

Brian Abbott
Colin McDermott

One major issue we had was being unable to merge our gradle branch into main.

Why 1: Gradle wouldn't merge because the code wasn't compiling and the tests were not passing.

Why 2: The code wasn't compiling because of various simple bugs, the tests were not passing because of Mockito issues

Why 3: Mockito was causing the tests to fail because it was implemented incorrectly in some of the tests, specifically the when() component.

Why 4: Mockito when() was performing this action on an instantiated java object, instead of a mocked object.

Why 5: This issue came from my misunderstanding of Mockito when(), upon reviewing the documentation I was able to come to the conclusion that it must be performed on a mocked object, upon fixing that the tests passed and gradle was able to merge.

Example:

Buggy code:

```
public class TestDataToFromComputeAPI {
    private DataToFromComputeAPI testAPI;

    @Test
    public void testWrite() throws Exception {
        SourceType mockSource = Mockito.mock(SourceType.class);
        ComputeResults mockResult = Mockito.mock(ComputeResults.class);

        when(testAPI.writeData(any(SourceType.class),
any(ComputeResults.class))).thenReturn("Key made");

        testAPI.writeData(mockSource, mockResult);
    }

    @Test
    public void testRead() throws Exception {
        SourceType mockSource = Mockito.mock(SourceType.class);

        when(testAPI.readData(any(SourceType.class))).thenReturn("Data read");

        testAPI.readData(mockSource);
    }
}
```

Fix:

```
public class TestDataToFromComputeAPI {  
    private DataToFromComputeAPI testAPI = mock(DataToFromComputeAPI.class);  
  
    @Test  
    public void testWrite() throws Exception {  
        List<Integer> mockResult = new ArrayList<Integer>();  
        mockResult.add(42);  
  
        when(testAPI.writeData(mockResult)).thenReturn(0);  
  
        int value = testAPI.writeData(mockResult);  
    }  
  
    @Test  
    public void testRead() throws Exception {  
        List<Integer> fakeList = new ArrayList<Integer>();  
        fakeList.add(42);  
  
        when(testAPI.readData(7)).thenReturn(fakeList);  
  
        List<Integer> result = testAPI.readData(1);  
    }  
}
```