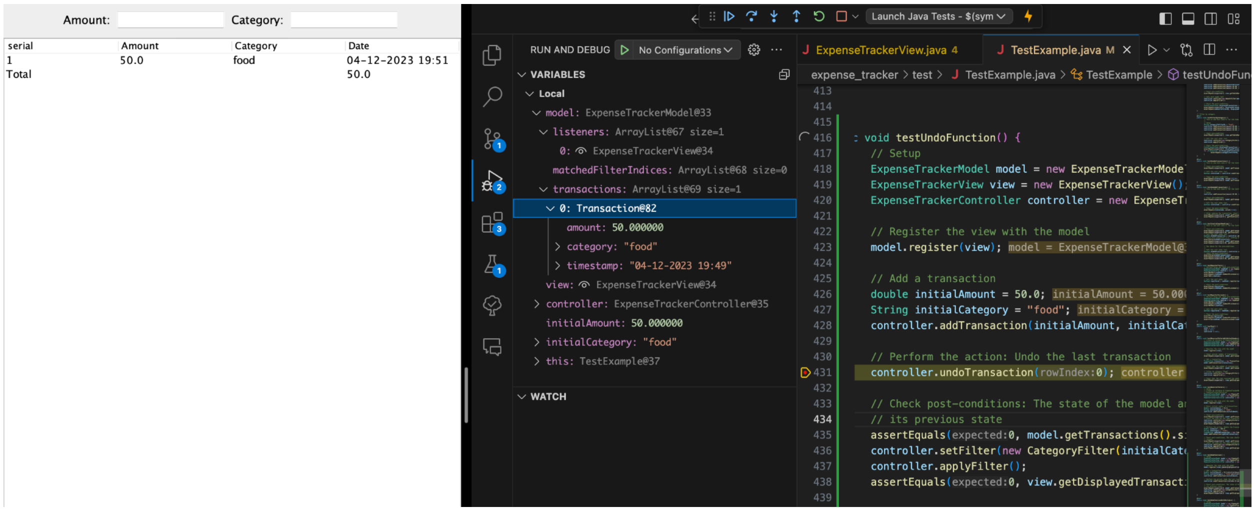
Debugger



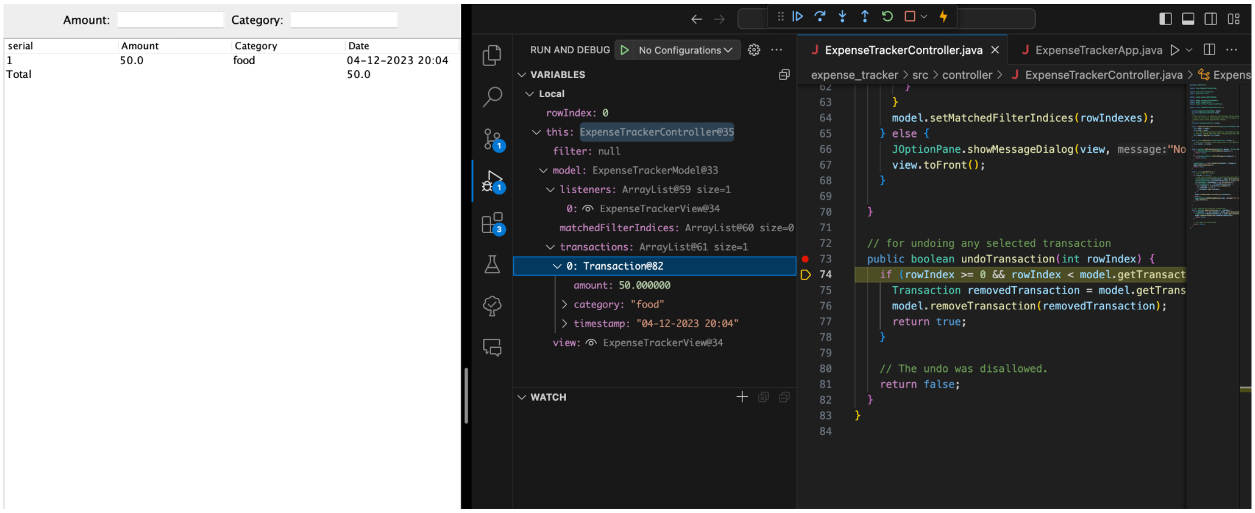
Amount: Category:

serial	Amount	Category	Date
1	50.0	food	04-12-2023 19:51
Total			50.0

RUN AND DEBUG
No Configurations
Launch Java Tests - $(sym
ExpenseTrackerView.java 4
TestExample.java M
expense_tracker > test > TestExample.java > TestExample > testUndoFun

```
VARIABLES
  Local
    model: ExpenseTrackerModel@33
      listeners: ArrayList@67 size=1
        0: ExpenseTrackerView@34
      matchedFilterIndices: ArrayList@68 size=0
      transactions: ArrayList@69 size=1
        0: Transaction@82
          amount: 50.000000
          category: "food"
          timestamp: "04-12-2023 19:49"
    view: ExpenseTrackerView@34
    controller: ExpenseTrackerController@35
    initialAmount: 50.000000
    initialCategory: "food"
    this: TestExample@37
WATCH
```

```
: void testUndoFunction() {
    // Setup
    ExpenseTrackerModel model = new ExpenseTrackerMode
    ExpenseTrackerView view = new ExpenseTrackerView();
    ExpenseTrackerController controller = new ExpenseTr

    // Register the view with the model
    model.register(view); model = ExpenseTrackerModel@

    // Add a transaction
    double initialAmount = 50.0; initialAmount = 50.000
    String initialCategory = "food"; initialCategory =
    controller.addTransaction(initialAmount, initialCat

    // Perform the action: Undo the last transaction
    controller.undoTransaction(rowIndex:0); controller

    // Check post-conditions: The state of the model ar
    // its previous state
    assertEquals(expected:0, model.getTransactions().si
    controller.setFilter(new CategoryFilter(initialCate
    controller.applyFilter();
    assertEquals(expected:0, view.getDisplayedTransact
```



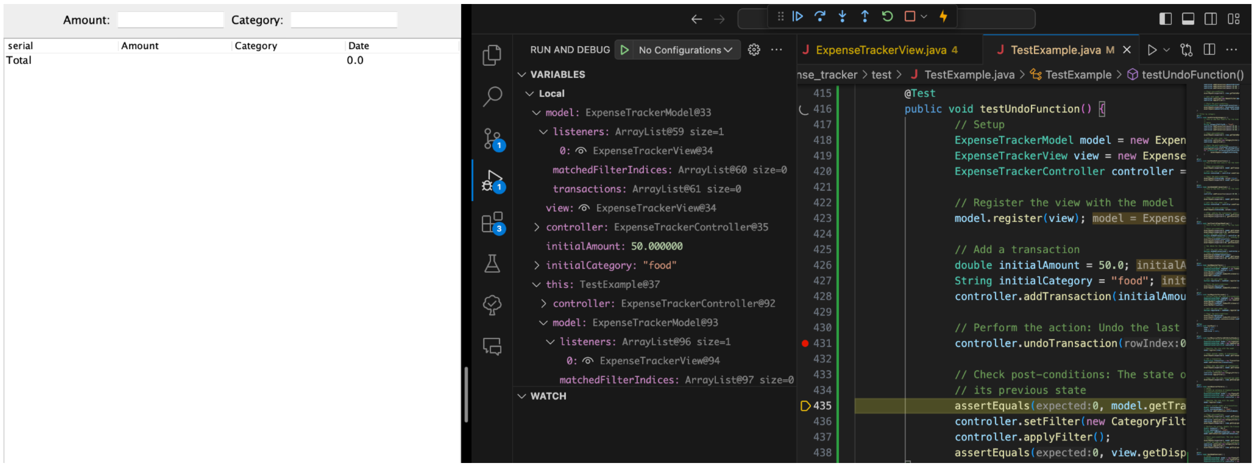
Amount: Category:

serial	Amount	Category	Date
1	50.0	food	04-12-2023 20:04
Total			50.0

RUN AND DEBUG
No Configurations
ExpenseTrackerController.java
ExpenseTrackerApp.java
expense_tracker > src > controller > ExpenseTrackerController.java > Expens

```
VARIABLES
  Local
    rowIndex: 0
    this: ExpenseTrackerController@35
      filter: null
      model: ExpenseTrackerModel@33
        listeners: ArrayList@59 size=1
          0: ExpenseTrackerView@34
        matchedFilterIndices: ArrayList@60 size=0
        transactions: ArrayList@61 size=1
          0: Transaction@82
            amount: 50.000000
            category: "food"
            timestamp: "04-12-2023 20:04"
      view: ExpenseTrackerView@34
WATCH
```

```
          }
        }
        model.setMatchedFilterIndices(rowIndexes);
      } else {
        JOptionPane.showMessageDialog(view, message:"No
        view.toFront();
      }

    }

    // for undoing any selected transaction
    public boolean undoTransaction(int rowIndex) {
      if (rowIndex >= 0 && rowIndex < model.getTransact
        Transaction removedTransaction = model.getTrans
        model.removeTransaction(removedTransaction);
        return true;
      }

      // The undo was disallowed.
      return false;
    }
}
```



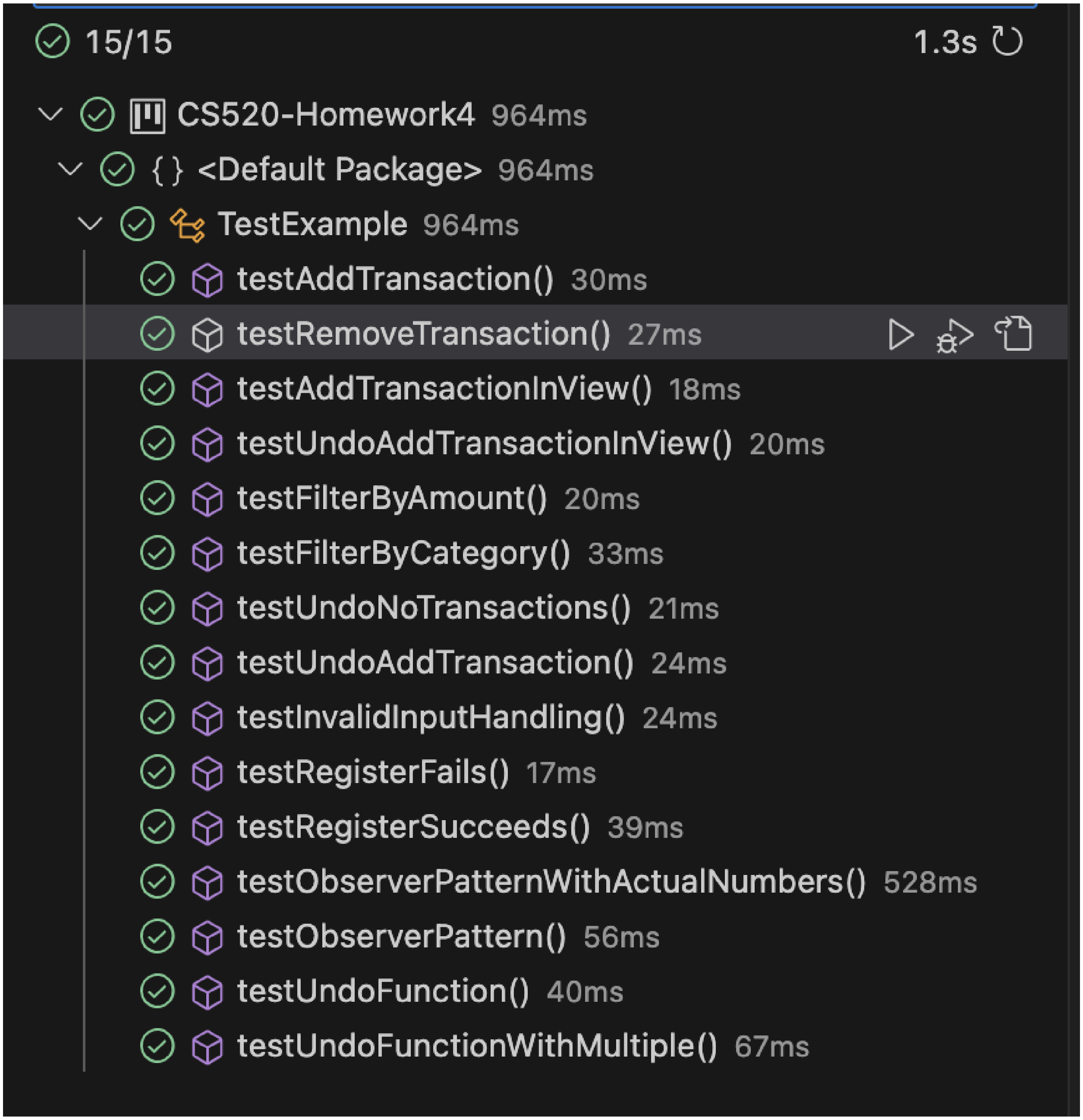
Amount: Category:

serial	Amount	Category	Date
Total			0.0

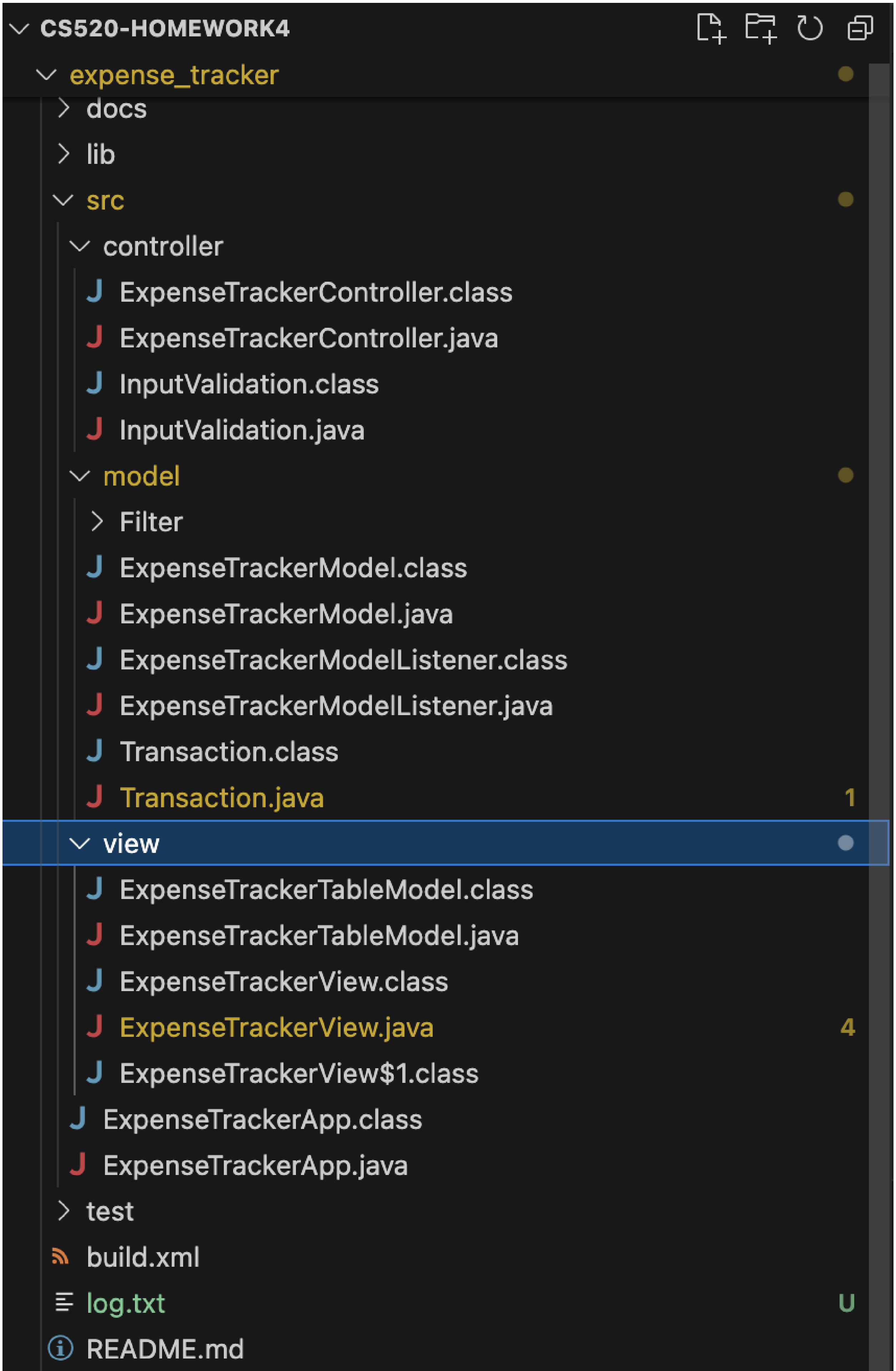
RUN AND DEBUG
No Configurations
ExpenseTrackerView.java 4
TestExample.java M
nse_tracker > test > TestExample.java > TestExample > testUndoFunction()

```
VARIABLES
  Local
    model: ExpenseTrackerModel@33
      listeners: ArrayList@59 size=1
        0: ExpenseTrackerView@34
      matchedFilterIndices: ArrayList@60 size=0
      transactions: ArrayList@61 size=0
    view: ExpenseTrackerView@34
    controller: ExpenseTrackerController@35
    initialAmount: 50.000000
    initialCategory: "food"
    this: TestExample@37
      controller: ExpenseTrackerController@92
      model: ExpenseTrackerModel@93
        listeners: ArrayList@96 size=1
          0: ExpenseTrackerView@94
        matchedFilterIndices: ArrayList@97 size=0
WATCH
```

```
    @Test
    public void testUndoFunction() {
        // Setup
        ExpenseTrackerModel model = new Expen
        ExpenseTrackerView view = new Expense
        ExpenseTrackerController controller =

        // Register the view with the model
        model.register(view); model = Expense

        // Add a transaction
        double initialAmount = 50.0; initialA
        String initialCategory = "food"; init
        controller.addTransaction(initialAmou

        // Perform the action: Undo the last
        controller.undoTransaction(rowIndex:0

        // Check post-conditions: The state o
        // its previous state
        assertEquals(expected:0, model.getTra
        controller.setFilter(new CategoryFilt
        controller.applyFilter();
        assertEquals(expected:0, view.getDisp
```



```
15/15                                          1.3s
CS520-Homework4 964ms
  { } <Default Package> 964ms
    TestExample 964ms
      testAddTransaction() 30ms
      testRemoveTransaction() 27ms
      testAddTransactionInView() 18ms
      testUndoAddTransactionInView() 20ms
      testFilterByAmount() 20ms
      testFilterByCategory() 33ms
      testUndoNoTransactions() 21ms
      testUndoAddTransaction() 24ms
      testInvalidInputHandling() 24ms
      testRegisterFails() 17ms
      testRegisterSucceeds() 39ms
      testObserverPatternWithActualNumbers() 528ms
      testObserverPattern() 56ms
      testUndoFunction() 40ms
      testUndoFunctionWithMultiple() 67ms
```



```
∨ CS520-HOMEWORK4
  ∨ expense_tracker                              ●
    > docs
    > lib
    ∨ src                                        ●
      ∨ controller
          ExpenseTrackerController.class
          ExpenseTrackerController.java
          InputValidation.class
          InputValidation.java
      ∨ model                                    ●
        > Filter
          ExpenseTrackerModel.class
          ExpenseTrackerModel.java
          ExpenseTrackerModelListener.class
          ExpenseTrackerModelListener.java
          Transaction.class
          Transaction.java                       1
      ∨ view                                     ●
          ExpenseTrackerTableModel.class
          ExpenseTrackerTableModel.java
          ExpenseTrackerView.class
          ExpenseTrackerView.java                4
          ExpenseTrackerView$1.class
        ExpenseTrackerApp.class
        ExpenseTrackerApp.java
    > test
      build.xml
      log.txt                                    U
      README.md
```