test.

Loading result dashboard...

```

┌────────── Summary ─────────┐
|                               |
| Test Name: postman-api-test  |
| Test Type: API-SPEC-TEST    |
| Status: COMPLETED          |
|                               |
└──────────────────────────┘

```

Git Details

Git Information	Value
Git Type	LOCAL
Project Path	.
Branch	roost-1763116002
Repo Path	.

AI Usage Statistics

AI Usage	Value
AI Type	bedrock_ai
Model	global.anthropic.claude-sonnet-4-20250514-v1:0
API Calls	6
Prompt Tokens	4,983
Completion Tokens	2,661

Generated Files

File Type	Path
-----------	------

| Postman Collection | postman_collections\roost-postman\prepaid-card-swagger\roost_postman_prepaid-card-swagger_1763116... |

Endpoints from: postman-api-test\prepaid-card-swagger.yaml

Endpoint (Method)	Test Data
✓ Register a new user [POST]	postman_collections\roost-postman\prepaid-card-swagger\datasets\Register_a_new_user.json
✓ User login [POST]	postman_collections\roost-postman\prepaid-card-swagger\datasets\User_login.json
✓ Submit KYC documents [POST]	postman_collections\roost-postman\prepaid-card-swagger\datasets\Submit_KYC_documents.json
✓ Issue a new card [POST]	postman_collections\roost-postman\prepaid-card-swagger\datasets\Issue_a_new_card.json
✓ Load funds onto a card [POST]	postman_collections\roost-postman\prepaid-card-swagger\datasets\Load_funds_onto_a_card.json
✓ Get card transaction history [GET]	postman_collections\roost-postman\prepaid-card-swagger\datasets\Get_card_transaction_history.json

Configuration

Configuration	Value
Test Type	api-spec-test
Framework	postman
API Spec Type	swagger
API Spec Source	file
HTTP Verbs	get,post,put,patch,delete
Endpoint Regex	.*

[14/11/2025, 3:58:12 PM] - [INFO] -

=====
 =====