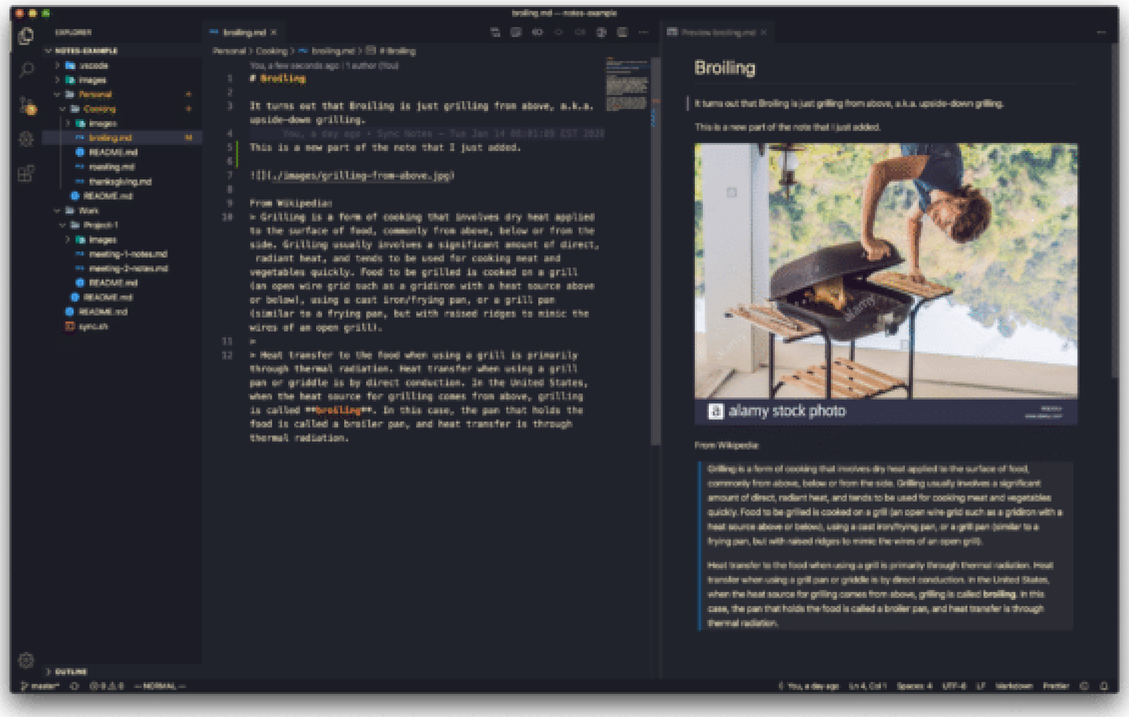
Turning VS Code into a DIY Notes App

spin.atomicobject.com/2020/02/27/text-editor-notes-app

February 27, 2020

February 27, 2020
by: Alex Zurek
11 Comments
I've written several posts, about my search for a great note-taking app with a text editor that uses Markdown. I've bounced around between a handful of different apps, but they all have flaws that make me consider switching to something else. So I've come back to VS Code, and I'm trying to tailor it to suit my needs.



```
# Broiling

It turns out that Broiling is just grilling from above, a.k.a.
upside-down grilling.

This is a new part of the note that I just added.

![](./images/grilling-from-above.jpg)

From Wikipedia:
> Grilling is a form of cooking that involves dry heat applied
to the surface of food, commonly from above, below or from the
side. Grilling usually involves a significant amount of direct,
 radiant heat, and tends to be used for cooking meat and
vegetables quickly. Food to be grilled is cooked on a grill
(an open wire grid such as a gridiron with a heat source above
or below), using a cast iron/frying pan, or a grill pan
(similar to a frying pan, but with raised ridges to mimic the
wires of an open grill).
>
> Heat transfer to the food when using a grill is primarily
through thermal radiation. Heat transfer when using a grill
pan or griddle is by direct conduction. In the United States,
when the heat source for grilling comes from above, grilling
is called **broiling**. In this case, the pan that holds the
food is called a broiler pan, and heat transfer is through
thermal radiation.
```

Note-taking in VS Code with a live-preview open.

VS Code has great built-in support for Markdown files, and there are a handful of other useful extensions to make it even better. I especially like the Markdown All in One extension, which offers auto-formatting and excellent IntelliSense.

But while VS Code is great for editing text, it lacks the organizational features and syncing capabilities that other note-taking apps offer. Or does it?

It turns out, a lot of these features are easy to achieve with VS Code with the help of a few other tools.

Project Setup

VS Code provides a lot of support for managing projects, especially those that use Git for version control. We can make use of these project-level features by creating a project to store our notes and initializing it as a new Git repository.

The "$" in these examples indicates a prompt in the terminal; it is not a part of each command.

Bash

```
$ mkdir ~/notes
$ cd ~/notes
$ git init
$ git add .
$ git commit -m "Initial commit"
```

Now we have a new directory (set up as a local Git repository) where we can store our notes. If we want to sync our notes with other devices or want them backed up, we can configure our preferred Git remote. In this example, I'm going to be using GitHub.

Once we've created a new repository, we can configure it as a remote repository for the `notes` project:

Bash

```
$ git remote add orgin git@github.com:YourUsername/new-notes-repo.git
$ git push --set-upstream origin master
```

We should now have an empty Git repository on our local machine that is synced with GitHub!

Project Structure

With any project, we need a way to structure all of our files. For a typical software project, there might be `src`, `spec`, and `lib` folders that hold different types of files. We can do the same thing with our notes, creating different folders for different types of notes.

Perhaps we want to keep our notes about cooking separate from our notes about work. We can do this easily by creating different folders:

```
.
├── Personal
│   ├── Cooking
│   │   ├── README.md
│   │   ├── broiling.md
│   │   ├── roasting.md
│   │   └── thanksgiving.md
│   └── README.md
├── README.md
└── Work
    ├── Project-1
    │   ├── README.md
    │   ├── meeting-1-notes.md
    │   └── meeting-2-notes.md
    └── README.md
```

You may have noticed the multiple `README.md` files in each directory. When we use GitHub as the git remote, it automatically detects a `README` file in a directory and displays the rendered form of it on the GitHub website. I typically use this as a landing page for the sub-directories of my notes.

Syncing

With the project configured with a Git remote, we can commit and push those changes to the remote repository. The typical workflow for this involves staging all of the changes that we made, adding a meaningful commit message, and pushing those changes to the remote repository:

Bash

```
$ git add .
$ git commit -m "This is my meaningful message"
$ git push
```

This certainly works, but it's a bit cumbersome to write all the time. And the commit message likely won't be as important as it would be for a typical software project.

We can easily write a script to do this for us and configure a VS Code task to execute it when we want to sync our notes. To start, we'll need to create a file called `sync.sh` at the root of our notes project.

Bash

```
#!/bin/bash

echo "Syncing Notes"
git pull

echo ""
echo "Staging changed files"
git add .

echo ""
echo "Committing staged files"
git commit -m "Sync Notes - $(date)"

echo ""
echo "Pushing changes to GitHub"
git push

echo ""
echo "Finished Syncing"
```

We could execute this file from the root of the project, but instead, we'll configure a task in VS Code to run it for us.

Using `Cmd-Shift-P` (`Ctrl-Shift-P` on Windows) to bring up the Command Palette in VS Code, search for `Configure Task` . This will create a new file called `tasks.json` in a new `.vscode` directory where you can define your new task.

Copy/Paste the following code into `tasks.json` :

Json

```
{
    "version": "2.0.0",
    "tasks": [
        {
            "label": "Sync",
            "type": "shell",
            "command": "./sync.sh",
            "problemMatcher": [],
            "presentation": {
                "echo": true,
                "reveal": "silent",
                "focus": false,
                "panel": "shared",
                "showReuseMessage": false,
                "clear": true
            }
        }
    ]
}
```

We should now be able to search for `Run Task` in the Command Palette and execute that task. If we want to trigger this task when pressing a keybinding, we can open the Command Palette and search for `Preferences: Open Keyboard Shortcuts (JSON)` and add the following entry.

Json

```
{
    "key": "ctrl+shift+s",
    "command": "workbench.action.tasks.runTask",
    "args": "Sync"
},
```

This will bind the `Sync` task to “ctrl-shift-s”. (This keybinding is not specific to our notes project.)

Please check out the example notes repository for the full project setup. I hope this helps you set up VS Code (or your preferred text editor) for note-taking with Markdown.

Let me know what you think!