IBM Power Week 2025

#pw2025

18 - 19 - 20 novembre 2025

IBM Innovation Studio Paris

S49 – VS Code for i : focus sur le débogage et les tests unitaires

19 novembre *14:45 - 15:45*

Philippe Bourgeois
IBM France
pbourgeois@fr.ibm.com

IBM

common FRANCE

Plan de la présentation

- 1. L'essentiel de VS Code for IBM i
- 2. Le débogage des programmes IBM i avec VS Code
- 3. Les tests unitaires IBM i avec VS Code

IBM Power Week

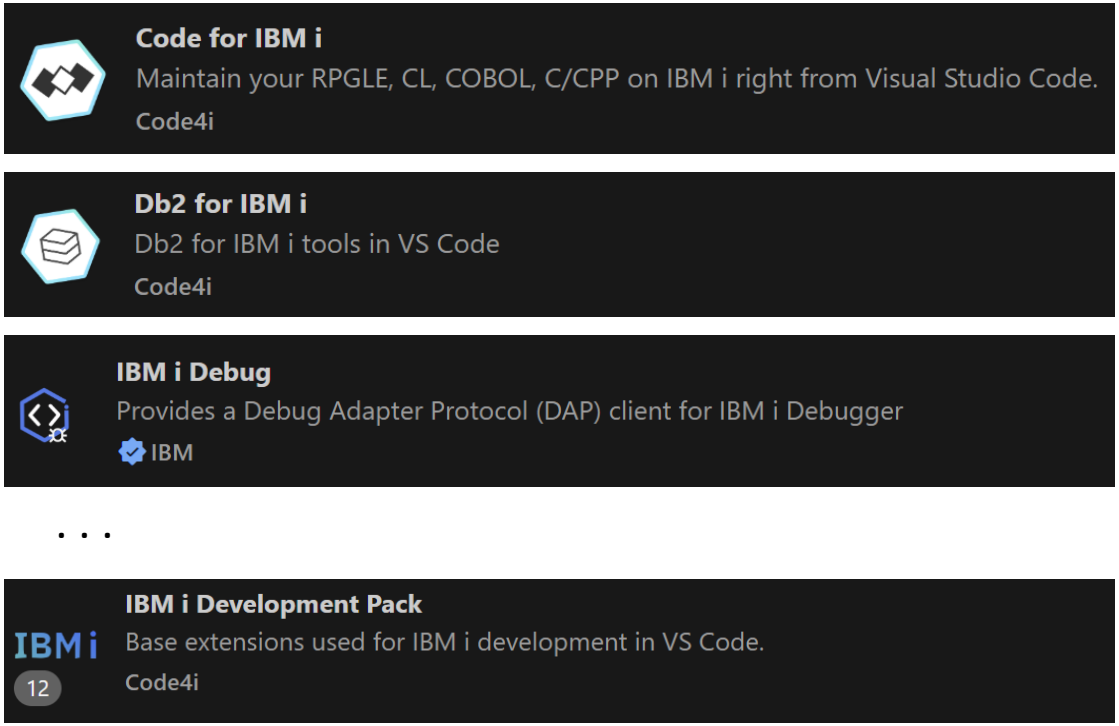
18-19-20 novembre 2025

1. L'essentiel de VS Code for IBM i

VS Code for IBM i

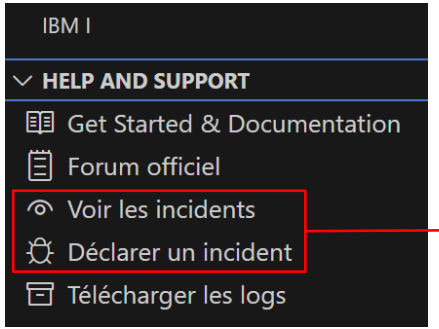
- VS Code for IBM i = VS Code + des **extensions** IBM i

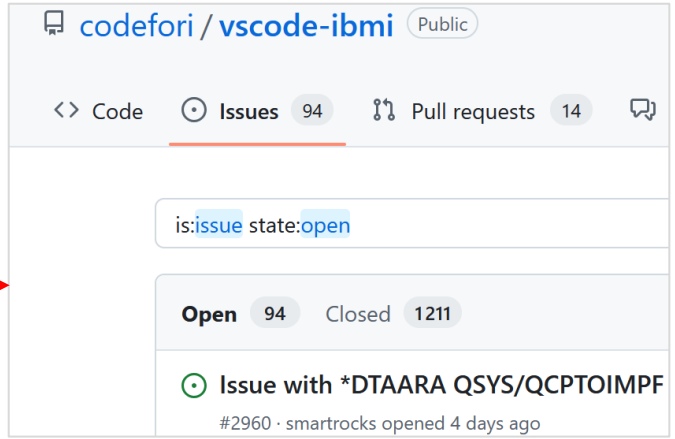
- Extensions IBM i :
  - Code for i
  - IBM i Languages
  - RPGLE
  - CL
  - COBOL
  - DB2 for i
  - IBM i Debug
  - IBM i Testing
  - IBM i Project Explorer
  - Source Orbit
  - …
  - IBM i Development Pack



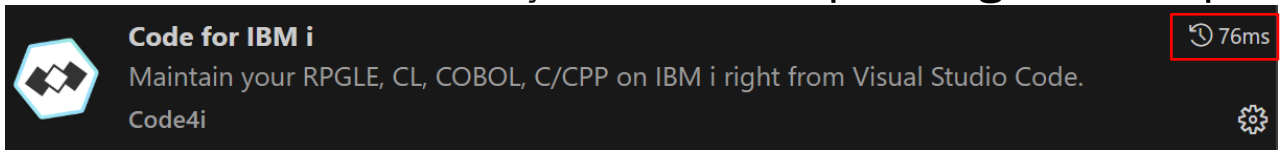
VS Code for IBM i

- Gratuit
- Support **communautaire**
  OU
- Support **IBM**
  - "IBM i extensions for VS Code Enterprise Support"
  - Disponible depuis le 31/10/2025
  - Pour toutes les extensions du "IBM i Development Pack"
  - Facturable en plus du support IBM i
  - https://www.ibm.com/docs/en/announcements/i-extensions-vs-code-enterprise-support



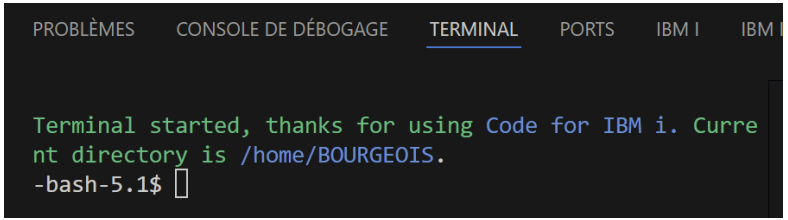


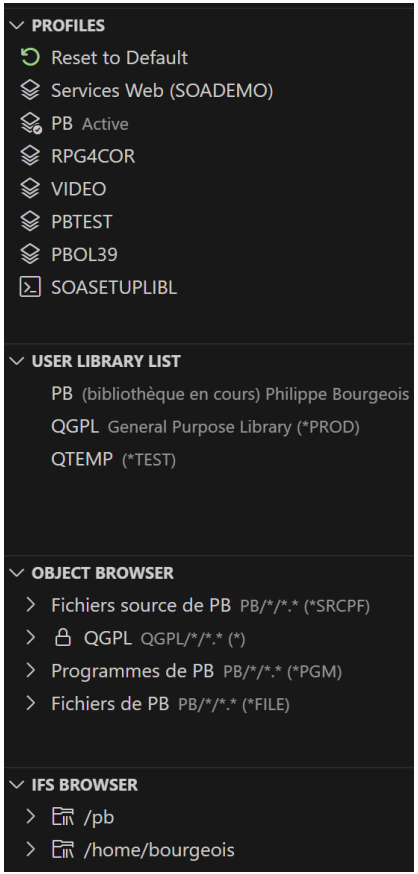
- Installation facile. Mises à jour **automatiques**, **légères** et **rapides**



VS Code for IBM i – Gestion des objets, gestion de l'IFS

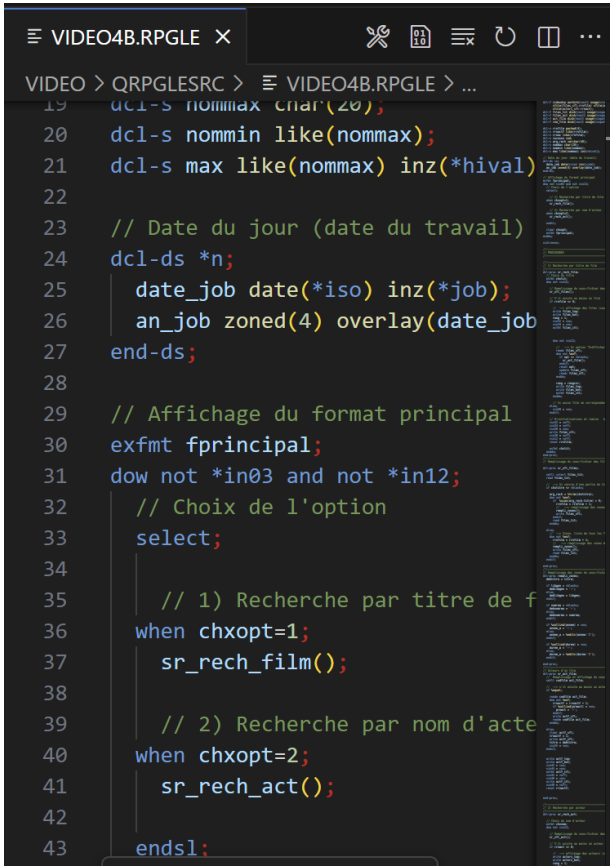
- Connexions sécurisées en SSH
- Gestion de la LIBL
- Création de filtres de bibliothèques, objets, membres
- Possibilité de filtres en lecture seule
- Création de raccourcis vers les répertoires de l'IFS
- Infobulles sur les bibliothèques, objets et repertoires IFS
- Création de différents environnement de travail (profils)
- Fonction de recherche IBM i (fichiers source et IFS)
- Terminal 5250
- Terminal PASE
- Download / Upload de fichiers





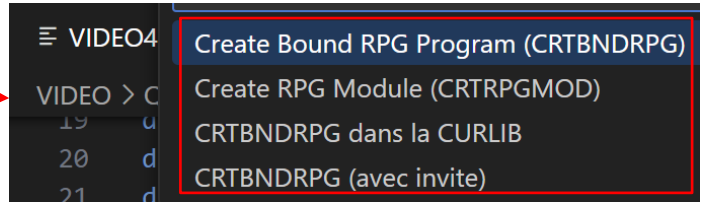
VS Code for IBM i – Edition des sources

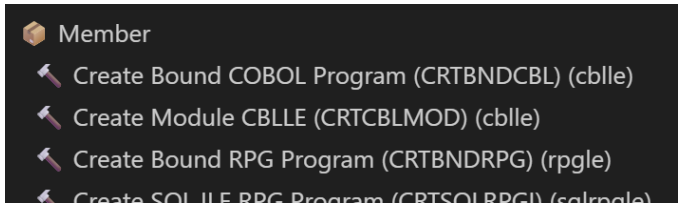
- Colorisation des sources
- Vérification de syntaxe
- Complétion de code
- Ouvertures multiples d'un même source en édition
- Minimap
- Fil d'Ariane
- Mise en commentaires rapide
- Affichage / masquage des blocs de programmation structurée
- Affichage des imbrications de blocs
- Renommage de chaînes de caractères avec contrôle avant validation
- Vue Structure
- Infobulles sur les variables, formats et fichiers
- Formatage de code
- Aperçu des, et accès aux, définitions, références et implémentations
- Refactorisation de code
- Modèles de code
- Prévisualisation des formats des DSPF
- Contrôle de la qualité du code
- Conversion RPG fixe vers RPG Full Free

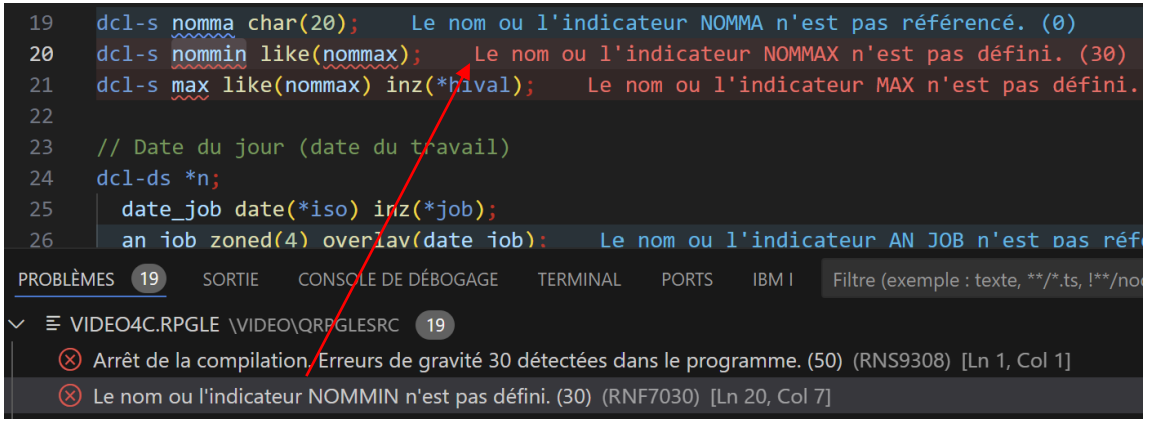


VS Code for IBM i – Compilation des sources

- Lancement par CTRL-E
- Commandes de compilation fournies
- Possibilité de se créer ses propres commandes et variables
- Affichage de la liste des erreurs :
  - Dans le source
  - Vue dédiée ("Problèmes")
- Affichage de la liste de compilation







VS Code for IBM i – Gestion de DB2 for i

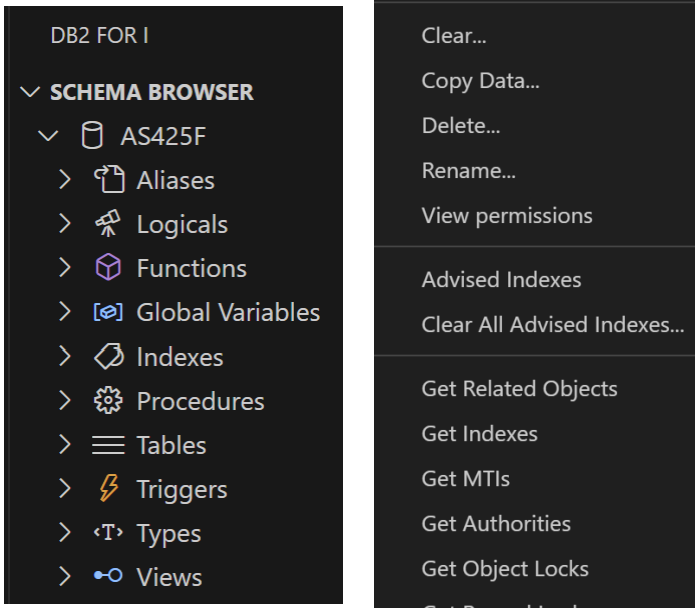
- SQL Job Manager
  - Gestion des jobs SQL et de leurs propriétés
  - Activation de SELF (SQL Error Logging Facility)

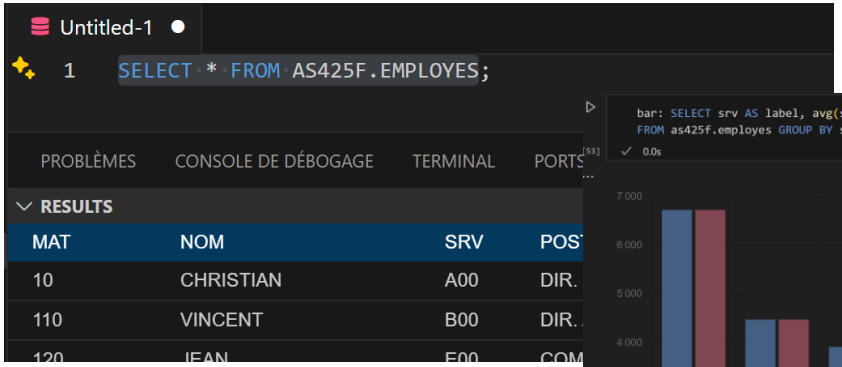
- Gestion des **objets** DB2 for i (Schema Browser)
  - Liste des objets DB2 d'un schéma
  - Génération du DDL
  - Affichage du contenu des tables
  - Liste des objets dépendants
  - Liste des index, MTI, index conseillés, droits, verrouillages sur une table et verrouillages sur les enregistrements

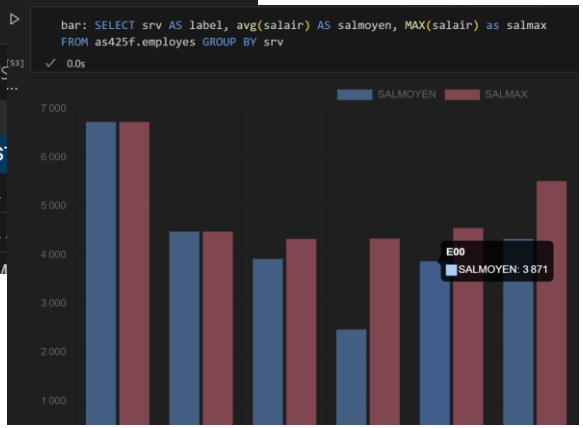
- Exécution de requêtes SQL
  - Exécution d'une ou plusieurs requêtes
  - Vérification de la syntaxe SQL
  - Affichage / modification des données
  - Sorties JSON, CSV, RPG...
  - Historique des requêtes
  - Exemples de requêtes
  - Complétion de code
  - Liage de variables
  - Exécution de commandes CL
  - Visual Explain
  - Notebooks
  - Sauvegarde des requêtes

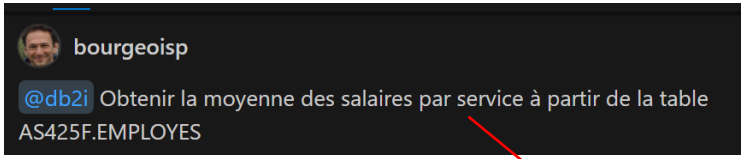
- DB2 for i Code Assistant
  - Intégration avec des assistants d'IA pour interroger DB2 for i

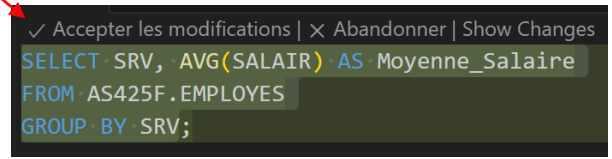
VS Code for IBM i – Gestion de DB2 for i





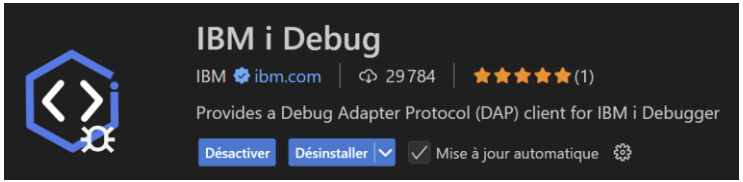


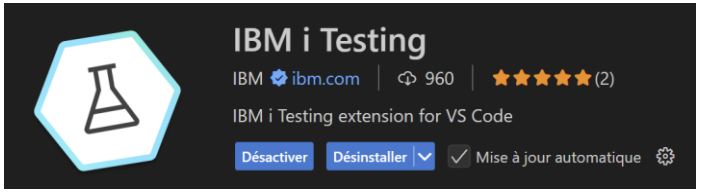




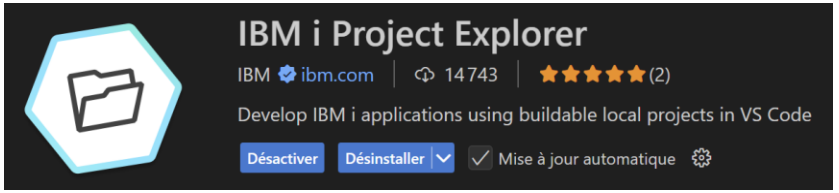
VS Code for IBM i – Autres fonctionnalités / extensions

- IBM i Debug
  - Débogage de programmes IBM i
- IBM i Testing
  - Tests unitaires IBM i

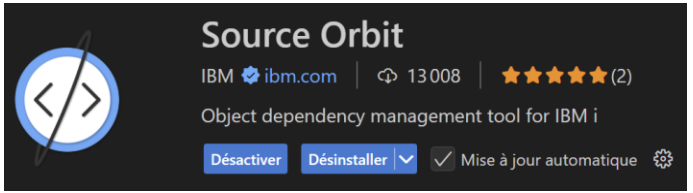




- IBM i Project Explorer
  - Développement IBM i avec des projets locaux



- Source Orbit
  - Gestion des dépendances IBM i applicatives



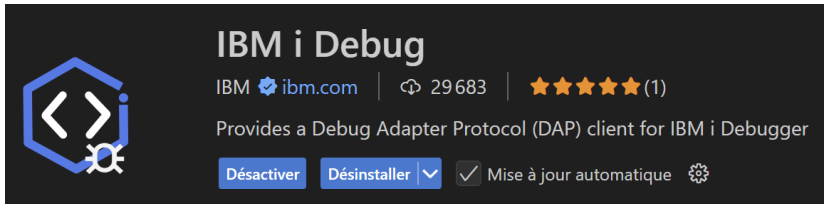
IBM Power Week

18-19-20 novembre 2025

2. Le débogage des programmes IBM i avec VS Code

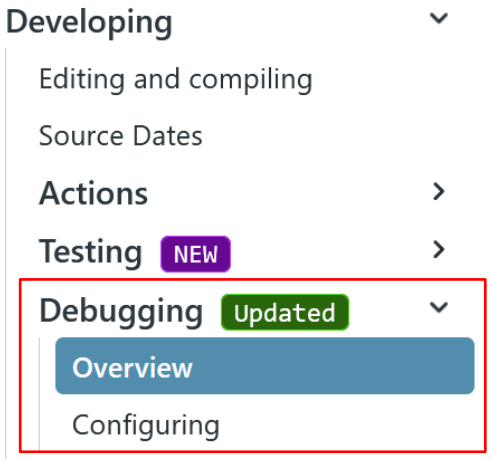
Débogueur IBM i de VS Code – Fonctionnalités

- Débogage de programmes RPG (OPM et ILE), COBOL, C/C++, CL
- Débogage pas à pas dans le source
  - STEP OVER, STEP INTO, STEP OUT
- Points d'arrêt simples ou conditionnés
  - Peuvent être ajoutés avant ou pendant le débogage
- Affichage et modification de variables
- Monitoring de variables
- Affichage de la pile d'appel des programmes
- Possibilité de mettre à jour les "fichiers de production" (~ UPDPRD(*YES))
- Points d'entrée de service
  - Permet de mettre le programme en "attente de débogage"
  - Quand le programme est exécuté (quel que soit le job), la session de débogage est lancée dans VS Code
  - Avec possibilité de choisir le profil utilisateur, une condition d'exécution et de rafraîchir le point d'entrée de service après recompilation

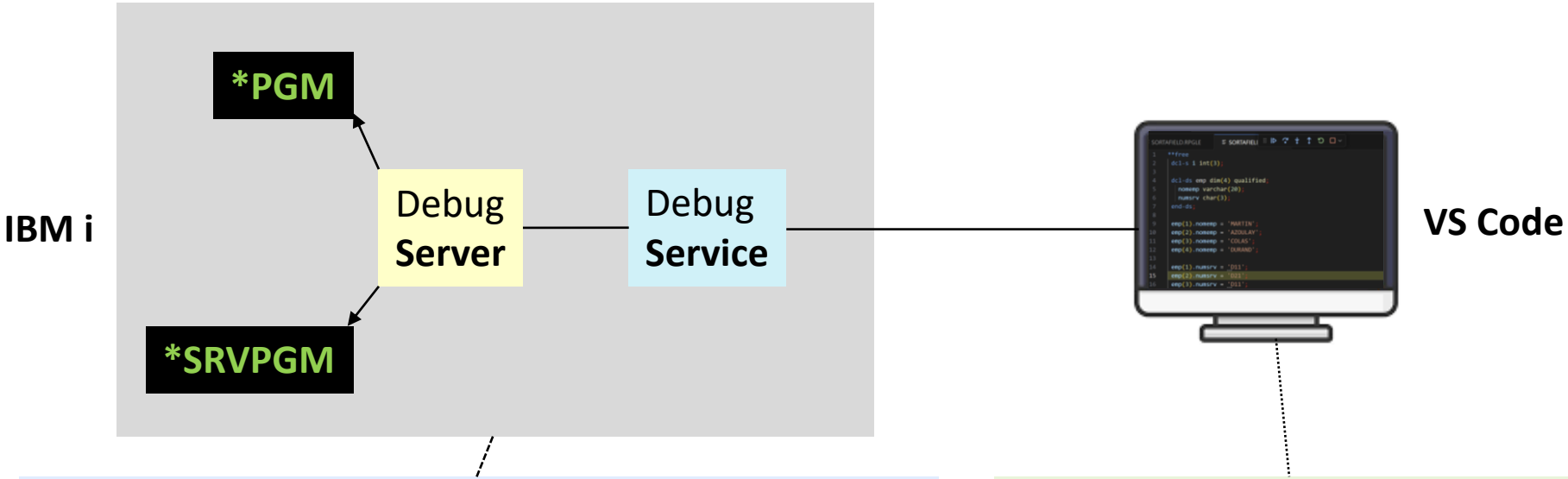


Débogueur IBM i de VS Code – Versions

- Au 19/11/2025 la version du débogueur est la **3.0.1** :
  - https://codefori.github.io/docs/developing/debug/overview/
  - https://codefori.github.io/docs/developing/debug/configure/



Débogueur IBM i – Composants et prérequis

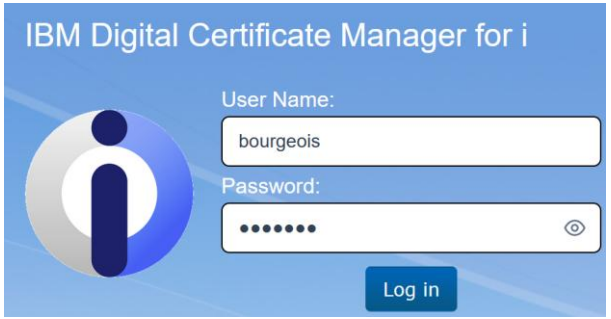


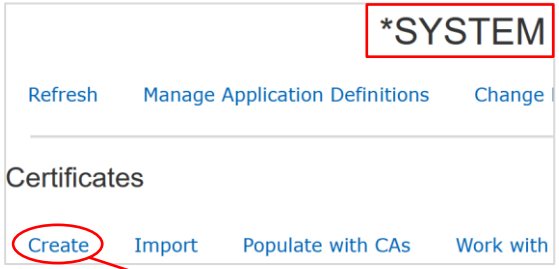
- 5770-JV1 Option 19 (Java 11) ou Option 20 (Java 17)
- 5770-WDS Option 60
- PTFs (https://codefori.github.io/docs/developing/debug/configure/)
- 1 certificat dans le magasin *SYSTEM de DCM et affecté au Debug Service
- Démarrer le Debug Server
- Démarrer le Debug Service

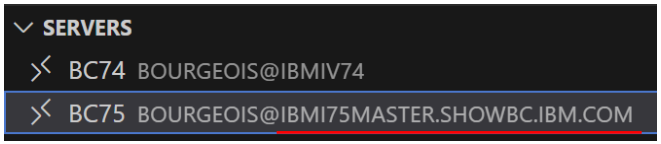
- Extension "IBM i Debug"
- BASH comme shell par défaut
- 1 connexion IBM i avec un nom de hôte et non pas une adresse IP
- Importer le certificat créé sur le serveur

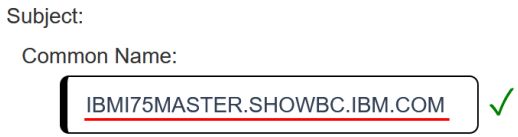
Débogueur IBM i – Mise en oeuvre

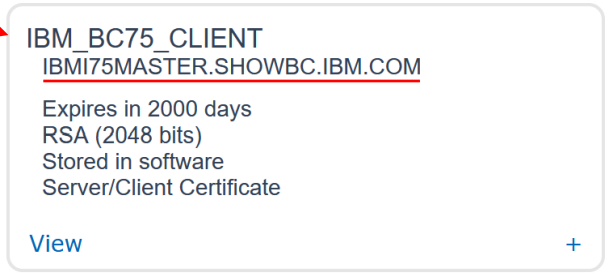
- Côté IBM i (1/2)
  - Le certificat créé dans DCM doit avoir comme CN (Common Name) le nom du hôte défini dans la connexion IBM i de VS Code





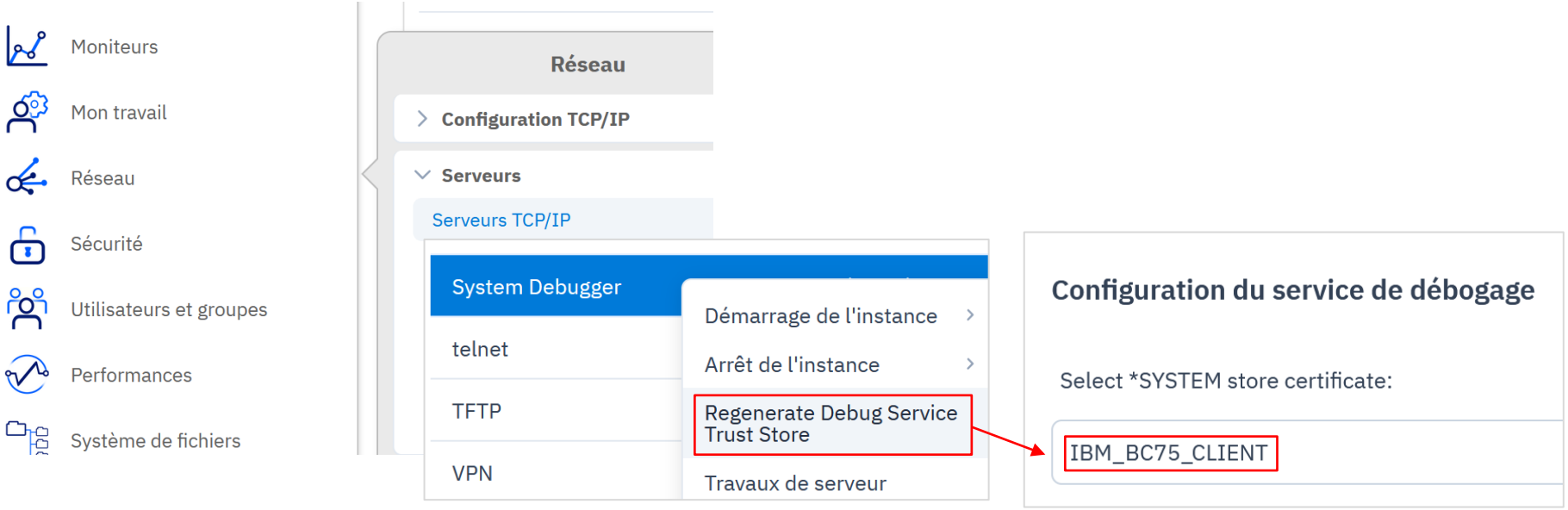






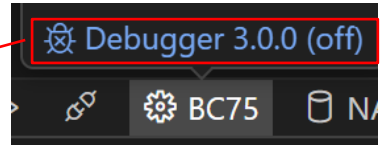
Débogueur IBM i – Mise en oeuvre

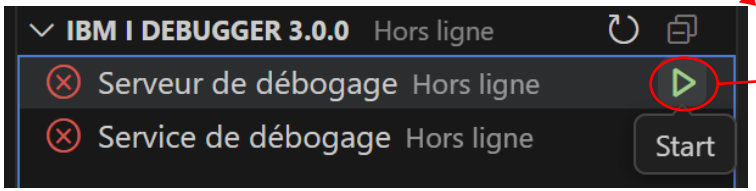
- Côté IBM i (2/2)
  - Le certificat doit être affecté au Debug Service

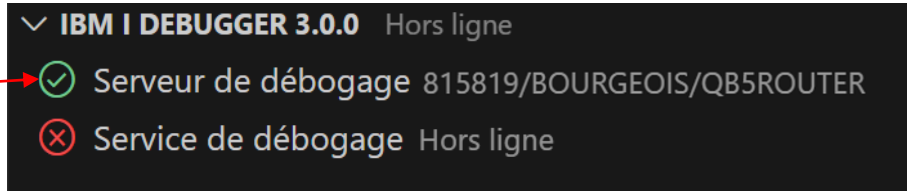


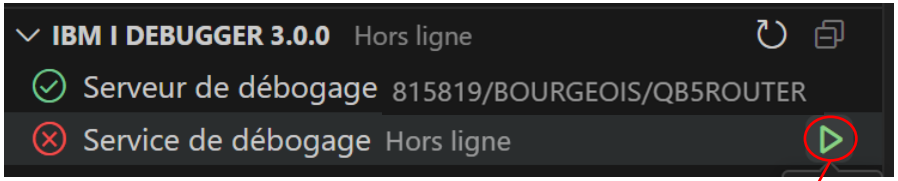
Débogueur IBM i – Mise en oeuvre

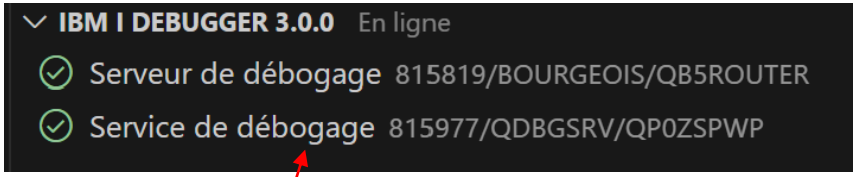
- Côté VS Code

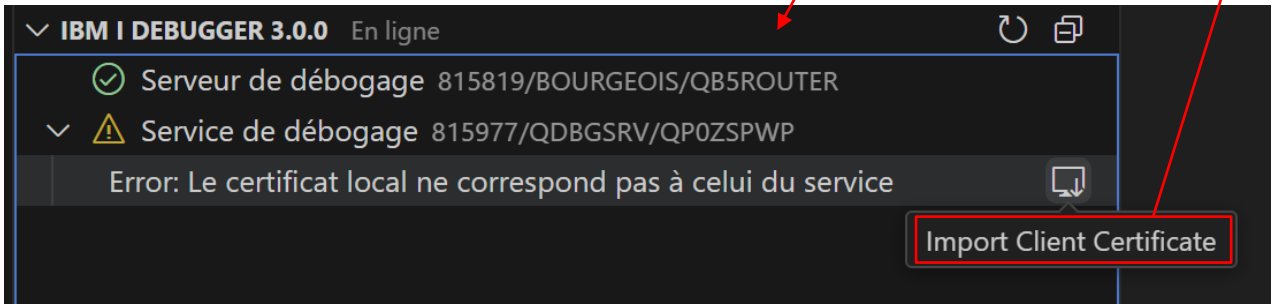












Exemple de débogage IBM i

- Les programmes

PGM1.CLLE

```
PB > QCLLESRC > PGM1.CLLE > ...
1   PGM        PARM(&ANNEE)
2   DCL        VAR(&ANNEE) TYPE(*DEC) LEN(4 0)
3   SBMJOB     CMD(CALL PGM(PGM2) PARM((&ANNEE (*DEC 4 0))))
4   ENDPGM
```

```
CALL PGM(PGM1)
```

```
PARM((2005 (*DEC 4 0)))
```

```
DSPLY  Année : 2005
DSPLY  D21 AZOULAY
DSPLY  D11 COLAS
DSPLY  D31 DURAND
DSPLY  D11 MARTIN
DSPLY  D11 COLAS
DSPLY  D11 MARTIN
DSPLY  D21 AZOULAY
DSPLY  D31 DURAND
```

PGM2.RPGLE

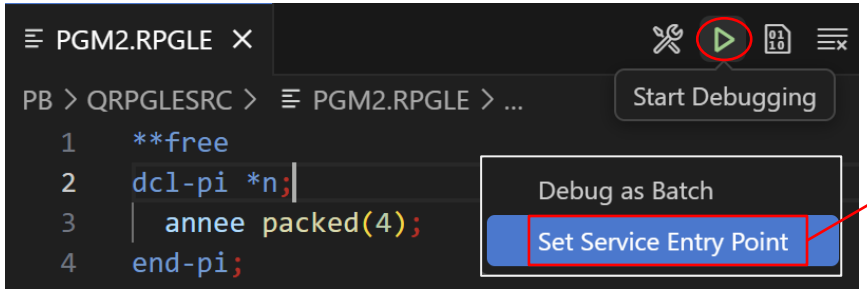
```
PB > QRPGLESRC > PGM2.RPGLE > ...
1   **free
2   dcl-pi *n;
3     annee packed(4);
4   end-pi;
5
6   dcl-s cpt int(3);
7
8   dcl-ds emp dim(4) qualified;
9     nomemp varchar(20);
10    numsrv char(3);
11  end-ds;
12
13  dsply ('Année : ' + %char(annee));
14
15  emp(1).nomemp = 'MARTIN';
```

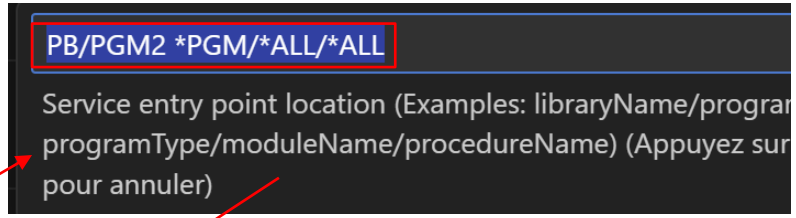
. . .

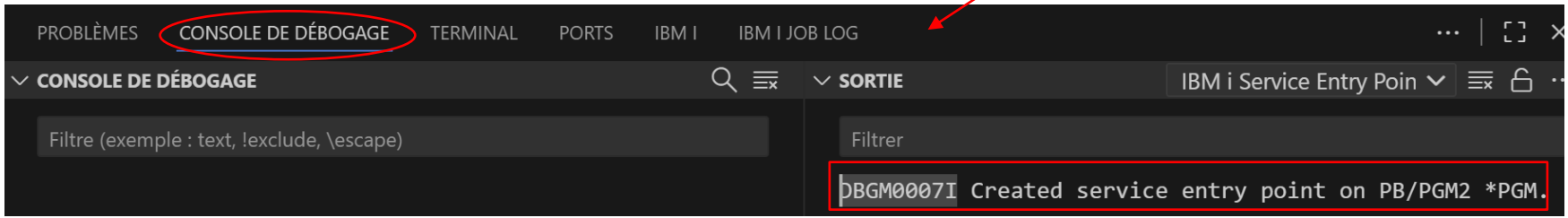
```
28  for cpt = 1 to %elem(emp);
29    dsply (emp(cpt).numsrv + '... ' + emp(cpt).nomemp);
30  endfor;
31
32  sorta(a) emp %fields(numsrv:nomemp); // Tri par numéro de
33
34  for cpt = 1 to %elem(emp);
35    dsply (emp(cpt).numsrv + '... ' + emp(cpt).nomemp);
36  endfor;
37
38  *inlr = *on;
```

Exemple de débogage IBM i

- On veut déboguer PGM2 et uniquement si le paramètre ANNEE est égal à 2025 :
  - On va ajouter un point d'entrée de service sur PGM2
  - Puis ajouter une condition sur le point d'entrée de service

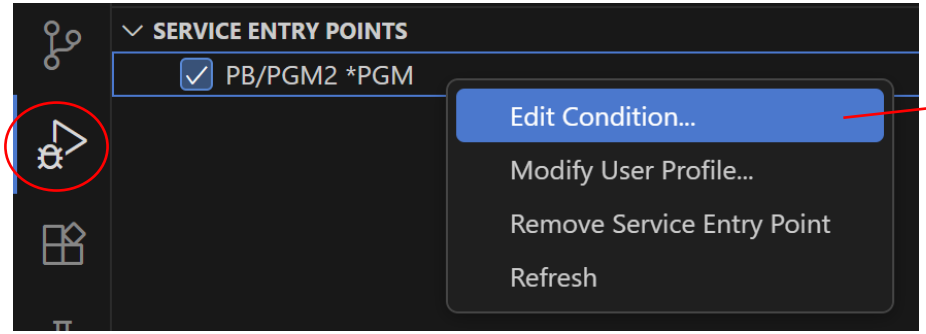


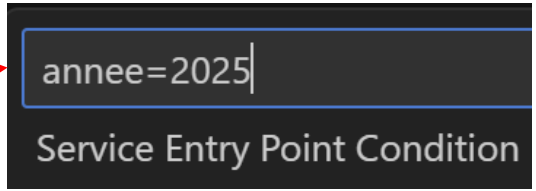


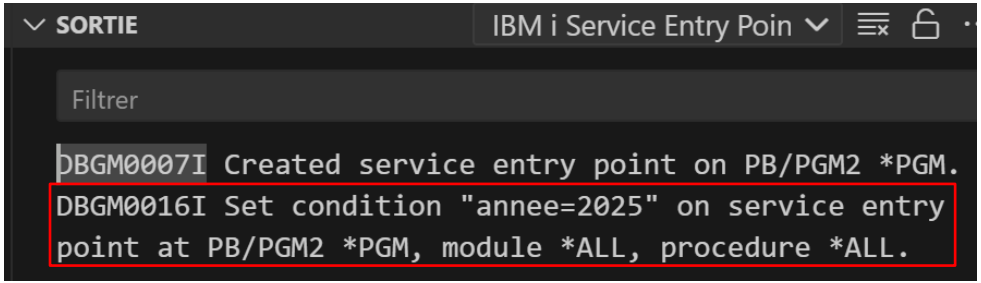


Exemple de débogage IBM i

- On veut déboguer PGM2 et uniquement si le paramètre ANNEE est égal à 2025 :
  - On va ajouter un point d'entrée de service sur PGM2
  - Puis ajouter une condition sur le point d'entrée de service





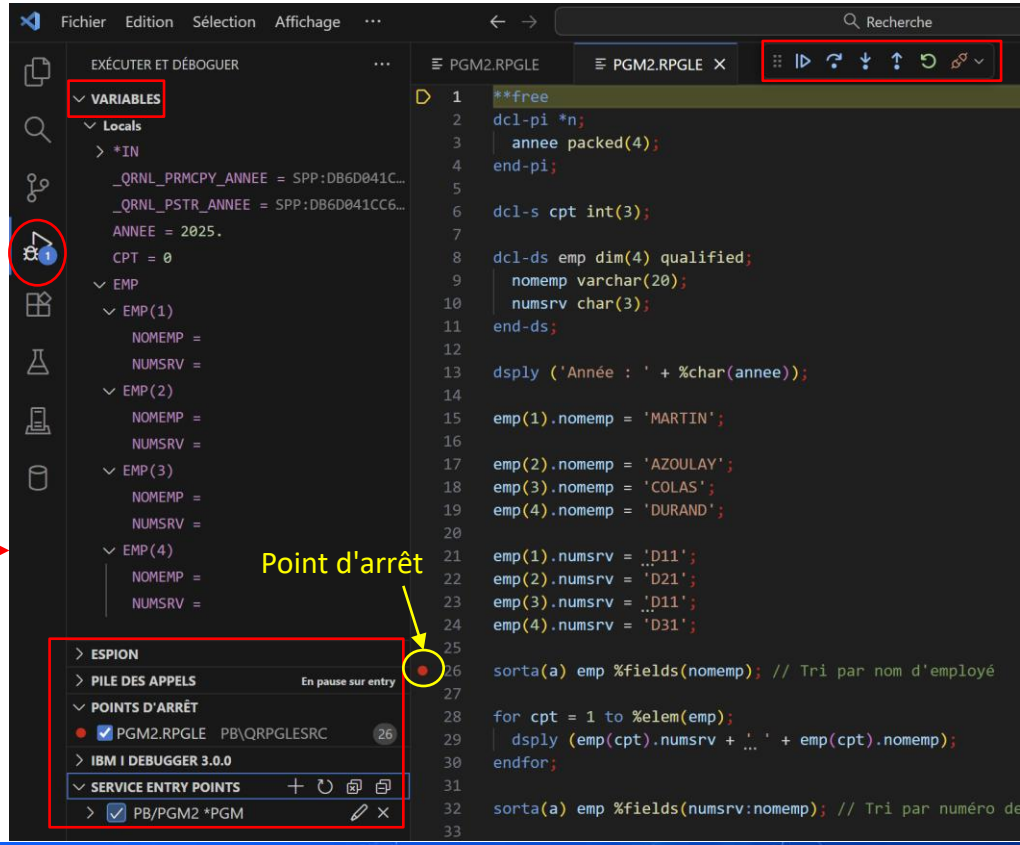


Exemple de débogage IBM i

- Exécution du programme PGM1 avec ANNEE = 2025
  - PGM1 fait un SBMJOB du CALL de PGM2
  - La session de débogage démarre pour PGM2

```
CALL PGM(PGM1)
```

```
PARM((2025 (*DEC 4 0)))
```



Débogueur IBM i – Fonctionnalités

Démarrer / reprendre l'exécution du programme (F5)

Pas à pas – STEP OVER (F10)

Pas à pas – STEP INTO (F11)

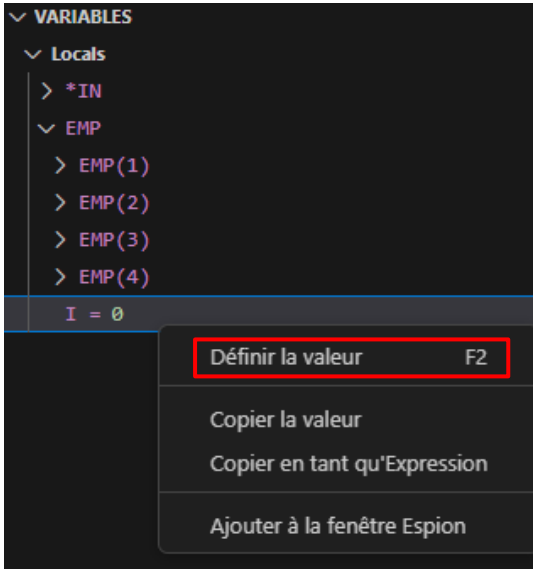
Pas à pas – STEP OUT (MAJ+F11)

Arrêter le débogage (MAJ+F5)

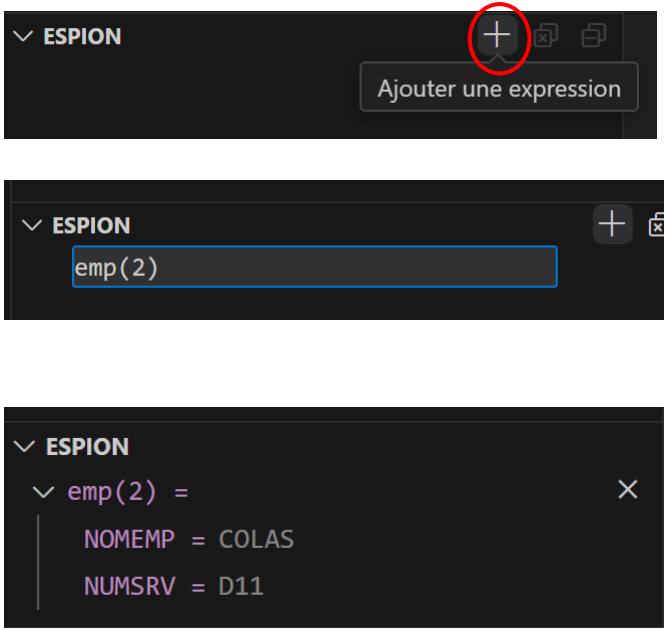
Redémarrer (CTLR+MAJ+F5)

Débogueur IBM i – Fonctionnalités

- **Modification** de variables



- **Monitoring** de variables



IBM Power Week

18-19-20 novembre 2025

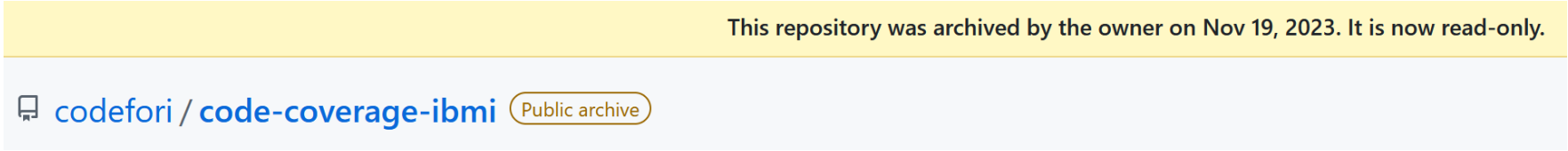
3. Les tests unitaires IBM i avec VS Code

Tests unitaires

- Tests unitaires
  - Tests d'unités de code isolées :
    - Logique métier ou technique (règles métier, calculs, formatage de données…)
  - Sur l'IBM i :
    - Test de procédures ou de programmes "métier" (non 5250)
    - Pas de test de programmes complets, qu'ils soient interactifs ou batch
      - Pour tester ce type de programme il existe des solutions comme Arcad Verifier (Arcad Software), ReplicTest (Polverini & Partners), X-DataTest (Fresche Solutions), etc.
    - On va développer (ou faire générer si VS Code) des procédures de test qui vont tester les procédures métier
  - Couverture de code
    - Donne le pourcentage du code qui a été exécuté dans la procédure de test
    - Permet de connaître l'efficacité des procédures de test
      - Plus le % est élevé, plus la procédure de test est efficace

Remarque : couverture de code de programmes IBM i

- L'extension "IBM i Code Coverage" pour VS Code, qui permet d'obtenir le taux de couverture de code d'un programme, n'est actuellement pas disponible

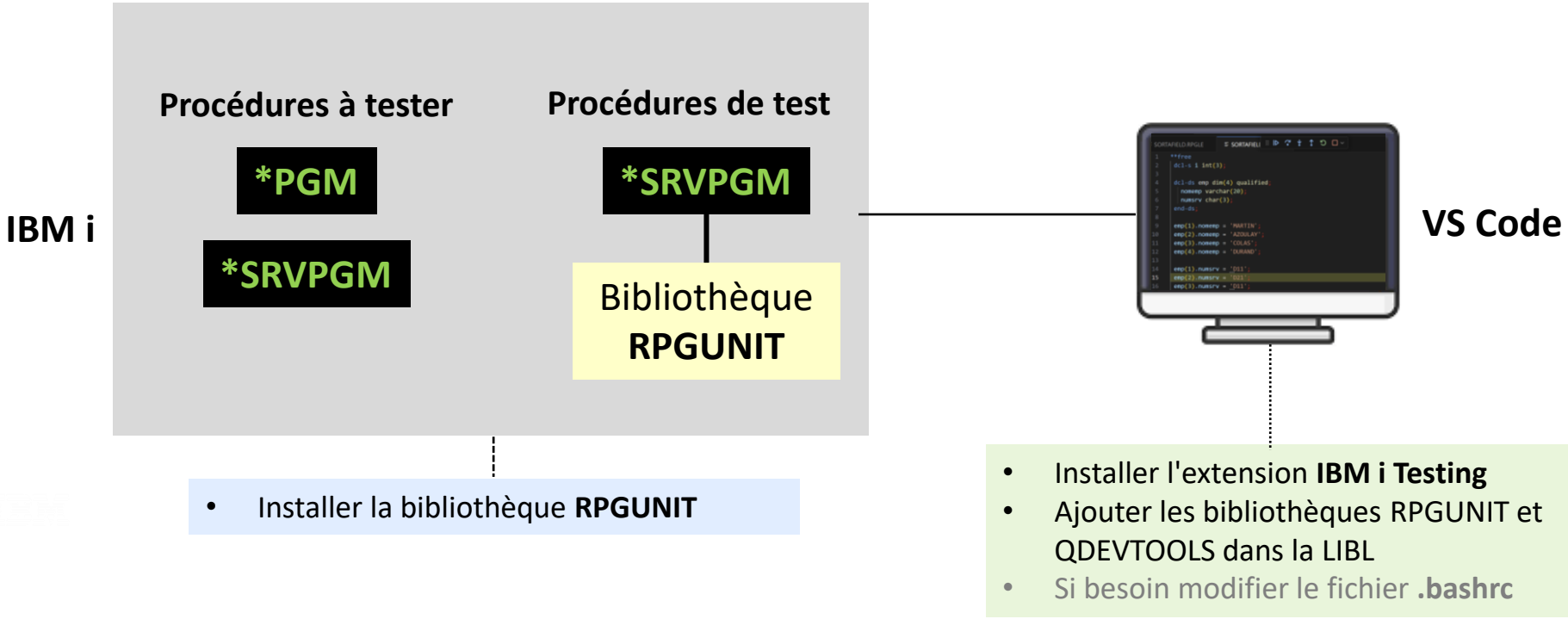


- La fonction de couverture de code proposée dans les tests unitaires permet de visualiser le taux de couverture de la procédure de test, pas de la procédure testée

Tests unitaires IBM i avec VS Code – Fonctionnalités

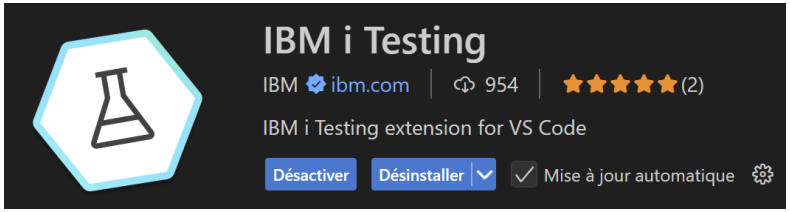
- L'extension **IBM i Testing** dans VS Code va permettre :
  - De générer automatiquement des **stubs** : génération des procédures de test
  - De **configurer** les tests (paramètres de compilation et d'exécution)
  - D'**exécuter** un test individuel ou une suite de tests
  - De **visualiser** les résultats des tests
  - De générer le résultat de la **couverture de code,** aux niveaux procédure et ligne
  - D'**automatiser** les tests pour une intégration dans un pipeline **CI/CD**

Tests unitaires IBM i avec VS Code – Composants et prérequis



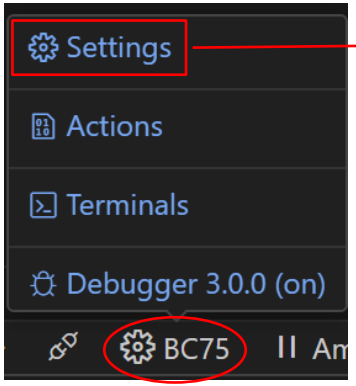
Tests unitaires IBM i avec VS Code – Composants et prérequis

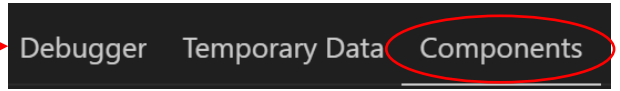
- Dans VS Code
  - Extension **IBM i Testing**

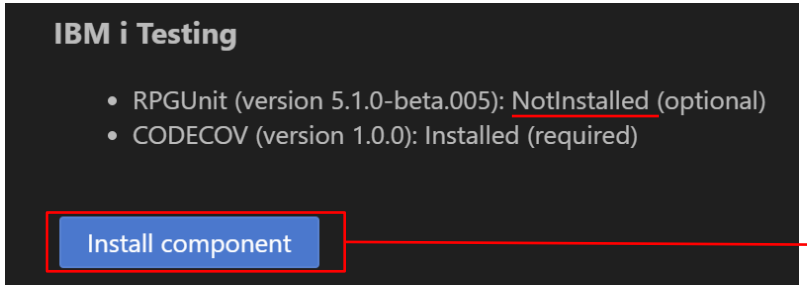


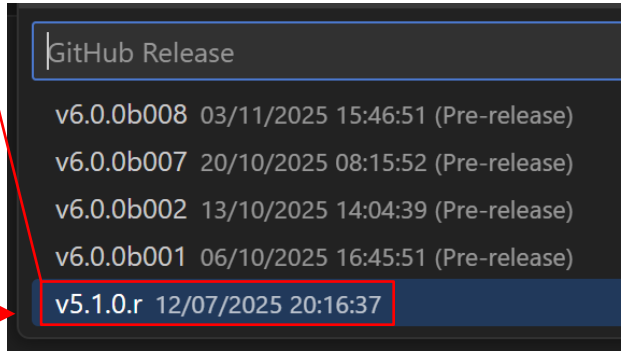
- Sur l'IBM i
  - Bibliothèque **RPGUNIT**
  - Peut être installée depuis VS Code

```
2025-11-12 14:41:01.541 [info] RPGUnit v5.1.0.r installed
successfully into RPGUNIT.LIB
```



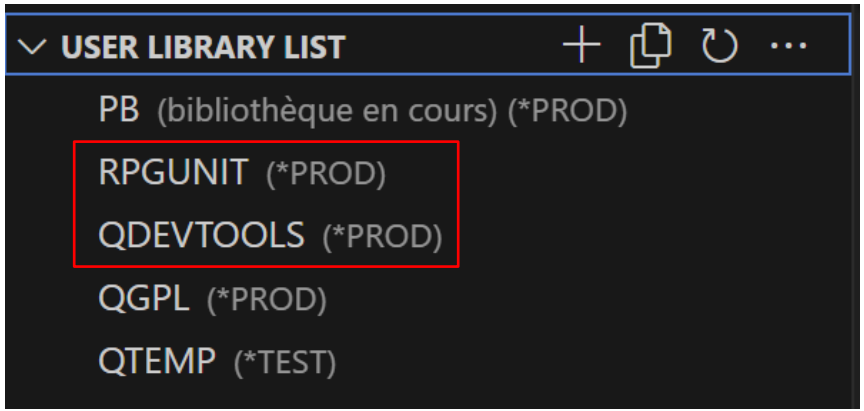






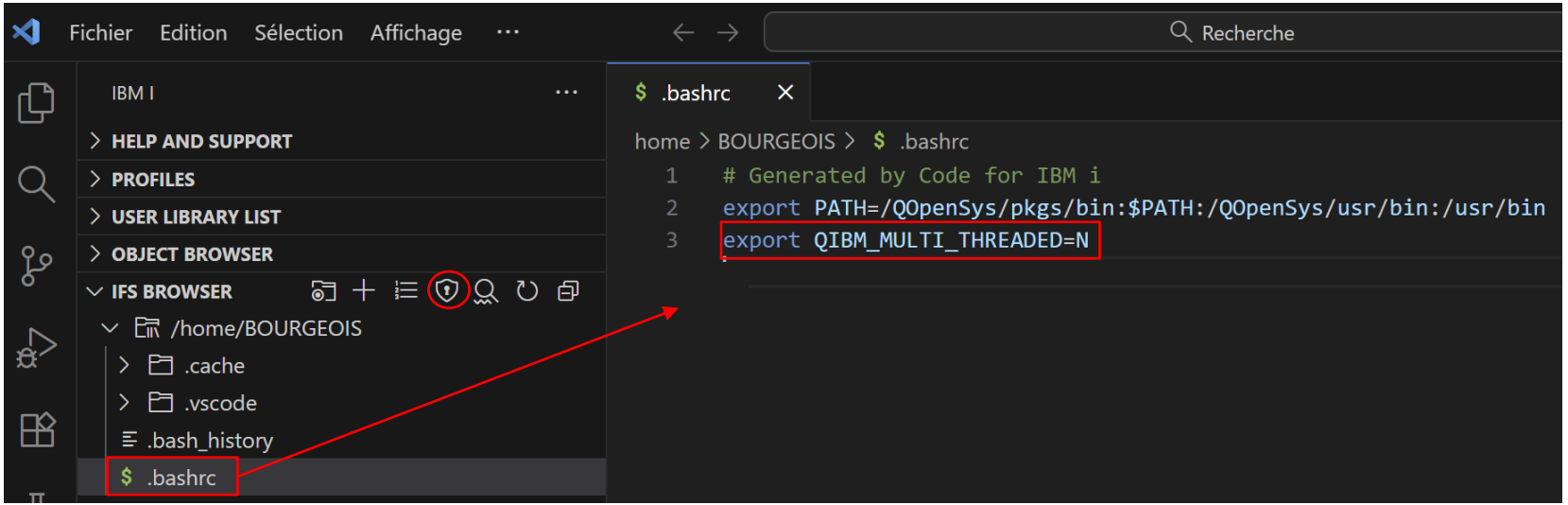
Tests unitaires IBM i avec VS Code – Composants et prérequis

- Ajouter les bibliothèques **RPGUNIT** et **QDEVTOOLS** dans la **LIBL**



Tests unitaires IBM i avec VS Code – Composants et prérequis

- **Optionnel** : si le message d'erreur *"Commande STRDBDG non admise dans les travaux admettant des unités d'exécution"* apparaît lors de l'exécution des tests (plus précisément la couverture de code), il faut modifier le fichier **.bashrc**

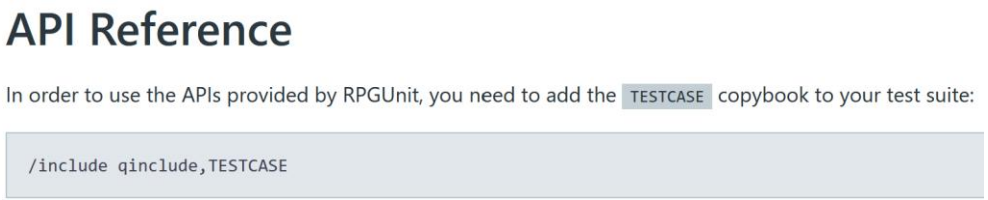


RPGUnit + extension IBM i Testing – Vocabulaire

- Stub
  - Le source qui contient la ou les procédure(s) de test
- Test Case
  - Cas de test individuel
  - Une procédure de test, dont le nom commence obligatoirement par **test,** permettant de tester une procédure ou un programme métier
    - Par exemple si la procédure à tester s'appelle **isPalindrome**, la procédure de test pourra s'appeler **test**isPalindrome, **test**_isPalindrome, etc.
- Test Suite
  - Suite entière de tests
  - Ensemble de cas de tests

RPGUnit + extension IBM i Testing – Stubs

- Les procédures de test (cas et suites de test) peuvent être :
  - Développées **manuellement**
    - En s'appuyant sur l'API Reference de RPGUnit
      - https://codefori.github.io/docs/developing/testing/writing/



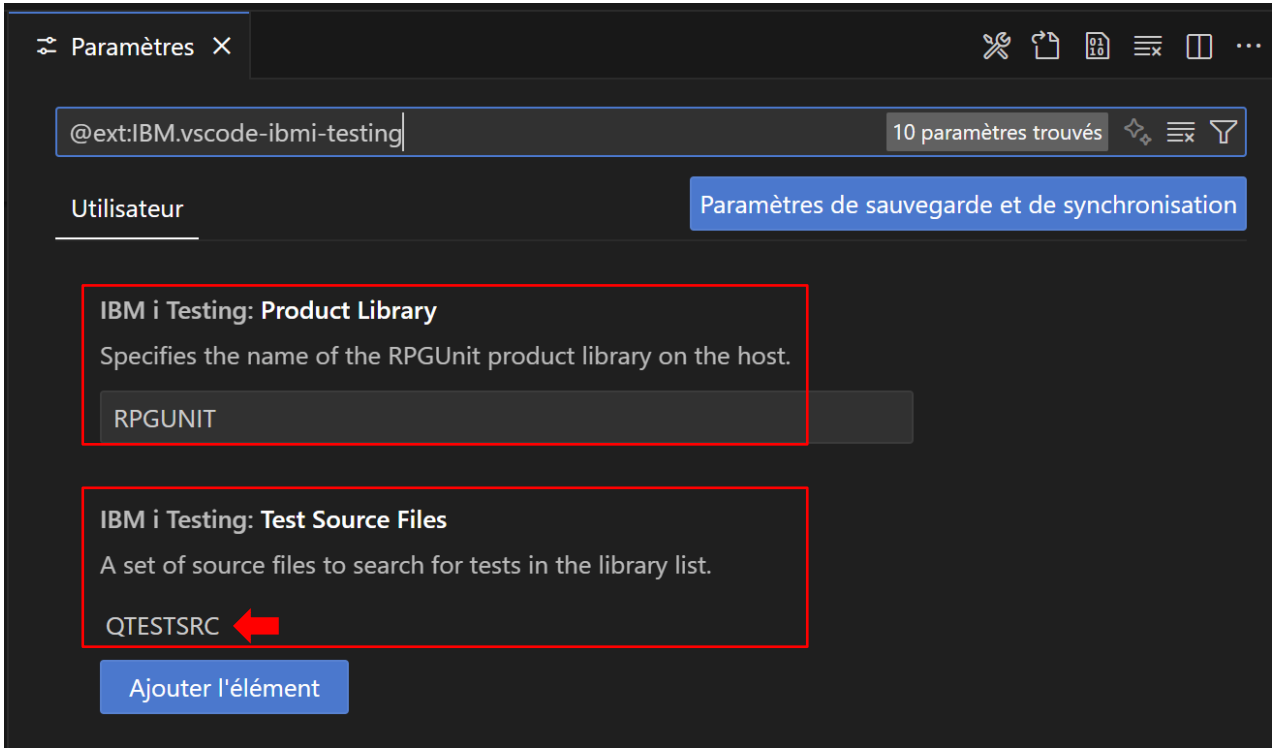
API Reference

In order to use the APIs provided by RPGUnit, you need to add the `TESTCASE` copybook to your test suite:

```
/include qinclude,TESTCASE
```

  - Générées **automatiquement** à partir de VS Code
    - Création de **stubs** à partir des procédures à tester
    - Modification possible en s'appuyant sur l'API Reference de RPGUnit

Extension "IBM i Testing" de VS Code – Paramètres



Extension "IBM i Testing" de VS Code – Paramètres

IBM i Testing: Test Stub Preferences

Preferences for how test stubs should be generated:

- `Show Test Stub Preview`: Controls whether to show a preview of the test stub before adding it to the file or source member. This refactoring preview can also be used to selectively insert portions of the stub.
- `Test Source File`: The default source file to generate new test members into.
- `Test Source Directory`: The default directory to generate new test files into.
- `Prompt For Test Name`: Controls whether to prompt for the name of the test including where it is located. The default source file is set using the `Test Source File` preference, the default directory is set using the `Test Source Directory` preference, and the test file or member will be named according to the rules described here.
- `Add Control Options and Directives`: Controls whether to add control options (`ctl-opt nomain ccsidcvt(*excp) ccsid(*char : *jobrun);`) and directives (`**free`) for new test files or members.
- `Add Includes`: Controls whether to generate relevant includes.
- `Add Prototypes`: Controls whether to generate a prototype for the procedure being tested (if it does not already exist).
- `Add Stub Comments`: Controls whether to add comments to the test stub to distinguish inputs, actual results, expected results, and assertions.

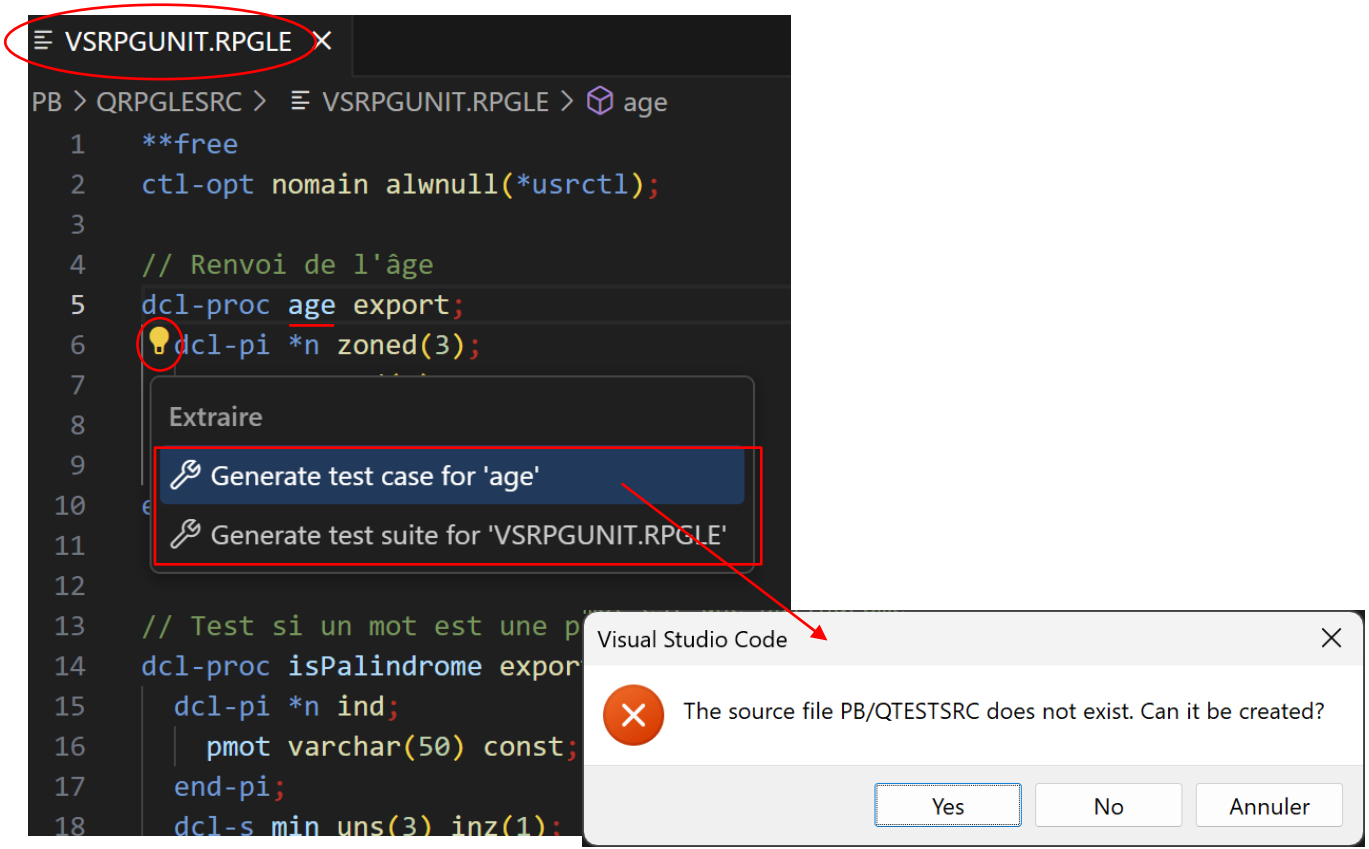
Élément	Valeur
Show Test Stub Preview	true
Test Source File	QTESTSRC
Test Source Directory	qtestsrc
Prompt For Test Name	false
Add Control Options and Directives	true
Add Includes	true
Add Prototypes	true
Add Stub Comments	false

Prompt For Test Name	true

. . .

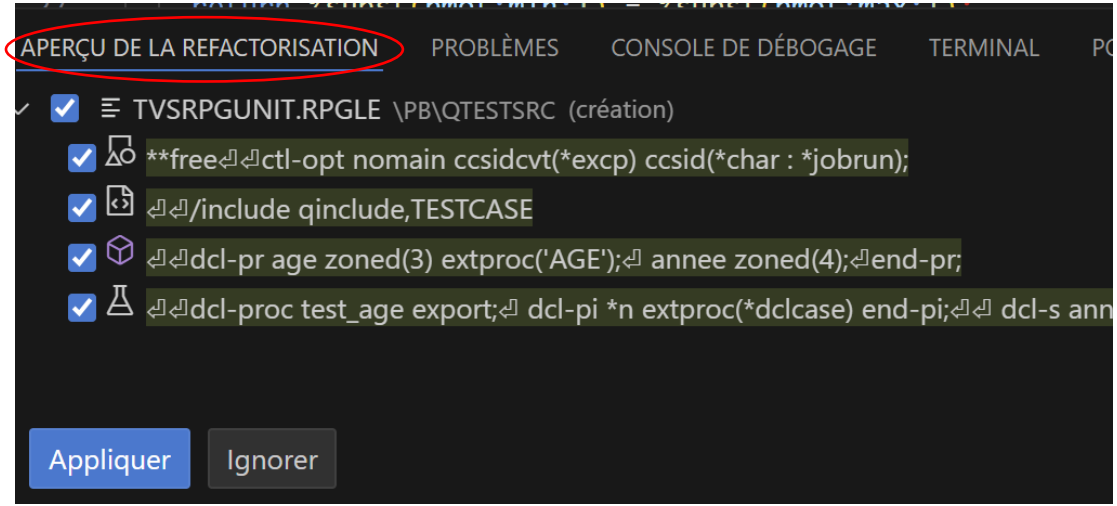
Exemple 1 – Génération d'un cas de test

- On souhaite tester la procédure AGE qui se trouve dans le programme de service VSRPGUNIT
  - La procédure à tester doit être indiquée en EXPORT



Exemple 1 – Génération d'un cas de test

- Mode **PREVIEW** (aperçu de la factorisation) disponible avant de générer le code



Élément	Valeur
Show Test Stub Preview	true
Test Source File	QTESTSRC
Test Source Directory	qtestsrc
Prompt For Test Name	true
Add Control Options and Directives	true
Add Includes	true
Add Prototypes	true
Add Stub Comments	false

Exemple 1 – Génération d'un cas de test

- Résultat :
  - Génération d'un membre source de test intitulé Txxx (xxx : nom du membre d'origine)
  - Ajout des directives /include nécessaires
  - Ajout d'une procédure de test intitulée **test_**yyy (yyy : nom de la procédure à tester)
    - Avec variables en entrée, appel de la procédure à tester, definition de ce qui est attendu…

VSRPGUNIT.RPGLE | TVSRPGUNIT.RPGLE

PB > QTESTSRC > TVSRPGUNIT.RPGLE > ...

```
1   **free
2
3   ctl-opt nomain ccsidcvt(*excp) ccsid(*char : *jobrun);
4
5   /include qinclude,TESTCASE
6
7   dcl-pr age zoned(3) extproc('AGE');
8     annee zoned(4);
9   end-pr;
10
11  dcl-proc test_age export;
12    dcl-pi *n extproc(*dclcase) end-pi;
13
14    dcl-s annee zoned(4);
15    dcl-s actual zoned(3);
16    dcl-s expected zoned(3);
17
18    annee = 0.0;
19
20    actual = age(annee);
21
22    expected = 0.0;
23
24    assert(expected = actual : 'actual');
25  end-proc;
```

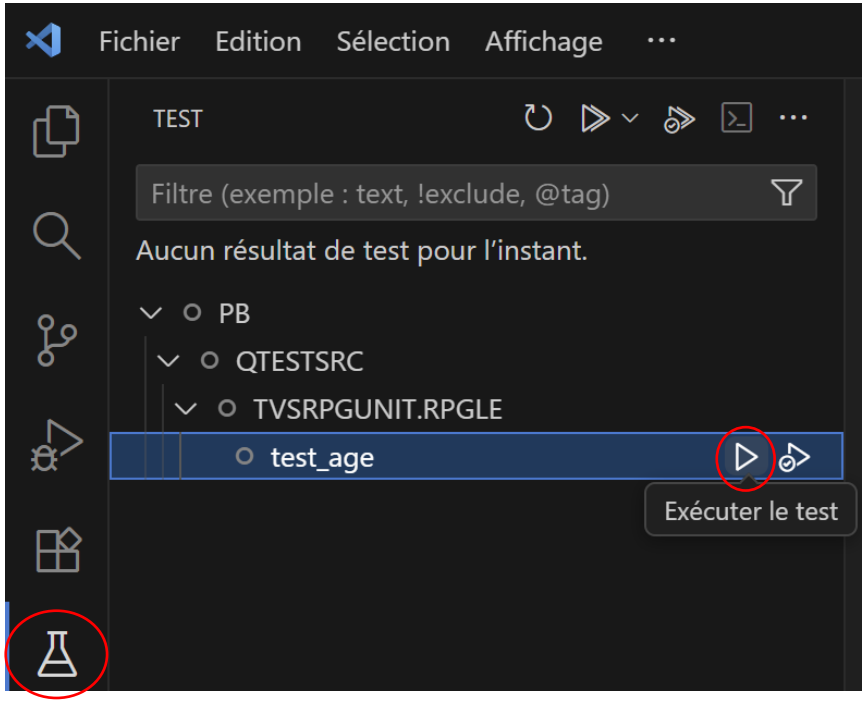
Exemple 1 – Exécution d'un cas de test

VSRPGUNIT.RPGLE | TVSRPGUNIT.RPGLE

PB > QTESTSRC > TVSRPGUNIT.RPGLE > ...

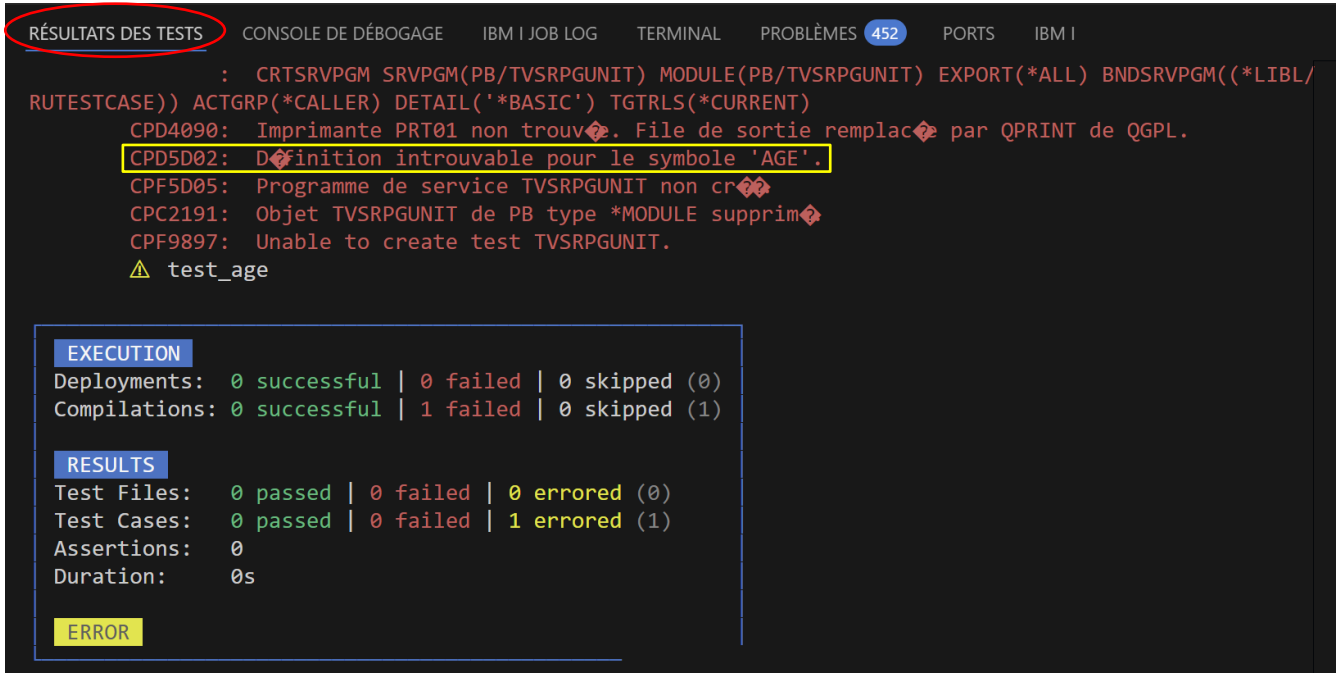
```
1   **free
2
3   ctl-opt nomain ccsidcvt(*excp) ccsid(*char : *jobrun);
4
5   /include qinclude,TESTCASE
6
7   dcl-pr age zoned(3) extproc('AGE');
8     annee zoned(4);
9   end-pr;
10
11  dcl-proc test_age export;
12    dcl-pi *n extproc(*dclcase) end-pi;
13
14    dcl-s annee zoned(4);
15    dcl-s actual zoned(3);
16    dcl-s expected zoned(3);
17
18    annee = 1966;
19
20    actual = age(annee);
21
22    expected = 59;
23
24    assert(expected = actual : 'actual');
25  end-proc;
```

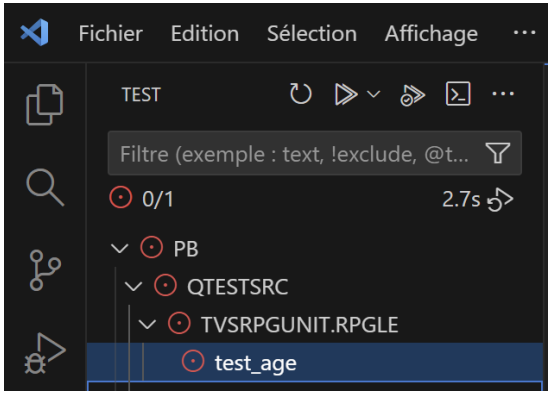
Détection automatique des stubs qui se trouvent dans les fichiers **QTESTSRC** de la LIBL et des procédures de test dont le nom commence par **test**



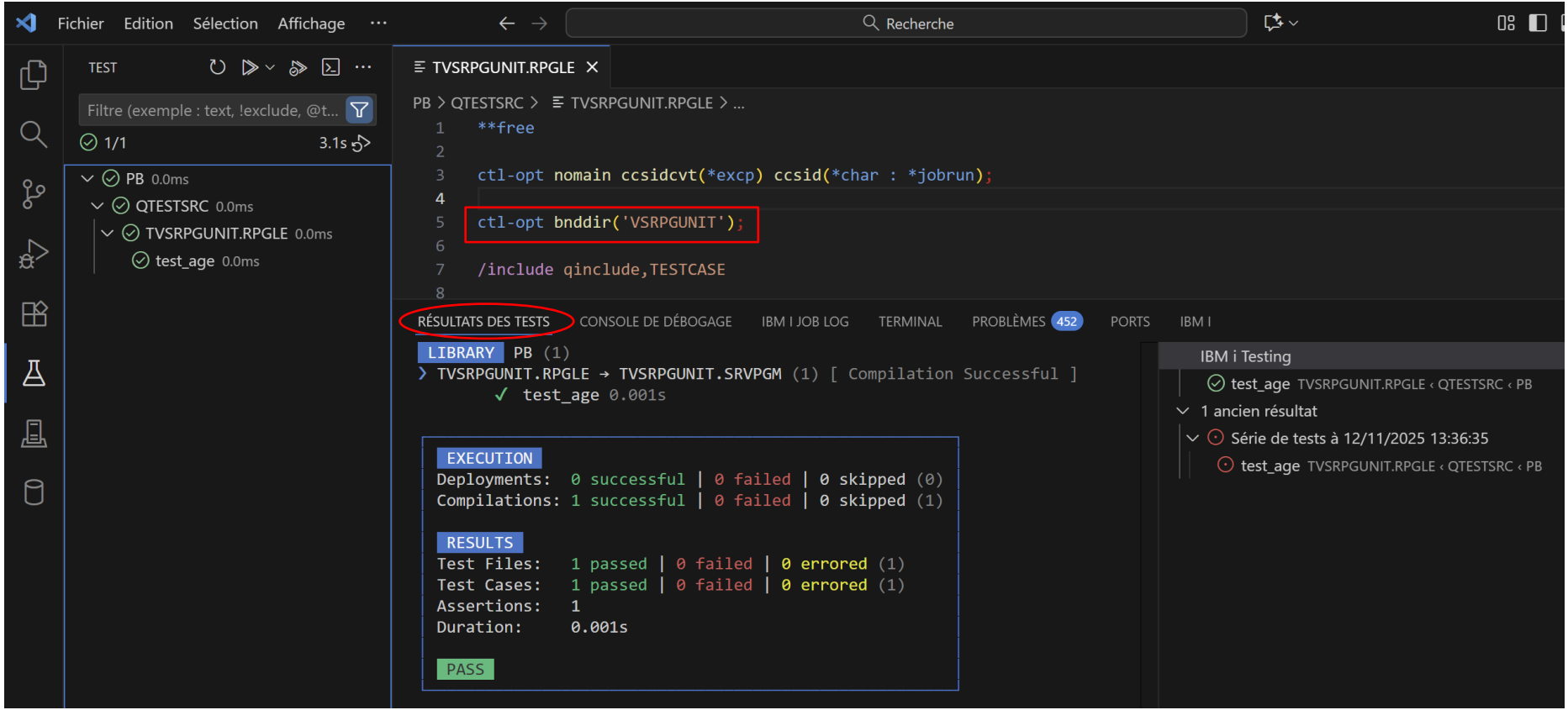
Exemple 1 – Exécution d'un cas de test

- Exécution des différentes étapes (creation du module, du programme de service...)

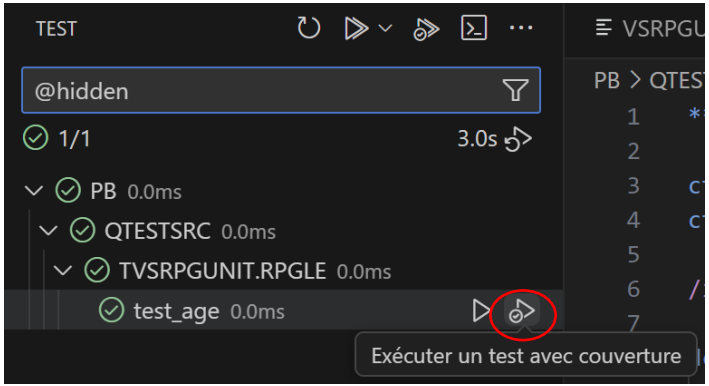


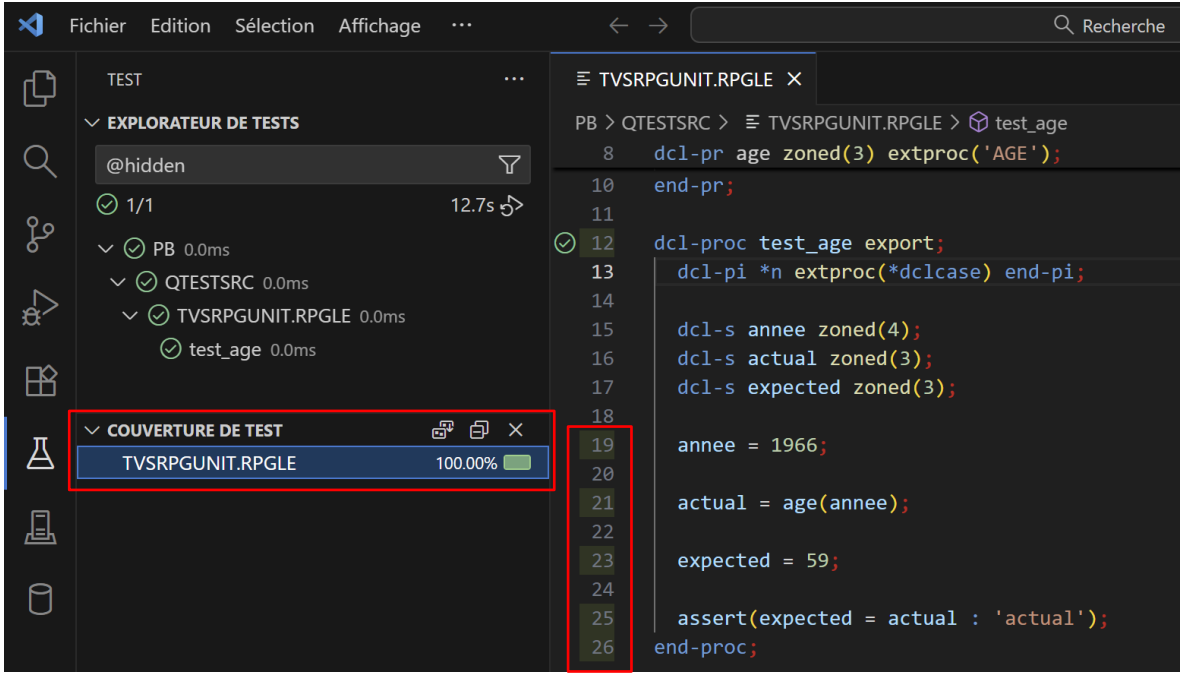


Exemple 1 – Exécution d'un cas de test



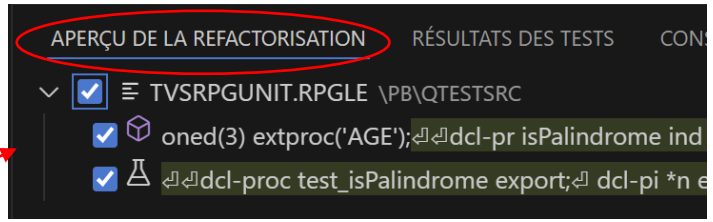
Exemple 1 – Exécution d'un cas de test avec couverture de code





Exemple 2 – Ajout d'une procédure à un cas de test

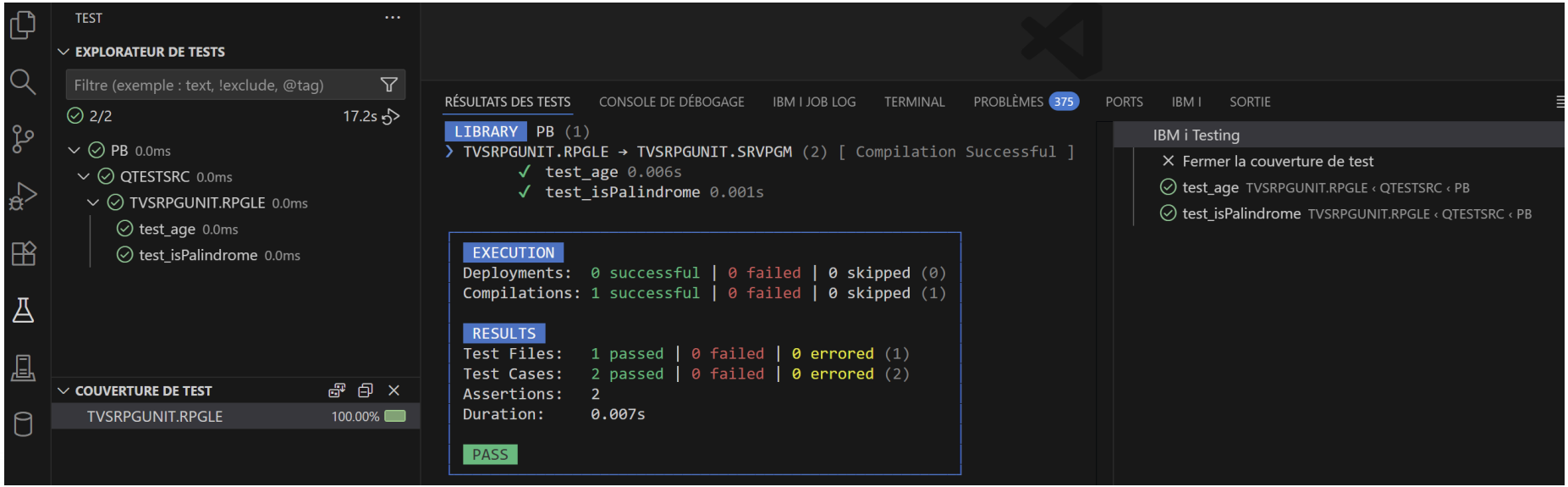
```
13   // Test si un mot est une palindrome
14   dcl-proc isPalindrome export;
15     dcl-pi *n ind;
16
17       Extraire
18       Generate test case for 'isPalindrome'
19
20       Generate test suite for 'VSRPGUNIT.RPGLE
21     max = %len(pmot);
22     if max <> 0;
23       dow min < max and %subst(pmot:mi
24         min += 1 ;
25         max -= 1 ;
26       enddo ;
27       return %subst(pmot:min:1) = %sub
28     else;
29       return 'Mot non fourni';
30     endif;
31   end-proc;
```




```
dcl-pr isPalindrome ind extproc('ISPALINDROME');
  pmot varchar(50) const;
end-pr;
```

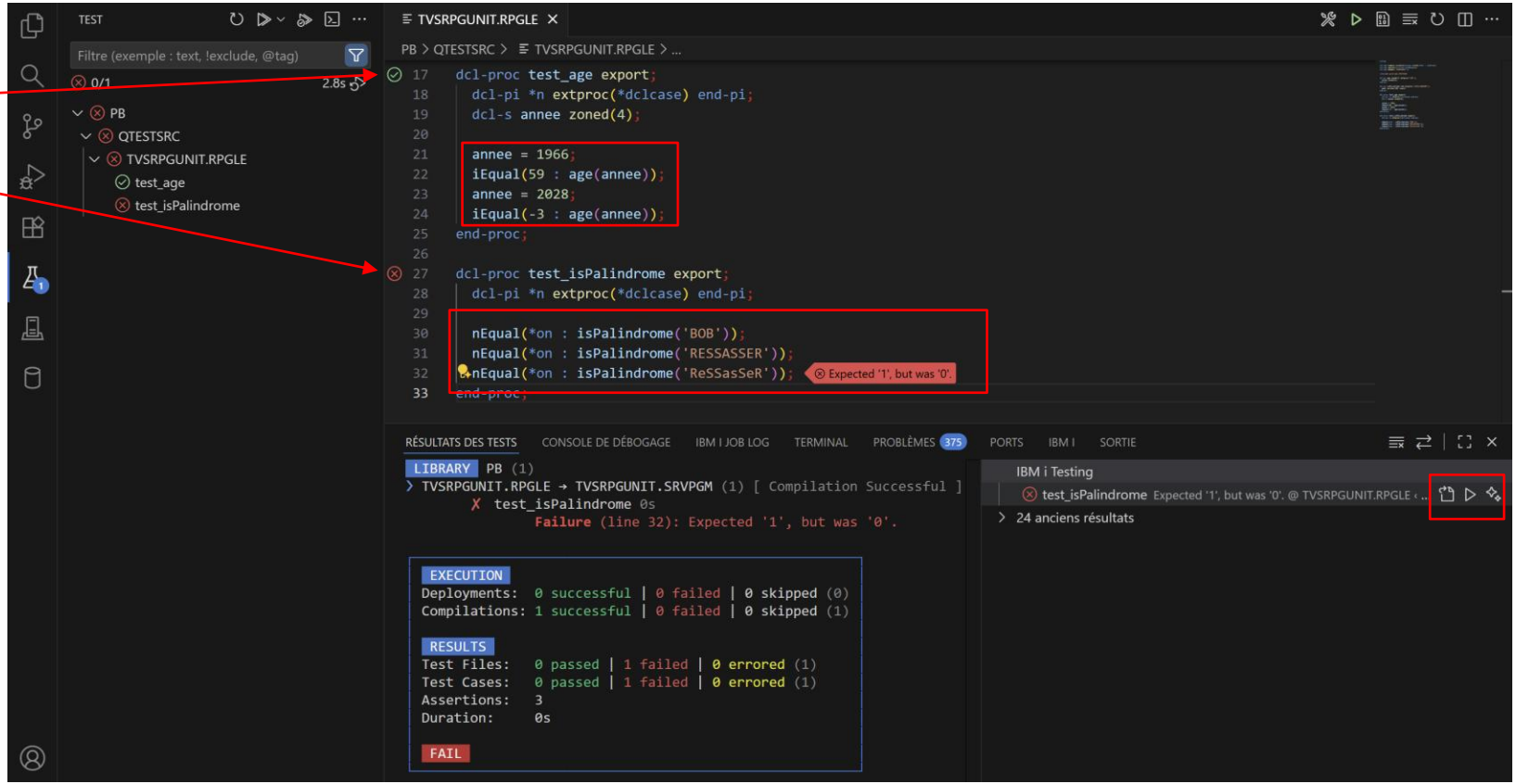
```
dcl-proc test_isPalindrome export;
  dcl-pi *n extproc(*dclcase) end-pi;
  dcl-s pmot varchar(50);
  dcl-s actual ind inz(*on);
  dcl-s expected ind inz(*on);

  pmot = 'RESSASSER';
  actual = isPalindrome(pmot);
  expected = *on;
  nEqual(expected : actual : 'actual');
end-proc;
```

Exemple 2 – Ajout d'une procédure à un cas de test



Cas de tests – Autres possibilités

Exécution directe des tests



RPGUnit – L'API Reference

- https://codefori.github.io/docs/developing/testing/writing/

Assertions

RPGUnit ≥ 5.1 | RPGUnit ≤ 5.1

assertEqual | assertThat | nEqual | assert | fail | assertJobLogContains | assertMessageQueueContains

Utilities

Commonly Used APIs

getFullTimeStamp | waitSeconds | getMonitoredMessage | setLowMessageKey

CL Command APIs

CLRPFM | RCLACTGRP | runCmd

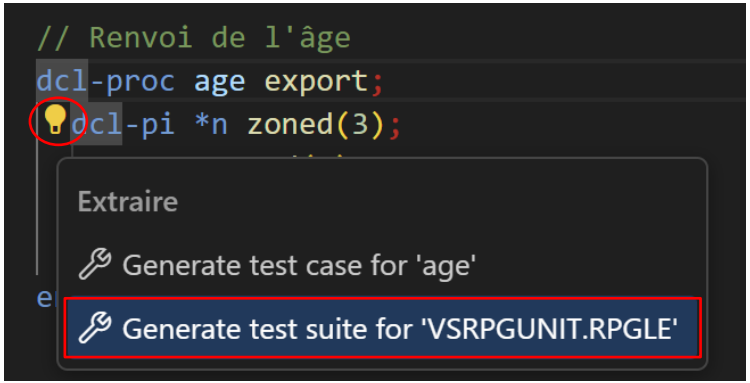
Status Message APIs

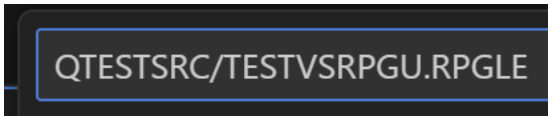
displayStatusMessage | restoreStatusMessage | clearStatusMessage

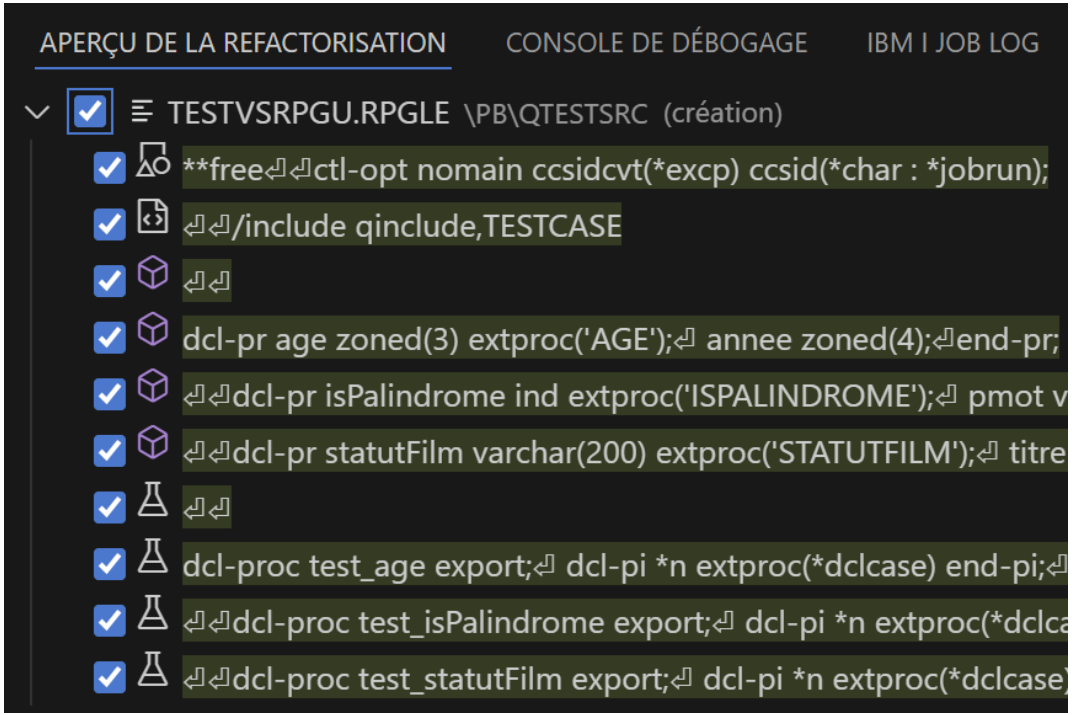
Source Type APIs

getMemberType | getStmfType

Exemple 3 – Génération d'une suite de tests







Exemple 3 – Génération d'une suite de tests

PB > QTESTSRC > TESTVSRPGU.RPGLE > ...

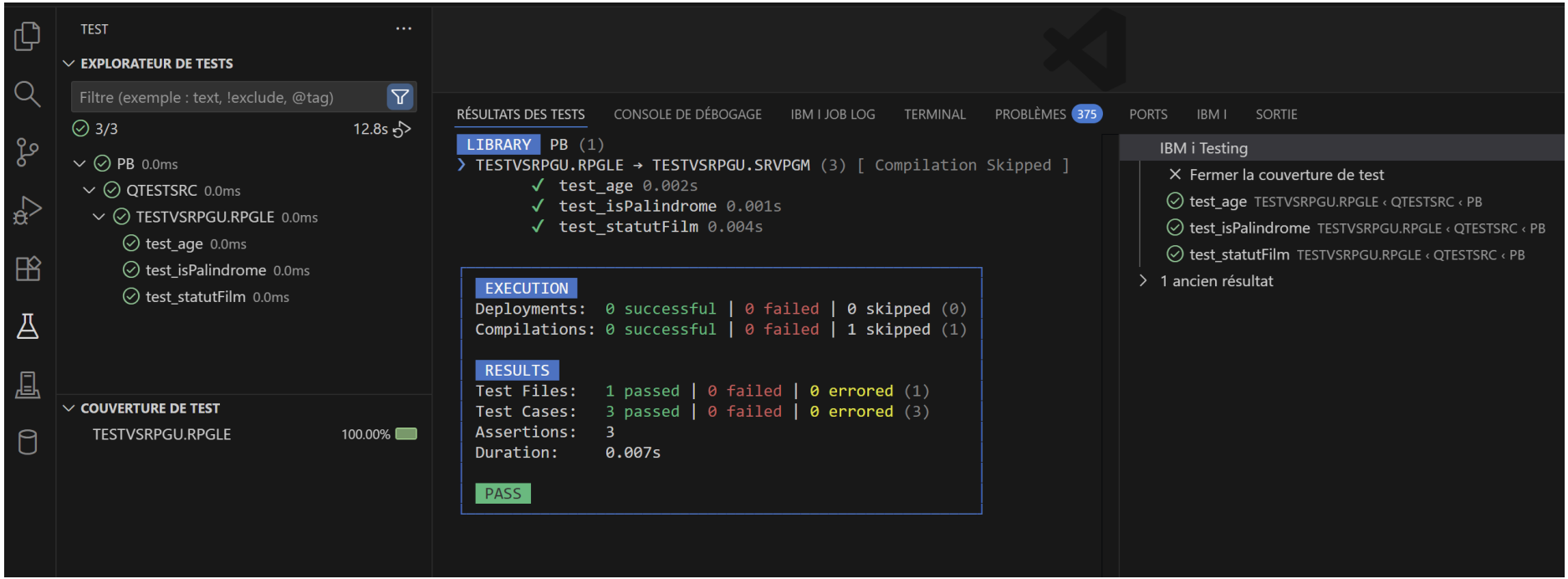
```
1   **free
2
3   ctl-opt nomain ccsidcvt(*excp) ccsid(*char : *jobrun);
4   ctl-opt option(*srcstmt:*nodebugio) alwnull(*usrctl);
5   ctl-opt bnddir('VSRPGUNIT');
6
7   /include qinclude,TESTCASE
8
9   dcl-pr age zoned(3) extproc('AGE');
10    annee zoned(4);
11  end-pr;
12
13  dcl-pr isPalindrome ind extproc('ISPALINDROME');
14    pmot varchar(50) const;
15  end-pr;
16
17  dcl-pr statutFilm varchar(200) extproc('STATUTFILM');
18    titre varchar(100) options(*trim) const;
19    etat varchar(10) options(*nullind) const;
20    dateSortie varchar(20) options(*nopass:*convert) value;
21  end-pr;
```

```
23  dcl-proc test_age export;
24    dcl-pi *n extproc(*dclcase) end-pi;
25    dcl-s annee zoned(4);
26    dcl-s actual zoned(3);
27    dcl-s expected zoned(3);
28
29    annee = 2020;
30    actual = age(annee);
31    expected = 5;
32    assert(expected = actual : 'actual');
33  end-proc;
34
35  dcl-proc test_isPalindrome export;
36    dcl-pi *n extproc(*dclcase) end-pi;
37    dcl-s pmot varchar(50);
38    dcl-s actual ind inz(*on);
39    dcl-s expected ind inz(*on);
40
41    pmot = 'IBM';
42    actual = isPalindrome(pmot);
43    expected = *off;
44    nEqual(expected : actual : 'actual');
45  end-proc;
```

Exemple 3 – Génération d'une suite de tests

```
47  dcl-proc test_statutFilm export;
48    dcl-pi *n extproc(*dclcase) end-pi;
49
50    dcl-s titre varchar(100);
51    dcl-s etat varchar(10) nullind;
52    dcl-s dateSortie date;
53    dcl-s actual varchar(200);
54    dcl-s expected varchar(200);
55
56    titre = '      AVATAR 4';
57    etat = 'P00';
58    dateSortie = %date + %years(4) + %months(1) + %days(10);
59
60    actual = statutFilm(titre : etat : dateSortie);
61
62    expected = 'AVATAR 4 - Tournage non démarré - Date de sortie : 2029-12-22';
63
64    assert(expected = actual : 'actual');
65  end-proc;
```

Exemple 3 – Génération d'une suite de tests



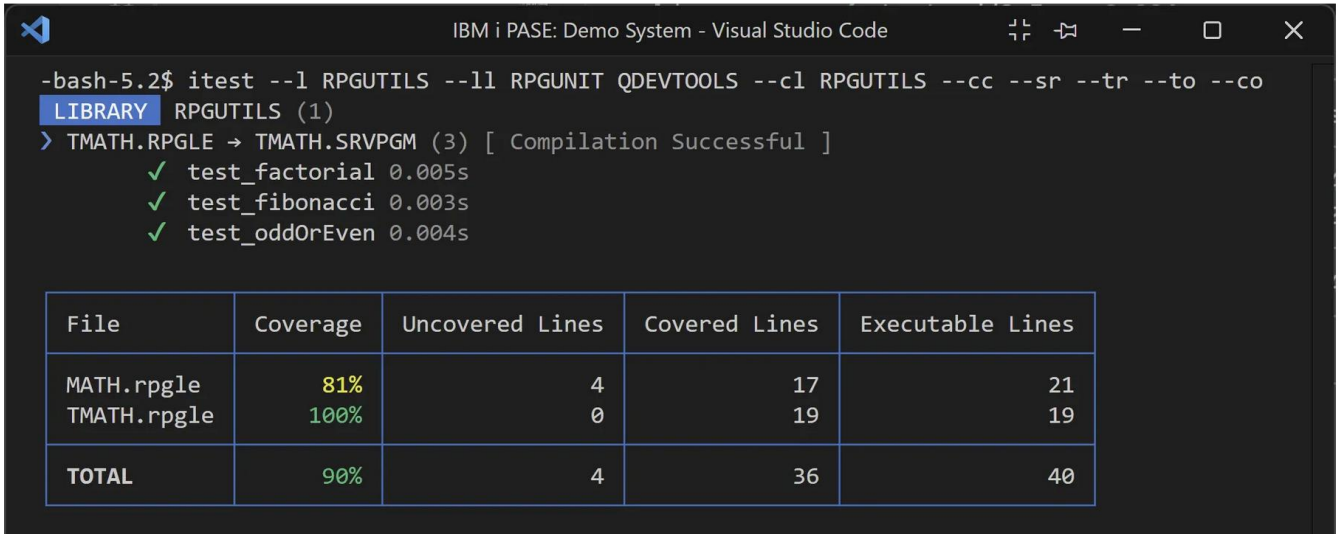
Tests automatisés et CLI

https://codefori.github.io/docs/developing/testing/cli/

CLI & Automated Tests

version v1.2.3 downloads 63/month

The IBM i Testing CLI (`itest`) is a companion to the IBM i Testing VS Code extension, which allows you to run unit tests and generate code coverage results for RPG and COBOL programs on IBM i. With this CLI, you can run tests in your terminal on your local PC or in PASE on IBM i. This enables you to even script the execution of tests in a CI/CD pipeline.



IBM Power Week

18-19-20 novembre 2025

4. Pour terminer

Pour vous former à VS Code : workshops IBM

- Actuellement disponibles : 2 workshops de 4 heures sur VS Code for IBM i :
  - Partie 1
    - Installation, configuration, personnalisation
    - Gestion des connexions IBM i, des filtres, raccourcis et profils
    - Edition de sources IBM i, compilation, actions et variables utilisateur
    - Autres fonctionnalités (comparaison, recherche, téléchargement, terminal, conversion RPG Free…)
  - Partie 2
    - Restructuration du code (refactoring)
    - Exécution, débogage de programmes et introduction aux tests unitaires
    - Autres fonctionnalités (linter : vérification de la qualité du code), recherche dans l'IFS)
    - Gestion de DB2 for i (gestion des schémas DB2, exécution SQL, SELF, notebooks, IA…)

- En 2026 : un 3ème workshop de 4 heures
  - IBM i Project Explorer
  - Source Orbit
  - Intégration avec GIT

- Me contacter si vous êtes intéressé(e) : pbourgeois@fr.ibm.com

Pour vous aider dans votre projet de modernisation IBM i

- 2 offres de service IBM Expert Labs France – A partir de 2026 – NOUVEAU

  - 1. IBM i Learning Connect – Formation de vos équipes aux nouveautés IBM i
    - Formation aux technologies, outils, nouveautés
    - Aide à la prise en main des outils
    - Accompagnement et suivi des équipes
    - Prestation au forfait, renouvelable

  - 2. IBM i Modernization Connect – Aide à la modernisation des applications IBM i
    - Audit de l'existant
    - Aide à la définition du projet de modernisation
    - POC des outils de modernisation
    - Lancement du projet
    - Prestation au forfait, renouvelable

- Me contacter si vous êtes intéressé(e) : pbourgeois@fr.ibm.com


MERCI