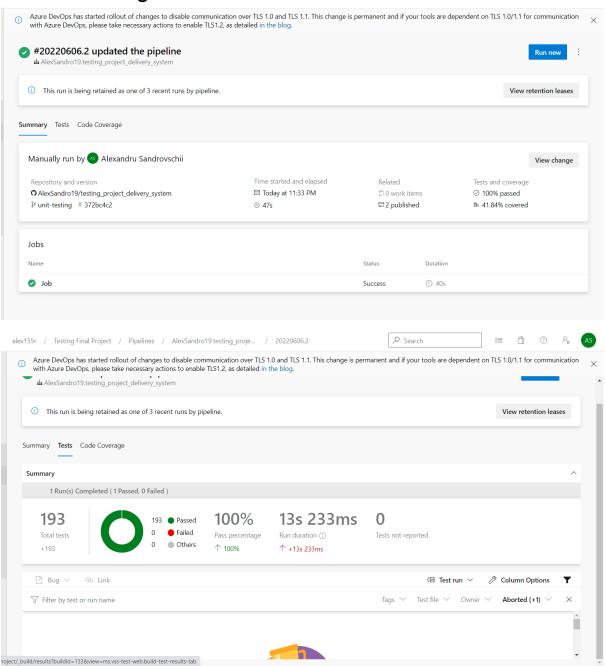
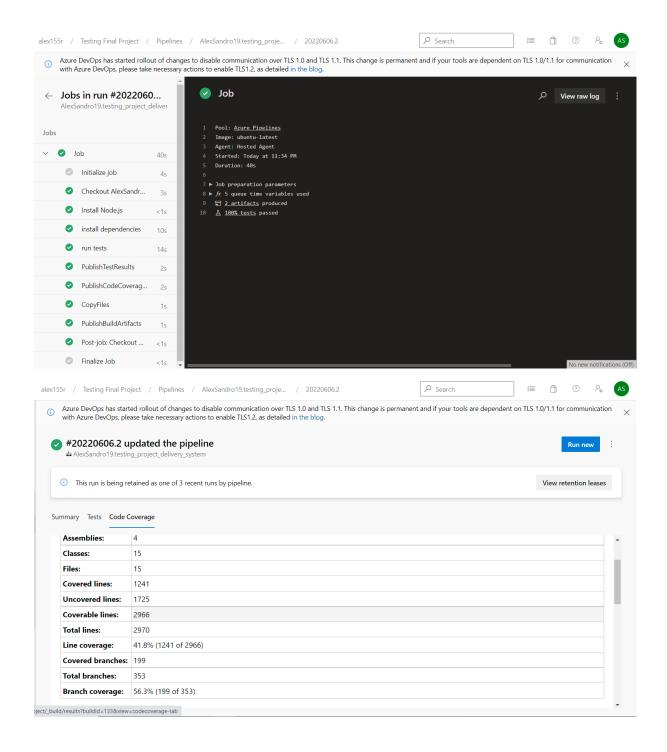
Code Coverage Results





How code coverage helped

```
const { execute } = require(".../database/mysql.connector.js");
        const getDateInSqlFormat = (date = null) => {
            // console.log("date: ", date)
// console.log("typeof date: ", typeof date)
             if (date instanceof Date && !isNaN(date)) {
                 const year = date.getFullYear(
                 const month = ((date.getMonth() + 1) >= 10) ? \{date.getMonth() + 1\} : 0\{date.getMonth() + 1\}
                const day = (date.getDate() >= 10) ? `${date.getDate()}` : `0${date.getDate()}`
const hours = (date.getHours() >= 10) ? `${date.getHours()}` : `0${date.getHours()}`
                const minutes = (date.getMinutes() >= 10) ? `${date.getMinutes()}` : `0${date.getMinutes()}`
const seconds = (date.getSeconds() >= 10) ? `${date.getSeconds()}` : `0${date.getSeconds()}`
                const res = `${year}-${month}-${day} ${hours}:${minutes}:${seconds}`
             } else {
                 return null
       module.exports ={
             getDateInSqlFormat
PROBLEMS OUTPUT GITLENS TERMINAL
 PASS tests/utility/utility.generator.test.js tests/utility/utility.functions.test.js
       tests/utility/utility.calculations.test.js
                                % Stmts
                                            % Branch | % Funcs |
                                                                     % Lines
                                                                                 Uncovered Line #s
All files
 utility.calculations.js
utility.functions.js
                                                            33.33
                                   55.88
100
                                                 100
                                                                        55.88
                                                                                 24-37,45-58,71-84,87-89
                                                                                 8-12
39-42
                                                44.44
                                                              100
                                                                          100
                                                92.85
 utility.generators.js
                                  93.65
Test Suites: 3 passed, 3 total
```

We have a function that converts the Date object into a string in the "YYYY-MM-DD hh:mm:ss" that MySql expects to receive.

Before implementing the code coverage, we were testing if a passed Date object returns the string in the correct format and what is the output with invalid input values.

```
PASS tests/utility/utility.functions.test.js
Run tests for getDateInSqlFormat function

V Pass a valid Date object and expect a string representation of it if format 'YYYY-MM-DD hh:mm:ss' (9 ms)

V Pass a valid Date object with month smaller than 10 and expect a string representation of it (1 ms)

V Pass an invalid Date object and expect null (1 ms)

V Pass a invalid object and expect null (1 ms)

V Pass a invalid string and expect null (1 ms)

V Pass a invalid number and expect null

V Pass a invalid null vallue and expect null

V Pass a nothing and expect null (1 ms)
```

However, in the code we have some logic that appends a 0 to the value if it is smaller than 10. This portion of code wasn't tested and code coverage showed that by specifying the % and 'Uncovered Lines #s': 8-12.

After adding additional tests that cover those use cases, we get 100% coverage.

```
PASS tests/utility/utility.functions.test.js
Run tests for getDateInSqlFormat function

V Pass a valid bate object and expect a string representation of it in format 'YYYY-MM-DD hh:mm:ss' (5 ms)

V Pass a valid Date object with month and day smaller than 10 and expect a string representation of it in format 'YYYY-MM-DD hh:mm:ss'

V Pass a valid Date object with month and day bigger than 10 and expect a string representation of it in format 'YYYY-MM-DD hh:mm:ss'

V Pass a valid Date object with hour, minutes and seconds smaller than 10 and expect a string representation of it in format 'YYYY-MM-DD hh:mm:ss'

V Pass a valid Date object with hour, minutes and seconds bigger than 10 and expect a string representation of it in format 'YYYY-MM-DD hh:mm:ss'

V Pass a invalid Date object and expect null (1 ms)

V Pass a invalid object and expect null (1 ms)

V Pass a invalid object and expect null (1 ms)

V Pass a invalid number and expect null

V Pass a invalid number and expect null
```

File	 % Stmts	 % Branch	% Funcs	 % Lines	Uncovered Line #s
All files utility.calculations.js utility.functions.js utility.generators.js	74.07 55.88 100 93.65	97.43 100 100 92.85	63.63 33.33 100 100	74.07 55.88 100 93.65	24-37,45-58,71-84,87-89 39-42

Test Suites: 3 passed, 3 total
Tests: 46 passed, 46 total

Snapshots: 0 total
Time: 3.175 s

Lo 40 Col 7