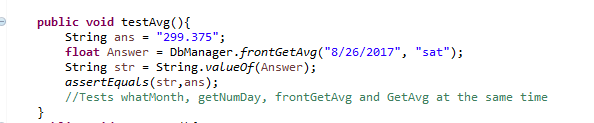
**Backend Junit Test Report**

**testAvg():**

testAvg will input a specific date and day of the week, then AssertEquals the returned value with the expected value. This test will test multiple methods within the frontGetAvg method.

Methods being tested:

* frontGetAvg
* GetAvg
* whatMonth
* getNumDay

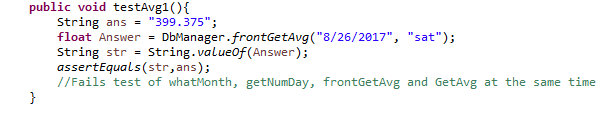




**testAvg1():**

testAvg1 is designed to cause the test to fail. It works the exact same way as testAvg, but AssertEquals the returned value with the wrong expected value to ensure that the test does not return true regardless of the input.

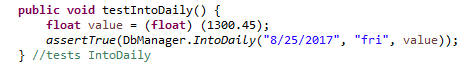
testAvg1 tests the exact same methods as testAvg.





**testIntoDaily():**

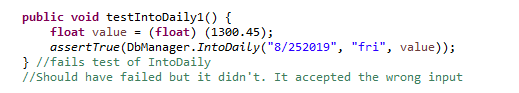
testIntoDaily will test the ability of the program to input new data into the database. It is inputting a date, day of the week, and a float gross sales value.





**testIntoDaily1():**

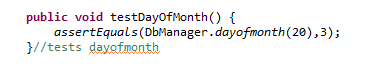
testIntoDaily1 is taking an invalid input and entering it into the database. This test is designed to fail, however it does not and IntoDaily is accepting the invalid date and entering it into the database. This highlights an issue that needs to be fixed.





**testDayOfMonth():**

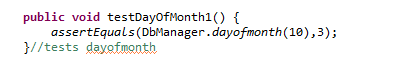
testDayOfMonth takes an input of the day of the month and tests the output of DayOfMonth against the expected value. DayOfMonth tests the 20th of any given month to return which iteration of the day of the week of a given month it is. The 20th should be the third time that specific day of the week has been seen that month.





**testDayOfMonth1():**

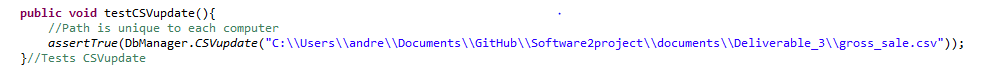
testDayOfMonth is designed to fail by testing against the wrong expected value. It fails as expected, validating the DayOfMonth works as expected.





**testCSVupdate():**

testCSVupdate tests that the CSV file is properly updated given a user specific file path. The test works as expected.





**testCSVupdate1():**

testCSVupdate1 attempts to run CSVupdate with an improper input. This test fails as expected.

**testMonthUpdate():**

testMonthUpdate tests the MonthUpdate method to update the table for that entire month. It inserts a set date into MonthUpdate and AssertTrue the returned value.





**testMonthUpdate1():**

testMonthUpdate1 works the same as testMonthUpdate but inserts an invalid input. This test predictably fails.



