/ask swimm

Cheat Sheet

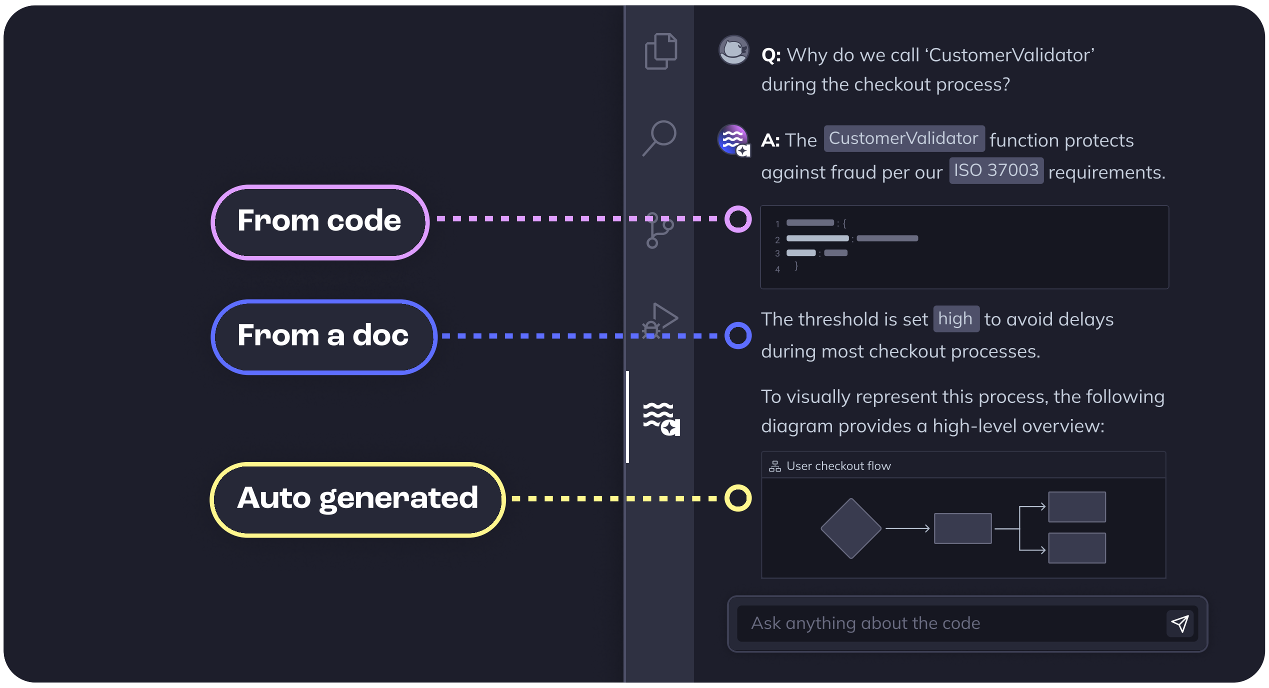
swimm.io

Introduction

/ask Swimm is an AI assistant that answers technical questions immediately, accurately, and completely contextualized to your codebase. Code doesn't always explain itself, in fact most times it doesn't. That's because a lot of the logic about code isn't evident from the code itself.

Swimm's knowledge base builds an understanding of your specific code and is enriched over time with the "human context" around the code added to your Swimm documents. "Human context" might include business logic, architectural decisions, reasons why we did or did not do something, etc. Since Swimm documents are connected with the code, all of the contextual knowledge will stay up to date over time – only feeding /ask Swimm with relevant, timely information.

This document describes best practices to get the most out of /ask Swimm.



Q: Why do we call 'CustomerValidator' during the checkout process?

A: The CustomerValidator function protects against fraud per our ISO 37003 requirements.

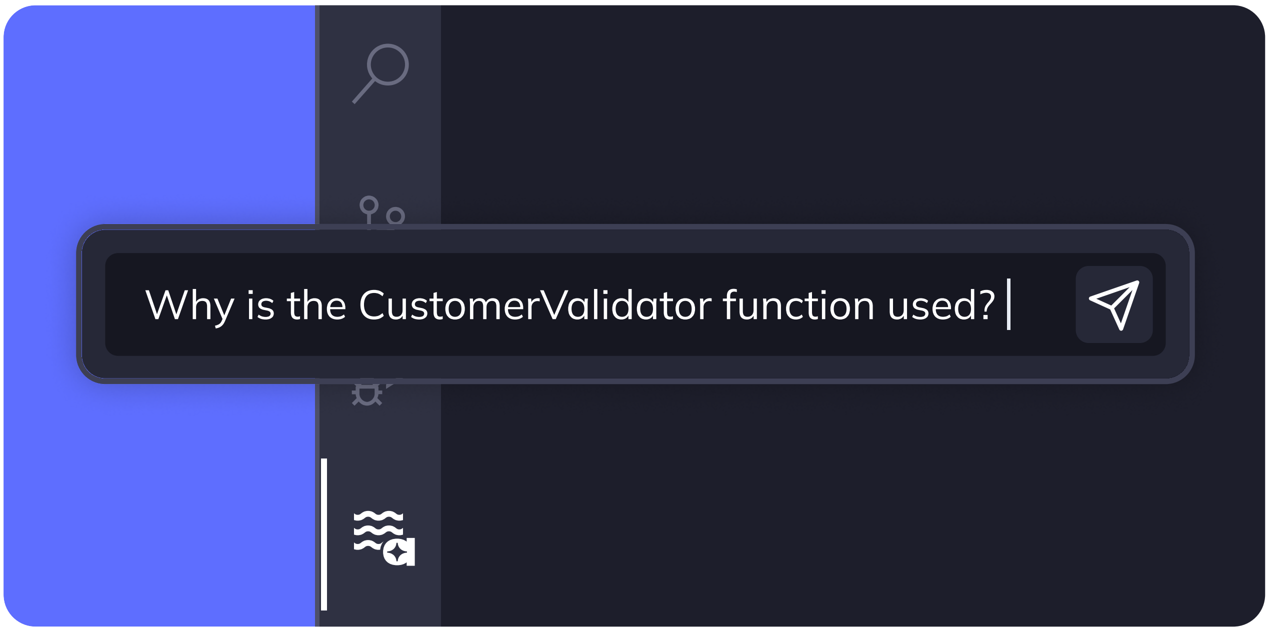
The threshold is set high to avoid delays during most checkout processes.

To visually represent this process, the following diagram provides a high-level overview:

Examples for Questions

/ask Swimm knows your codebase and can help you answer a variety of questions. These are some common types of questions that you can ask:

- **What is X?**
  Example: “what is a Playlist in this repository?”
- **Where is X implemented?**
  Example: “where is the editor implemented?”
- **How do we X in Y?**
  Example: “how do we send analytics in the editor?”
- **Where do we use X?**
  Example: “where do we use the Firestore API in the code?”
- **How to add X?**
  Example: “how should I add a new plugin to the system?”


Why is the CustomerValidator function used?

swimm

Best Practices

Be specific 🎯

Make sure you ask a single question each time.

✓ ‘Where is the Playlist feature implemented?’

✗ ‘Where is the Playlist feature implemented and how do we use it effectively?’

✓ ‘How do we trigger a run of the CLI?’

✗ ‘How do we trigger a run of the CLI and why would we do it in the backend?’

Keep it short

Don’t overload your question with fancy words. Keeping the question short improves /ask Swimm’s ability to answer your question.

✓ ‘Where is the Playlist feature implemented?’

✗ ‘Where is the code that implements the feature of Playlists resides in this repository?’

✓ ‘How do we validate input from the user in forms?’

✗ ‘When the user provides information, it can be in many places, and we need to validate the information that the user provided. How do we do that?’

swimm

Best Practices

Provide context to avoid ambiguity

Codebases tend to be big and complex. Many times, a certain term can mean different things in different context.

Consider the following question:

✗ 'How do we log an error?'

It is common to have many different implementations for logging an error in different parts of your system. If that's the case, you should explain which module you expect the answer to be based on. For example:

✓ 'In the utils module, how do we log an error?'

Use follow up questions

After asking a question, you may receive an answer that was incomplete, or didn't answer what you meant. No worries - you can keep asking follow up questions on the same thread - /ask Swimm will be aware of your discussion. Of course, the same applies if you got an accurate answer that then led you to another question.

Follow up with instructions

Before, we mentioned that you can use follow up questions to improve answers. You can also follow up on an answer with an instruction, for example:

→ 'Draw a diagram of the flow you described'

→ 'Explain this in a short paragraph'

swimm

Why did /ask not answer my question?

Asking complex questions about codebases can be challenging.
While /ask Swimm excels at this, it may not always find the relevant answer in two main scenarios:

- The answer exists in the code - in some cases, /ask Swimm was not able to find the answer even though it exists in the code. This may be due to a configuration error or other issue. We would love to learn more about your specific case and provide support as needed.

  Please contact your sales representative, our support inbox at info@swimm.io, or post an inquiry on our shared Slack channel for personalized assistance. Additionally, improving /ask with documents helps Swimm provide the right answer the next time someone asks a relevant question.

- The answer doesn't exist in the code - oftentimes, there is information that simply does not exist in your codebase. It may be business logic (why we did something), an architectural decision someone took, what a certain magic number in the code means, even why something was not implemented in another way.

  In these cases, /ask Swimm does not invent an answer, but might tell you it doesn't exist in the code. If you /ask with documents, Swimm will be able to provide the right answer the next time someone asks that question.

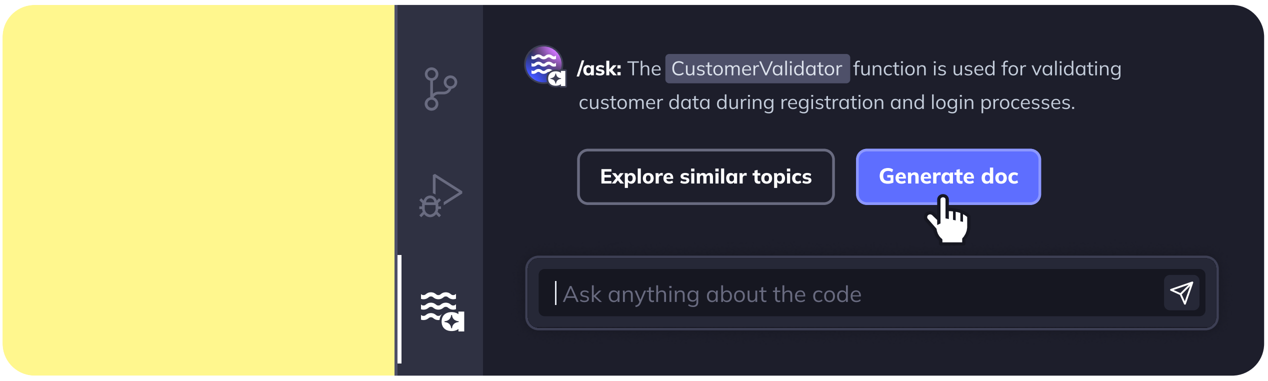
Improve /ask with documents

When you create a document with Swimm, your knowledge is fed into /ask Swimm's knowledge. When someone asks a question relevant to the information you wrote, it may be used as part of the answer.

This is particularly important when it comes to information that AI can't extract directly from the code. The more you act as the "human in the loop," and codify your knowledge into the Swimm knowledge base, the higher the quality of /ask Swimm. For example, consider feeding /ask Swimm with an explanation of why you didn't implement a certain functionality within a feature. If someone ever asks about this feature, they may get this information as well.

The challenge in relying on human-generated knowledge in any other system is that it might be outdated. For example, you may have previously written something about a part of the system that is no longer accurate today as the code changed. With Swimm's patented Auto-sync technology, you don't have to worry about feeding AI with outdated knowledge.

Remember, whenever you provide input to /ask Swimm, you contextualize it more and more to your organization, thus **improving future answers for you and your team**.



/ask: The CustomerValidator function is used for validating customer data during registration and login processes.