

The Power of Obsidian for Personal Finance Accounting

[Geet Duggal](#) • 9 mins

[Jan 5th](#)

A direct approach to managing your money in plain text

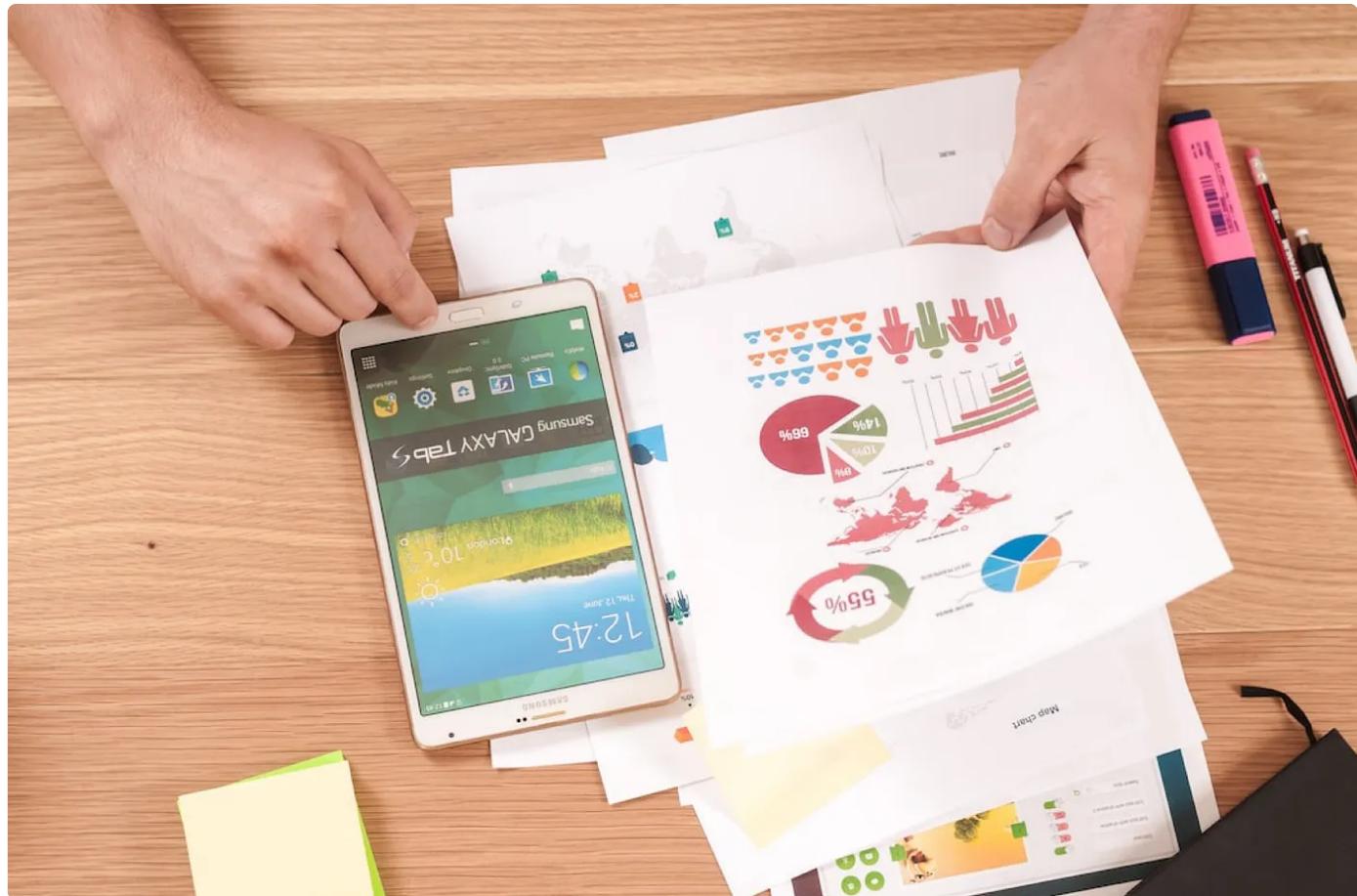


Photo by Firmbee.com on Unsplash

I built a basic personal finance accounting system for myself in Obsidian's mobile app in less than a day over my holiday break. It is already super useful to me. But what excites me the most is the potential for what it could become using a variety of tools in Obsidian.

When I was in high school (twenty-five-ish years ago 😱), I landed an internship at a rapidly growing health information technology company called Cerner. On my first day, I was thrown into typical corporate culture with a whole onboarding program, spending \$ with credit for swag, and ... setting up my financials.

I quickly learned that a personal finance management program called ‘Microsoft Money’ could automatically grab transactions and other information directly from my bank. It sported a much cleaner, slick interface than the various ‘Web 1.0’ sites of yore, and it could combine my financial data from multiple sources into one single view. Most importantly, it provided me an enjoyable way to regularly check on the activity of my spending and saving.

If there’s anything you should regularly check on, it’s probably your finances. A journal of how you earn and spend your money may be as close as you can get to a truthful, auditable record of your activities. It may even be an indicator of what you *really* value vs what you’d *like* to value. This is a lesson I am still learning, and I’m convinced that the right medium to explore my finances can be a big motivator to pay attention to them in the first place.

In the past Microsoft Money was that medium. I eventually abandoned it as it ended up having its own shortcomings (bank account compatibility being the biggest issue I had). I subsequently tried to use Mint.com but ultimately ran into similar issues and instabilities with synchronization of my accounts. Worse, I was in ‘Web 2.0 advertisement hell’ and in an interface that I really didn’t find much more appealing or useful to me than going to individual account websites. (Microsoft Money is sunset

and Mint.com is following suit literally as I write this.)

So I stopped using personal finance management software and, in turn, stopped paying attention to my personal finances. The consequences of this for me were certainly negative. Subscription fees added up without my knowledge. Lack of planning and/or budgeting led to short-term vs. long-term thinking with respect to spending. Crucially, I think I subconsciously beat myself up for not paying attention to my finances. Frankly, I thought it would be boring and potentially not useful to “look at my books” or even bookkeep in the first place.

“And some things that should not have been forgotten were lost. History became legend. Legend became myth. And for two and a half thousand years, the ring passed out of all knowledge.” – Galadriel in The Lord of the Rings: The Fellowship of the Ring

A couple of months ago I decided this negligence needed to end. Somehow I should be motivated to track my finances and have fun in the process. In the last quarter century or so (💡), I’ve learned I thing or two that helped me come up with the following list of requirements for a new personal finance management system.

I need to be able to:

- track spending by category (eg. family childcare, home accessories, home improvement, ...),
- look up specific transactions (eg. drill down any particular category to see what expenses contributed to them),
- visualize an overview of assets, expenses, liabilities via tables and

charts,

- ensure asset accounts have enough money for expected expenses at any time of the month,
- rely on the underlying format to stand the test of time (ideally it is readable and plain text-centric),
- analyze and enter data in a frictionless and mobile-friendly way, and
- share this information in an understandable and customized way for myself, my family, or others as appropriate.

Basics of accounting & brief introduction to Ledger

This section is, at its core, a shameless advertisement to watch this excellent presentation:

Some content could not be imported from the original document.

[View content ↗](#)

I'm a newbie when it comes to accounting, and I really like how this presentation overviews some basics of accounting. The presentation also illustrates how they apply when using a Unix command line interface (CLI) utility for personal finance called 'Ledger'.

At its core, you account or "keep books" to have an accurate understanding of "what you have". Bookkeeping broadly involves keeping track of

what you're owed and what you owe others: how various streams of your income contribute to your expenses and what you can spend money on.

For the purposes of this article, your **Ledger journal** can be thought of as 'your books'. It is a record of all your transactions in a useful and human-friendly plain text format.

A **transaction** is a collection of credit and debits to and from various accounts that occurs at a particular time.

An **account** is an abstract label along with a total amount of something in it.

A simple example of a transaction in Ledger journal format would be:

```
2024-01-01 Amazon (Tetely Tea and Dark Chocolates)
  Expenses:Groceries $15.54
  Liabilities:Credit:VisaChasePrime $-15.54
```

This transaction simply encodes that we took money (debited) from a 'Liabilities' account (in this case with a 'sub account' labeled with the credit card used) and added (credited) it to an 'Expenses' account (in this case with a 'sub account' labeled with the type of expense).

Notice that the sum of credits and debits in the transaction is 0. This practice of book keeping is called **double entry accounting**, and it is useful to get an accurate view of where money goes to and from and to catch errors in accounting.



I use the term ‘abstract’ to describe the account name because when you keep your books you have flexibility to create accounts “from thin air” however you like. For example, you may not have an actual bank account you keep for tracking grocery expenses (here abstractly referred to as ‘Expenses:Groceries’), but it is useful for you to understand how you spend your money.

Generally speaking, in “Ledger land”, there are a 5 broad types of accounts:

- Assets (eg. bank accounts, cash on person, investments)
- Income (eg. paychecks, gifts, interest earned)
- Expenses (eg. groceries, taxes, subscriptions, clothing)
- Liabilities (eg. mortgages, credit cards, car loans)
- Equity (opening balances, general catch-all)

The presentation linked above and other documentation online (including the Ledger documentation) provide more useful detail and examples. In short, if you have a list of transactions like the one shown above for all of your accounts, then you possess the foundation for managing your personal finances in a principled way using the Ledger CLI tool.

The beauty of the Ledger CLI is that it follows the UNIX philosophy of solving one problem well. If you have your transactions listed in the simple and readable format above (one of my criteria above!), this tool will provide you the basic nuts and bolts you need to help better analyze your

finances. For more on keeping a Ledger journal, see the ‘Keeping a Journal’ section of the Ledger documentation. Here are a few excerpts from the documentation from the documentation for motivation:

Ledger is an accounting tool with the moxie to exist. It provides no bells or whistles, and returns the user to the days before user interfaces were even a twinkling in their father’s CRT.

What it does offer is a double-entry accounting journal with all the flexibility and muscle of its modern day cousins, without any of the fat. Think of it as the Bran Muffin of accounting tools.

...

The most important part of accounting is keeping a good journal. If you have a good journal, tools can be written to work whatever mathematical tricks you need to better understand your spending patterns. Without a good journal, no tool, however smart, can help you.

...

Ledger has been designed to make data transactions as simple as possible, by keeping the journal format easy, and also by automatically determining as much information as possible based on the nature of your transactions.

...

The power of Obsidian and Plain Text Accounting

A tool like Ledger falls under the broader category of [plain text account-](#)

ing and its journal format is similar across many tools. Ledger itself satisfies most of my requirements. However, in my exploration there are two key requirements that are left unsatisfied:

- analyze and enter data in a frictionless and *mobile-friendly* way, and
- share this information in an understandable and *customized* way for myself, my family, or others as appropriate.

Ledger's CLI is great, but it's not meant for use on the phone and lacks other visuals that may be nice to have (e.g. charts). In addition, there is specific information on my Ledger journal data that I would like to extract that is better suited to be customized just for me.

For example, I would like to share with my wife, at a glance, the sum of money we should transfer from investment accounts to our checking account based on expected upcoming expenses. A detailed explanation of why should be easily available and auditable as well. There have been numerous times where we both have essentially manually done this (myself related to remodeling expenses). The ability to create this summary in a relatively automated and precise way would be amazingly useful.

At first I thought spreadsheets would satisfy a lot of these concerns, but I started to realize that spreadsheets, as intuitive as they are, can get complicated and the functionality I desire is often more cleanly and directly accessed from my Ledger journal coupled with some **basic code and notes**.

This is where Obsidian starts to flex its muscles. Due to the tool's power and flexibility, I tend to go about things in two broad steps: (1) think

clearly about the problem I'm trying to solve, and (2) piece together the right files and tools in Obsidian to do the thing.

For example, if all I want is a simple summary of all my account & sub-account balances and how they add up (one problem I'm trying to solve), I could directly use the [Obsidian Ledger plugin](#). However, it currently lacks visualization in a mobile-friendly way. I realized I could piece together my Ledger journal file with the [Obsidian dataview plugin](#) to do just that.

The dataview plugin is powerful in the sense that you can execute ‘arbitrary javascript code’ directly in your Obsidian note and see the results in a friendly way. While the plugin has no built-in capabilities to access a Ledger journal, it is fairly straightforward to obtain the summary of account balances in a data pipeline of three steps:

1. Parse ledger file transactions and impute necessary data for double entry accounting
2. Group transactions by account and sum credits and debits
3. Visualize the result in a table in my Obsidian note

For example, a Ledger journal could look like:

```
alias A=Assets  
alias B=Assets:Banking  
alias C=Liabilities:Credit  
alias E=Expenses  
alias I=Income
```

```
2023-12-15 Opening balances  
C:ChasePrime $-200
```

C:AmexPlatinum \$-100
B:BofASavings \$18000
B:BofAChecking \$1200
Equity:OpeningBalances

2023-12-15 Paycheck

B:BofAChecking \$1000
I:Work

2023-12-18 Poirot on Amazon Prime

E:Family:Entertainment \$6
C:ChasePrime

2023-12-19 Common Savings to Checking

B:BofAChecking \$1000
B:BofASavings

2023-12-19 Monthly earnings

B:CapitalOneChecking \$800
I:SideHustleLLC

2023-12-21 Trader Joes

E:Family:Groceries \$50
C:AmexPlatinum

2023-12-25 Apple

E:Family:Subscriptions \$10
C:AmexPlatinum

2023-12-26 Netflix

E:Family:Subscriptions \$15
C:AmexPlatinum

2023-12-27 Home Depot

E:Family:Homebuilding \$45
C:ChasePrime

2023-12-27 Tetley Tea

E:Family:Groceries \$16

C:AmexPlatinum

2023-12-28 Readwise

E:Family:Subscriptions \$9

C:ChasePrime

2023-12-29 Safeway

E:Family:Groceries \$15

C:AmexPlatinum

1. Parse ledger file transactions and impute necessary data



2. Group transactions by account and sum credits, debits



3. Visualize the result in a table in my Obsidian note

2023-12-15 Paycheck
B:BofAChecking \$1000
I:Work

2023-12-18 Poirot on Amazon Prime
E:Family:Entertainment \$6
C:ChasePrime

2023-12-19 Common Savings to Checking
B:BofAChecking \$1000
B:BofASavings



2023-12-15 Paycheck
B:BofAChecking \$1000
I:Work \$-1000

2023-12-18 Poirot on Amazon Prime
E:Family:Entertainment \$6
C:ChasePrime \$-6

2023-12-19 Common Savings to Checking
B:BofAChecking \$1000
B:BofASavings \$-1000



C:ChasePrime sum[\$-200, \$-6]
B:BofASavings sum[\$18000, -\$1000]
B:BofAChecking sum[\$1200, \$1000, \$1000]
I:Work sum[\$-1000]



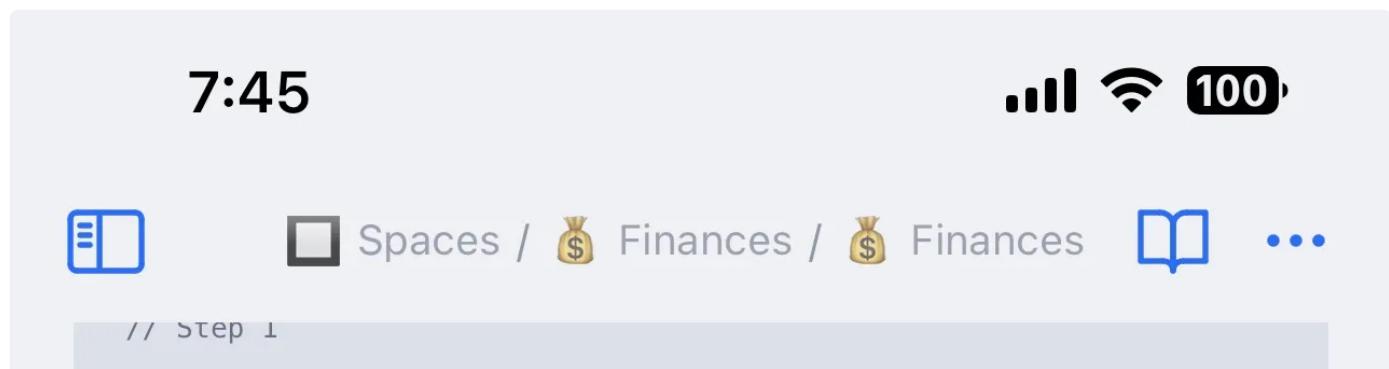
Account (18)	Balance
B:BofAChecking	\$21000
B:BofASavings	\$3200
B:CapitalOneChecking	\$17000
C	\$800
C:AmexPlatinum	\$466
C:ChasePrime	\$206
E	\$166
E:Family	\$166
E:Family:Entertainment	\$6
E:Family:Groceries	\$81
E:Family:Homebuilding	\$45
E:Family:Subscriptions	\$34
Equity	\$18900
Equity:OpeningBalances	\$-18900
I	\$1800
I:SideHustleLLC	\$800
I:Work	\$1000

Sketch of a pipeline for creating an account balance summary in Obsidian

And the final resulting table can look something like:

Account (18)	Balance
B	\$21000
B:BofAChecking	\$3200
B:BofASavings	\$17000
B:CapitalOneChecking	\$800
C	\$-466
C:AmexPlatinum	\$-206
C:ChasePrime	\$-260
E	\$166
E:Family	\$166
E:Family:Entertainment	\$6
E:Family:Groceries	\$81
E:Family:Homebuilding	\$45
E:Family:Subscriptions	\$34
Equity	\$-18900
Equity:OpeningBalances	\$-18900
I	\$-1800
I:SideHustleLLC	\$-800
I:Work	\$-1000

The amount of code to do this is so small it can be almost viewed in one screenshot on my phone:



```

const ledgerFile = await dv.io.load("Spaces/$Finances/example.ledger.md")
const allTransactions = ledgerFile.split(/\n\n/).
  filter((transactionText) => {
    return !(transactionText.startsWith(";") ||
transactionText.startsWith("alias") || transactionText.match(/^\s*$/))
  }).
  map((transactionText) => {
    let transactionLines = transactionText.split("\n")
    let transaction = {}
    transaction.date = transactionLines[0].slice(0,10)
    transaction.description = transactionLines[0].slice(10)
    transaction.entries = transactionLines.slice(1).map((e)=>{
      let eSplit = e.trim().split("$")
      return {account: eSplit[0].trim(), amount:
parseFloat(eSplit[1])}
    })
    let transactionSum =
transaction.entries.map((e)=>e.amount).filter((x)=>!isNaN(x)).reduce((a,b)=>a+b)
    transaction.entries = transaction.entries.map((e)=>{
      if (isNaN(e.amount)) { e.amount = -1*transactionSum}
      return e
    })
    return transaction
  })

// Step 2

let accountsSummary =
allTransactions.map((t)=>t.entries).flat().reduce((acc, t)=>{
  let subAccounts = t.account.split(":")
  let prefixAccount = ""
  subAccounts.forEach((a)=>{
    prefixAccount += a
    if (!acc[prefixAccount]) { acc[prefixAccount] = 0 }
    acc[prefixAccount] += t.amount
    prefixAccount += ":"
  })
  return acc
}, {})

// Step 3

dv.table(
  ["Account", "Balance"],
  accountsSummary
)

```

```
Object.keys(accountsSummary).sort().map((a)=>{
  let balance = accountsSummary[a]
  balance = "$"+Math.round(balance)
  function bold(x) {return `<b>$\{x\}</b>`}
  if(a.split(":").length == 1) {
    a = bold(a)
    balance = bold(balance)
  }
  return [a, balance]
})
```



The code is far from perfect but it does produce the output I was looking for and I didn't have to try very hard.

I also want to emphasize that I am of course not limited to using only the Ledger journal format. I'm just amazed that I could get what I wanted from my Ledger journal directly from my phone. I didn't have to develop a new module. I didn't need to request features for another module/tool. And I got to have fun while doing it.

This is where Obsidian shines. It allows for customization that goes way beyond spreadsheet formulas. For example, I could conceivably bypass Ledger all together and use the basic features of the dataview plugin and store entries in files representing transactions, or use the [Advanced Tables](#) and/or [CalcCraft](#) plugins to store my data. Or, better yet, use these plugins to *compliment* and *supplement* analysis of the data in my Ledger journal.

PTPL o85: CalcCraft is the Obsidian Tables Plugin I

Was Looking For!

Did you know there's a plugin for doing simple calculations in Markdown tables in Obsidian? And that there's a serious...

www.blog.plaintextpaperless.com

The possibilities seem so endless that it is almost inevitable that I have to start from the basics. So I repeat for myself:

1. think clearly about the problem I'm trying to solve, and
2. piece together the right files and tools in Obsidian to do the thing.

Perhaps I'll write another article with snippets on how I solve other specific problems for me later, but that not why I'm inspired to be writing this right now. The reason I posted the code as a screenshot above is to emphasize that the code itself is not the point.

The point is that I feel like with Obsidian I've been empowered in a special way to solve the problems that matter to me related to personal finance management. And I get to have fun in the process. The screenshot is therefore intended for inspiration and a reminder of the power of Obsidian.

On a similar note, my hope in writing this article is that I will find some camaraderie with others in the plain text accounting community. I am excited to hear about how you solve your personal finance problems with Obsidian, and if there are interesting alternative ways to think about personal finance management. Until then, I will excuse myself from further writing and make sure my books are up to date.

