
Why was I charged twice for an upgrade and now have 2 separate accounts?

Quinn (Support) <support@replit.com>
Reply-To: Support <support@replit.com>
To: TMH Global <tmhglobal@gmail.com>

Tue, Sep 9, 2025 at 5:32 PM

Hello Thomas,

Thank you for contacting Replit Support.

Let me first clarify the \$80 charges you're seeing. Looking at your account, there is one actual charge of \$80 from September 7th for your Teams subscription. The second \$80 amount you're seeing is a draft invoice for October 7th - this is not an actual charge yet, but rather a preview of your upcoming renewal.

Regarding the AI-generated apps and usage concerns, let me share some important information about how our AI tools work:

Best Practices

To get the most out of the Agent:

- Use clear, specific prompts that include relevant context.
- Test all code thoroughly before using it in production.
- Use the AI iteratively—review its outputs, make adjustments, and continue the conversation.
- Avoid copy-pasting large sections of generated code without understanding its function.

How Agent Works

Our Agent is built on LLMs, which are powerful statistical models trained on vast datasets of natural language and code. While they are capable of remarkable feats—like writing entire functions, assisting with debugging, or generating architectural suggestions—they do not "understand" code in the way a human does. Their outputs are generated probabilistically based on input prompts, context, and learned patterns.

This means the Agent:

- Does not guarantee correctness in the code it generates.
- May produce different results for the same prompt at different times.
- Cannot reason about project-specific goals, business logic, or edge cases unless explicitly told to do so.

Why We Can't Guarantee Outcomes

- Prompt Quality: The Agent relies entirely on the instructions and context provided by the user. Vague or incomplete prompts often lead to incorrect or suboptimal results.
- Project Complexity: Complex software projects often involve dependencies, nuanced logic, and architecture-specific constraints that the Agent may not fully comprehend from limited context.
- Debugging & Supervision: The Agent still requires human oversight. It's up to the user to review, test, and iterate on the code suggestions provided.
- Randomness: LLMs are non-deterministic by design—meaning the same prompt can lead to slightly different outputs depending on internal randomness and current context.

Agent Limitations

While we continuously strive to improve the reliability and usefulness of our Agent, we do not assume responsibility for the correctness, safety, or effectiveness of any code generated by the Agent. This includes, but is not limited to:

- Security vulnerabilities
- Logical bugs
- Performance regressions
- Violations of best practices or style guide

For better results with the Agent, I recommend using this prompting method:

```
START OF 1ST PROMPT
```

```
I am trying to (Explain your problem or goal in deep detail here)I want you to do the following:
```

```
- Research deeply across my codebase
```

```
- Find what files and functions are related to the problems and goals I outlined above
```

```
- Assess reasons for why the feature might not be working or why it might not work. If I am asking for something impossible or a task you do not have the tools to accomplish, let me know.
```

```
- Develop a plan to fix it
```

```
- Write this entire plan and report into a file called "Instructions.md" in my project
```

```
END OF 1ST PROMPT
```

Once the Assistant is done writing this new file ^ (Instructions.md) Start a new Assistant chat or new Agent chat (I would go with Assistant if your project is further along) and prompt it with:

```
START OF 2ND PROMPT
```

```
I want you to fix and implement (short description about goal or bug you have). Before you begin, please carefully read and adhere to the plan
```

and insights found in the "Instructions.md" file and follow those instructions.If there is something in those Instructions that might stop you, let me know and guide me on how to unblock you.

END OF 2ND PROMPT

While we can't rollback your local code to your deployment code, you are welcome to use the history feature in your code to go back to previous versions:

<https://docs.replit.com/replit-workspace/workspace-features/file-history>

It is a bit manual of a process since you have to revert the files individually. If you were connected to Git and maintaining version control, that is another way you can revert to a previous version of your code. Git is the recommended solution for future rollbacks.

The only other option is the Agent rollback if you are using the Replit Agent feature, but Git management is still recommended.

You can view your detailed AI usage and charges at <https://replit.com/usage>.

Would you like me to help you with:

1. Cancelling your Teams subscription?
2. Looking into a potential refund for your most recent subscription payment?

Please let me know how you'd like to proceed.

Regards,
Quinn

This email is a service from Replit. Delivered by **Zendesk**

[Quoted text hidden]
[LZ4J3D-312RY]