P2P Code Editor Extension for VS Code

Lima IoT project proposal

Idea

A P2P Code Sharing Extension for VS Code would allow developers on different machines in different physical locations to connect with each other and share their editor, similarly to how screen-sharing works. Sharing an editor is better than sharing your screen, because that would allow the other connected peers to copy, save, and edit the code – all things you cannot do with a shared screen.

Use Cases

TA/Lectures

You are a TA giving a live demo of a new algorithm or how to set up your new IoT device. The students usually ask you to share the code, or they take pictures during class. Instead – have the students follow along with the code, where they can copy it at any time, save it locally or make suggestions on how to proceed.

Working on a Chromebook or a tablet

You are working together with your friend over the IoT project over lunch, but you only have your tablet with you, so you cannot run the code. Wouldn't it be nice to be able to connect to a network with other peers with more powerful machines and execute your code in a docker container on them? Or even have your own cloud hosted latest model Ryzen, 256GB RAM beast join the network as a worker node so you and your colleagues can all run your code from anywhere in the world, working on any kind of device.

Giving a live demo at an IT event

Have you seen people give a live demo of their really cool new library or technology during some IT event? What if they could have everyone present join them in their editor and tinker along. This would make for great workshops and very interactive demos.

Architecture

VS Code extension. Details are unclear at this point.

Milestones

This project can be extended in many ways, so we'll start simple and see how far we can go with the Secondary goals and the Nice-to-haves.

Main goal

- Create a bare-bones VS Code extension
- Create a P2P network, which connects peers (handshaking and basic protocols)
- Allow peers on the network to share files
- Allow peers on the network to share binary data
- Watch the local files for change and push updates to the network
- Allow the host to reconnect after losing connection and reclaim their host position

Secondary goal – Remote execution using Docker

- Create default Dockerfile (template) for launching standalone apps
- Add option in the extension to share the Dockerfile together with the source code
- Have peers announce if they can run docker
- Add option to choose 1 of the docker owning peers to run the code on

Nice to have – Edit permissions

- Allow the host to set edit permissions to the file to Read-only
- Allow the host to give edit permissions to a specific peer
- Allow the host to revoke edit permissions

Nice to have – Forking

- Allow peers to make their own local copy of the code, which they can edit
- Add a mechanism to allow peers to make edit suggestions from the forks
- Add a merging mechanism for forks using automerge

Comparison to similar solutions

The idea borrows from the Live Share extension, which sets up a server at the host and the peers connect to that server. The issue with it is that if the host's network is interrupted, and they are disconnected even for a microsecond, the server dies and disconnects everyone. If the extension was implemented over P2P, the peers would remain connected and the host would rejoin moments later to continue where they left off.
The project also borrows ideas from the Remote - SSH extension. Which allows clients to connect to remote servers over SSH and use the editor only as an API, while the code underneath lives on the remote machine. This allows for remote code execution and "shared memory"-kind of collaboration between developers.