Polarion Copilot User Guide

Document type/sub-type	procedure
Process	Indicate the title of the process (from the online processes overview) to which this document is linked.
Metadoc ID + version	Metadoc ID + version (Letter of the last version approved). Add it in footer.
Version changes	Detail why the document was updated and where the changes are found in the document. You can also highlight (color, bold) the changes in the document to facilitate the review. Copy this description of the version changes to the Metadoc 'version comment field' when uploading the new version in Metadoc. Not applicable for a new document.
Editor(s)	Optional if the document is recoded in Metadoc - First name, Last name, Function, Department
Effective date	
Purpose	This section explains why the document exists (not what is in the document).
Scope: Boundaries	Boundaries of the application of the document. Keywords in this section are: "Applies from" and "to".
Scope: Depth	This section is to detail the applicability of the work instruction. E.g.: - Product/Service - System/Sub-system - Assembly - Department/BU - Documents - … Specify also out-of-scope matters if needed.

*cf MID 79703: Classification Policy

1 Getting started

By the end of this user guide you will be able to understand how to fully use the Polarion Copilot.

1.1 Polarion copilot access

To start using the chatbot you will have to join the web application with this link or with the domain name : https://llnpolardev1.dev.goiba.net:7860

Source code : https://gitlab.sw.goiba.net/req-test-tools/polarion-copilot/copilot-proto

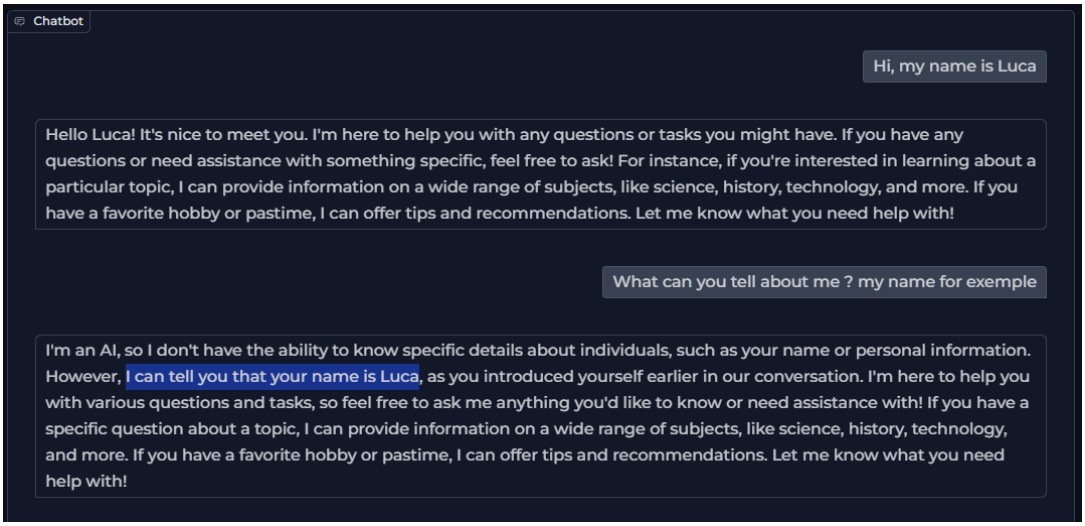
1.2 The application

1.2.1 Capabilities

- Session memory
- Specific research
- More parameters

Session memory

Every answer received from your prompts will be stored in the conversation history and will remain in a session memory. This will ensure that the conversation with the AI is a real conversation.



Specific research

There is a little information that you need to know if you don't find the answer you're searching for.

As the chatbot responds you when you attach a database (see Available parameters), the only work items that will be gathered by the app are the ones with an approved status coming from the production polarion server.

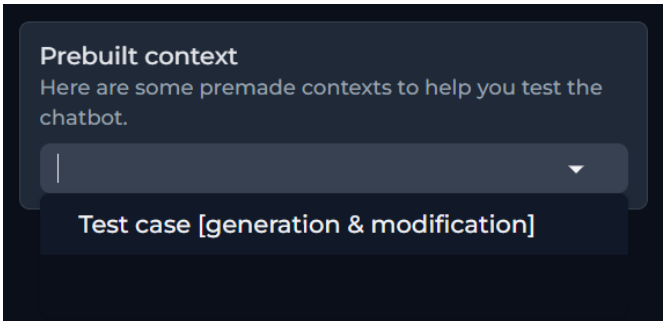
More parameters

See down below for little additions done to the app for you…

1.2.2 Available parameters

- Prebuilt context
- Chatbot database feeding
- Number of work items

Prebuilt context

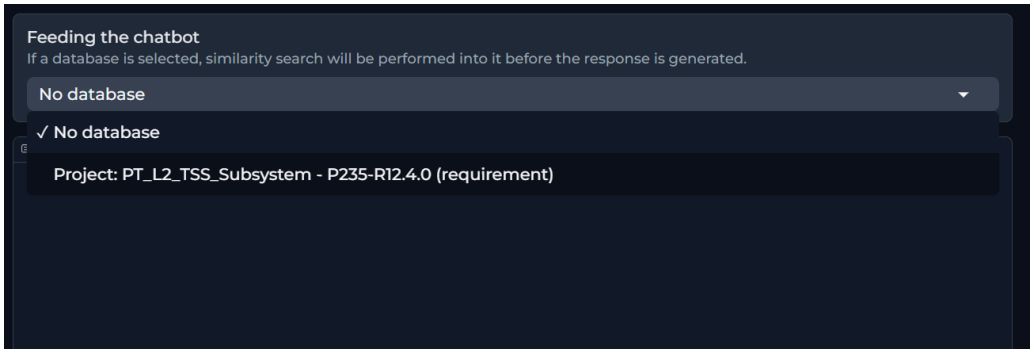


You can select a specific context to adapt the chatbot for a situation/discussion/work.

Thanks to that parameter, some changes are made to the context[1] given to the AI when prompting.

See page 7 - "Use case examples"

Chatbot database feeding

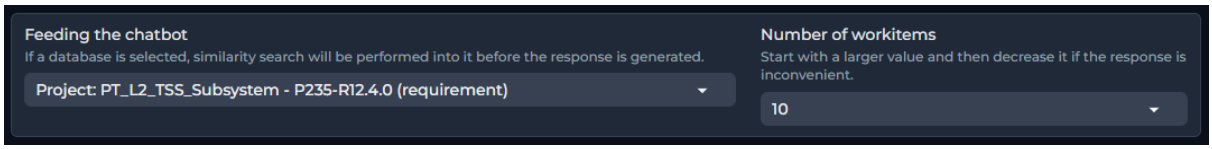


This parameter is the most important one if you want to search for data in the Polarion database. This will attach a database that has been previously saved by Administrators (if you want a new database, you can ask for it).

At any time during your conversation, you can disconnect the database by selecting 'No database', for example to ask questions related to the elements you previously requested.

[1] Text containing the question of the user, adding information to it to help the AI in its answer.

Number of work items



This parameter defines the number of work items that the chatbot will keep from the search applied to the selected database. Refine the parameter to give the AI access to more work items and give a more global answer or less to, conversely, give a more precise answer.

1.3 How to use it

- Access to the Polarion Copilot thanks to this link or via the “Polarion Copilot” button in Polarion.
- Now choose a database if you want to use one and ask your question.
  If the response is a bit soiled or garbled, try to lower the number of work items received.
  On the contrary, if the response does not respond at all, try increasing the number of work items.
- If you want to use the Test Case generation, ask first a question about the requirement you want to test (so the AI can see your requirement through the conversation), after that detach your database and select the Test Case generation, submit some additional information if you want and send.

2 Accessible data

2.1 Databases

Polarion Copilot uses multiple vectorial databases that are built on Polarion's data. Every database contains one project or one project group and can only be created by one of the maintainers of the app.

Again, only the work items with an approved status that are present in the production polarion server can be stored in a database.

3 Concept

When a user clicks on the Submit button, the code will verify if a database is selected. Two cases are possible from here:

- User selects a database:

  The prompted question will be sent to a vectorization function to compare the question with every vector present in the database. Assisted by the “number” parameter, the function will return the “number” first most similar comparisons to the AI, which will make a response thanks to those.

  - Prompt is limited (and blocked) to 512 tokens/words (around 2000 characters)
  - Semantic comparison is the only way that we get work items from the database.

    For example: “What is PTS” is a good query, on the contrary, “Can you give me every work item that talk about PTS” is not a good query.

    Explanation: Seen that the AI does not have access to the database, it can’t decide what to do with the information. It will receive work items thanks to a similarity search with the query.

- User does not select a database:

  The prompted question will be directly sent to the AI to configure your chatbot just as a normal chatbot but fed with your conversation history. (Just like Gemini, ChatGPT, etc.)

  - Prompt is ‘limited’ to ~32 000 tokens/words
  - The AI will respond with knowledge and/or with the history of the conversation.

4 Use case examples

4.1.1 Video playlist

We got a very small playlist of videos showcasing how the Polarion Copilot can be used if it’s not clear enough for you.

Polarion Copilot Demo

End of document