

BAT - Quinterac Assignment #4  
Tamir Arnesty, Ben Durston, and Aaron Safer-Rosenthal

Overall Structure:

The backend of our project uses the Accounts class from the frontend, to allow us to create, update, and delete accounts that will be used for the next day's frontend session. When the backend is run, all of the accounts that exist in the old master accounts file are made into a list of Account objects. Then all of the transaction summary files from the day are merged into one. From there, the merged transaction summary file gets read and the appropriate modifications to the Account objects are performed. Once all of the transactions have occurred, the master accounts file gets updated with the modified Account objects and a new copy of the valid accounts list file gets created for the next day's frontend session.

BackEnd.py - Class: BackEnd

Method Name	Purpose	Arguments	Return
<code>__init__</code>	Declares class attributes, such as <code>self.accounts</code> which will be used in other methods.	The class instance, the old <code>masterAccountsFile</code> , and the transaction summary file.	New master accounts file and new valid accounts list file.
<code>loadAccounts</code>	Get all of the accounts that are stored in the <code>masterAccountsFile</code> , and store them in a list. Each account is stored as an object, so it's account number, name, balance and other attributes can be accessed.	The class instance.	A list of account objects.
<code>update</code>	Update the daily transaction summary files into one merged transaction summary file. The method goes through each line in the transaction summary file, and performs the necessary actions	The class instance, and the transaction summary file that is being used to update the necessary accounts.	None. Calls <code>writeMaster</code> once finished.

	<p>according to the transaction code. Once all lines are checked. The list of account objects stored in the self.accounts list is sorted from highest to lowest account number.</p>		
writeMaster	<p>This method goes through each account object stored in the self.accounts list and updates the masterAccountsFile.</p>	The class instance.	<p>None. Calls generateNewAccountsFile once finished.</p>
formatLine	<p>This method is called before each account is written or updated into the masterAccountsFile. The purpose of the method is to return a string in the proper masterAccountsFile format.</p>	<p>accountNumber, balance, and accountName. All from the account object.</p>	<p>A string in the masterAccountsFile format of account number, balance, and account name.</p>
generateNewAccountsFile	<p>The method takes the accounts that were previously updated by the update function, and adds account numbers to the valid accounts list file. This is the validAccountListFile that will be used for the next day's front-end session. Once finished, the mergedTransactionSummaryFile gets cleared so no</p>	The class instance.	<p>None.</p>

	transactions occur twice.		
updateAccount	This method updates the accounts balance from withdrawal, deposit, and transfer transactions.	The class instance, the account number, and the amount.	None.
createAccount	This method creates an account with the information from the create account transaction.	The class instance, the account number, and the name of the account.	None.
deleteAccount	This method deletes an account with the information from the delete account transaction.	The class instance, the account number, and the name of the account.	None.