

# 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=github # Default is github; Supports [github, gitlab, azure, bitbucket, local]
HOSTED_TYPE=cloud # Default is cloud; Supports [cloud, hosted]
GIT_HOSTED_URL= # Required if HOSTED_TYPE=hosted
USE_SSH=false # 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=true # 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
ENABLE_SECRET_SCANNING_BYPASS= # Optional; Default is true; Supports [true, false]

# Open AI env vars
```

```
OPENAI_API_MODEL=gpt-4                # Optional; Default is gpt-4
OPENAI_BASE_URL=                       # Optional; Default is "https://api.openai.com/v1"
OPENAI_API_KEY=                         # Required if AI_TYPE=openai

# Vertex AI env vars
VERTEX_FINE_TUNE=false                 # Optional; Default is false, Supports [true, false]
VERTEX_CREDENTIAL_TYPE=               # Required if AI_TYPE=vertexai; Supports [file, token]
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 and
VERTEX_CREDENTIAL_TYPE=token
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]
VERTEX_SERVICE_ACCOUNT_KEY=           # Required if AI_TYPE=vertexai and
VERTEX_CREDENTIAL_TYPE=file; path to the service account key file

# 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=2024-02-15-preview # Optional Default is 2024-02-15-preview
AZURE_INFERENCE_AI=                  # Optional; Default is false; Supports [true, false]

# 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
```

```
# AWS Bedrock env vars
AWS_BEDROCK_MODEL=                # Required if AI_TYPE=AWS_BEDROCK
AWS_DEFAULT_REGION=                # Required if AI_TYPE=AWS_BEDROCK
AWS_ACCESS_KEY_ID=                 # Optional if AI_TYPE=AWS_BEDROCK
AWS_SECRET_ACCESS_KEY=             # Optional if AI_TYPE=AWS_BEDROCK
AWS_SESSION_TOKEN=                 # Optional if AI_TYPE=AWS_BEDROCK

# Advanced AI env vars
AI_TEMPERATURE=0.2                 # Optional; Default value is 0.2
USE_ASSISTANT=false                 # 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
CREATE_EXCEL=                         # Optional; Default is false; Supports [true, false]
TICKET_INPUT_TYPE=                   # Optional; Default is id; Supports [id, file]
TICKET_FILES=                         # Required if TICKET_INPUT_TYPE is file, List of comma separated files

# Log env vars
LOG_SOURCE=elks                      # Optional; Default is elks
LOG_SOURCE_PATH=                     # Optional; the log file path
LOG_FILE_ENCODING=                   # Optional; the log file encoding; Default is utf-8
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=

# Proto Spec env cars
PROTO_SPEC_SOURCE=                     # Optional; Supports [file, gitpath, url, gitrepo]
PROTO_SPEC_FILE_PATH=                  # Optional; path of the source file if source is file/gitpath. Relative
path in case of gitpath
PROTO_SPEC_URL=                         # Optional; URL of the source file if source is url

PROTO_SPEC_GIT_REPO_URL=                # Optional; the proto repo clone URL
PROTO_SPEC_GIT_BRANCH=                  # Required: branch of git repo of proto file
PROTO_SPEC_REPO_NAME=                   # Required; Repo name of proto file
PROTO_SPEC_REPO_ID=                     # Required if GIT_TYPE=gitlab
PROTO_SPEC_PROJECT_NAME=                # Required for GIT_TYPE=azure
PROTO_SPEC_PROJECT_ID=                  # Required for GIT_TYPE=bitbucket

API_TEST_SERVER_CONFIG=                 # Optional; The api server config

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

# Additional env vars
TEST_NAME=roost-test            # Optional; Default is roost-test
ROOST_DIR=/var/tmp/Roost/RoostGPT # Optional; Default is /var/tmp/Roost/RoostGPT
LANGUAGE=java                   # Optional; Default is java; Supports [java, go, python, csharp, nodejs,
cpp]
AI_TYPE=openai                  # Optional; Default is openai; Supports [openai, vertexai, azure_open_ai,
open_source_ai, sagemaker_model, claude_ai, dbrx, bedrock_ai, deepseek_ai]
JAVA_BUILD_TOOL=maven           # 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=1                     # Optional; Set VERIFY_TEST=true to enable iteration; Default value of
ITERATION is 1
TIMEOUT=1                       # Optional; Default is 1 hour
ROOST_USER_INPUT=               # Optional; The user input will be used to generate tests.
TEST_TYPE=unit                  # Optional; Default is unit Supports [unit, functional, artillery, integration,
api-spec-test, ui-test, functional-source]
BOARD_TYPE=jira                 # Optional; Default is jira; Supports [jira, azure, none]
MAX_DEPTH=-1                    # 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, karate, playwright] 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
VERIFY_TEST=false               # Optional; Default is false, Supports [true, false]
VULNERABILITY_GEN=false        # Optional; Default is false, Supports [true, false]
CUSTOM_TAGS=                    # Optional; The tags to be attached to test code, so that the test's can
run based on tags
```

```

CONSIDER_EXISTING_TEST=false          # Optional; Default is false, 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
CREATE_UNIQUE_BRANCH_FOR_REPORT=      # Optional; Default is false; Supports [true, false]
ONE_TEST_FILE_PER_SOURCEFILE=        # Optional; valid for LANGUAGE=go , if user want to generate a
single test file per source file
GRADING_NOTES=
USER_SCENARIOS=
USE_CACHE=

# UI Test vars
BROWSER_USE_HEADLESS=true            # Optional; Default is false; Supports [true, false]
URLS_TO_TEST=                        # Required if TEST_TYPE="ui-test" and TEST_FRAMEWORK="playwright"

# User can provide additional env var starting with "UI_" prefix
UI_LOGIN_TYPE=                      # Optional; use only if TEST_TYPE="ui-test" and
TEST_FRAMEWORK="playwright"
UI_SITE_USERNAME=                   # Optional; use only if TEST_TYPE="ui-test" and
TEST_FRAMEWORK="playwright"
UI_SITE_PASSWORD=                   # Optional; use only if TEST_TYPE="ui-test" and
TEST_FRAMEWORK="playwright"          #

# 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=17                     # Optional; Default is 17, Used for TEST_FRAMEWORK "JUnit4, "JUnit5",
"karate", "rest-assured"
MAVEN_VERSION=3.8.6                 # Optional; Default is 3.8.6, Used for TEST_FRAMEWORK "JUnit4,
"JUnit5", "karate", "rest-assured"
GRADLE_VERSION=8.6                  # Optional; Default is 8.6, Used for TEST_FRAMEWORK "JUnit4,
"JUnit5", "karate", "rest-assured"
GO_VERSION=1.19.9                   # Optional; Default is 1.19.9, Used for TEST_FRAMEWORK "gotest"
PYTHON_VERSION=3                     # Optional; Default is 3, Used for TEST_FRAMEWORK "pytest",
"unittest"
DOTNET_VERSION=8                     # Optional; Default is 8, Used for TEST_FRAMEWORK "nunit"
NODE_VERSION=18.18.0                # Optional; Default is 18.18.0, Used for TEST_FRAMEWORK "artillery",
"postman", "rest-assured", "karate", "jest", "mocha"
KARATE_VERSION=1.5.1                 # Optional; Default is 1.5.1, Used for TEST_FRAMEWORK "karate",

```

**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"

---

Revision #1

Created 5 September 2025 06:51:31 by Harish

Updated 5 September 2025 06:51:31 by Harish