CLI

Installation Command

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

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-macos` (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 config create` : Creates roostgpt configuration file. The configuration file contains environment variables used by RoostGPT application.
- `--ui` : 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 code.
- `--config,-c <path>` : Path to the configuration file if provided, otherwise the configuration will retrieved from Environment variables .

Instructions for creating .env file

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


# 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]


# 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


# 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


# 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]
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


# License env vars
ROOST_DOMAIN=                 # Optional; Default is app.roost.ai
ROOST_TOKEN=                  # Required, the Roost token
TELEMETRY=                    # Optional; Default is true, Supports [true, false], send
telemetry data to roost, no private information is shared.


# Additional 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]
AI_TYPE=                      # Optional; Default is openai; Supports [openai, vertexai,
azure_open_ai, open_source_ai]
PACKAGES_TO_SCAN=             # Required for LANGUAGE=java, the package to scan and generate
test for example - com.demo.sample
ITERATION=                    # Optional; Default is 2
TIMEOUT=                      # Optional; Default is 1 hour
TEST_TYPE=                    # Optional; Default is unit Supports [unit, functional,
artillery, artillery-api-spec, integration]
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]
GIT_PR_URL=                   # The Git PR URL  of the generated test
```

```
# Improve test env vars
IMPROVE_TEST=                    # Optional; Default is false, Supports [true, false]
FILE_PATH=                       # Required if IMPROVE_TEST=true
USER_CONTENT=                    # Required if IMPROVE_TEST=true
```

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")

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".

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")

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")

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)

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".
  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: "2".
- `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".
- `TEST_TYPE` : The type of test to generate. (Optional)
  Supported values: "unit", "functional", "artillery", "artillery-api-spec","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".
- `BOARD_TYPE` : The type of board to be used. (Optional)
  Supported values: "jira", "azure","none".
  Default value: "jira".
- `GIT_PR_URL` : Git PR URL of the generated test. (Optional)

Improve Test env vars

- `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.)