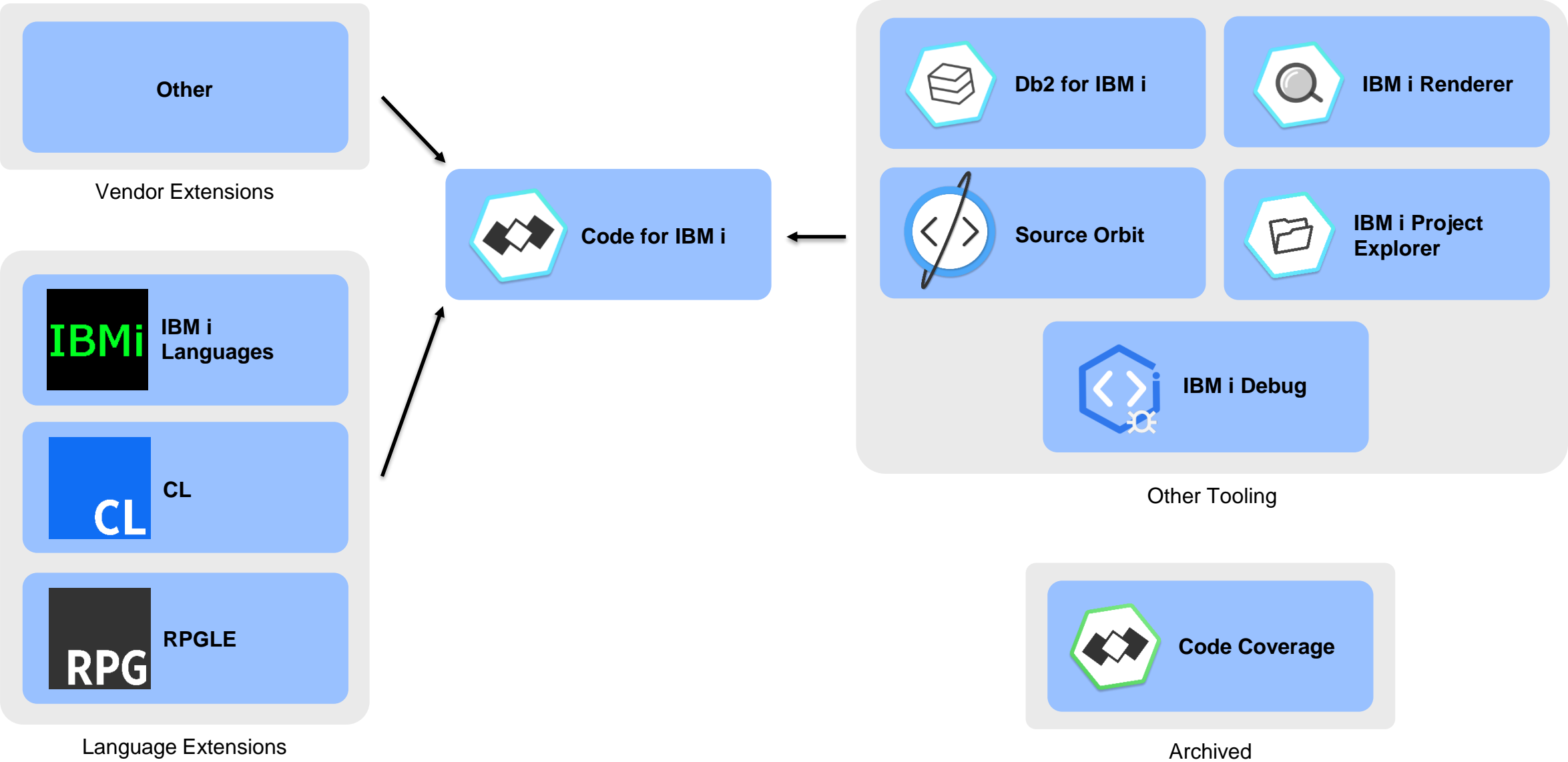


IBM i Unit Testing Now in VS Code!



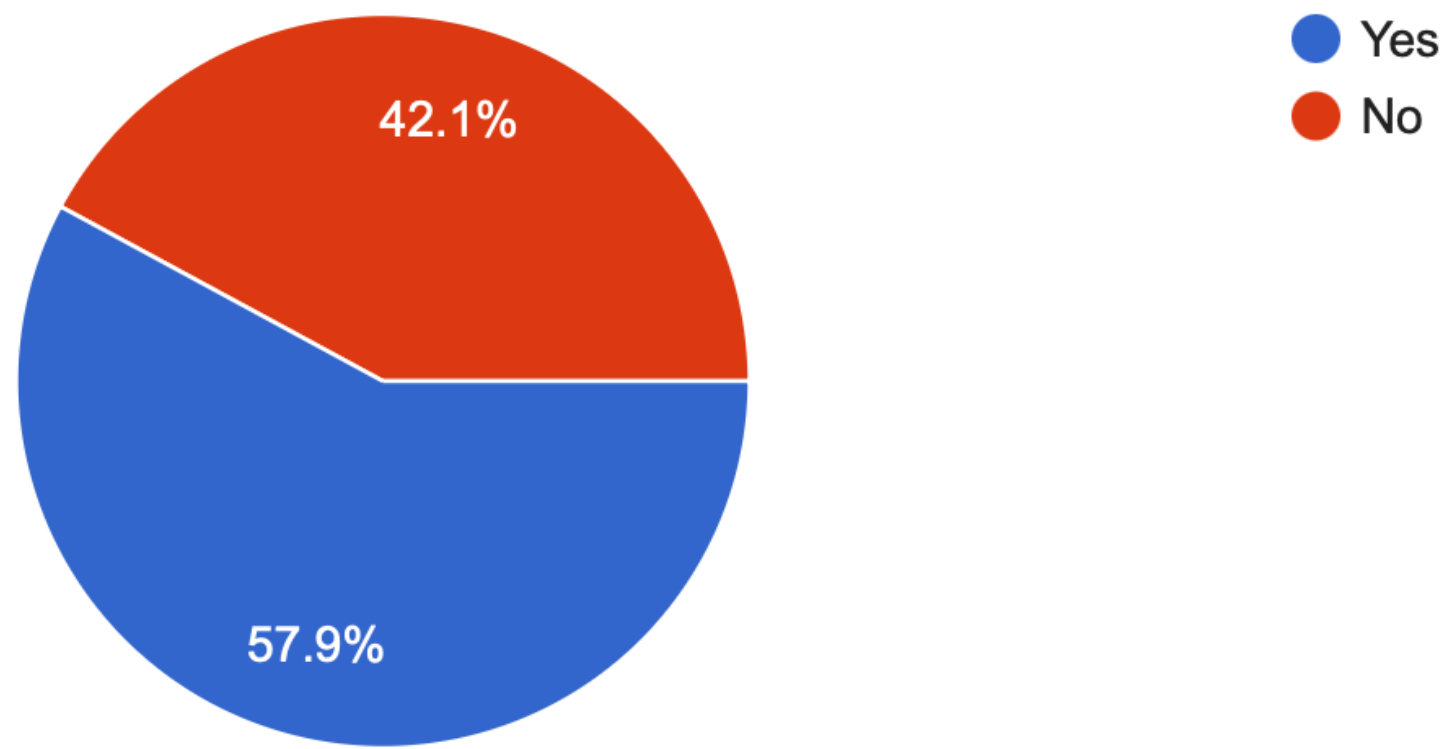
What is missing in the VS Code stack?



We did a survey...

Have you done unit testing with IBM i before?

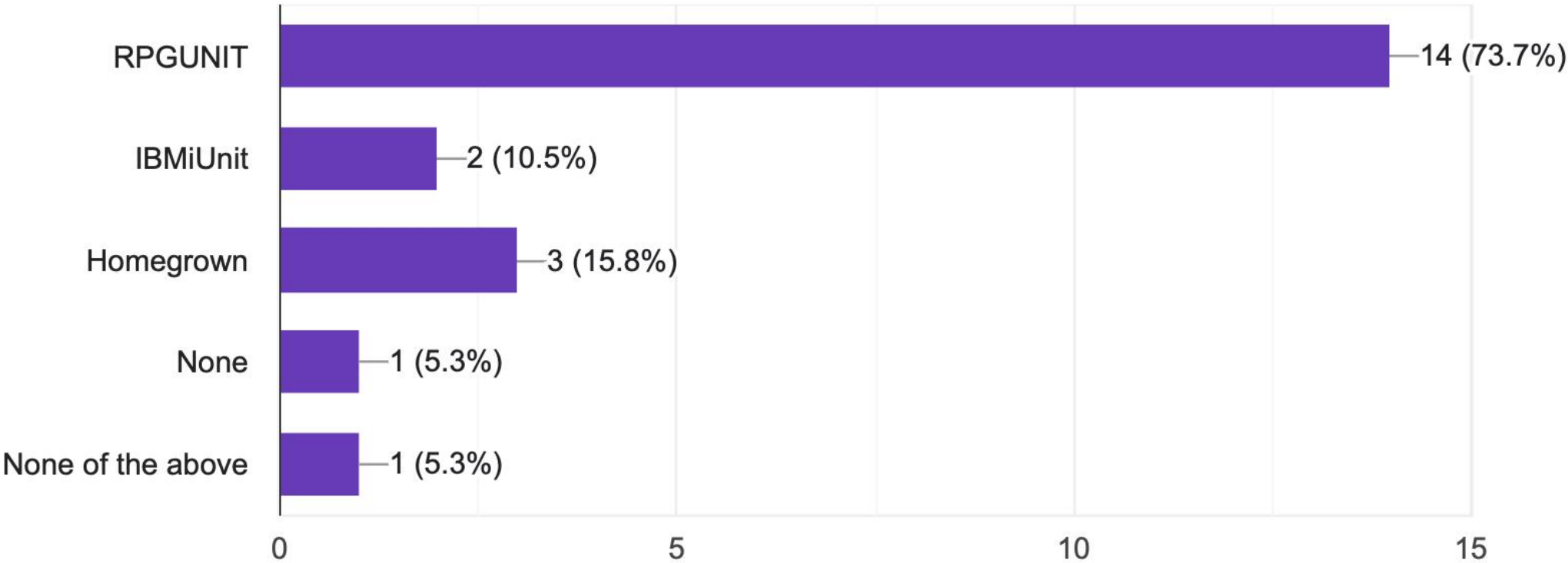
19 responses



We did a survey...

What framework did you use for IBM i testing?

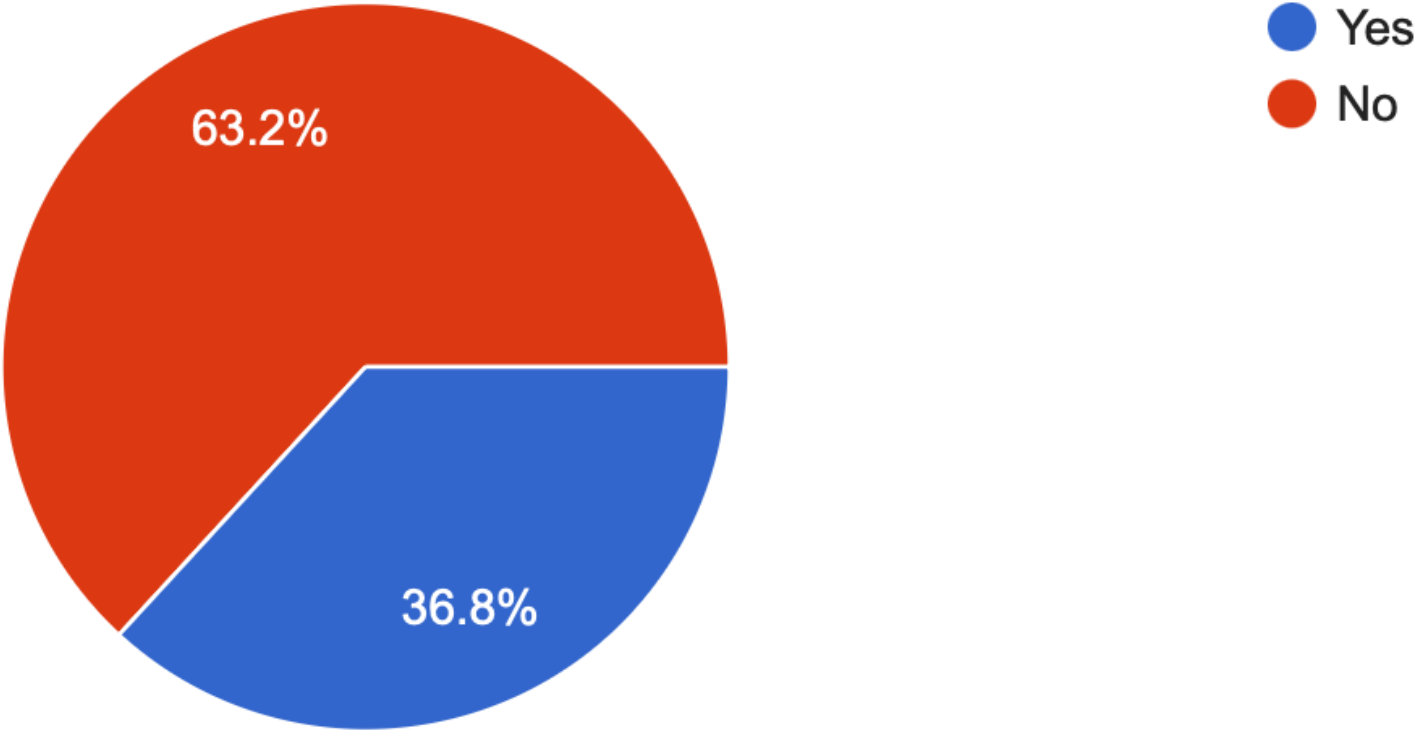
19 responses



We did a survey...

Have you used IBM i Code Coverage? (CODECOV)

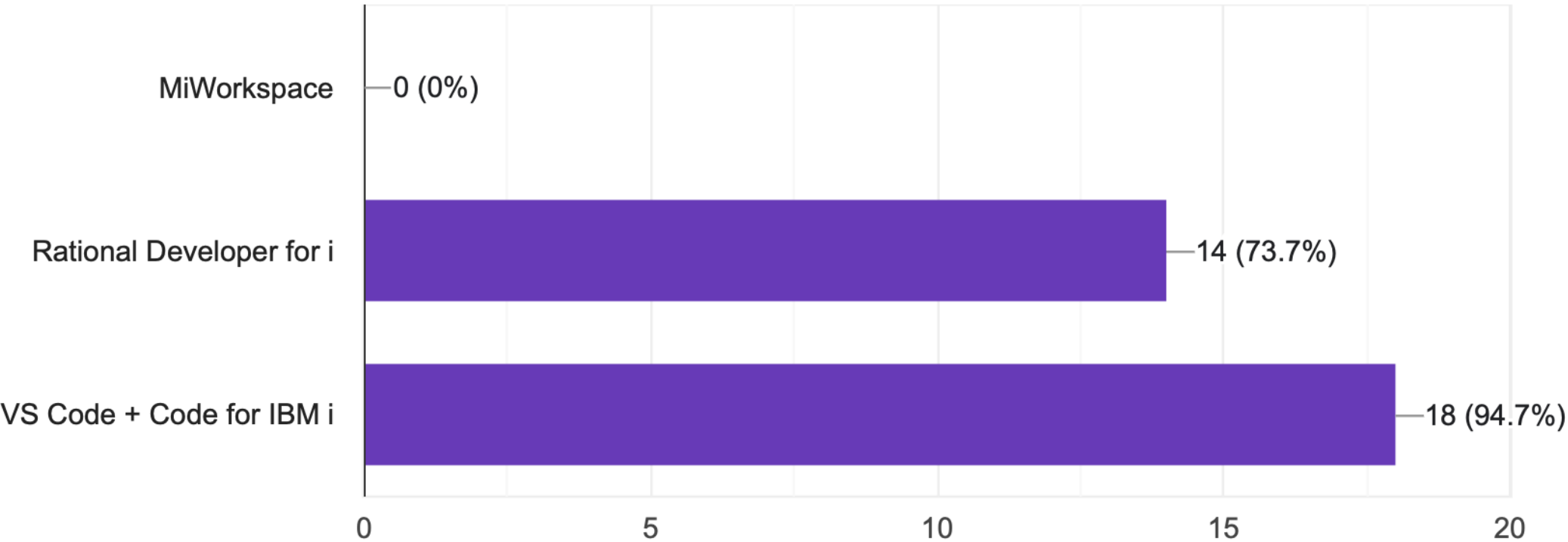
19 responses



We did a survey...

Which IDEs for IBM i have you used?

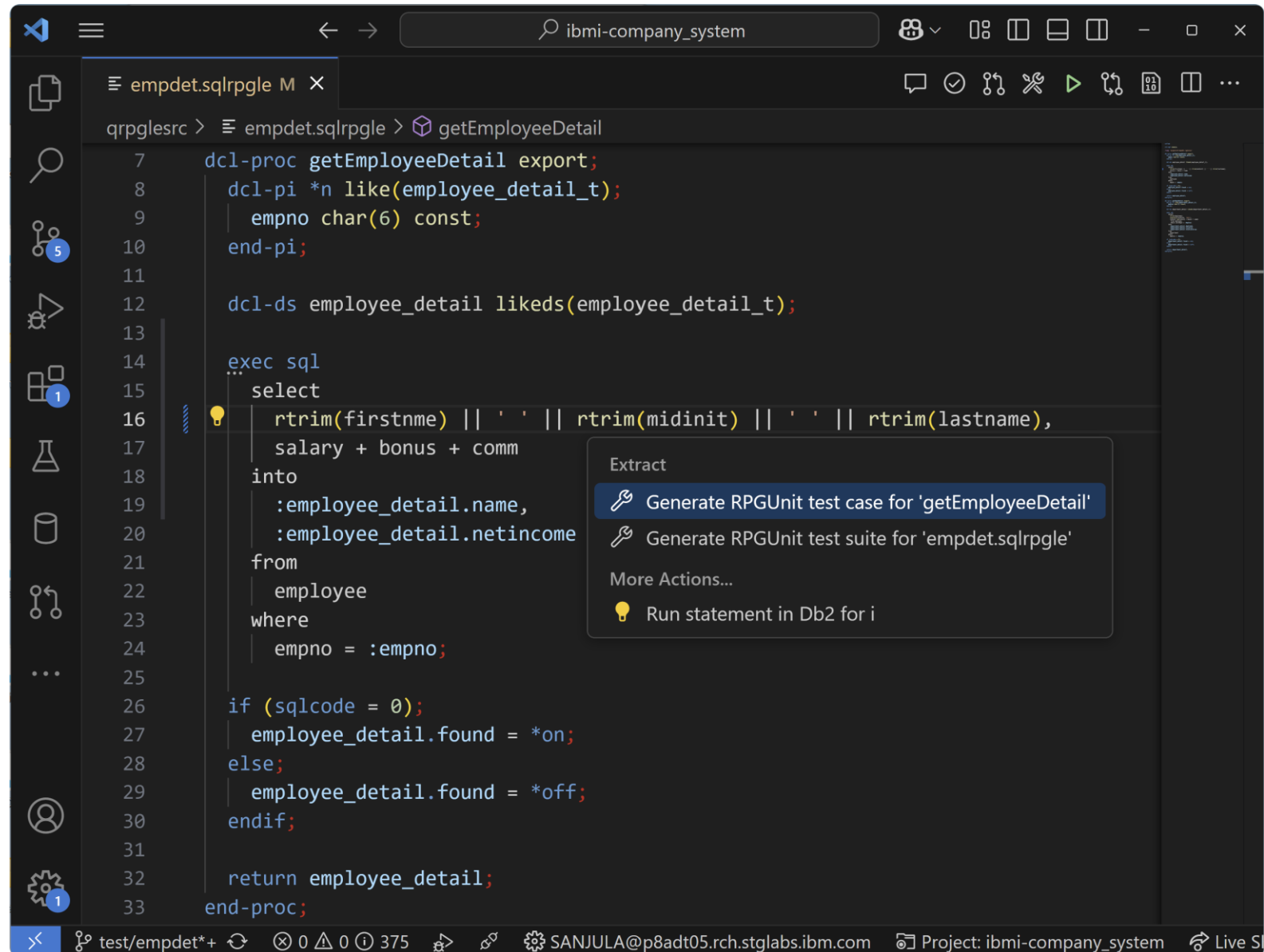
19 responses



Let's look at the new IBM i Testing Extension



Generating test stubs



The screenshot shows an IBM i development environment with a code editor displaying the following code:

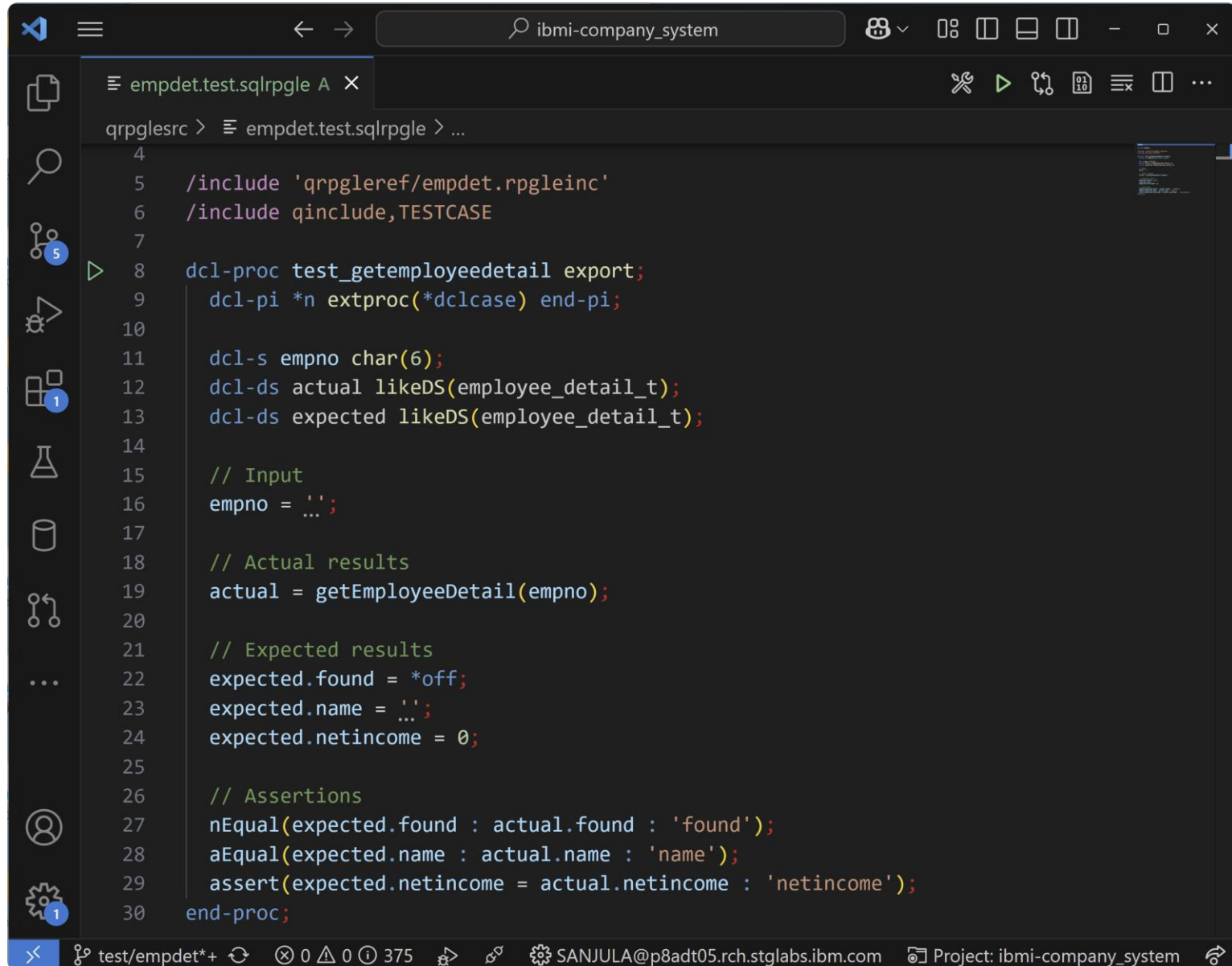
```
7 dcl-proc getEmployeeDetail export;  
8 dcl-pi *n like(employee_detail_t);  
9   empno char(6) const;  
10 end-pi;  
11  
12 dcl-ds employee_detail likeds(employee_detail_t);  
13  
14 exec sql  
15   select  
16     rtrim(firstname) || ' ' || rtrim(midinit) || ' ' || rtrim(lastname),  
17     salary + 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;
```

A context menu is open over the code, showing the following options:

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

The status bar at the bottom indicates the current file is 'test/empdet*+' and the user is 'SANJULA@p8adt05.rch.stglabs.ibm.com'.

Generating test stubs

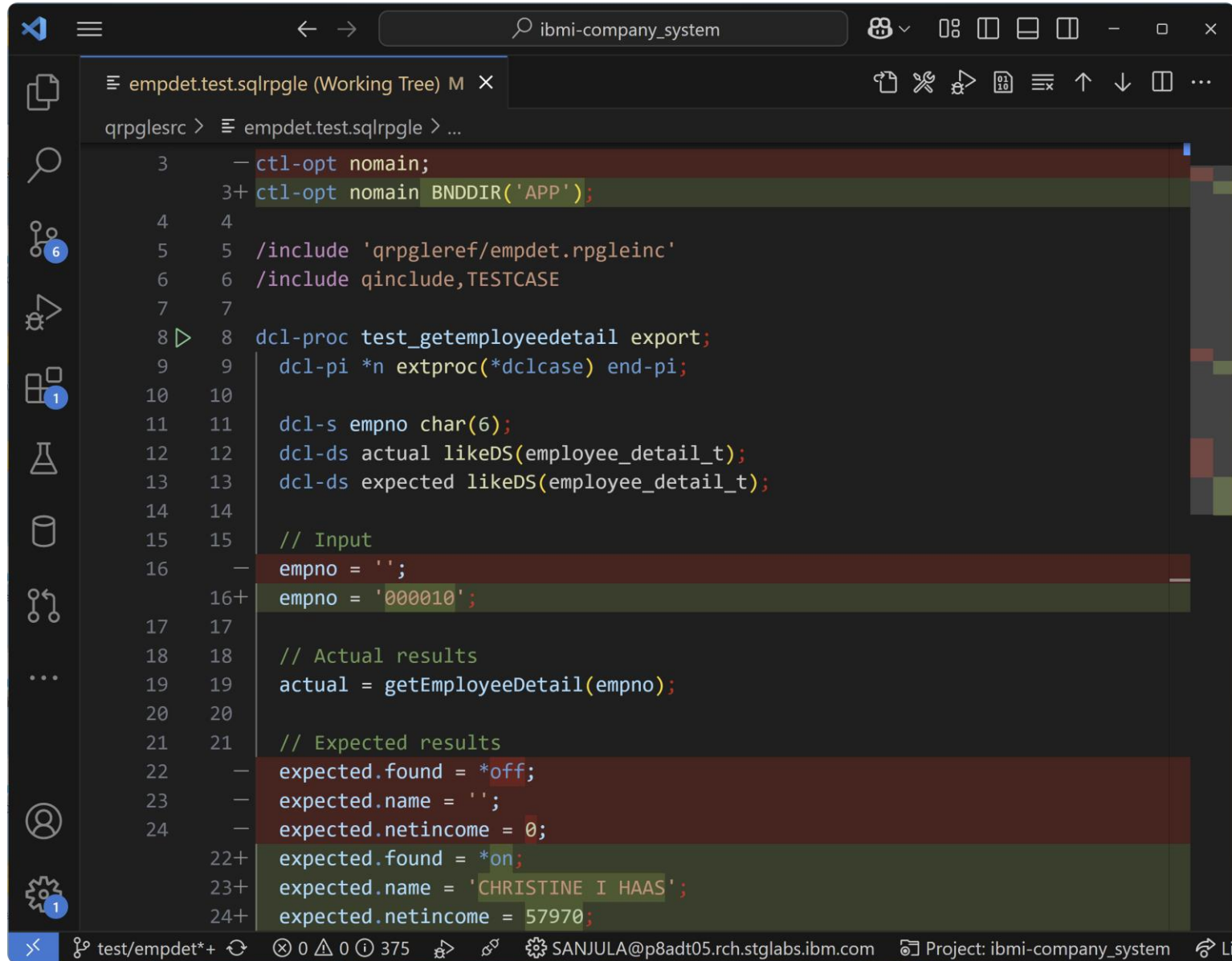


The screenshot shows a Visual Studio Code editor window with a dark theme. The title bar at the top indicates the project is 'ibmi-company_system'. The editor has a single tab open named 'empdet.test.sqlrpgle A'. The code is written in a QRPGL dialect and is as follows:

```
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;
```

The status bar at the bottom of the editor shows 'test/empdet*+' with a refresh icon, '0 0 375' with a magnifying glass icon, the user 'SANJULA@p8adt05.rch.stglabs.ibm.com' with a gear icon, and 'Project: ibmi-company_system' with a folder icon.

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 and several lines of COBOL syntax, such as `ctl-opt`, `/include`, `dcl-proc`, `dcl-pi`, `dcl-s`, `dcl-ds`, and `actual = getEmployeeDetail(empno);`. The code is color-coded, and the editor interface includes a sidebar with icons for Explorer, Search, Run and Debug, and Extensions. The status bar at the bottom shows the project name `ibmi-company_system` and the user `SANJULA@p8adt05.rch.stglabs.ibm.com`.

```
3      - ctl-opt nomain;  
3+    3+  ctl-opt nomain BNDDIR('APP');  
4      4  
5      5  /include 'qrpgleref/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_getemployeedetail 68ms

RPGUNIT

empdet.test.sqlrpgle M X

qrpglesrc > empdet.test.sqlrpgle > test_getemployeedetail

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 = '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_getemployeedetail 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_getemployeede...

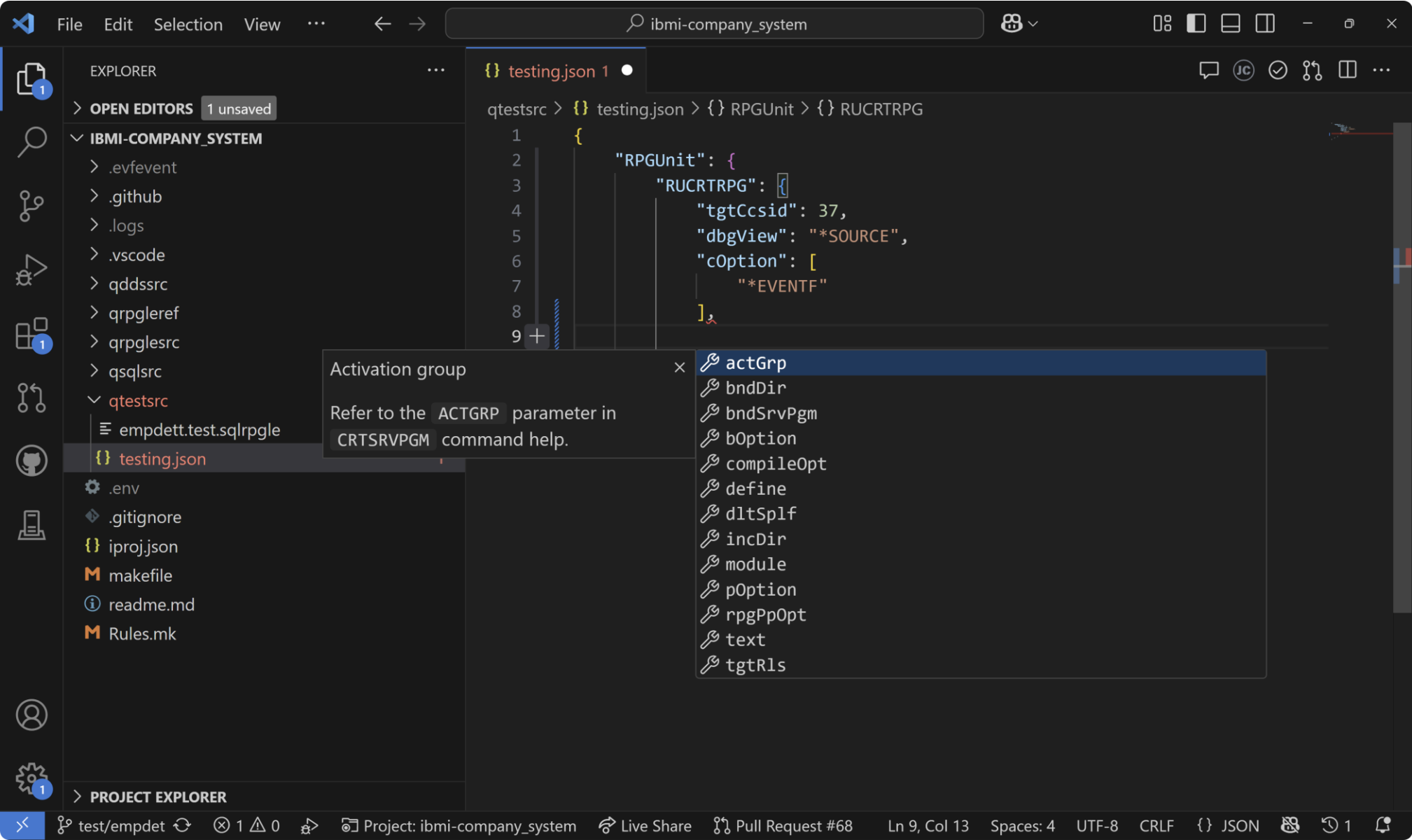
test/empdet*+ 0 0 375 SANJULA@p8adt05.rch.stglabs.ibm.com Project: ibmi-company_system Live Share Deploy New job (1) Pul

Viewing failures and errors

The screenshot displays the IBM i Testing interface within a code editor. The interface is divided into several panels:

- Left Panel (Test Explorer):** Shows a tree view of test cases. The 'examples' folder is expanded, showing 'cbl.test.cbllc' (failed) and 'rpgsqlstmf.test.sqlrpgle' (passed). The 'cbl.test.cbllc' folder is further expanded, showing 'TEST_TRUE', 'TEST_FALSE', 'TEST_FAIL0', and 'TEST_FAIL1'. The 'rpgstmf.test.rpgle' folder is also expanded, showing 'testWhatever_1' (failed) and 'testWhatever_2' (passed).
- Top Panel (Code Editor):** Displays the source code for 'rpgstmf.test.rpgle'. The code includes comments and assertions. A red error message is visible on line 231: 'Name of the test suite should be 'TEMPLATE''.
- Bottom Panel (Test Results):** Shows the results of the test run. The 'WORKSPACE' section lists the test cases and their status: 'examples (4) [Deployment Successful]', 'rpgstmf.test.rpgle (2) [Compilation Skipped]', 'rpgsqlstmf.test.sqlrpgle (2) [Compilation Skipped]', and 'cbl.test.cbllc (4) [Compilation Failed]'. The 'IBM i Testing' section lists the test cases and their status: 'TEST_TRUE cbl.test.cbllc < examples', 'TEST_FALSE cbl.test.cbllc < examples', 'TEST_FAIL0 cbl.test.cbllc < examples', 'TEST_FAIL1 cbl.test.cbllc < examples', 'testWhatever_1 Name of the test suite should be b...', 'testWhatever_1 Name of the test suite should b...', 'testWhatever_2 rpgstmf.test.rpgle < examples', 'testWhatever_1 rpgsqlstmf.test.sqlrpgle < exemp...', 'testWhatever_2 rpgsqlstmf.test.sqlrpgle < exemp...', and 'testWhatever_2 rpgstmf.test.rpgle < sub < exemp...'.

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 end-proc;
189
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

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>

Sample Projects With Unit Tests

Company System

https://github.com/IBM/ibmi-company_system

RPGUTILS

<https://github.com/IBM/vscode-ibmi-testing/tree/main/examples/rpg-utils>

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