

Public SaaS

RoostGPT UI :

Access Public SaaS at <https://app.roost.ai/login>

Roost GPT allows the user to automate their test against their code repository.

Below is the UI structure of RoostGPT which has two header tabs Test and Events.

- The "**Events Tab**" shows the individual test generation trigger details and is explained in detail later.
- The table under "**Test Tab**" shows details such as
 - Test Name,
 - Created By,
 - GenAI Model used for test generation ,
 - Test Type whether it is Unit Test, Functional Test, API(using swagger) and API(using source code)
 - Test create date.
- Actions are available in "**Test Tab**"
 - to trigger the test generation,
 - to view the test configuration or workflow and
 - to delete the test
- Search box on the page allows to search tests by name
- Project Admins can view and edit anyone's tests workflows
- Regular project members can view only their test workflows

View of the Test Tab

TEST NAME	CREATED BY	GEN AI MODEL	TEST TYPE	CREATED ON	ACTIONS
demoTest2	Bhavya Goswami bhavya.goswami	Open AI	API (using swagger)	Jul 12 2023 17:33	
demoTest	Bhavya Goswami bhavya.goswami	Open AI	API (using swagger)	Jul 12 2023 16:38	
divyesh-Integration	Divyesh Maheshwari divyesh	Open AI	API (using swagger)	Jul 12 2023 10:43	
camundan-engine-rest	Divyesh Maheshwari divyesh	Vertex AI	Unit	Jul 11 2023 16:08	
camunda-engine-unit	Divyesh Maheshwari divyesh	Open AI	Unit	Jul 10 2023 17:20	
test	Bhavya Goswami bhavya.goswami	Open AI	API (using swagger)	Jul 10 2023 16:20	
rahul-test-unit-07072023	Rahul Goyal rahulgoyal436	Open AI	Unit	Jul 07 2023 17:47	
go-http-dem	Afsal A Azeez afsal	Open AI	API (using source code)	Jul 07 2023 15:24	

Add Test

In case of Add Test it opens up the below page, which has 5 sections ->

1. **Provide** a Test Name
2. **Select** the Test Type (Unit Tests , Functional and Non Functional Tests, (API using swagger), API(using source code) and integration tests.
3. **Choose GenAI Models** which has OpenAI Model and VertexAI Model and field to input their respective tokens. After the token verification, model specific details will be available for selection, such as
 - For openAI, you can choose GPT-3.5 or GPT-4,
 - For Vertex AI, you need to specify vertex project id, region and vertex model.

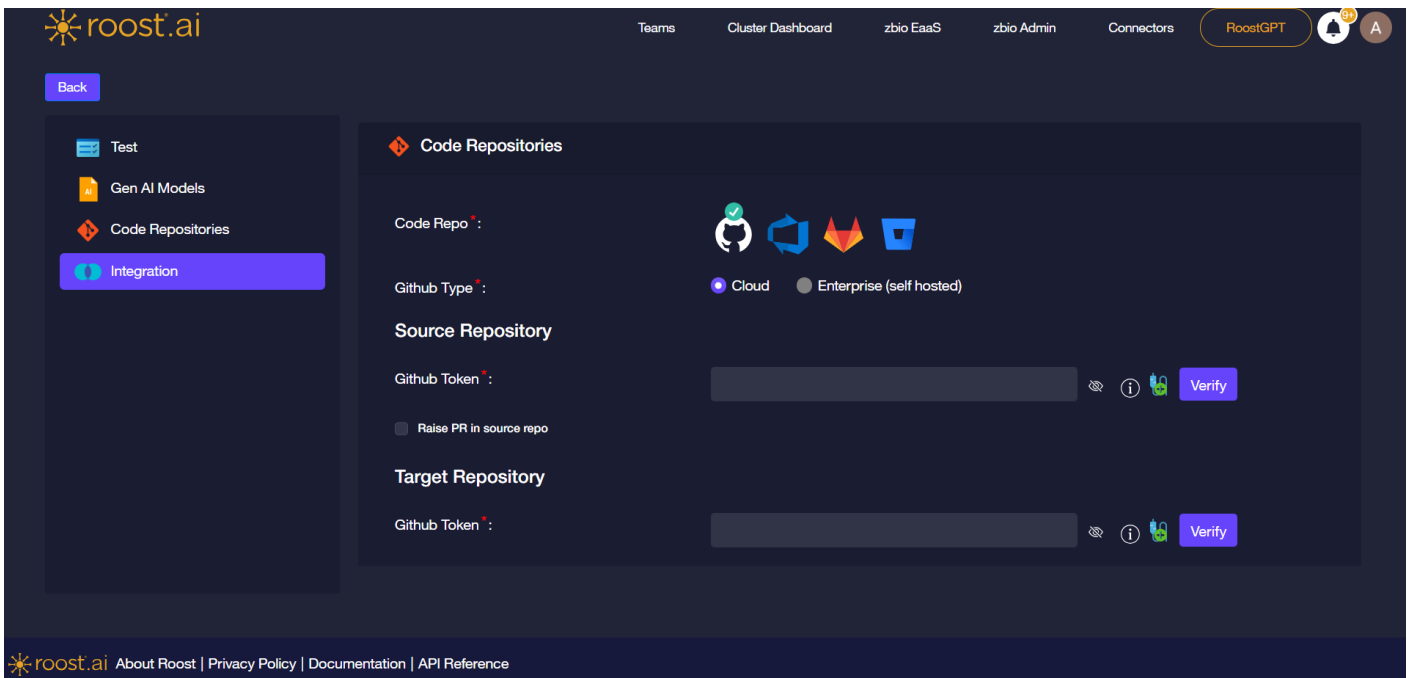
The screenshot shows the 'Add Test' form with the following fields and options:

- Test Name ***: Text input field.
- Test Type ***: Dropdown menu with 'Unit Test' selected.
- Iterations ***: Dropdown menu with '1' selected.
- Gen AI Models**: Section containing:
 - AI Model ***: Selection area with icons for OpenAI, Vertex AI, and another model.
 - Open AI Token ***: Text input field with a 'Verify' button.

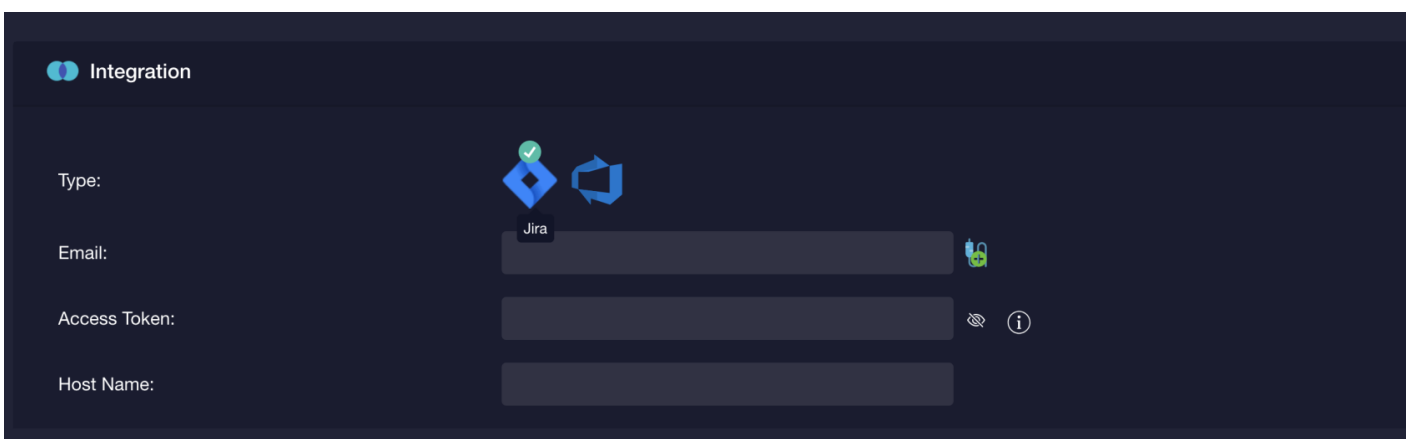
At the bottom of the form, there is a footer with the Roost.ai logo and links: About Roost | Privacy Policy | Documentation | API Reference.

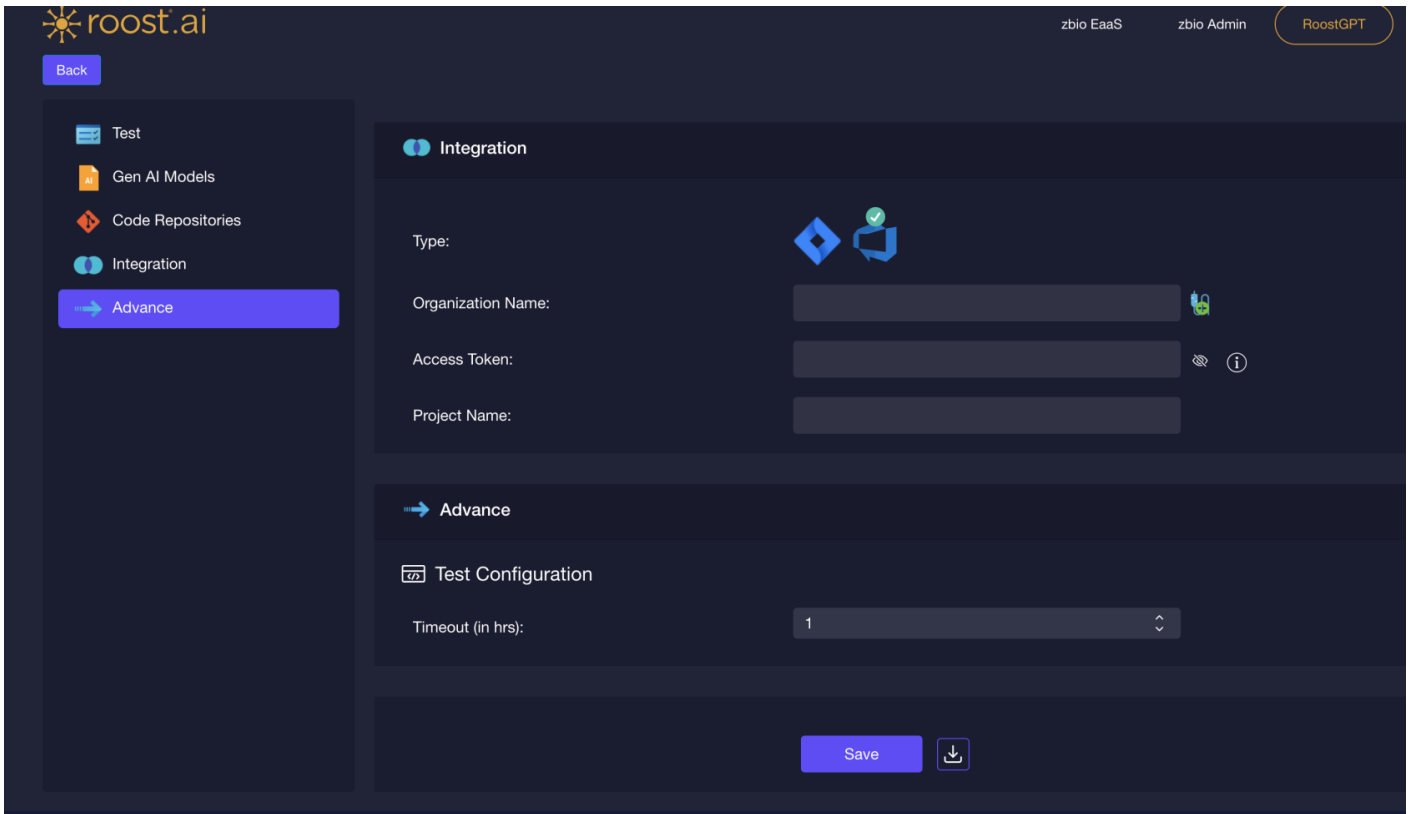
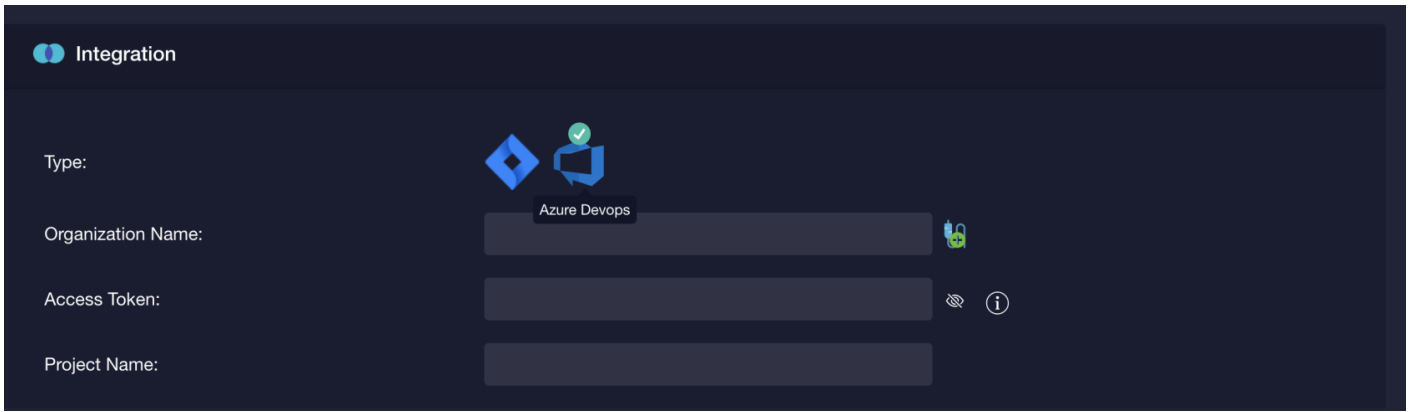
4. **Select SCCS (Source Code Repositories)** from available
 - **Github** (Cloud and self hosted) ,

- **Gitlab**, (cloud and self managed),
- **Azure Devops** and
- **Bitbucket** (cloud and self hosted).



1. After the SCCS token verification, you can provide the code repo, branch, language and their versions.
2. Languages supported are Java, Python, C#, Go
3. **Optional Integration** with ticketing tools like Jira and Azure DevOps is available.
4. **Jira** requires email address associated with jira account, jira hostname and the access token for your jira account.
5. **Azure Devops** requires the Organization Name associated with Azure Devops account, access token for authenticating with Azure Devops and the Project Name for Azure Devops.



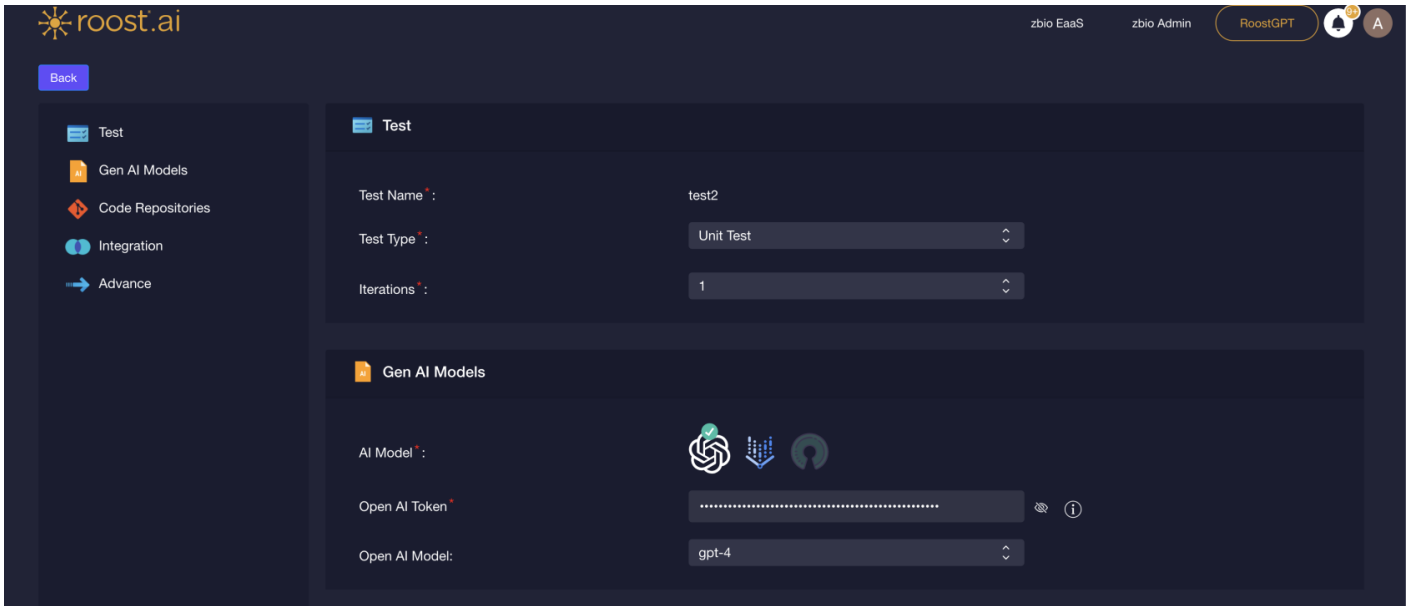


5. In **Advance** Timeout (in hrs) can be specified, for which the time the test will be on triggered mode

On click of save Button you can save your test and see the test you saved in the test header section. On click of download icon configuration for the test will be downloaded.

Workflow View

This will show you the workflow of your test, details you filled while adding your test.



Events View

Events view contains status of all the triggered test. Event Status filter shows the status of these events whether it is in Progress, timed out, aborted, completed or failed and the fields in the table show the test name with the status of the events and the status icons also the information of the repo, type of test, creation and completion time, user information and action icons for re-triggering the test again, logs view and the insights

TEST NAME	REPO INFO	TEST TYPE	CREATION TIME	COMPLETION TIME	USER INFO	ACTIONS
✗ demo-test	bhavyashah1226/ballot bhavyashah1226-patch-1	API (using swagger)	Jul 13 2023 12:13	Jul 13 2023 12:14	Bhavya Shah bhavya.shah	
✓ csharpapitest	Yashupadhyaya/RestfulApiBestPracticesAspNetCore master	API (using source code)	Jul 13 2023 11:56	Jul 13 2023 11:58	Yash Upadhyay yash.upadhyay	
✗ test-unit-gpt	afsalazeez/spring-boot-examples master	Unit	Jul 13 2023 11:49	Jul 13 2023 11:50	Afsal A Azeez afsal	
✓ csharpunit	Yashupadhyaya/CSharpBeginnerProjects main	Unit	Jul 13 2023 11:31	Jul 13 2023 11:36	Yash Upadhyay yash.upadhyay	
✗ camunda-engine-unit	Divyeshzb/camunda-bpm-platform master	Unit	Jul 12 2023 20:08	Jul 12 2023 21:14	Divyesh Maheshwari divyesh	
✓ divyesh-Integration	roost-io/java-springboot main	API (using swagger)	Jul 12 2023 19:54	Jul 12 2023 20:00	Divyesh Maheshwari divyesh	
✓ my-products-test	roost-io/java-springboot main	API (using source code)	Jul 12 2023 19:24	Jul 12 2023 19:28	Divyesh Maheshwari divyesh	

Logs View

Logs View contains the Event Information, AI Model Information and the Test Result Information having PR URL of your generated test on the left side and the Logs View container on the right side which has search Bar for searching in the logs and copy icon and download icons to copy and

download logs respectively.

The screenshot shows the Roost.ai interface with the 'Stream Logs' tab selected. On the left, there is a sidebar with 'Event Information' and 'AI Model Information'. The 'Event Information' section includes details like Test Name (csharpapitest), Test Type (API), Code Language (csharp), Git Type (github), Source (Yashupadhyaya/RestfulApiBestPracticesAspNetCore), Source Branch (master), Selected Path (ApiClient/Controllers), API Calls To AI (4), and Tokens Used (3268). The 'AI Model Information' section shows AI Model (Open AI) and GPT Model (gpt-4). The main area displays a log stream with various messages, including warnings about JavaScript experimental features and a series of successful actions like cloning a repo, generating test cases, writing tests, fetching user email, adding git config, checking out a new branch, adding test files, committing code, pushing to a branch, and creating a PR. The log ends with a GitHub PR URL: <https://github.com/Yashupadhyaya/RestfulApiBestPracticesAspNetCore/pull/3>.

Insights View

Insights view gives you the details about your test, here it will show your test files, their completion and creation time and corresponding action icons for Analyze and improve for further analysis and improvement of your tests and download, for downloading the test file.

The screenshot shows the Roost.ai interface with the 'Insights View' tab selected. It displays a table with the following columns: FILE NAME, COMPLETION TIME, USER INFO, CREATION TIME, and ACTIONS. The table contains five entries, all with a completion time of Jul 06 2023 17:07 and a creation time of Jul 06 2023 17:01. The user info for all entries is Akshat Jain (akshat.jain). The ACTIONS column contains an eye icon for each entry. At the bottom, it shows 'Showing 1 to 10 of entries' and a pagination control with 'First', '1', and 'Last' buttons.

FILE NAME	COMPLETION TIME	USER INFO	CREATION TIME	ACTIONS
...ntroller/CourseController.java	Jul 06 2023 17:07	A Akshat Jain akshat.jain	Jul 06 2023 17:01	👁
src/main/java/alahaoui/curriculum/controller/CourseRestController.java	Jul 06 2023 17:07	A Akshat Jain akshat.jain	Jul 06 2023 17:01	👁
...controller/HomeController.java	Jul 06 2023 17:07	A Akshat Jain akshat.jain	Jul 06 2023 17:01	👁
...troller/ProgramController.java	Jul 06 2023 17:07	A Akshat Jain akshat.jain	Jul 06 2023 17:01	👁
...ler/ProgramRestController.java	Jul 06 2023 17:07	A Akshat Jain akshat.jain	Jul 06 2023 17:01	👁

on the Click of eye icon in Actions, it will show you the test Files and Corresponding Actions. In the Actions, on click of Analyze and Improve it will open the modal which will show you the test and ask for the feedback for improvement of the test.

On click of download icon, it will download the test file generated.

The screenshot shows the Roost.ai interface with a table of test files. The table has columns for FILE NAME, COMPLETION TIME, USER INFO, CREATION TIME, and ACTIONS. A modal window is open, displaying the content of a selected test file: `...ntroller_getCourses_43b25856c6_Test.yaml`. The modal also includes a feedback section with a text input field and 'Cancel' and 'Improve' buttons.

FILE NAME	COMPLETION TIME	USER INFO	CREATION TIME	ACTIONS
...ntroller/CourseController.java	Jul 06 2023 17:07	A Akshat Jain akshat.jain	Jul 06 2023 17:01	👁
TEST FILE			ACTIONS	
...etCourses_4bce7788a4_Test.yaml				📄 ⬇
...llCourses_705d3563bc_Test.yaml				📄 ⬇
...onCourses_edcde750a4_Test.yaml				📄 ⬇
Page 1 of 1				Record 1-3 of 3
src/main/java/alahyaoui/curriculum/controller/CourseRestController.java	Jul 06 2023 17:07	A Akshat Jain akshat.jain	Jul 06 2023 17:01	👁
...controller/HomeController.java	Jul 06 2023 17:07	A Akshat Jain akshat.jain	Jul 06 2023 17:01	👁
...troller/ProgramController.java	Jul 06 2023 17:07	A Akshat Jain akshat.jain	Jul 06 2023 17:01	👁
...ler/ProgramRestController.java	Jul 06 2023 17:07	A Akshat Jain akshat.jain	Jul 06 2023 17:01	👁

```
1 # Test generated by RoostGPT for test curriculum-simulator-backend-test using AI Type Open AI and AI Model
2
3 config:
4   target: "http://localhost:8080"
5   phases:
6     - duration: 60
7       arrivalRate: 20
8
9 scenarios:
10  - name: "Successful API request"
11    flow:
12      - get:
13          url: "/api/courses"
14          headers:
15            Content-Type: "application/json"
16          expect:
17            - statusCode: 200
18              hasProperty: "courses"
19
20  - name: "API request with filter"
21    flow:
22      - get:
23          url: "/api/courses?filter=math"
```

Feedback ⓘ

Enter details here....

Cancel Improve

Revision #6

Created 12 July 2023 03:46:24 by Mayank Gaikwad

Updated 1 August 2023 07:52:06 by Harish