

Software Quality Lab Deliverable 2

John Howe - 100785128

GITHUB: [johnh-otu/sofe3980Ulab2: software quality lab 2 \(github.com\)](https://github.com/johnh-otu/sofe3980Ulab2:software%20quality%20lab%202)

VIDEO:  2024-02-16 22-15-03.mp4

BinaryAPIController.java Tests (BinaryAPIControllerTest.java)

Test Function	Target Function	Description
blankParams()	addJSON()	Provide the API with two blank ("") parameters. The Binary class should automatically convert these parameters to 0 when calculating the result, but the operands will be saved in the json as blank.
stringParams()	addJSON()	Provide the API with two letter strings. The Binary class should automatically convert these parameters to 0 when calculating the result, and the operands in the json will also be saved as 0.
negativeParams()	addJSON()	Provide the API with two negative numbers. Since the operands include the "-" sign, they will be treated as character/letter strings and converted to 0.
multiply()	multiplyString()	Provides the API with two binary numbers and tests if the result is the product of the two.
multiply2()	multiplyJSON()	Provides the API with two binary numbers, then tests if the result is the product of two and if the original parameters, including the operator, are saved in the JSON.
or()	orString()	Provides the API with two binary numbers and tests if the result is the bitwise OR of the two.
or2()	orJSON()	Provides the API with two binary numbers, then tests if the result is the bitwise OR of two and if the original parameters, including the operator, are saved in the JSON.
and()	andString()	Provides the API with two binary numbers and tests if the result is the bitwise AND of the two.
and2()	andJSON()	Provides the API with two binary numbers, then tests if the result is the bitwise AND of two and if the original parameters, including the operator, are saved in the JSON.

[INFO] Tests run: 11, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 3.037 s - in com.ontariotechu.sofe3980U.BinaryAPIControllerTest

BinaryController.java Tests (BinaryControllerTest.java)

Test Function	Target Function	Description
getBlankParameter()	getCalculator()	Perform a GET request with a blank "operand1" parameter. The operand1 parameter is blank by default, so the app serves the Calculator view without focusing operand1.
postDefault()	result()	Perform a POST request with no parameters. The server returns the Error view since no operator is provided.
postIncomplete()	result()	Perform a POST request with only the "operand1" parameter provided. The server returns the Error view since no operator is provided.
postBlankParameterAdd()	result()	Perform a POST request with the "+" operator and a blank operand parameter. The Binary class interprets this parameter as a binary 0 and runs the operation as normal.
postParameterMultiply()	result()	Perform a POST request with the "*" operator and two valid operands, then check if the result is the product of the two operands.
postBlankParameterMultiply()	result()	Perform a POST request with the "*" operator and a blank operand parameter. The Binary class interprets this parameter as a binary 0 and runs the operation as normal.
postParameterAnd()	result()	Perform a POST request with the "&" operator and two valid operands, then check if the result is the bitwise AND of the two operands.
postBlankParameterAnd()	result()	Perform a POST request with the "&" operator and a blank operand parameter. The Binary class interprets this parameter as a binary 0 and runs the operation as normal.

postParameterOr()	result()	Perform a POST request with the “ ” operator and two valid operands, then check if the result is the bitwise OR of the two operands.
postBlankParameterOr()	result()	Perform a POST request with the “ ” operator and a blank operand parameter. The Binary class interprets this parameter as a binary 0 and runs the operation as normal.

```
[INFO] Tests run: 13, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.974 s - in com.ontariotechu.sofe3980U.BinaryControllerTest
```

I have also included my test cases for the Binary class that I used in the first deliverable.

```
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 52, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 10.273 s
[INFO] Finished at: 2024-02-16T21:04:48-05:00
[INFO] -----
```