***Algorithmic Trading System Regression Test***

Algorithmic Trading Software

2023-12-06

Version 1.1

Authors: Jacob Rawlings,Jake Fischer**,** Alan Abdollahzadeh, Devon Volberg, William Blake, Benjamin Carrier, Vanessa Dubouzet, Dakota Flath, Parker Green, Yuan Hu, Jacob Labelle, Isaac Lengacher-Bergeron, Dominic Presch, Dylan Soares, Jaeden Soukoroff

**Revision Sheet**

| Revision | Date | Brief Summary of Changes |
| --- | --- | --- |
| Version 1.0 | 2023-12-04 | Baseline document draft |
| Version 1.1 | 2023-12-06 | Updated formatting |
|  |  |  |

[**1. INTRODUCTION 1**](#_heading=h.gjdgxs)

[**2. REQUIREMENTS 1**](#_heading=h.qkxngo1dzha2)

[**3. STRUCTURE 1**](#_heading=h.rv52yxolckml)

[**4. FUNCTIONALITY 2**](#_heading=h.6b4vu5foot5)

# INTRODUCTION

The regression test is designed to provide an automated library of unit tests for each script that can be easily run after changes or bug fixes to the insertion scripts to ensure that new changes haven’t introduced unintended side effects.

# REQUIREMENTS

Since output data varies, a simple sample of data was hardcoded into each test to ensure consistency and to make the code more portable. The scripts require access to *connect.py*, as well as a *credentials.py* file, both properly configured for the intended environment and database. Ensure *insertion\_regression\_test.sh* has execute permissions via running *sudo chmod +x* on it.

# STRUCTURE

The current tests consist of the Python source files *realtime\_stock\_insert\_UnitTest.py* and *company\_statements\_insert\_UnitTest.py*. To automate the tests a bash shell script, *insertion\_regression\_test.sh*, runs both scripts sequentially and echoes the results to the terminal.

# FUNCTIONALITY

***insertion\_regression\_test.sh***

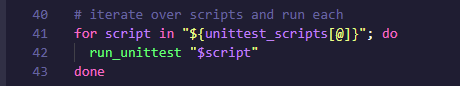
By default, it runs in ‘silent’ mode, i.e. the Python output is redirected to */dev/null*. For full output, include the flag *-v* or *--verbose*.



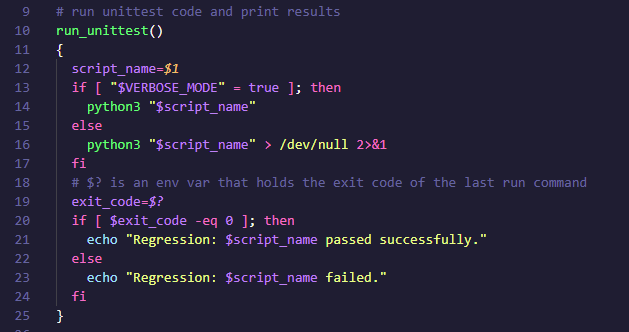
The script has been written to be easily expanded with tests for new scripts in the future, simply add them to the array of file paths:



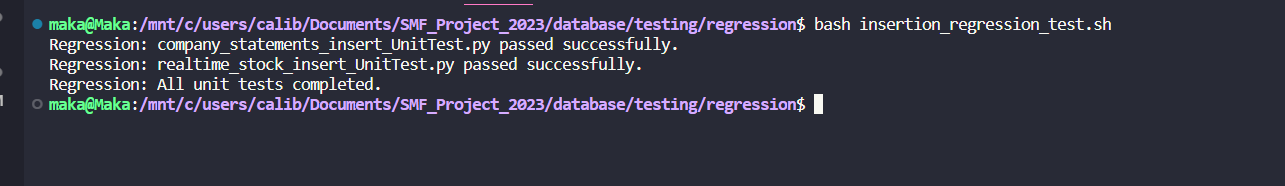
The array of paths is iterated over, passed to the *run\_unittest()* function sequentially:

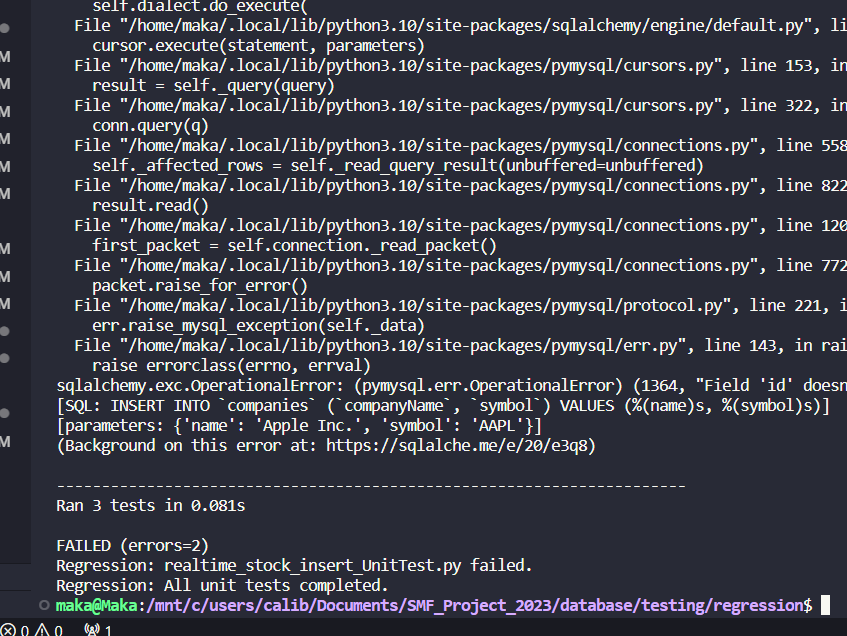


The *run\_unittest()* function starts the Python script after deciding whether output should be silenced, and checks the error code returned for the script, printing success or failure:



Output when all tests pass (silent mode):



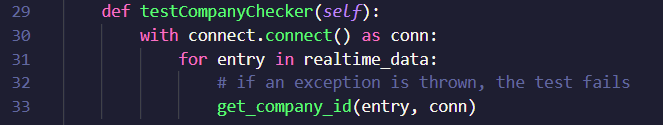
Output on a fail (verbose mode):

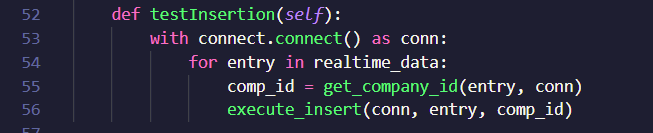
***realtime\_stock\_insert\_UnitTest.py***

An entry of typical output data is provided for testing, but may be changed if needed:

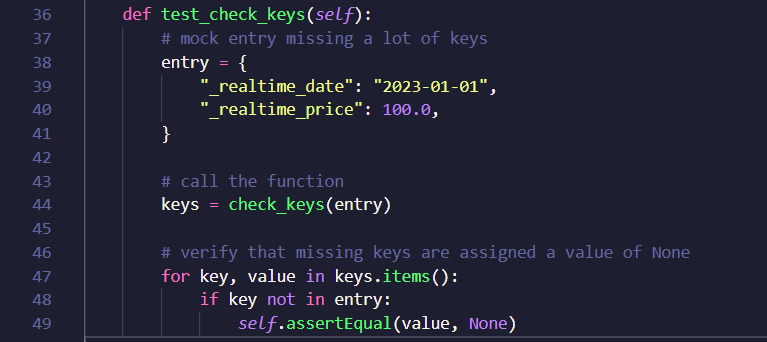


A selection of functions are tested. *testCompanyChecker()* and *testInsertion()* test for any query exceptions:





*test\_check\_keys()* tests the *check\_keys(entry)* function to ensure all missing keys have been assigned a value of *None*.



***company\_statements\_insert\_UnitTest.py***

Tests for *company\_statements\_insert\_UnitTest.py* possess the same functionality as those of *realtime\_stock\_insert\_UnitTest.py*.