

# 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.
  - `--web`: It opens the web form in order to create downloadable configuration file.
  - `--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).

# 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]

# 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

# 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

# Log env vars
LOG_SOURCE=                     # Optional; Default is elks
LOG_SOURCE_PATH=               # Optional; the log file path
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

# 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.

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.

# License env vars

ROOST\_DOMAIN= # Optional; Default is app.roost.ai

ROOST\_TOKEN= # Optional; the Roost token or license filepath, default

\$HOME/.roost/license.ral

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 separated values. E.g.

"com.demo.sample,"com.example.products"

ITERATION= # Optional; Default is 1

```

TIMEOUT=                                # Optional; Default is 1 hour
CREATE_API_SPEC=                        # Optional; Used for TEST_TYPE=functional
ROOST_USER_INPUT=                      # Optional; The user input will be used to generate tests.
TEST_TYPE=                             # Optional; Default is unit Supports [unit, functional,
artillery, integration, api-spec-test]
RETRIGGER=                             # Optional; Default is false, Supports [true, false]
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
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]
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_ASSISTANT=                         # Optional; Use Assistant feature in openai. Default:false
VERIFY_TEST=                           # Optional; Default is true, Supports [true, false]
VULNERABILITY_GEN=                     # Optional; Default is false, Supports [true, false]

# Improve test env vars
GIT_PR_URL=                            # The Git PR URL of the generated test
IMPROVE_TEST=                          # Optional; Default is false, Supports [true, false]
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,
https://docs.roost.ai/topics/roostgpt/page/supported-features to check the supported language
versions
JAVA_VERSION=                          # Optional; Used for TEST_FRAMEWORK "JUnit4, "JUnit5", "karate",
"rest-assured"
MAVEN_VERSION=                         # Optional; Used for TEST_FRAMEWORK "JUnit4, "JUnit5", "karate",
"rest-assured"
GRADLE_VERSION=                        # Optional; Used for TEST_FRAMEWORK "JUnit4, "JUnit5", "karate",
"rest-assured"
GO_VERSION=                            # Optional; Used for TEST_FRAMEWORK "gotest"
PYTHON_VERSION=                        # Optional; Used for TEST_FRAMEWORK "pytest", "unittest"
DOTNET_VERSION=                        # Optional; Used for TEST_FRAMEWORK "nunit"

```

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

## Here's a description of each variable:

### # Git env vars

- `GIT_TYPE`: Specifies the type of Git repository. Default value is "github"  
Supported values: "github", "gitlab", "azure", "bitbucket". (Optional)
- `HOSTED_TYPE`: Where the git platform is accessible. Default is "cloud", supports "cloud" and "hosted". (Optional)
- `GIT_HOSTED_URL`: URL of the hosted git platform. (Required if `HOSTED_TYPE`="hosted")
- `USE_SSH`: SSH based auth for Git. Default is "false". Supports "true", "false". (Optional)
- `LOCAL_PROJECT_PATH`: Path to your workspace (Required if `GIT_TYPE` is "local")
- `SOURCE_GIT_TOKEN`: A token for authenticating the specific source Git repository. (Required)
- `SOURCE_GIT_CLONE_URL`: the source repo clone URL (Optional)
- `SOURCE_OWNER_NAME`: The name of the source repository owner (Git username). (Required)
- `SOURCE_REPO_NAME`: The name of the source repository. (Required)
- `SOURCE_REPO_ID`: The ID of the source repository. (Required for `GIT_TYPE`="gitlab")
- `SOURCE_PROJECT_NAME`: The name of the source project. (Required for `GIT_TYPE`="azure")
- `SOURCE_PROJECT_ID`: The ID of the source project . (Required for `GIT_TYPE`="bitbucket")
- `SOURCE_BASE_BRANCH`: The base branch that is to be tested in the source repository.  
(Required)
- `SOURCE_RELATIVE_DIRECTORY`: The relative directory path within the source repository.
- `SAME_TARGET_DETAIL`: Set to true if the destination git details are same as source git details.  
Default value is "true", Supports "true" and "false". (Optional)
- `DESTINATION_GIT_TOKEN`: A token for authenticating the specific destination Git repository.  
(Required if `SAME_TARGET_DETAIL`=false)
- `DESTINATION_GIT_CLONE_URL`: the destination repo clone URL (Optional)
- `DESTINATION_OWNER_NAME`: The name of the destination repository owner (Git username).  
(Required if `SAME_TARGET_DETAIL`=false)
- `DESTINATION_REPO_NAME`: The name of the destination repository. (Required if  
`SAME_TARGET_DETAIL`=false)
- `DESTINATION_REPO_ID`: The ID of the destination repository. (Required if  
`SAME_TARGET_DETAIL`=false and `GIT_TYPE`="gitlab")
- `DESTINATION_PROJECT_NAME`: The name of the destination project. (Required if  
`SAME_TARGET_DETAIL`=false and `GIT_TYPE`="azure")
- `DESTINATION_PROJECT_ID`: The ID of the destination project . (Required if  
`SAME_TARGET_DETAIL`=false and `GIT_TYPE`="bitbucket")
- `DESTINATION_BASE_BRANCH`: The base branch in which the test code is to be pushed in the  
destination repository. (Required if `SAME_TARGET_DETAIL`=false)

### # Open AI env vars

- `OPENAI_API_MODEL`: Specifies the type of the OpenAI API model.  
Supported values: "gpt-4", "gpt-3.5-turbo" etc. ( This models availability depends on the `OPENAI_API_KEY` ).  
Default value: "gpt-4". (Optional)
- `OPENAI_API_KEY`: The API key for accessing the OpenAI API. (Required if `AI_TYPE="openai"`)

## # AZURE Open AI env vars

- `AZURE_OPENAI_ENDPOINT`: API Endpoint to access Azure Open AI. (Required if `AI_TYPE="azure_open_ai"`)
- `AZURE_DEPLOYMENT_NAME`: Name of Azure Open AI deployment (Required if `AI_TYPE="azure_open_ai"`)
- `AZURE_OPENAI_KEY`: The API key for accessing the Azure Open AI API (Required if `AI_TYPE="azure_open_ai"`)
- `AZURE_OPENAI_VERSION`: The version of Azure Open AI to be used. (Optional)

## # Vertex AI env vars

- `VERTEX_PROJECT_ID`: The ID of the Vertex project. (Required if `AI_TYPE="vertexai"`)
- `VERTEX_REGION`: The region where the Vertex project is located (example "us-central1").  
(Required if `AI_TYPE="vertexai"` and `VERTEX_FINE_TUNE="true"`)
- `VERTEX_BEARER_TOKEN`: The bearer token for accessing the Vertex API. (Required if `AI_TYPE="vertexai"`)
- `VERTEX_MODEL`: The name of the Vertex model to use. (Required if `AI_TYPE="vertexai"`)  
Supported values: "text-bison", "code-bison". (Required if `AI_TYPE="vertexai"`)
- `VERTEX_FINE_TUNE`: Indicates whether fine-tuning is enabled for the model.  
Supported values: "true", "false". (Optional)  
Default value: "false".

## # Open Source AI env vars

- `OPEN_SOURCE_MODEL_ENDPOINT`: API Endpoint to access Open Source AI model. (Required if `AI_TYPE="open_source_ai"`)
- `OPEN_SOURCE_MODEL`: The name of the Open Source model to use. (Optional)  
Supported values: "meta-llama/Llama-2-13b-chat", "HuggingFaceH4/starchat-beta".

## # SageMaker Model env vars

- `SAGEMAKER_MODEL_ENDPOINT`: The endpoint where the sagemaker model is hosted.(Required if `AI_TYPE=sagemake_model`)

## # Claude AI env vars

- `CLAUDE_AI_MODEL`: Specifies the type of the Claude AI model.  
Supported values: "claude-3-opus-20240229", "claude-3-sonnet-20240229", and "claude-3-haiku-20240307" (Required if `AU_TYPE = "claude_ai"`)



- `CLAUDE_AI_API_KEY`: The API key for accessing the Claude AI API. (Required if `AI_TYPE="claude_ai"`)

## # DBRX AI env vars

- `DBRX_MODEL_ENDPOINT`: The endpoint where the DBRX AI model is hosted.
- `DBRX_AUTH_TOKEN`: The auth token for accessing the DBRX model. (Required if `AI_TYPE="dbrx"`)
- `DBRX_MODEL_NAME`: The name of your DBRX model.(Required if `AI_TYPE="dbrx"`)

## # Advanced AI env vars

- `AI_TEMPERATURE`: The AI Temperature to be used during test generation.(Optional; Default value is 0.6)

## # Jira board env vars

- `JIRA_EMAIL`: The email address associated with your Jira account. (Required if `TEST_TYPE="functional"` and `BOARD_TYPE="jira"`)
- `JIRA_HOST_NAME`: The hostname of your Jira instance. (Required if `TEST_TYPE="functional"` and `BOARD_TYPE="jira"`)
- `JIRA_PASSWORD`: The password for your Jira account. (Required if `TEST_TYPE="functional"` and `BOARD_TYPE="jira"`)
- `JIRA_ID`: The Jira ID associated with your account. (Required if `TEST_TYPE="functional"` and `BOARD_TYPE="jira"`)

## # Azure board env vars

- `AZURE_ORG`: The organization associated with your Azure Devops account. (Required if `TEST_TYPE="functional"` and `BOARD_TYPE="azure"`)
- `AZURE_TOKEN`: The access token for authenticating with Azure Devops. (Required if `TEST_TYPE="functional"` and `BOARD_TYPE="azure"`)
- `AZURE_PROJECT`: The name of the Azure Devops project. (Required if `TEST_TYPE="functional"` and `BOARD_TYPE="azure"`)
- `AZURE_WORK_ITEM_ID`: The ID associated with the relevant work item. (Required if `TEST_TYPE="functional"` and `BOARD_TYPE="azure"`)

## # Log env vars

- `LOG_SOURCE`: origin from where the logs are generated. (Optional)  
Default value: "elks".
- `LOG_SOURCE_PATH`: location where the log files are stored. (Optional)
- `LOG_ELASTICSEARCH_URL`: The URL for the Elasticsearch. (Optional)
- `LOG_ELASTICSEARCH_USER`: A user name for accessing Elasticsearch resources. (Optional)
- `LOG_ELASTICSEARCH_TOKEN`: Authentication used for accessing Elasticsearch resources. (Optional)

- `LOG_ELASTICSEARCH_API_KEY`: The API key to authenticate and authorize access to Elasticsearch APIs. (Optional)

## # Behavioural Test cases env vars

- `BEHAVIORAL_TEST_TYPE`: The type of behavioural tests. Supports gherkin. Supports "gherkin". (Optional)
- `BEHAVIORAL_TEST_SOURCE`: location or the source of behavioural test source. Supports file, gitpath and url. (Optional)
- `BEHAVIORAL_TEST_FILE_PATH`: Path of the source file if source is file/gitpath, Relative path in case of gitpath. (Optional)
- `BEHAVIORAL_TEST_URL`: The URL of the source file if the source is URL. (Optional)

## # API Spec env vars

- `API_SPEC_TYPE`: The type of API specification. Supports Swagger and Postman. (Optional)
- `API_SPEC_SOURCE`: location or the source of api. Supports file, gitpath and url. (Optional)
- `API_SPEC_FILE_PATH`: Path of the source file if source is file/gitpath, Relative path in case of gitpath. (Optional)
- `API_SPEC_URL`: The URL of the source file if the source is URL. (Optional)

## # Application test env variables

- `APPLICATION_ENTRYPOINT`: The relative path of the application's entry point from project root. If application test generation is also required along with Unit tests. Only supported for gpt-4-turbo models. (Optional)

## # API test env vars

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

## # License env vars

- `ROOST_DOMAIN`: This is the Roost Domain, default value for this is app.roost.ai. Default is app.roost.ai. (Optional)
- `ROOST_TOKEN`: Authentication token for RoostGPT CLI. (Required)
- `TELEMETRY`: Send telemetry data to roost, no private information is shared. Default is "true", Supports "true", "false". (Optional)

## # Additional vars

- **TEST\_NAME**: The name of the test. (Optional)  
Default value: "roost-test".
- **ROOST\_DIR**: The directory path for Roost. (Optional)  
Default value: "/var/tmp/Roost/RoostGPT".
- **LANGUAGE**: The programming language of the source code. (Optional)  
Supported values: "java", "go", "python", "csharp", "node.js".  
Default value: "java".
- **AI\_TYPE**: The AI model being used to generate tests. (Optional)  
Supported values: "openai", "vertexai", "open\_source\_ai", "azure\_open\_ai", "sagemaker\_model", "claude\_ai", and "dbrx".  
Default value: "openai".
- **PACKAGES\_TO\_SCAN**: The packages to be scanned, supports single package or comma separated values.  
( Required for JAVA ex. com.example.product or com.example.product, com.example.controller )
- **ITERATION**: The iteration number to improve and run the test. (Optional)  
Default value: "1".
- **TRIGGER\_ID**: The ID of the trigger. Unique id to identify multiple triggers. Default is epoch. (Optional)
- **TIMEOUT**: The timeout duration, after this the test generation will automatically stop. (Optional)  
Default value: "1 hour".
- **CREATE\_API\_SPEC**: For functional tests, will also create an API spec file.
- **ROOST\_USER\_INPUT**: This user Input will be used to generate tests.
- **TEST\_TYPE**: The type of test to generate. (Optional)  
Supported values: "unit", "functional", "api-spec-test", "integration".  
Default value: "unit".
- **RETRIGGER**: Indicates whether to re-trigger existing test generated that from where it left. (Optional)  
Supported values: "true", "false".  
Default value: "false".
- **TRIGGER\_ID**: Unique ID to identify triggers, default is epoch.
- **BOARD\_TYPE**: The type of board to be used. (Optional)  
Supported values: "jira", "azure", "none".  
Default value: "jira".
- **MAX\_DEPTH**: Maximum depth of directories to look for test files, if set to -1 it will traverse all subdirectories. (Optional)
- **TEST\_FRAMEWORK**: Supports [pytest, unittest] for Python, [gotest] for Golang, [JUnit4, JUnit5] for Java, [nunit] for Csharp, [jest, mocha] for Node.js, [postman, artillery, rest-assured, karate] for [test\_type: api-spec-test] or [test\_type: integration]. (Optional)
- **FUNCTIONS\_TO\_TEST**: list of function names to be tested, separated by comma. Name to be specified as module.[class.]method OR module.function. (Optional)
- **USE\_ASSISTANT**: Use the Assistant feature in Openai. (Optional)
- **VERIFY\_TEST**: Verify if the generated tests are running correctly. (Optional)
- **VULNERABILITY\_GEN**: Whether to generate a vulnerability report of the tests or not. (Optional)

## # Improve Test env vars

- `GIT_PR_URL`: Git PR URL of the generated test. (Optional).
- `IMPROVE_TEST`: This is used when we want to improve the generated test. (Optional)  
Supported values: "true","false"  
Default value: "false".
- `FILE_PATH`: location of the file containing test. (Required if `IMPROVE_TEST` = true.)
- `USER_CONTENT`: Content Provided by the user for improving the generated test. (Required if `IMPROVE_TEST` = true.)
- `TESTSCRIPT_ENDPOINT`: The endpoint to be improved (Required if `TEST_TYPE` = "postman")

## # Improve Test env vars

- `JAVA_VERSION`: Used for `TEST_FRAMEWORK` "JUnit4", "JUnit5", "karate", "rest-assured".
- `MAVEN_VERSION`: Used for `TEST_FRAMEWORK` "JUnit4", "JUnit5", "karate", "rest-assured".
- `GRADLE_VERSION`: Used for `TEST_FRAMEWORK` "JUnit4", "JUnit5", "karate", "rest-assured".
- `GO_VERSION`: Used for `TEST_FRAMEWORK` "gotest".
- `PYTHON_VERSION`: Used for `TEST_FRAMEWORK` "pytest", "unittest".
- `DOTNET_VERSION`: Used for `TEST_FRAMEWORK` "nunit".
- `NODE_VERSION`: Used for `TEST_FRAMEWORK` "artillery", "postman", "rest-assured", "karate", "jest", "mocha".

---

Revision #29

Created 14 June 2023 08:07:49 by Divyesh

Updated 22 July 2024 10:57:10 by Divyesh