

CLI

Installation Command

RoostGPT binary is available at <https://github.com/roost-io/roost-support/releases>

The RoostGPT binary allows for unmatched flexibility during test generation, by allowing you to trigger test generation both locally and through git through the use of one simple .env file.

Add the "roostgpt-<linux|macos|win.exe>" binary to your PATH environment variable

For Linux/MacOS, you can execute the below command to install roostgpt

```
curl https://raw.githubusercontent.com/roost-io/roost-support/master/roostgpt.sh | bash -
```

CLI Commands

`roostgpt` (for mac) & `./roostGPT-win.exe` (for windows)

This command is used to run the RoostGPT CLI on your operating system.

- `-h, --help`: Displays the help message that provides information about the available options and usage.
- `roostgpt version update`: Update roostgpt version.
- `roostgpt config create`: Creates roostgpt configuration file. The configuration file contains environment variables used by RoostGPT application.
 - `--output-dir, -o <path>`: Creates a configuration file in a specific directory. The configuration file contains environment variables used by the RoostGPT application. Default path is pwd.
 - `--name <name>`: Creates the configuration file with specific name. Default name is default.env.
- `roostgpt config update`: Updates the roostgpt configuration file.

- `--config, -c <path>` : Specifies the location of the environment configuration file. The configuration file contains environment variables used by the RoostGPT application.
 - `--name <name>` : loads `pwd/$name.env`. It is ignored if the `--file` is provided.
- `roostgpt test create` : Triggers the test generation.
 - `--config,-c <path>` : Path to the configuration file if provided, Default value is `pwd/default.env`.
 - `--docker,-d` : Generate tests in a docker environment, see [supported features](#) to see what tool/language versions are supported.
 - `--verbose,-v` : Enable verbose logging(Debug mode).
 - `roostgpt test improve` : Improve the selected test based on user feedback.
 - `--docker,-d` : Improve tests in a docker environment, see [supported features](#) to see what tool/language versions are supported.
 - `--verbose,-v` : Enable verbose logging(Debug mode).
 - `roostgpt test retrigger` : Re-trigger the chosen test from where it left off during its last execution.
 - `--docker,-d` : Re-trigger tests in a docker environment, see [supported features](#) to see what tool/language versions are supported.
 - `--verbose,-v` : Enable verbose logging(Debug mode).
 - `roostgpt analysis` : Analyze the existing tests and scenarios and generate a report.
 - `--config,-c <path>` : Path to the configuration file if provided, Default value is `pwd/default.env`.
 - `--verbose,-v` : Enable verbose logging(Debug mode).

Instructions for creating .env file

An easy way to download the .env file is to use <https://app.roost.ai/gptCLIForm>

The .env file should be in the same folder path as RoostGPT binary.

Here is what the content of your `.env` file will look like:

```
# GIT env vars

GIT_TYPE=                # Default is github; Supports [github, gitlab, azure, bitbucket, local]

HOSTED_TYPE=             # Default is cloud; Supports [cloud, hosted]

GIT_HOSTED_URL=          # Required if HOSTED_TYPE=hosted

USE_SSH=                 # Optional; Default is false; Supports [true, false]; SSH based auth for Git

LOCAL_PROJECT_PATH=      # Required if GIT_TYPE is local, Path to your workspace

SOURCE_GIT_CLONE_URL=    # Optional; the source repo clone URL
```

| | |
|-----------------------------|---|
| SOURCE_GIT_TOKEN= | # Required |
| SOURCE_OWNER_NAME= | # Required |
| SOURCE_REPO_NAME= | # Required |
| SOURCE_REPO_ID= | # Required for GIT_TYPE=gitlab |
| SOURCE_PROJECT_NAME= | # Required for GIT_TYPE=azure |
| SOURCE_PROJECT_ID= | # Required for GIT_TYPE=bitbucket |
| SOURCE_BASE_BRANCH= | # Required |
| SOURCE_RELATIVE_DIRECTORY= | # Optional; the relative path of the repository |
| SAME_TARGET_DETAIL= | # Optional; Default true, Supports [true, false] |
| DESTINATION_GIT_CLONE_URL= | # Optional; the destination repo clone URL |
| DESTINATION_GIT_TOKEN= | # Required if SAME_TARGET_DETAIL=false |
| DESTINATION_OWNER_NAME= | # Required if SAME_TARGET_DETAIL=false |
| DESTINATION_REPO_NAME= | # Required if SAME_TARGET_DETAIL=false |
| DESTINATION_REPO_ID= | # Required if SAME_TARGET_DETAIL=false and GIT_TYPE=gitlab |
| DESTINATION_PROJECT_NAME= | # Required for GIT_TYPE=azure |
| DESTINATION_PROJECT_ID= | # Required if SAME_TARGET_DETAIL=false and GIT_TYPE=bitbucket |
| DESTINATION_BASE_BRANCH= | # Required if SAME_TARGET_DETAIL=false |
| | |
| # Open AI env vars | |
| OPENAI_API_MODEL= | # Optional; Default is gpt-4 |
| OPENAI_API_KEY= | # Required if AI_TYPE=openai |
| | |
| # Vertex AI env vars | |
| VERTEX_FINE_TUNE= | # Optional; Default is false, Supports [true, false] |
| VERTEX_PROJECT_ID= | # Required if AI_TYPE=vertexai |
| VERTEX_REGION= | # Required if AI_TYPE=vertexai and VERTEX_FINE_TUNE=true |
| VERTEX_BEARER_TOKEN= | # Required if AI_TYPE=vertexai |
| VERTEX_MODEL= | # Required if AI_TYPE=vertexai; Supports [text-bison, code-bison, codechat-bison, text-bison-32k, code-bison-32k, codechat-bison-32k, gemini-pro] |
| | |
| # Azure Open AI env vars | |
| AZURE_OPENAI_ENDPOINT= | # Required if AI_TYPE=azure_open_ai |
| AZURE_DEPLOYMENT_NAME= | # Required if AI_TYPE=azure_open_ai |
| AZURE_OPENAI_KEY= | # Required if AI_TYPE=azure_open_ai |
| AZURE_OPENAI_VERSION= | # Optional Default is 2023-12-01-preview |
| | |
| # Open source env vars | |
| OPEN_SOURCE_MODEL_ENDPOINT= | # Required if AI_TYPE=open_source_ai |
| OPEN_SOURCE_MODEL= | # Optional; Supports [meta-llama/Llama-2-13b-chat, HuggingFaceH4/starchat-beta] |

```
# Sagemaker model env vars
SAGEMAKER_MODEL_ENDPOINT=          # Required if AI_TYPE=sagemake_model

# Claude AI env vars
CLAUDE_AI_MODEL=                    # Required if AI_TYPE=claude_ai
CLAUDE_AI_API_KEY=                  # Required if AI_TYPE=claude_ai

# DBRX AI env vars
DBRX_MODEL_ENDPOINT=                # Required if AI_TYPE=dbrx
DBRX_AUTH_TOKEN=                    # Required if AI_TYPE=dbrx
DBRX_MODEL_NAME=                    # Required if AI_TYPE=dbrx

# Advanced AI env vars
AI_TEMPERATURE=                     # Optional; Default value is 0.6
USE_ASSISTANT=                      # Optional; Use Assistant feature in openai. Default:false

# Jira board env vars
JIRA_HOST_NAME=                     # Required if TEST_TYPE=functional and BOARD_TYPE=jira
JIRA_EMAIL=                         # Required if TEST_TYPE=functional and BOARD_TYPE=jira
JIRA_TOKEN=                         # Required if TEST_TYPE=functional and BOARD_TYPE=jira
JIRA_ID=                            # Optional if TEST_TYPE=functional and BOARD_TYPE=jira

# Azure board env vars
AZURE_ORG=                          # Required if TEST_TYPE=functional and BOARD_TYPE=azure
AZURE_PROJECT=                      # Required if TEST_TYPE=functional and BOARD_TYPE=azure
AZURE_TOKEN=                        # Required if TEST_TYPE=functional and BOARD_TYPE=azure
AZURE_WORK_ITEM_ID=                 # Optional if TEST_TYPE=functional and BOARD_TYPE=azure

# Advanced Jira/Azure board env vars
CREATE_API_SPEC=                    # Optional; Used for TEST_TYPE=functional

# Log env vars
LOG_SOURCE=                         # Optional; Default is elks
LOG_SOURCE_PATH=                    # Optional; the log file path
LOG_FILE_ENCODING=                  # Optional; Default is utf-8, supports [utf-8, utf-16le]
LOG_ELASTICSEARCH_URL=              # Optional; the Elastic search URL
LOG_ELASTICSEARCH_USER=             # Optional; the Elastic search user name
LOG_ELASTICSEARCH_TOKEN=            # Optional; the Elastic search token
LOG_ELASTICSEARCH_API_KEY=          # Optional; the Elastic search api key
```

Behavioural test cases env vars

BEHAVIORAL_TEST_TYPE= # Optional; Supports [gherkin]
BEHAVIORAL_TEST_SOURCE= # Optional; Supports [file, gitpath, url]
BEHAVIORAL_TEST_FILE_PATH= # Optional; path of the source file if source is file/gitpath. Relative
path in case of gitpath
BEHAVIORAL_TEST_URL= # Optional; URL of the source file if source is url

API Spec env vars

API_SPEC_TYPE= # Optional; Supports [swagger, postman, raml]
API_SPEC_SOURCE= # Optional; Supports [file, gitpath, url]
API_SPEC_FILE_PATH= # Optional; path of the source file if source is file/gitpath. Relative path
in case of gitpath
API_SPEC_URL= # Optional; URL of the source file if source is url
API_SPEC_ALIAS=

Cypress test env vars

SCREENSHOTS_SOURCE= # Optional; Used for cypress test. Supports [file, gitpath, url]
SCREENSHOTS_URL= # Optional; URL of the source file if source is url
SCREENSHOTS_FILE_PATH= # Optional; path of the source file if source is file/gitpath. Relative
path in case of gitpath

HTTP_VERBS_FOR_TESTING= # Optional; Default is "get,post,put,patch,delete", Supports all the
combinations of http verbs. The specific http methods to be tested for all APIs. Enter "," separated values here.
REGEX_HTTP_ENDPOINTS_FOR_TESTING= # Optional; Regex string that matches to specific endpoints
that needs to be tested. If empty, all the endpoints will be tested.

Application test env vars

APPLICATION_ENTRYPOINT= # Required; if you need to generate application tests along with
unit tests. This feature is supported only for GPT Turbo models. Use a relative path from the project root.

License env vars

ROOST_DOMAIN= # Optional; Default is app.roost.ai
ROOST_TOKEN= # Optional; the Roost token or license filepath, default
\$HOME/.roost/license.rai
TELEMETRY= # Optional; Default is true, Supports [true, false], send telemetry data to
roost, no private information is shared.

Additional env vars

TEST_NAME= # Optional; Default is roost-test

```

ROOST_DIR=                                # Optional; Default is /var/tmp/Roost/RoostGPT
LANGUAGE=                                # Optional; Default is java; Supports [java, go, python, csharp, nodejs,
reactjs, cpp]
AI_TYPE=                                # Optional; Default is openai; Supports [openai, vertexai, azure_open_ai,
open_source_ai, sagemaker_model, claude_ai, dbrx]
JAVA_BUILD_TOOL=                          # Optional; Default is maven; Supports [maven, gradle]
PACKAGES_TO_SCAN=                        # Optional; Default is "*"; used for LANGUAGE=java, for generating
tests for specific packages; Enter comma seperated values. E.g. "com.demo.sample,"com.example.products"
ITERATION=                              # Optional; Set VERIFY_TEST=true to enable iteration; Default value of
ITERATION is 1
TIMEOUT=                                # Optional; Default is 1 hour
TEST_TYPE=                              # Optional; Default is unit Supports [unit, functional, artillery, integration,
api-spec-test]
TEST_FRAMEWORK=                          # Optional; Default is "pytest" for Python, "gotest" for Golang, "JUnit4"
for Java, "nunit" for CSharp, "jest" for Node.js, "postman" for Postman test-script. Supports [pytest, unittest] for
Python, [gotest] for Golang, [JUnit4, JUnit5] for Java, [nunit] for Csharp, [jest] for Node.js, [postman, artillery,
rest-assured] for [test_type: api-spec-test]
TRIGGER_ID=                              # Unique id to identify multiple triggers; Default is epoch
BOARD_TYPE=                              # Optional; Default is jira; Supports [jira, azure, none]
MAX_DEPTH=                              # Optional; Default is -1; if MAX_DEPTH is -1, it will traverse all the sub
directories else the maximum depth directories to look for
FUNCTIONS_TO_TEST=                       # Optional; List of function names to be tested, separated by comma.
Name to be specified as module.[class.]method OR module.function
USE_CACHE=                              # Optional; Used for only developer. Default:false
VERIFY_TEST=                             # Optional; Default is true, Supports [true, false]
VULNERABILITY_GEN=                       # Optional; Default is false, Supports [true, false]
CONSIDER_EXISTING_TEST=                  # Optional; Default is true, Supports [true, false], used for Java,
python, Golang unit test generation, It scans your repository to identify any existing tests for the specified
function. If tests are found, it incorporates them into its consideration and generates additional test cases as
needed
GRADING_NOTES=                           # Optional; default is true
CUSTOM_TAGS=                             # Optional; The tags to be attached to test code, so that the test's can
run based on tags
ROOST_USER_INPUT=                        # Optional; The user input will be used to generate tests

# Improve test env vars
FILE_PATH=                               # Required if IMPROVE_TEST=true
USER_CONTENT=                            # Required if IMPROVE_TEST=true
TESTSCRIPT_ENDPOINT=                     # Required if IMPROVE_TEST=true and TEST_FRAMEWORK=postman

```

```
# Language version env vars, Used when RoostGPT is run in docker mode, visit
https://docs.roost.ai/topics/roostgpt/page/cli to check the supported language versions

JAVA_VERSION=                # Optional; Default is 17, Used for TEST_FRAMEWORK "JUnit4", "JUnit5",
"karate", "rest-assured"

MAVEN_VERSION=                # Optional; Default is 3.8.6, Used for TEST_FRAMEWORK "JUnit4",
"JUnit5", "karate", "rest-assured"

GRADLE_VERSION=              # Optional; Default is 8.6, Used for TEST_FRAMEWORK "JUnit4", "JUnit5",
"karate", "rest-assured"

GO_VERSION=                  # Optional; Default is 1.19.9, Used for TEST_FRAMEWORK "gotest"

PYTHON_VERSION=              # Optional; Default is 3, Used for TEST_FRAMEWORK "pytest",
"unittest"

DOTNET_VERSION=              # Optional; Default is 8, Used for TEST_FRAMEWORK "nunit"

NODE_VERSION=                # Optional; Default is 18.18.0, Used for TEST_FRAMEWORK "artillery",
"postman", "rest-assured", "karate", "jest", "mocha"
```

Note: If a value in your .env file contains the '#' character, enclose the entire value in quotes. This prevents the '#' from being interpreted as the start of a comment.

Example: SECRET_KEY="my#secret#key"

Here's a description of each variable:

Git env vars

- **GIT_TYPE**

Description: Specifies the type of Git repository.

Default Value: "github"

Supported Values: "github", "gitlab", "azure", "bitbucket", "local"

Optional/Required: Optional

- **HOSTED_TYPE**

Description: Indicates where the Git platform is accessible.

Default Value: "cloud"

Supported Values: "cloud", "hosted"

Optional/Required: Optional

- **GIT_HOSTED_URL**

Description: The URL of the hosted Git platform.

Optional/Required: Required if HOSTED_TYPE is "hosted"

- **USE_SSH**

Description: Specifies whether to use SSH-based authentication for Git.

Default Value: "false"

Supported Values: "true", "false"

Optional/Required: Optional

- **LOCAL_PROJECT_PATH**

Description: Path to your workspace.

Optional/Required: Required if GIT_TYPE is "local"

- **SOURCE_GIT_CLONE_URL**

Description: Clone URL for the source repository.

Optional/Required: Optional

- **SOURCE_GIT_TOKEN**

Description: Token for authenticating the specific source Git repository.

Optional/Required: Required

- **SOURCE_OWNER_NAME**

Description: Name of the source repository owner (Git username).

Optional/Required: Required

- **SOURCE_REPO_NAME**

Description: Name of the source repository.

Optional/Required: Required

- **SOURCE_REPO_ID**

Description: ID of the source repository.

Optional/Required: Required if GIT_TYPE is "gitlab"

- **SOURCE_PROJECT_NAME**

Description: Name of the source project.

Optional/Required: Required if GIT_TYPE is "azure"

- **SOURCE_PROJECT_ID**

Description: ID of the source project.

Optional/Required: Required if GIT_TYPE is "bitbucket"

- **SOURCE_BASE_BRANCH**

Description: Base branch to be tested in the source repository.

Optional/Required: Required

- **SOURCE_RELATIVE_DIRECTORY**

Description: Relative directory path within the source repository.

Optional/Required: Optional

- **SAME_TARGET_DETAIL**

Description: Indicates if the destination Git details are the same as the source Git details.

Default Value: "true"

Supported Values: "true", "false"

Optional/Required: Optional

- **DESTINATION_GIT_CLONE_URL**

Description: Clone URL for the destination repository.

Optional/Required: Optional

- **DESTINATION_GIT_TOKEN**

Description: Token for authenticating the specific destination Git repository.

Optional/Required: Required if SAME_TARGET_DETAIL is "false"

- **DESTINATION_OWNER_NAME**

Description: Name of the destination repository owner (Git username).

Optional/Required: Required if SAME_TARGET_DETAIL is "false"

- **DESTINATION_REPO_NAME**

Description: Name of the destination repository.

Optional/Required: Required if SAME_TARGET_DETAIL is "false"

- **DESTINATION_REPO_ID**

Description: ID of the destination repository.

Optional/Required: Required if SAME_TARGET_DETAIL is "false" and GIT_TYPE is "gitlab"

- **DESTINATION_PROJECT_NAME**

Description: Name of the destination project.

Optional/Required: Required if SAME_TARGET_DETAIL is "false" and GIT_TYPE is "azure"

- **DESTINATION_PROJECT_ID**

Description: ID of the destination project.

Optional/Required: Required if SAME_TARGET_DETAIL is "false" and GIT_TYPE is "bitbucket"

- **DESTINATION_BASE_BRANCH**

Description: Base branch in which the test code is to be pushed in the destination repository.

Optional/Required: Required if SAME_TARGET_DETAIL is "false"

Open AI env vars

- **OPENAI_API_MODEL**

Description: Specifies the type of the OpenAI API model to be used.

Default Value: "gpt-4"

Supported Values: "gpt-4", "gpt-3.5-turbo", etc. (Availability of models depends on the OPENAI_API_KEY).

Optional/Required: Optional

- **OPENAI_API_KEY**

Description: The API key required for accessing the OpenAI API.

Optional/Required: Required if AI_TYPE is "openai"

AZURE Open AI env vars

- **AZURE_OPENAI_ENDPOINT**

Description: The API endpoint used to access Azure OpenAI services.

Optional/Required: Required if AI_TYPE is "azure_open_ai"

- **AZURE_DEPLOYMENT_NAME**

Description: The name of the Azure OpenAI deployment you wish to use.

Optional/Required: Required if AI_TYPE is "azure_open_ai"

- **AZURE_OPENAI_KEY**

Description: The API key required for authenticating requests to Azure OpenAI.

Optional/Required: Required if AI_TYPE is "azure_open_ai"

- **AZURE_OPENAI_VERSION**

Description: Specifies the version of the Azure OpenAI API to be used.

Default Value: "2024-02-15-preview"

Optional/Required: Optional

Vertex AI env vars

- **VERTEX_FINE_TUNE**

Description: Indicates whether fine-tuning is enabled for the model.

Default Value: "false"

Supported Values: "true", "false"

- **VERTEX_PROJECT_ID**

Description: The ID of the Vertex project you are working with.

Optional/Required: Required if AI_TYPE is "vertexai"

- **VERTEX_REGION**

Description: The region where the Vertex project is located. For example, "us-central1".

Optional/Required: Required if AI_TYPE is "vertexai" and VERTEX_FINE_TUNE is "true"

- **VERTEX_BEARER_TOKEN**

Description: The bearer token used for authenticating requests to the Vertex API.

Optional/Required: Required if AI_TYPE is "vertexai"

- **VERTEX_MODEL**

Description: The name of the Vertex model to be used.

Supported Values: "text-bison", "code-bison"

Optional/Required: Required if AI_TYPE is "vertexai"

Open Source AI env vars

- **OPEN_SOURCE_MODEL_ENDPOINT**

Description: The API endpoint used to access the Open Source AI model.

Optional/Required: Required if AI_TYPE is "open_source_ai"

- **OPEN_SOURCE_MODEL**

Description: The name of the Open Source AI model you wish to use.

Supported Values: "meta-llama/Llama-2-13b-chat", "HuggingFaceH4/starchat-beta"

Optional/Required: Optional

SageMaker Model env vars

- **SAGEMAKER_MODEL_ENDPOINT**

Description: The endpoint where the SageMaker model is hosted.

Optional/Required: Required if AI_TYPE is "sagemaker_model"

Claude AI env vars

- **CLAUDE_AI_MODEL**

Description: Specifies the type of Claude AI model to be used.

Supported Values: "claude-3-opus-20240229", "claude-3-sonnet-20240229", "claude-3-haiku-20240307"

Optional/Required: Required if AI_TYPE is "claude_ai"

- **CLAUDE_AI_API_KEY**

Description: The API key required for accessing the Claude AI API.

Optional/Required: Required if AI_TYPE is "claude_ai"

DBRX AI env vars

- **DBRX_MODEL_ENDPOINT**

Description: The endpoint where the DBRX AI model is hosted.

Optional/Required: Optional

- **DBRX_AUTH_TOKEN**

Description: The authentication token required for accessing the DBRX model.

Optional/Required: Required if AI_TYPE is "dbrx"

- **DBRX_MODEL_NAME**

Description: The name of the DBRX model you wish to use.

Optional/Required: Required if AI_TYPE is "dbrx"

Advanced AI env vars

- **AI_TEMPERATURE**

Description: Specifies the temperature setting for AI responses, which influences the creativity and variability of the test generation. A higher value results in more creative and varied outputs, while a lower value produces more deterministic responses.

Default Value: 0.6

Supported Values: Any numeric value, typically between 0.0 and 1.0.

Optional/Required: Optional

- **USE_ASSISTANT**

Description: Indicates whether to use the Assistant feature in OpenAI.

Supported Values: "true", "false"

Optional/Required: Optional

Jira board env vars

- **JIRA_EMAIL**

Description: The email address associated with your Jira account.

Optional/Required: Required if TEST_TYPE is "functional" and BOARD_TYPE is "jira"

- **JIRA_HOST_NAME**

Description: The hostname of your Jira instance (e.g., jira.company.com).

Optional/Required: Required if TEST_TYPE is "functional" and BOARD_TYPE is "jira"

- **JIRA_PASSWORD**

Description: The password for your Jira account.

Optional/Required: Required if TEST_TYPE is "functional" and BOARD_TYPE is "jira"

- **JIRA_ID**

Description: The Jira ID associated with your account, often used as a unique identifier.

Optional/Required: Required if TEST_TYPE is "functional" and BOARD_TYPE is "jira"

Azure board env vars

- **AZURE_ORG**

Description: The organization associated with your Azure DevOps account.

Optional/Required: Required if TEST_TYPE is "functional" and BOARD_TYPE is "azure"

- **AZURE_TOKEN**

Description: The access token for authenticating with Azure DevOps.

Optional/Required: Required if TEST_TYPE is "functional" and BOARD_TYPE is "azure"

- **AZURE_PROJECT**

Description: The name of the Azure DevOps project.

Optional/Required: Required if TEST_TYPE is "functional" and BOARD_TYPE is "azure"

- **AZURE_WORK_ITEM_ID**

Description: The ID associated with the relevant work item in Azure DevOps.

Advanced Jira/Azure board env vars

- **CREATE_API_SPEC**

Description: Used for functional test type

Optional/Required: Required if TEST_TYPE=functional

Log env vars

- **LOG_SOURCE**

Description: Origin from where the logs are generated.

Default Value: "elks"

Optional/Required: Optional

- **LOG_SOURCE_PATH**

Description: Location where the log files are stored.

Optional/Required: Optional

- **LOG_ELASTICSEARCH_URL**

Description: The URL for the Elasticsearch instance.

Optional/Required: Optional

- **LOG_ELASTICSEARCH_USER**

Description: Username for accessing Elasticsearch resources.

Optional/Required: Optional

- **LOG_ELASTICSEARCH_TOKEN**

Description: Authentication token used for accessing Elasticsearch resources.

Optional/Required: Optional

- **LOG_ELASTICSEARCH_API_KEY**

Description: API key used to authenticate and authorize access to Elasticsearch APIs.

Optional/Required: Optional

- **LOG_FILE_ENCODING**

Description: Used to encode the log file, default is utf-8, supports [utf-8, utf-16le].

Optional/Required: Optional

Behavioural Test cases env vars

- **BEHAVIORAL_TEST_TYPE**

Description: Specifies the type of behavioral tests to be used.

Supported Values: "gherkin"

Optional/Required: Optional

- **BEHAVIORAL_TEST_SOURCE**

Description: Specifies the source location for the behavioral test data.

Supported Values: "file", "gitpath", "url"

Optional/Required: Optional

- **BEHAVIORAL_TEST_FILE_PATH**

Description: Path to the source file if the BEHAVIORAL_TEST_SOURCE is "file" or "gitpath". For "gitpath", this should be a relative path within the repository.

Optional/Required: Optional

- **BEHAVIORAL_TEST_URL**

Description: The URL of the source file if the BEHAVIORAL_TEST_SOURCE is "url".

Optional/Required: Optional

API Spec env vars

- **API_SPEC_TYPE**

Description: Specifies the type of API specification to be used.

Supported Values: "Swagger", "Postman"

Optional/Required: Optional

- **API_SPEC_SOURCE**

Description: Specifies the source location for the API specification data.

Supported Values: "file", "gitpath", "url"

Optional/Required: Optional

- **API_SPEC_FILE_PATH**

Description: Path to the source file if API_SPEC_SOURCE is "file" or "gitpath". For "gitpath", this should be a relative path within the repository.

Optional/Required: Optional

- **API_SPEC_URL**

Description: The URL of the source file if API_SPEC_SOURCE is "url".

Optional/Required: Optional

- **API_SPEC_ALIAS:**

Optional/Required: Optional

Cypress test env vars

- **SCREENSHOTS_SOURCE**

Description: Source for ScreenShot for which to generate test.

Supported Values: File, Gitpath, URL

Optional/Required: Required

- **SCREENSHOTS_URL**

Description: URL of the screenshot.

Optional/Required: Optional

- **SCREENSHOTS_FILE_PATH**

Description: Specifies the path of screenshot.

Optional/Required: Optional

Application test env variables

- **APPLICATION_ENTRYPOINT**

Description: Specifies the relative path of the application's entry point from the project root. This is used when application test generation is required in addition to unit tests.

Supported Values: Any valid relative path to the entry point file or directory within the project. Only supported for GPT-4-turbo models.

Optional/Required: Optional

API test env vars

- **HTTP_VERBS_FOR_TESTING**

Description: Specifies the HTTP methods to be tested for all APIs. This allows you to filter which HTTP verbs (methods) should be included in the tests.

Default Value: "get,post,put,patch,delete"

Supported Values: Any combination of HTTP verbs, separated by commas (e.g., "get,post,put,patch,delete").

Optional/Required: Optional

- **REGEX_HTTP_ENDPOINTS_FOR_TESTING**

Description: A regular expression string used to match specific endpoints that need to be tested. If this variable is left empty, all endpoints will be included in the testing.

Supported Values: Any valid regex pattern to filter endpoints.

Optional/Required: Optional

License env vars

- **ROOST_DOMAIN**

Description: Specifies the Roost domain to be used. This defines the base URL for RoostGPT services.

Default Value: "app.roost.ai"

Supported Values: Any valid Roost domain URL.

Optional/Required: Optional

- **ROOST_TOKEN**

Description: The authentication token required for accessing RoostGPT CLI. This token is necessary for authenticating and authorizing your requests to RoostGPT.

Optional/Required: Required

- **TELEMETRY**

Description: Determines whether telemetry data is sent to Roost. Telemetry data helps improve the service but does not include private information.

Default Value: "true"

Supported Values: "true", "false"

Optional/Required: Optional

Additional vars

- **TEST_NAME**

Description: The name assigned to the test.

Default Value: "roost-test"

Supported Values: Any valid string.

Optional/Required: Optional

- **ROOST_DIR**

Description: The directory path where RoostGPT files and configurations are stored.

Default Value: "/var/tmp/Roost/RoostGPT"

Supported Values: Any valid directory path.

Optional/Required: Optional

- **LANGUAGE**

Description: The programming language of the source code to be tested.

Default Value: "java"

Supported Values: "java", "go", "python", "csharp", "node.js"

Optional/Required: Optional

- **AI_TYPE**

Description: Specifies the AI model used for generating tests.

Default Value: "openai"

Supported Values: "openai", "vertexai", "open_source_ai", "azure_open_ai", "sagemaker_model", "claude_ai", "dbrx"

Optional/Required: Optional

- **JAVA_BUILD_TOOL**

Description: Specifies the java build tool for which to generate test.

Default Value: "maven"

Supported Values: "maven", "gradle".

Optional/Required: Optional

- **PACKAGES_TO_SCAN**

Description: Specifies the packages to be scanned for test generation. Supports single package or comma-separated values.

Supported Values: Comma-separated list of package names (e.g., "com.example.product" or "com.example.product, com.example.controller").

Optional/Required: Required for Java

- **ITERATION**

Description: The number of iterations to run and improve the tests.

Default Value: "1"

Supported Values: Any positive integer.

Optional/Required: Optional

- **TIMEOUT**

Description: The duration after which test generation will automatically stop.

Default Value: "1 hour"

Supported Values: Any valid duration string (e.g., "30 minutes", "2 hours").

Optional/Required: Optional

- **TEST_TYPE**

Description: The type of test to generate.

Default Value: "unit"

Supported Values: "unit", "functional", "api-spec-test", "integration"

Optional/Required: Optional

- **TEST_FRAMEWORK**

Description: Specifies the testing framework to be used based on the programming language and test type.

Supported Values:

For Python: "pytest", "unittest"

For Golang: "gotest"

For Java: "JUnit4", "JUnit5"

For Csharp: "nunit"

For Node.js: "jest", "mocha"

For API spec tests or integration tests: "postman", "artillery", "rest-assured", "karate"

Optional/Required: Optional

- **TRIGGER_ID**

Description: Unique identifier for triggers. Used to distinguish between multiple triggers.

Default Value: Epoch timestamp

Supported Values: Any unique string or timestamp.

Optional/Required: Optional

- **BOARD_TYPE**

Description: Specifies the type of board to be used for tracking.

Default Value: "jira"

Supported Values: "jira", "azure", "none"

Optional/Required: Optional

- **MAX_DEPTH**

Description: Maximum depth of directories to search for test files. If set to -1, it will traverse all subdirectories.

Supported Values: Any non-negative integer or -1.

Optional/Required: Optional

- **FUNCTIONS_TO_TEST**

Description: List of function names to be tested, specified as module.[class.]method or module.function.

Supported Values: Comma-separated list of function names.

Optional/Required: Optional

- **USE_CACHE**

Description: Use local cache to generate test.(Only for Developers)

Default Value: "false"

Supported Values: "true","false".

Optional/Required: Optional

- **VERIFY_TEST**

Description: Indicates whether to verify if the generated tests are running correctly.

Supported Values: "true", "false"

Optional/Required: Optional

- **VULNERABILITY_GEN**

Description: Specifies whether to generate a vulnerability report based on the tests.

Supported Values: "true", "false"

Optional/Required: Optional

- **CONSIDER_EXISTING_TEST**

Description: Used for Java, python, Golang unit test generation, It scans your repository to identify any existing tests for the specified function. If tests are found, it incorporates them into its consideration and generates additional test cases as needed.

Default Value: "true"

Supported Values: "true", "false"

Optional/Required: Optional

- **GRADING_NOTES**

Description: Grading notes are audit criteria used to evaluate and improve AI generated response for better accuracy and effectiveness.

Default Value: "true"

Supported Values: "true", "false"

Optional/Required: Optional

- **CUSTOM_TAGS**

Description: The tags to be attached to test code, so that the test's can run based on tags

Optional/Required: Optional

- **ROOST_USER_INPUT**

Description: User input to be used for generating tests.

Supported Values: Any valid input string.

Optional/Required: Optional

Improve Test env vars

- **FILE_PATH**

Description: Location of the file containing the test if the IMPROVE_TEST variable is set to "true." This file will be used as a basis for improving the generated test.

Supported Values: Any valid file path.

Optional/Required: Required if IMPROVE_TEST is set to "true."

- **USER_CONTENT**

Description: Content provided by the user to be used for improving the generated test.

This content will be incorporated into the improvement process.

Supported Values: Any valid text or data provided by the user.

Optional/Required: Required if IMPROVE_TEST is set to "true."

- **TESTSCRIPT_ENDPOINT**

Description: The endpoint to be improved when the TEST_TYPE is set to "postman." This specifies which API endpoint the Postman test script will target.

Supported Values: Any valid API endpoint URL.

Optional/Required: Required if TEST_TYPE is set to "postman."

Language version env vars

- **JAVA_VERSION**

Description: Specifies the version of Java to be used when the TEST_FRAMEWORK is "JUnit4", "JUnit5", "karate", or "rest-assured".

Default Value: "17"

Supported Values

Optional/Required: Optional

- **MAVEN_VERSION**

Description: Specifies the version of Maven to be used for Java projects when the TEST_FRAMEWORK is "JUnit4", "JUnit5", "karate", or "rest-assured".

Default Value: "3.8.6"

Supported Values

Optional/Required: Optional

- **GRADLE_VERSION**

Description: Specifies the version of Gradle to be used for Java projects when the TEST_FRAMEWORK is "JUnit4", "JUnit5", "karate", or "rest-assured".

Default Value: "8.6"

Supported Values

Optional/Required: Optional

- **GO_VERSION**

Description: Specifies the version of Go to be used when the TEST_FRAMEWORK is "gotest".

Default Value: "1.19.9"

Supported Values

Optional/Required: Optional

- **PYTHON_VERSION**

Description: Specifies the version of Python to be used when the TEST_FRAMEWORK is "pytest" or "unittest".

Default Value: "3"

Supported Values

Optional/Required: Optional

- **DOTNET_VERSION**

Description: Specifies the version of .NET to be used when the TEST_FRAMEWORK is "nunit".

Default Value: "8"

Supported Values

Optional/Required: Optional

- **NODE_VERSION**

Description: Specifies the version of Node.js to be used when the TEST_FRAMEWORK is "artillery", "postman", "rest-assured", "karate", "jest", or "mocha".

Default Value: "18.18.0"

Supported Values

Optional/Required: Optional

Revision #43

Created 14 June 2023 08:07:49 by Divyesh

Updated 23 October 2024 09:24:13 by Yash