

CS308: Testing

Group 27: Ross Lyons (201724258), Syeda Ayela Gilani (201810821), Karl Rivett (201727785), Ismael Ahmed (201707978), Koni Watson (201710203)

createFolioTest():

PASSED

There are five assertions in this method. Firstly, two FolioInterface objects are created.

- The first assertion checks the size of the list of folios is equal to 2.
- The next four check the Folio name and ID (in the list of Folios in FolioHolder) match the created Folios.

This test ensures Folios are created and stored correctly.

deleteFolioTest():

PASSED

A folio is created and then deleted.

- The only assertions in this test, check that **true** is returned when calling deleteFolio(folioID) and that **null** is returned when trying to get a folio – using its folio ID – that has been deleted.

This test showing that a folio can be deleted from the system as a whole.

addHoldingsToFolioTest():

PASSED

To begin this test, a folio is created.

- The first two assertions check if **true** is returned when the addHolding method is called with correct parameters.
- The third assertion checks if there are two Holdings in the array list of Holdings in Folio.
- The last assertions check the list of Holdings contain the Holdings initially added, shown by getStocks() returning **true**.

This highlighting, that when adding a holding its information is correctly added and stored.

stockHoldingValueTest():

PASSED

There is only one assertion in the following test.

- This assertion is repeated 10 times with random share amounts.

The purpose of this is to ensure that the method getValue() in the Holding class returns the correct amount.

getTotalTest():

PASSED

In this test a Folio is created and three holdings are added with a random number of share. The total value of the Folio is then manually calculated.

- The assertion is repeated 10 times and compares the getTotalHolding() method in Folio class with the manual calculation.

This test that the total value of a Folio is being calculated correctly.

saveAndLoadTest():

PASSED

Initially a Folio is created with one holding and saved to disk.

- The first assertion checks if loadFolio() returns true given the saved Folios ID.

This shows the Folio was found and added to the list of Folios in FolioHolder.

- The following assertions compare details of the Folio(Name, ID, Total Holding) and the Stock within the Folio(Ticker ID, Price).

This proves that the details of the Folio were loaded in correctly.

testFolioID():

PASSED

A Folio is created.

- The assertion checks if the length of the ID is 4.

This proves each Folio is given a four-digit ID.

refreshStockPrices():

PASSED

A folio is loaded and the size of the list of prices for a Stock is saved. The refreshPrices() method is then called.

- The assertion checks that the size of the list of prices is one bigger than before the function call.

This shows that a new price has been added to the list of prices, therefore ensuring the refresh works.

testHighsAndLows():

PASSED

A Stock is created, and numerous prices are set, the highest being 200, and the lowest being -105.

- The two assertions check that getHigh() and getLow() return the respective prices.

This shows that the correct values for the lowest and highest price of a stock is reached and returned.

isPriceUpOrDownTest():

PASSED

Stock is created with a price of -£1 (Stock doesn't exist, there is a check in project for this). The price is set at 8.

- The assertion checks that isPriceUpOrDown() returns 9. The price is then set to -5,
- The next assertion checks if isPriceUpOrDown() returns -4 (-1 to -5 is a decrease of 4).

This shows the correct change in price is calculated.

profitMadeTest():

PASSED

A Stock is made with price -£1. The price is set to £101, then £100.

- The following assertion checks if currentProfitOnSingleUnit() returns 101. The price is then set at -£45, then -£41,
- The next assertion checks the same method returns -40.

This shows the profit from the initially purchased price of a Stock is calculated correctly. The difference between this and the previous test is, this checks against the initial price, where the previous test attempts to check in the last 24 hours before checking against the second most recent update.

increaseSharesTest():

PASSED

A Folio is created with a Holding of 30 shares. 25 more shares are purchased.

- The assertions check if the number of shares and the total value of the Holding are updated.

This shows that the full Holding is updated following a purchase of more shares.

decreaseSharesTest():

PASSED

A Folio is created with a Holding with 10 shares. 5 shares are then sold.

- The assertions check if the getNoOfShares() method for the Holding returns 5, and the getValue() on the holding returns the same as the price of one share * 5.

This shows the full Holding details are updated following the selling of a share.

tickerNotFoundTest():

PASSED

A Folio is created, and a Holding is added where a ticker is not found.

- The first assertion checks the getStock() method returns **null** with the fake ticker as a parameter.
- The second assertion checks that addHolding returns **false** when attempting to add the fake Stock holding.

This ensures that only holdings that actually exist in the server are allowed to be added to the folio.

folioNotFoundTest():

PASSED

This test is simply made up of one assertion.

- This assertion checks that loadFolio() returns **null** if no Folio with the entered ID exists on disk.

This ensures the correct working of the loadFolio function by showing that only folios that had been saved initially are loaded back in.

openFolioWithSameNameTest():

PASSED

Creates three Folios with the same name. As you need a Name(1), Name(2), Name(3) etc, Name and Name(1) are considered the same name.

- This assertion checks the function to see if Name(1) and Name(2) are considered the same name, which would lead to a new Folio with its name "Name" being named Name(3).

This ensures that Folios cannot be created (if both open) with the same name. If a Folio called "PF" was open and the user tried to create a new one with the same name, the second would be named PF(1), for duplicate 1, and the next PF(2) and so on.

addHoldingThatExistsTest():

PASSED

A Folio is created and one Holding is added to it. A second Holding is added to it but with the same ticker symbol.

- Instead of creating a new HoldingInterface, the assertion ensures the existing Holding is updated with the attempted shares purchase, and the new shares entered are added on top of the existing Holding.

This is to ensure that a unique Ticker symbol is maintained for the Holdings and data with duplicate Ticker symbols is considered to be shared and thus, added or updated accordingly.