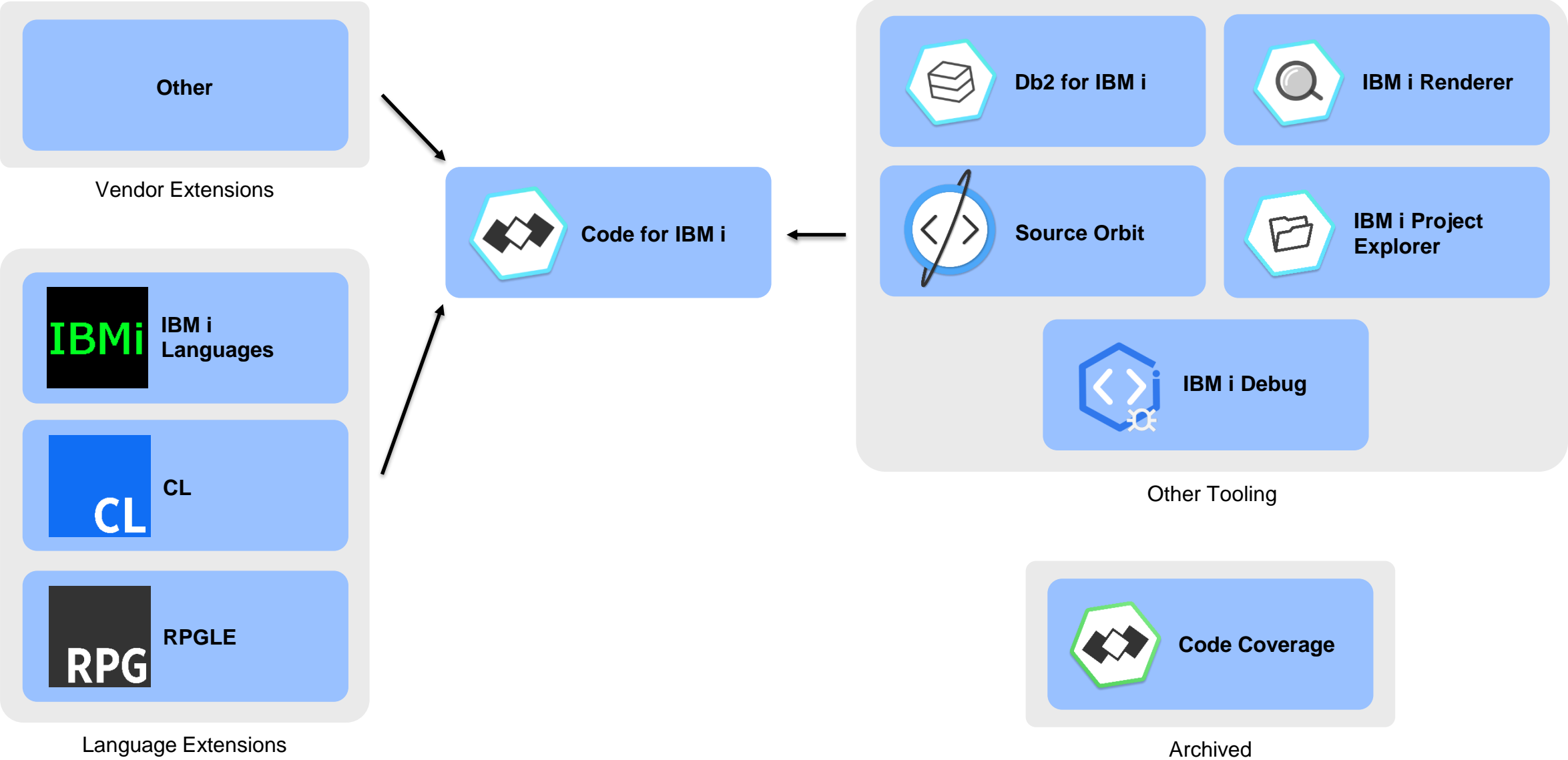


IBM i Unit Testing Now in VS Code!



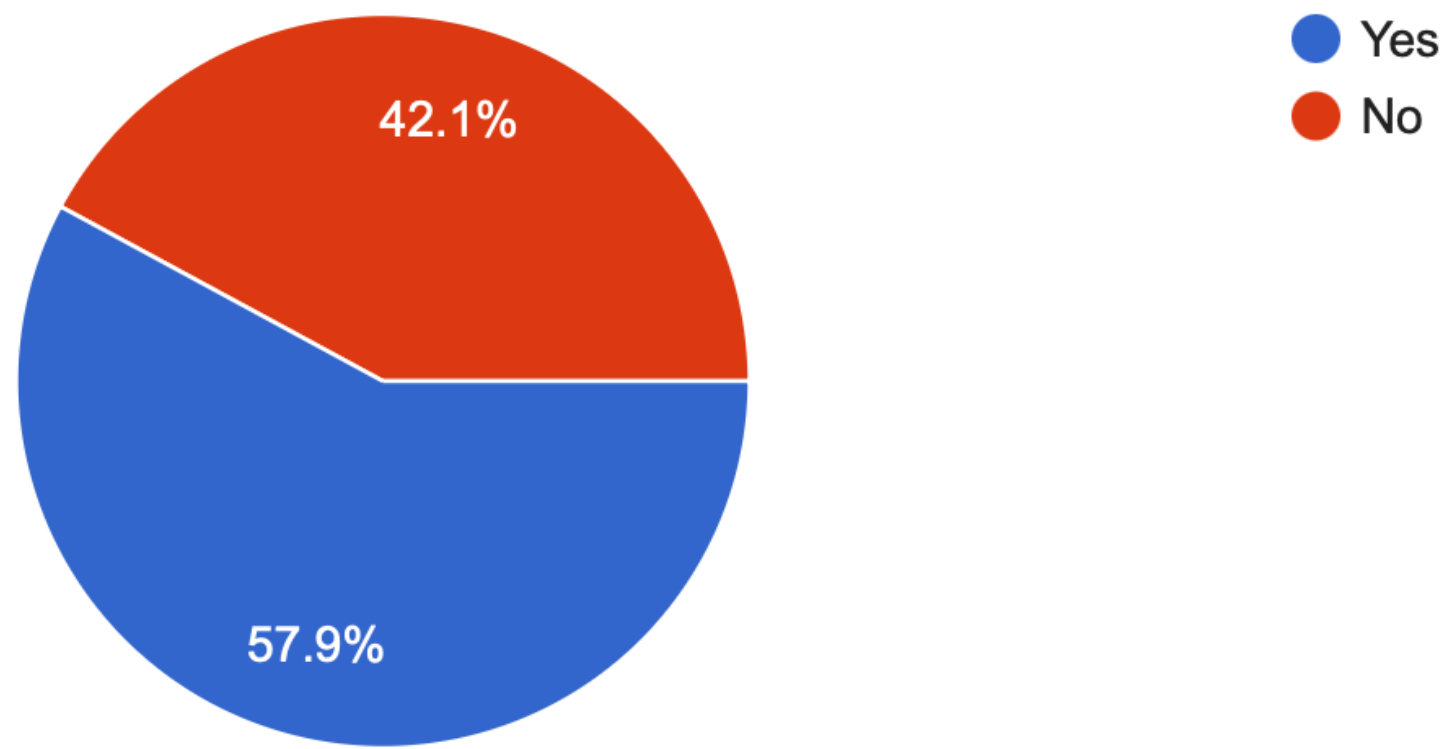
What is missing in the VS Code stack?



We did a survey...

Have you done unit testing with IBM i before?

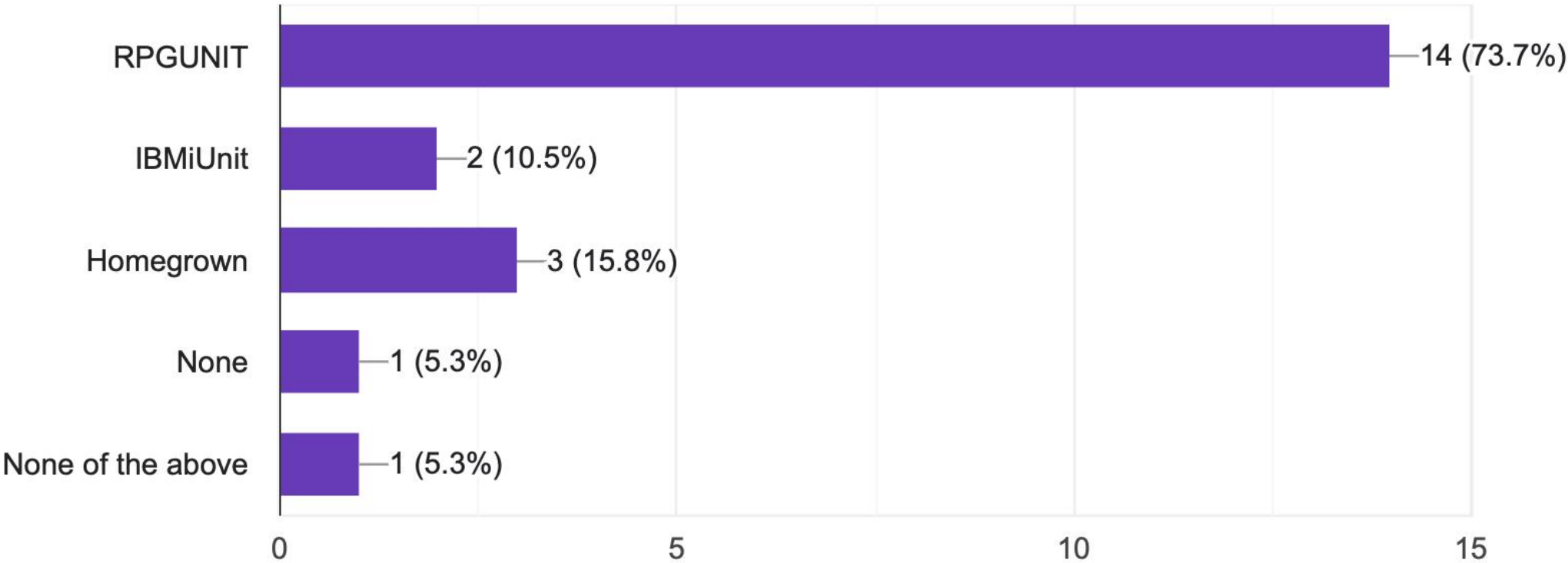
19 responses



We did a survey...

What framework did you use for IBM i testing?

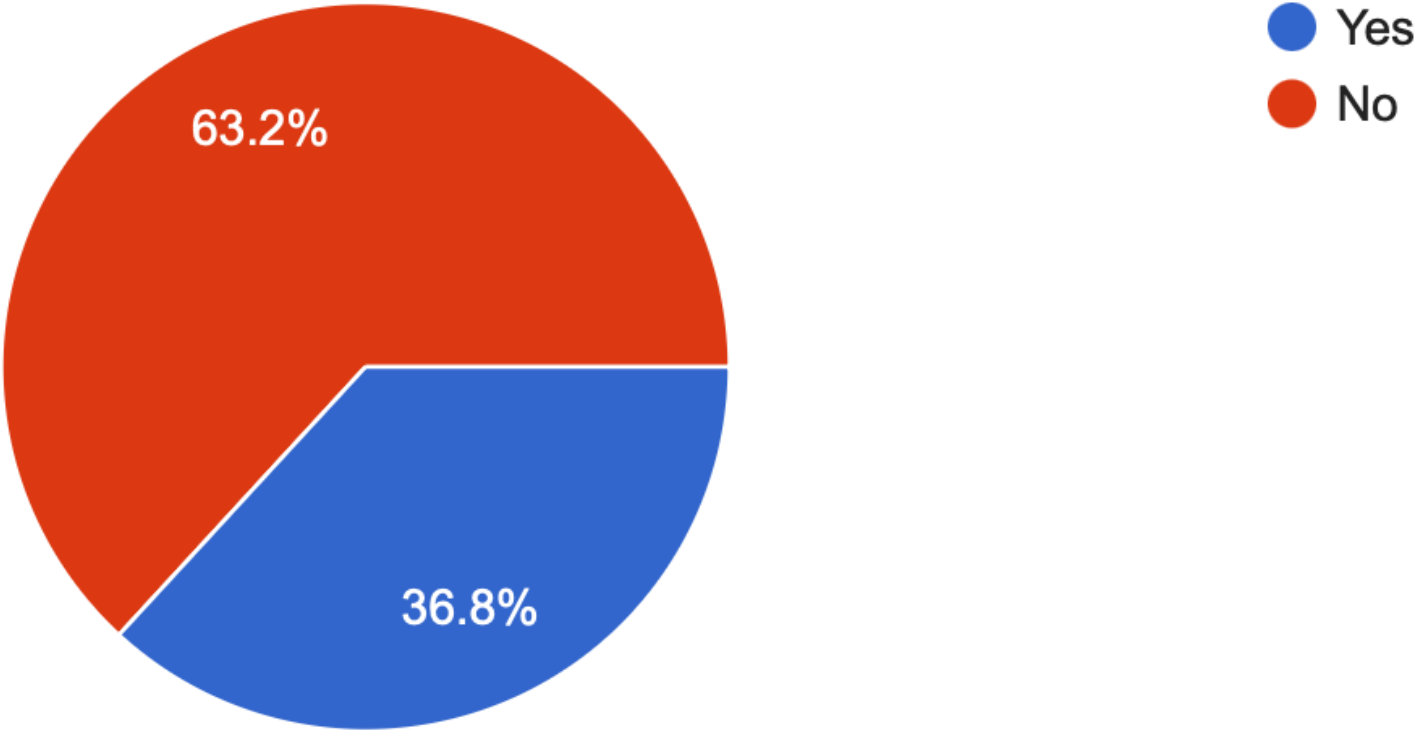
19 responses



We did a survey...

Have you used IBM i Code Coverage? (CODECOV)

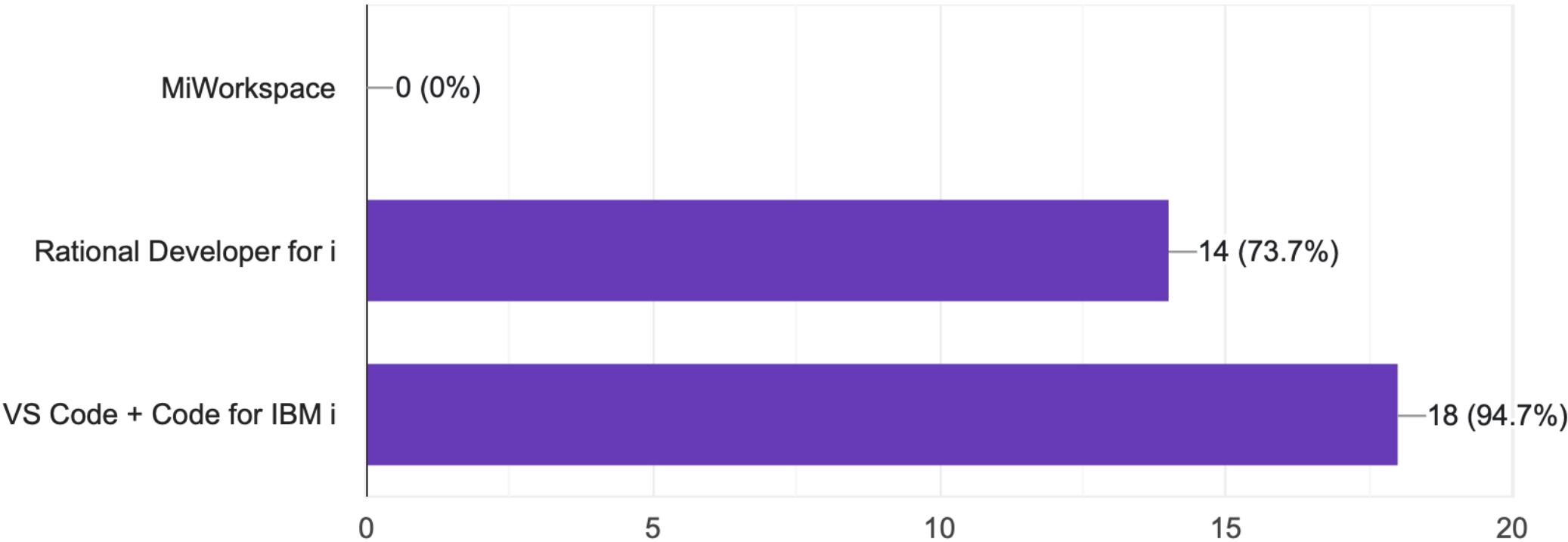
19 responses



We did a survey...

Which IDEs for IBM i have you used?

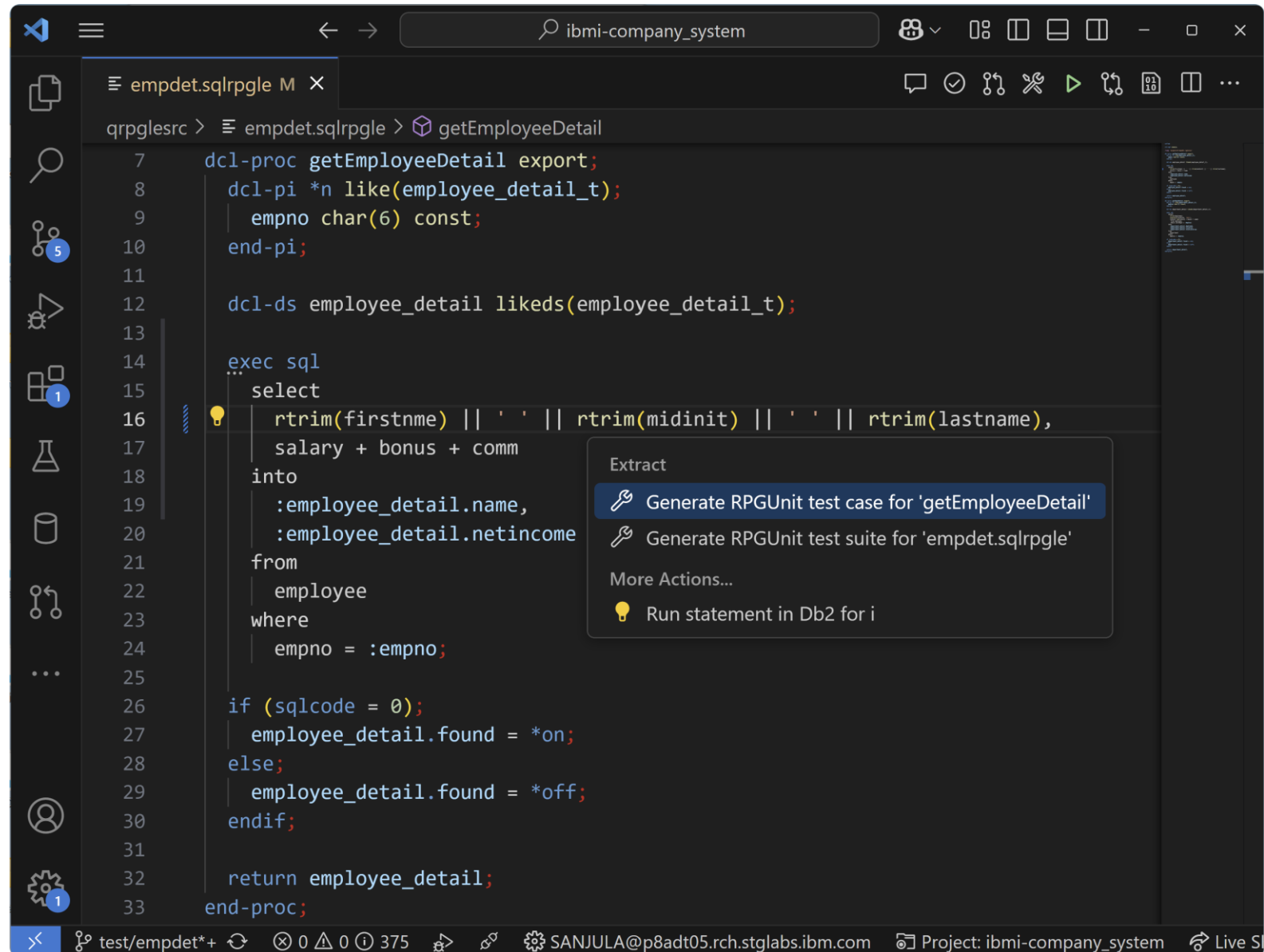
19 responses



Let's look at the new IBM i Testing Extension



Generating test stubs



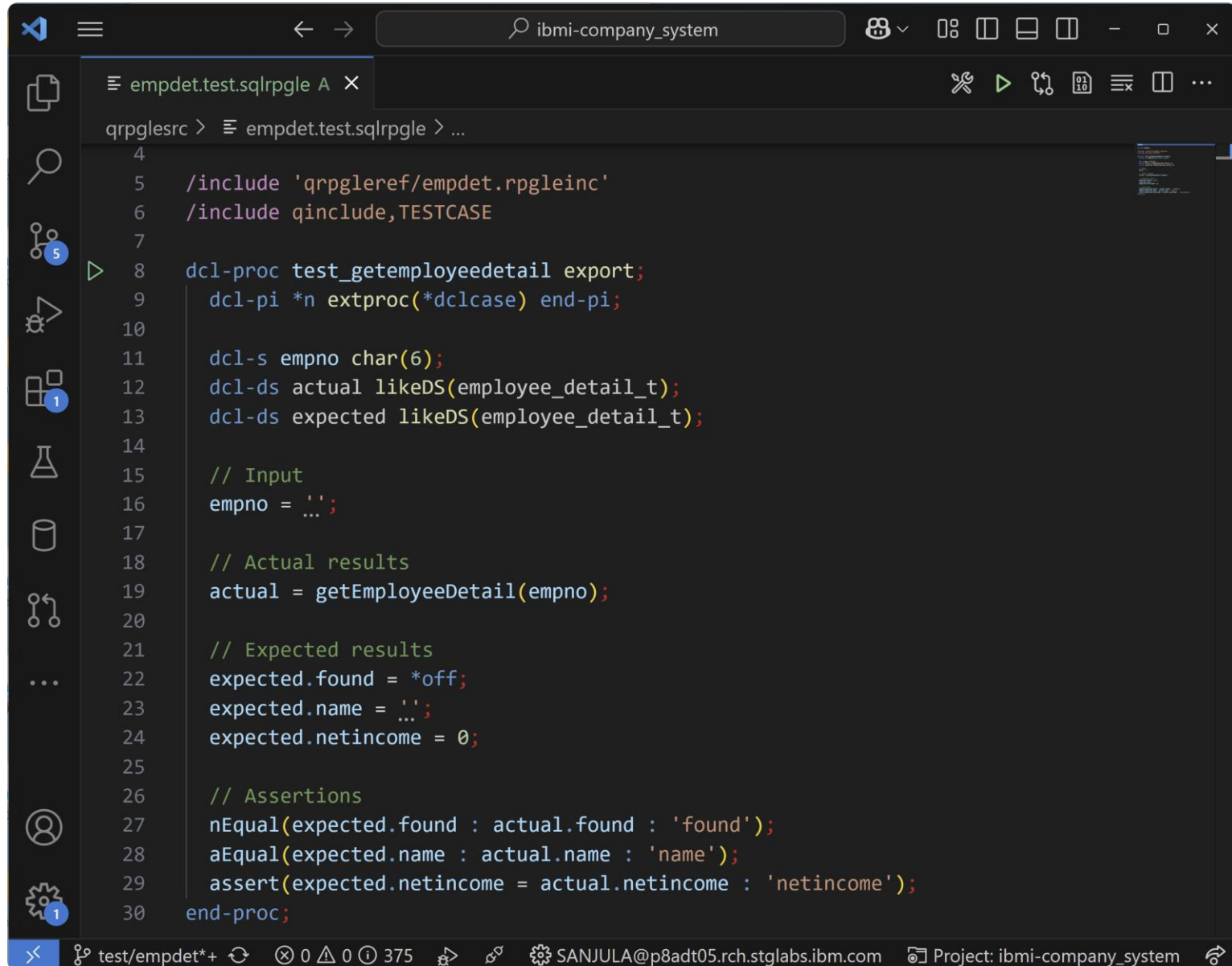
The screenshot shows an IDE window titled 'ibmi-company_system' with a file named 'empdet.sqlrpgle' open. The code is in RPG format and defines a procedure 'getEmployeeDetail'. A context menu is open over the 'select' statement, offering options to generate test cases or suites. The code is as follows:

```
7 dcl-proc getEmployeeDetail export;  
8   dcl-pi *n like(employee_detail_t);  
9     empno char(6) const;  
10  end-pi;  
11  
12  dcl-ds employee_detail likeds(employee_detail_t);  
13  
14  exec sql  
15  select  
16    rtrim(firstname) || ' ' || rtrim(midinit) || ' ' || rtrim(lastname),  
17    salary + bonus + comm  
18  into  
19    :employee_detail.name,  
20    :employee_detail.netincome  
21  from  
22    employee  
23  where  
24    empno = :empno;  
25  
26  if (sqlcode = 0);  
27    employee_detail.found = *on;  
28  else;  
29    employee_detail.found = *off;  
30  endif;  
31  
32  return employee_detail;  
33 end-proc;
```

The context menu options are:

- Extract
 - Generate RPGUnit test case for 'getEmployeeDetail'
 - Generate RPGUnit test suite for 'empdet.sqlrpgle'
- More Actions...
 - Run statement in Db2 for i

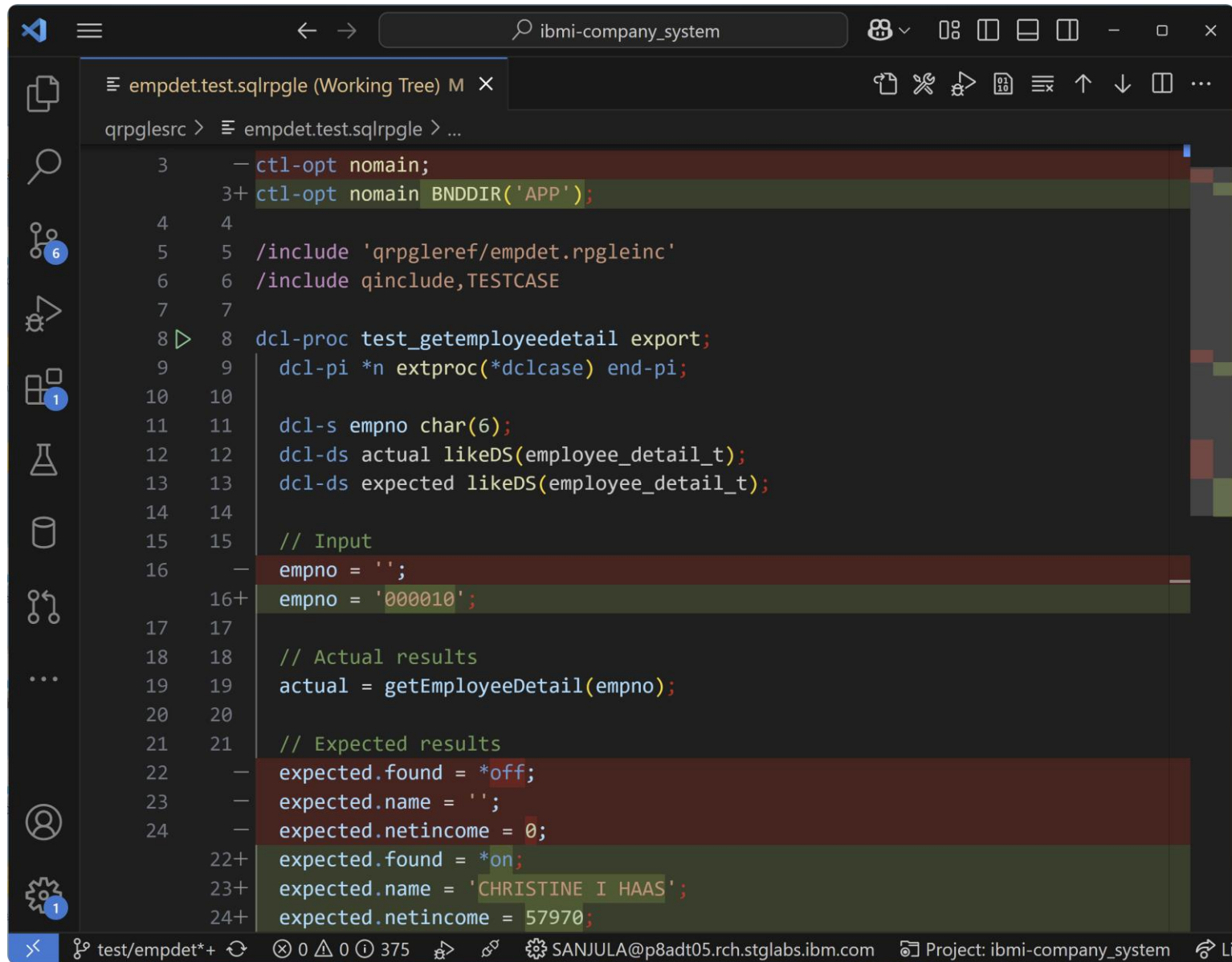
Generating test stubs



The image shows a Visual Studio Code editor window with a dark theme. The title bar at the top indicates the project is 'ibmi-company_system'. The editor is open to a file named 'empdet.test.sqlrpgle A'. The code is written in RPG and is a test stub for a program named 'test_getemployeedetail'. It includes headers for 'qrpgleref/empdet.rpgleinc' and 'qinclude,TESTCASE'. The test procedure 'test_getemployeedetail' is defined with an 'export' attribute and a parameter 'extproc(*dclcase)'. Inside the procedure, it declares a character variable 'empno' of length 6, and two data structures 'actual' and 'expected' of type 'employee_detail_t'. It then sets 'empno' to an empty string. The 'actual' data structure is populated by calling 'getEmployeeDetail(empno)'. The 'expected' data structure is initialized with 'found' set to *off, 'name' to an empty string, and 'netincome' to 0. Finally, assertions are made: 'nEqual' for 'found', 'aEqual' for 'name', and 'assert' for 'netincome'. The procedure ends with 'end-proc;'. The left sidebar shows the Explorer view with a file tree containing 'test/empdet*+'. The bottom status bar shows the file path 'test/empdet*+', a refresh icon, a search icon, a line count of 375, a user profile icon, the email 'SANJULA@p8adt05.rch.stglabs.ibm.com', the project name 'Project: ibmi-company_system', and a share icon.

```
4  
5  /include 'qrpgleref/empdet.rpgleinc'  
6  /include qinclude,TESTCASE  
7  
8  dcl-proc test_getemployeedetail export;  
9      dcl-pi *n extproc(*dclcase) end-pi;  
10  
11     dcl-s empno char(6);  
12     dcl-ds actual likeDS(employee_detail_t);  
13     dcl-ds expected likeDS(employee_detail_t);  
14  
15     // Input  
16     empno = '';  
17  
18     // Actual results  
19     actual = getEmployeeDetail(empno);  
20  
21     // Expected results  
22     expected.found = *off;  
23     expected.name = '';  
24     expected.netincome = 0;  
25  
26     // Assertions  
27     nEqual(expected.found : actual.found : 'found');  
28     aEqual(expected.name : actual.name : 'name');  
29     assert(expected.netincome = actual.netincome : 'netincome');  
30 end-proc;
```

Writing Tests



The screenshot shows a Visual Studio Code editor window with a file named `empdet.test.sqlrpgle` open. The editor is displaying COBOL code for a test program. The code includes comments, includes, and data declarations. The test logic involves setting input values, calling a function to get employee details, and comparing the actual results with expected values. The status bar at the bottom indicates the current file is `test/empdet*+` and the project is `ibmi-company_system`.

```
3      - ctl-opt nomain;  
3+    3+  ctl-opt nomain BNDDIR('APP');  
4      4  
5      5  /include 'qrpqleref/empdet.rpgleinc'  
6      6  /include qinclude,TESTCASE  
7      7  
8      8  dcl-proc test_getemployeedetail export;  
9      9  dcl-pi *n extproc(*dclcase) end-pi;  
10     10  
11     11  dcl-s empno char(6);  
12     12  dcl-ds actual likeDS(employee_detail_t);  
13     13  dcl-ds expected likeDS(employee_detail_t);  
14     14  
15     15  // Input  
16     -  empno = '';  
16+    16+  empno = '000010';  
17     17  
18     18  // Actual results  
19     19  actual = getEmployeeDetail(empno);  
20     20  
21     21  // Expected results  
22     -  expected.found = *off;  
23     -  expected.name = '';  
24     -  expected.netincome = 0;  
22+    22+  expected.found = *on;  
23+    23+  expected.name = 'CHRISTINE I HAAS';  
24+    24+  expected.netincome = 57970;
```

Running unit tests

TEST EXPLORER

Filter (e.g. text, !exclude, @tag)

1/1 9.3s

ibmi-company_system 68ms

qrpglesrc 68ms

empdet.test.sqlrpgle 68ms

test_getemployeeedetail 68ms

RPGUNIT

empdet.test.sqlrpgle M

qrpglesrc > empdet.test.sqlrpgle > test_getemployeeedetail

8 dcl-proc test_getemployeeedetail export;

9 dcl-pi *n extproc(*dclcase) end-pi;

10

11 dcl-s empno char(6);

12 dcl-ds actual likeDS(employee_detail_t);

13 dcl-ds expected likeDS(employee_detail_t);

14

15 // Input

16 empno = '000010';

PROBLEMS 375

COMMENTS

DB2 FOR I

DEBUG CONSOLE

OUTPUT

TERMINAL

TEST RESULTS

WORKSPACE

ibmi-company_system (1) [Deployment Successful]

> empdet.test.sqlrpgle (1) [Compilation Successful]

✓ test_getemployeeedetail 0.068s

EXECUTION

Deployments: 1 successful | 0 failed (1)

Compilations: 1 successful | 0 failed | 0 skipped (1)

RESULTS

Test Files: 1 passed | 0 failed | 0 errored (1)

Test Cases: 1 passed | 0 failed | 0 errored (1)

Assertions: 3

Duration: 0.068s

PASS

IBM i Testing

✓ test_getemployeeede...

test/empdet*+

0 0 0 375

SANJULA@p8adt05.rch.stglabs.ibm.com

Project: ibmi-company_system

Live Share

Deploy

New job (1)

Pul

Viewing failures and errors

TEST EXPLORER

Filter (e.g. text, !exclude, @tag)

6/16

17.3s

examples 35ms

- cbl.test.cbllle
 - TEST_TRUE
 - TEST_FALSE
 - TEST_FAIL0
 - TEST_FAIL1
- rpgsqlstmf.test.sqlrpgle 26ms
 - testWhatever_1 25ms
 - testWhatever_2 1.0ms
- rpgstmf.test.rpgle 3.0ms
 - testWhatever_1 3.0ms
 - testWhatever_2 0.0ms
- sub 6.0ms
- RPGUNIT

rpgstmf.test.rpgle

rpgstmf.test.rpgle > ...

```
222 // =====
223 // RPGUnit test case.
224 // =====
225 dcl-proc testWhatever_1 export;
226   dcl-pi *n extproc(*dclcase) end-pi;
227
228   print('Executing:      * testWhatever_1()');
229
230   // Run
231   assert(sds.pgmName = 'TEMPLATE': 'Name of the test suite should be 'TEMPLATE''');
232
233   // Place your assertions here.
234
235 end-proc;
```

PROBLEMS

DB2 FOR I

DEBUG CONSOLE

OUTPUT

TERMINAL

PORTS

IBM I JOB LOG

TEST RESULTS

WORKSPACE

examples (4) [Deployment Successful]

> rpgstmf.test.rpgle (2) [Compilation Skipped]

X testWhatever_1 0.003s

Failure (line 231): Name of the test suite should be 'TEMPLATE'

✓ testWhatever_2 0s

> rpgsqlstmf.test.sqlrpgle (2) [Compilation Skipped]

✓ testWhatever_1 0.025s

✓ testWhatever_2 0.001s

> cbl.test.cbllle (4) [Compilation Failed]

CPFA0A9: Object not found. Object is EN_US.UTF-8.

: CRTCBMOD MODULE(*PGMID) SRCSTMF('/home/SANJULA/builds/examples/cbl.test.cbllle') OPTION(*EVENTF) DBGVIEW(*SOURCE) TGTRLS(*CURRENT) INCDIR('/home/SANJULA/builds/examples') DEFINE(*NONE) TGTCCSID(37)

CPFA0A9: Object not found. Object is EN_US.UTF-8.

CPFA0A9: Object not found. Object is EN_US.UTF-8.

CPFA0A9: Object not found. Object is EN_US.UTF-8.

CPD4090: Printer device PRT01 not found. Output queue changed to QPRINT i

n library QGPL.

CPFA0A9: Object not found. Object is EN_US.UTF-8.

IBM i Testing

TEST_TRUE cbl.test.cbllle < examples

TEST_FALSE cbl.test.cbllle < examples

TEST_FAIL0 cbl.test.cbllle < examples

TEST_FAIL1 cbl.test.cbllle < examples

TEST_TRUE cbl.test.cbllle < sub < examples

TEST_FALSE cbl.test.cbllle < sub < examples

TEST_FAIL0 cbl.test.cbllle < sub < examples

TEST_FAIL1 cbl.test.cbllle < sub < examples

testWhatever_1 Name of the test suite should b...

testWhatever_1 Name of the test suite should b...

testWhatever_2 rpgstmf.test.rpgle < examples

testWhatever_1 rpgsqlstmf.test.sqlrpgle < examp...

testWhatever_2 rpgsqlstmf.test.sqlrpgle < examp...

testWhatever_2 rpgstmf.test.rpgle < sub < examp...

main*

0 0

SANJULA@p8adt05.rch.stglabs.ibm.com

Live Share

Deploy

New job (1)

Ln 239, Col 45

Spaces: 2

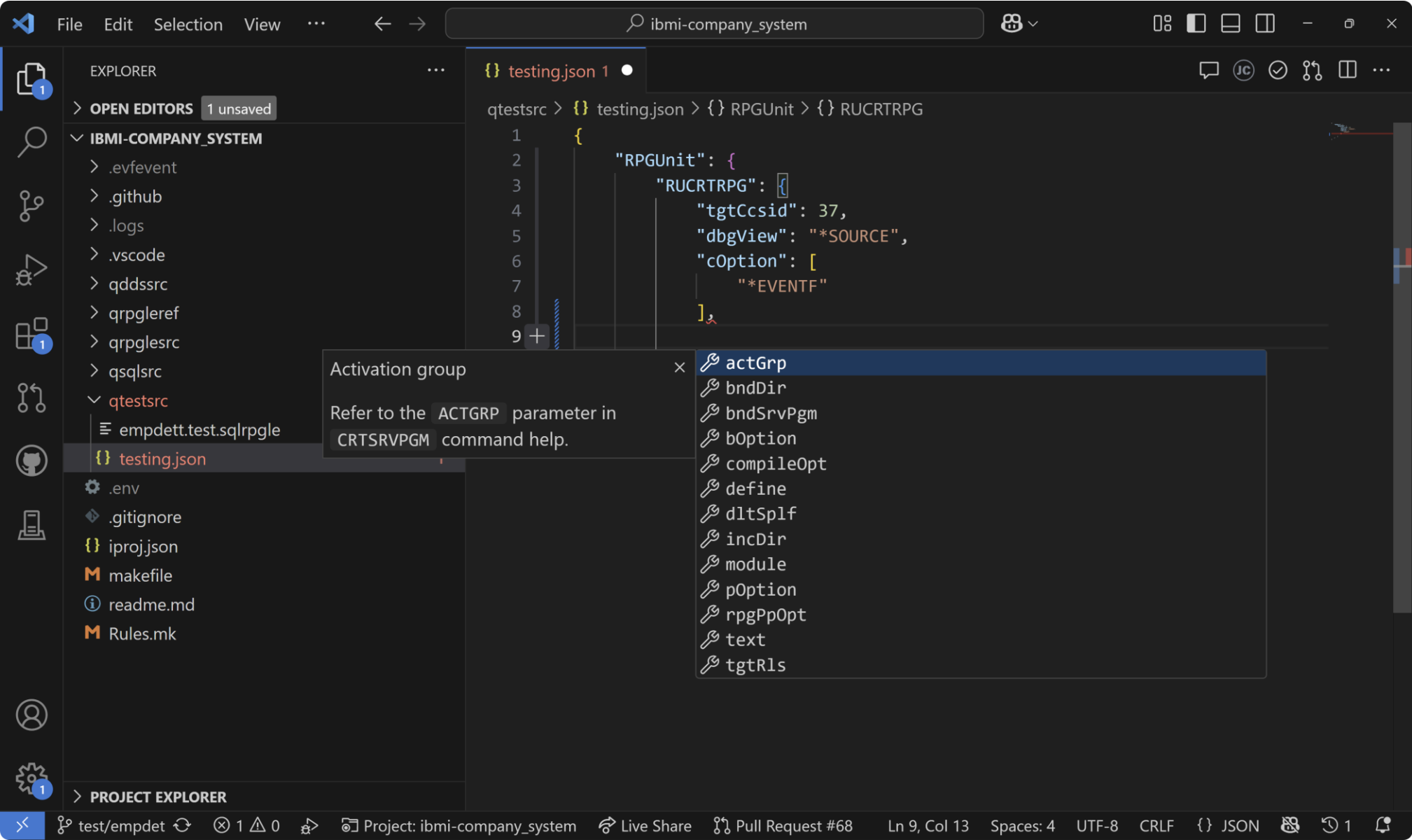
UTF-8

CRLF

{ } RPGLE

1

Configuring tests



Generating code coverage

TESTING

TEST EXPLORER

Filter (e.g. text, !exclude, @tag)

1/2 6.0s

examples 195ms

- cbl.test.cbll
- ✓ rpgsqlstmf.test.sqlrpgle 93ms
- ✗ rpgstmf.test.rpgle 4.0ms

sub 98ms

- cbl.test.cbll
- ✓ rpgsqlstmf.test.sqlrpgle 94ms
- ✗ rpgstmf.test.rpgle 4.0ms
 - ✗ testWhatever_1 3.0ms
 - ✓ testWhatever_2 1.0ms

Run Test with Coverage

TEST COVERAGE

rpgstmf.test.rpgle 60.00%

rpgstmf.test.rpgle

Hide Inline Coverage Rerun

```
165 // =====
166 // Set up test suite. Executed once per RUCALLTST.
167 // =====
168 dcl-proc setUpSuite export;
169   dcl-pi *n extproc(*dclcase) end-pi;
170
171   dcl-s rc char(1);
172
173   runCmd('OVRPRTF FILE(QSYSPRT) TOFILE(*FILE) +
174         SPLFNAME(PROC_FLOW) OVRSCOPE(*JOB)');
175   monitor;
176     openPrinter();
177     print('Executing:  setUpSuite()');
178   on-error;
179     // ignore errors ...
180   endmon;
181
182   // ... but try to remove the override.
183   monitor;
184     runCmd('DLTOVR FILE(QSYSPRT) LVL(*JOB)');
185   on-error;
186     dsply '*** Failed to delete QSYSPRT override! ***' rc;
187   endmon;
188
189 end-proc;
190
191 // =====
192 // Tear down test suite.
193 // =====
194 dcl-proc tearDownSuite export;
```

PROBLEMS

COMMENTS

DB2 FOR I

DEBUG CONSOLE

OUTPUT

TERMINAL

PORTS

IBM I JOB LOG

TEST RESULTS

> rpgstmf.test.rpgle (2) [Compilation Skipped]

✗ testWhatever_1 0.003s

Failure: Name of the test suite should be 'TEMPLATE'

✓ testWhatever_2 0.001s

Test Cases: 1 passed | 1 failed | 0 errored (2)

Duration: 0.004s

IBM i Testing

✗ Close Test Coverage

✗ testWhatever_1 Name of the test suite should be 'TEMPLATE'

✓ testWhatever_2

> 7 older results

© Copyright IBM Corporation 2024

14

Important Links

IBM i Testing

VS Code Marketplace

<https://marketplace.visualstudio.com/items?itemName=IBM.vscode-ibmi-testing>

Documentation

<https://codefori.github.io/docs/developing/testing/overview/>

GitHub Repository

<https://github.com/IBM/vscode-ibmi-testing>

RPGUNIT

Documentation

<https://irpgunit.sourceforge.io/help/>

GitHub Repository

<https://github.com/tools-400/irpgunit>

VS Code

Documentation

<https://code.visualstudio.com/docs/editor/testing>

Sanjula Ganepola - IBM i Unit Testing Now in VS Code!

Please take the last minute of this session to complete the evaluation. A direct link to the evaluation can be found using the QR code to the right.



IBM i