

From Business requirement to Unit Test, API Test & Integration Test

1. Integration Test (via UI)

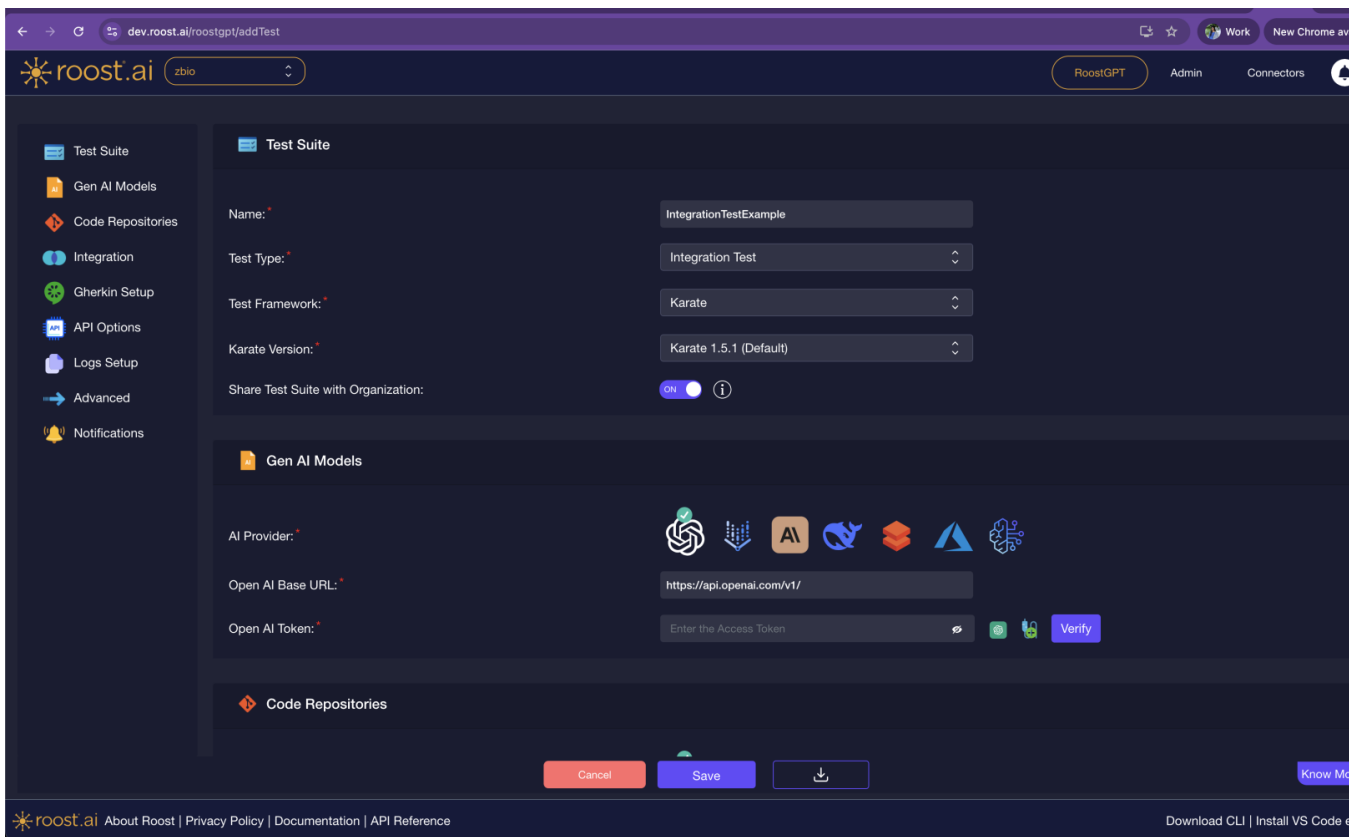
Test Suite Setup

- **Test Name:** Provide a descriptive name (e.g. `IntegrationTestExample`).
- **Test Type:** Select **Integration Test** (options: Unit / Functional / Integration).
- **Test Framework:** Choose from supported frameworks:
 - Rest Assured
 - Karate (*example used here*)
 - Postman
- **Selected Test Framework Version:** Choose from supported versions

The screenshot displays the 'Test Suite' configuration page in the Roost.ai application. The interface is dark-themed with a purple header. On the left, a sidebar lists navigation options: Test Suite, Gen AI Models, Code Repositories, Integration, Gherkin Setup, API Options, Logs Setup, Advanced, and Notifications. The main content area is titled 'Test Suite' and contains several sections:

- Name:** A text input field containing 'IntegrationTestExample'.
- Test Type:** A dropdown menu set to 'Integration Test'.
- Test Framework:** A dropdown menu set to 'Artillery', with a list of options (Artillery, Rest Assured, Karate) visible below it.
- Share Test Suite with Organization:** A dropdown menu set to 'Artillery'.
- Gen AI Models:** A section with fields for 'AI Provider' (showing icons for OpenAI, Anthropic, etc.), 'Open AI Base URL' (set to 'https://api.openai.com/v1/'), and 'Open AI Token' (with a 'Verify' button).
- Code Repositories:** A section with a 'Code Repo' field and icons for GitHub, GitLab, and others.

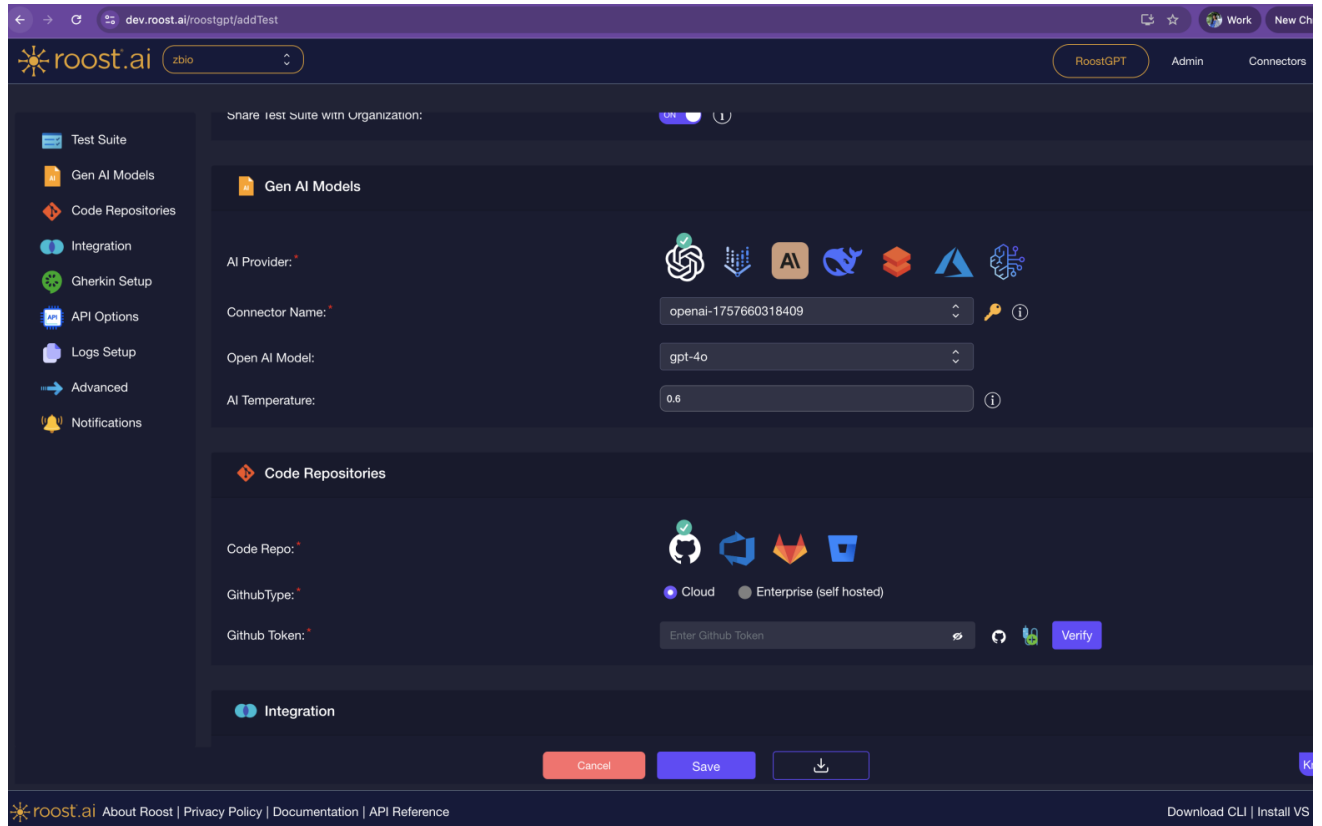
At the bottom, there are 'Cancel', 'Save', and 'Download' buttons. The footer includes the Roost.ai logo, links to 'About Roost', 'Privacy Policy', 'Documentation', and 'API Reference', along with 'Download CLI' and 'Install VS Code' links.



GenAI Models Setup

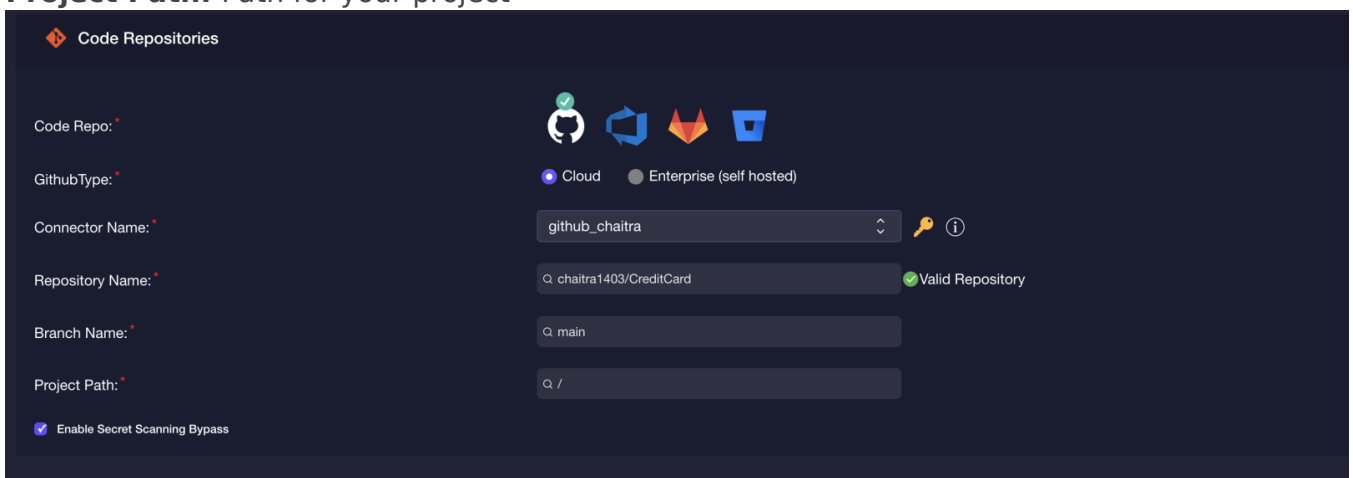
- **Select AI Provider:** Choose an AI provider (e.g. OpenAI, Vertex AI, Azure OpenAI, Claude, LLaMA).
 - Enter required credentials:
 - **API Key / Token** (from provider)
 - **Model Name** (e.g. `gpt-4`, `claude-3`, etc.)

- o Or choose connector



Code Repository Setup

- **Select Repository Type:** GitHub, GitLab, Bitbucket, Azure Repos.
- **Repository Details(Or choose connector):**
 - o **Access Token / PAT:** Token with valid permissions.
 - o **Username/Organization Name:** for Azure/Bitbucket give username/organization name
- **Branch:** Branch where test PR will be created
- **Project Path:** Path for your project



Business Requirement Source

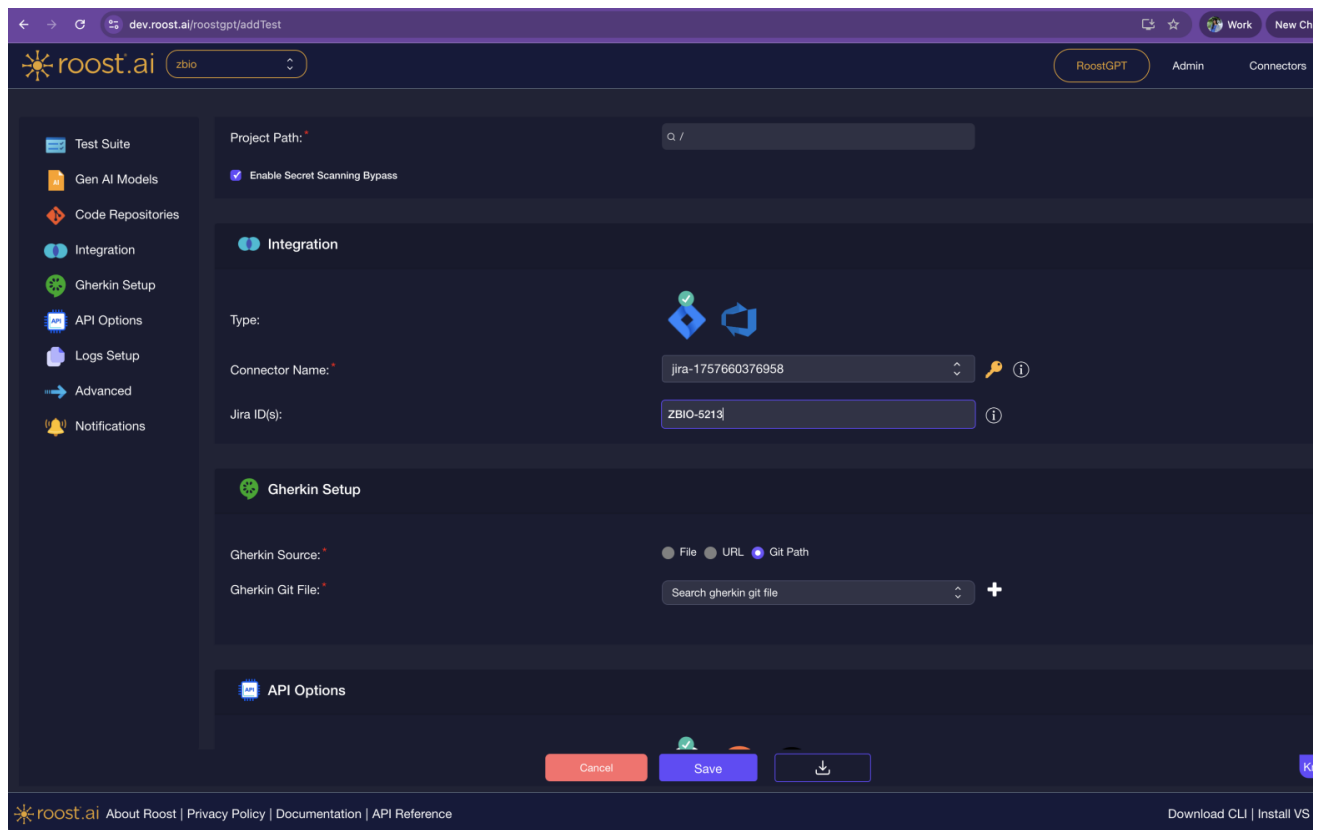
Integration with requirement management tools:

- **Choose Source:** Jira or Azure DevOps.
- **Authentication:**
 - Jira: Base URL, Email/Username, API Token.
 - Azure DevOps: Org URL, PAT token.
- **Requirement Reference:**
 - Provide **Ticket ID** (e.g. PROJ-1234) for Jira.
 - System will fetch business requirement details for context.

The screenshot shows the 'Integration' setup page in the Roost.ai web application. The interface is dark-themed. On the left is a sidebar with navigation links: Test Suite, Gen AI Models, Code Repositories, Integration (highlighted), Gherkin Setup, API Options, Logs Setup, Advanced, and Notifications. The main content area has a header with 'Project Path' and a search icon, and a checkbox for 'Enable Secret Scanning Bypass'. The 'Integration' section includes a 'Type' selector with icons for Jira and Azure DevOps, a 'Connector Name' field with the value 'jira-1757660376958', and a 'Jira ID(s)' field with the value 'ZBIO-5213'. Below this is the 'Gherkin Setup' section with a 'Gherkin Source' selector (File, URL, Git Path) and a 'Gherkin Git File' field with a search icon. At the bottom of the main area are 'Cancel', 'Save', and a download icon. The footer contains the Roost.ai logo, links for About Roost, Privacy Policy, Documentation, and API Reference, and buttons for Download CLI and Install VS.

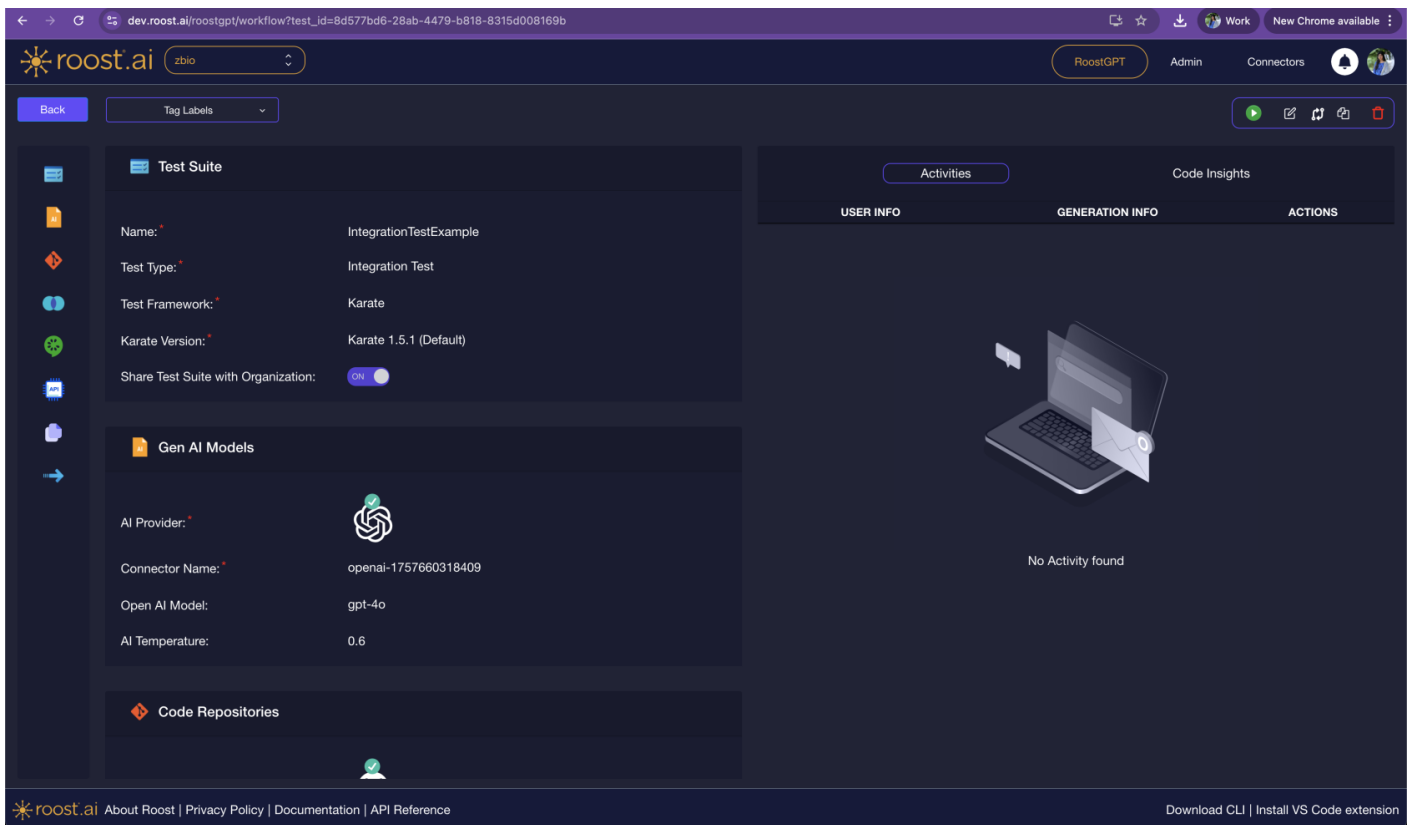
Gherkin Setup

- **Gherkin Source:** Choose how to provide acceptance criteria. Options:
 - **File Upload:** Upload .feature file.
 - **URL:** Provide direct URL to Gherkin spec.
 - **Git Path:** Specify repository and file path to .feature file.



Save the Test:

1. Once you have configured the test Save configuration and your test will be ready for the trigger



Test Execution

1. Click on execute test button to trigger the test
2. UI will prompt with Test Server Configuration details this provide the details and click on generate test

Test Server Configuration

File Name: functional_tests/ZBIO-5213.yaml

API Host (Key):
API_HOST

API Host (value):
http://localhost:3000

Showing 1 to 1 of 1 entries

First

«

1

2

»

Last

Test Server Configuration

AI Temperature: *

0.22.0

0.6

Timeout (in hrs): *

0 hrs24 hrs

1 hrs

Iterations: *

05

3

Back

Generate Test

- Test will be triggered after tokens are verified
- Logs will be available in the UI in logs tab.

Activities		Code Insights
USER INFO	GENERATION INFO	ACTIONS
Chaitra B V chaitra	- Sep 15 2025 21:04	Logs

- Once test creation is complete:
 - A **Pull Request (PR)** will be automatically generated in the selected repository.
 - PR link will be displayed.
 - You can view test insights from the generated PR.

Improving Tests

- Select the created test.
- Provide improvement instructions (e.g. "Add negative test for invalid payment ID").

- Re-run improvement process.
 - Updated PR will reflect the improved test suite.
-

Revision #2

Created 15 September 2025 10:24:11 by Divyesh

Updated 15 September 2025 15:47:53 by Chaitra