Lemonade Stand Tutorial: Transactions

By John McKee

Hello again everyone! In our last tutorial (Getting Started) we learned how to set up the basic structure in Subledger. In this tutorial we'll cover how to get information into the system with journal entries, and we'll cover some common journal entry patterns. Just like in the first tutorial we will build on the lemonade stand idea. You can see a working version of the lemonade stand using the web and iOS apps. So, lets get started!

Create an Opening Balance

For our first journal entry lets just post an opening balance to cash:

```
{
{
}
}
,
{
}
}
]
}
```

and the response:


```
{
}
}
```


Effective_at is a string in ISO 8601 format. Subledger will store the effective_at down to the millisecond and can handle thousands of journal entries per second. References are a URL, just like in previous blogs. The overall journal entry has a reference and each line has a reference too. Since we did not specify a reference for each line Subledger automatically sets the line reference to the overall journal entry reference, in this case "http://receipt.com." Journal entries must have at least one line, but normally have two or more lines, however, in all cases debits must equal credits. If you have a journal entry with one line for documentation purposes the value of that line must be zero.
This journal entry deals with an asset account and an equity account. You debit an asset account to increase its balance, and you credit an equity account to increase its balance, so things look pretty good.

Stock the Shelves

Ok, so now we have an opening balance posted of $100 to both our cash and shareholder equity accounts. Next we might want to use some of that cash to buy some inventory so we can make some lemonade. Let's take $60 and stock the shelves. What might that journal entry look like? Well, if we think about it, we are really just moving money from the cash account to the inventory accounts. We aren't adding or deleting anything to the system overall. So, the journal entry would look like this:


```
{
```

```
[redacted]
[redacted]
[redacted]
[redacted]
{
[redacted]
[redacted]
[redacted]
[redacted]
}
},
{
[redacted]
[redacted]
[redacted]
[redacted]
}
},
{
[redacted]
[redacted]
[redacted]
[redacted]
}
},
{
[redacted]
[redacted]
[redacted]
[redacted]
}
}
]
}
[redacted]
```


and the response:


```
{
[redacted]
```

[redacted]
[redacted]
[redacted]
[redacted]
[redacted]
[redacted]
}
}

In this case all the accounts are asset accounts so keeping track of debits and credits is easy, credits reduce the account value and debits increase the account value.

Lemonade sold Journal Entry

Ok, so lets get down to the heart of the matter. Lets create a journal entry to capture the sale of a lemonade. We will say we are going to sell a single lemonade for $1.25, and that sales tax in our area is 7% which equates to $0.09. Being good lemonade stand operators we know that it costs us $0.25 in lemons, $0.15 in sugar, and $0.10 for the cup, so in total it costs us $0.50 to make one lemonade.

[redacted]
[redacted]
[redacted]
[redacted]
{
[redacted]
[redacted]
[redacted]
[redacted]
{
[redacted]
[redacted]
[redacted]
[redacted]
}
}
{
[redacted]
[redacted]
[redacted]

[redacted]
}
}
,
{
[redacted]
}
}
,
{
[redacted]
}
}
,
{
[redacted]
}
}
,
{
[redacted]
}
}
,
{
[redacted]
}
}
,
{

and the response:

Ok, so lets break down this journal entry. I think of this journal entry in two separate parts, one part that tracks money received and another part that tracks profit. Lets start with the track money received part which correlate to the first three lines in the lines array. Those lines effect cash, sales, and sales tax. We sold one lemonade, and our customer gave us $.134 which is reflected in line 1 which increases cash $1.34. Line two increases the sales account $1.25 (total cost - tax). And finally, line three add .09 to the sales tax account. At this point we can check, 1.34 debits = 1.34 credits. Great! So far, so good.

Now lets look at the second part, tracking profit. The second part is really moving money from the inventory accounts to the cost of goods sold accounts. We decrease the

Inventory Lemons account 0.25 and increase the Cost of Lemons account 0.25, decrease the Inventory Sugar account 0.15 and increase the Cost of Sugar account, and finally, decrease the Inventory Cup account and increase the Cost of Cups account. Now lets double check ourselves, 0.50 debits = 0.50 credits. Life is good!

So far we have learned how to get identities, orgs, books, and accounts set up. We have seen the importance of the Chart of Accounts which spells out what we want to track, and we have seen how to correctly capture financial events, like the sale of lemonade, in the system. My next blog post is going to close the basic skills loop and show you how to get information out of Subledger using reports.