Building VS Code Extensions and Contributing to Code for IBM i

Sanjula Ganepola
Software Developer – IBM i App Dev & AI Toolchain

Sanjula.Ganepola@ibm.com

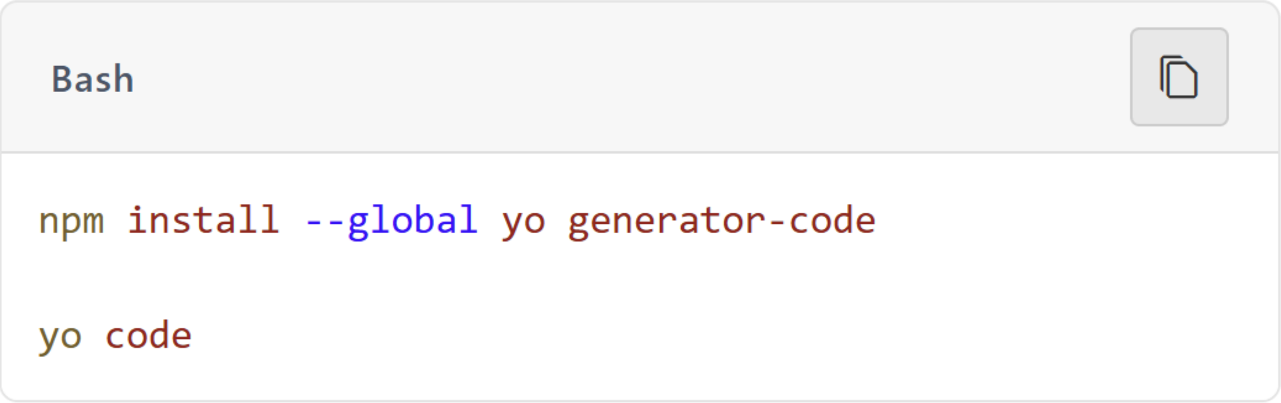
2025 September

Agenda

- Building VS Code Extensions
  - Extension anatomy
  - Extension manifest
  - VS Code API
  - Language extensions
- Contributing to Code for IBM i
  - Extension structure
  - Core logic
  - Integration with your own extension

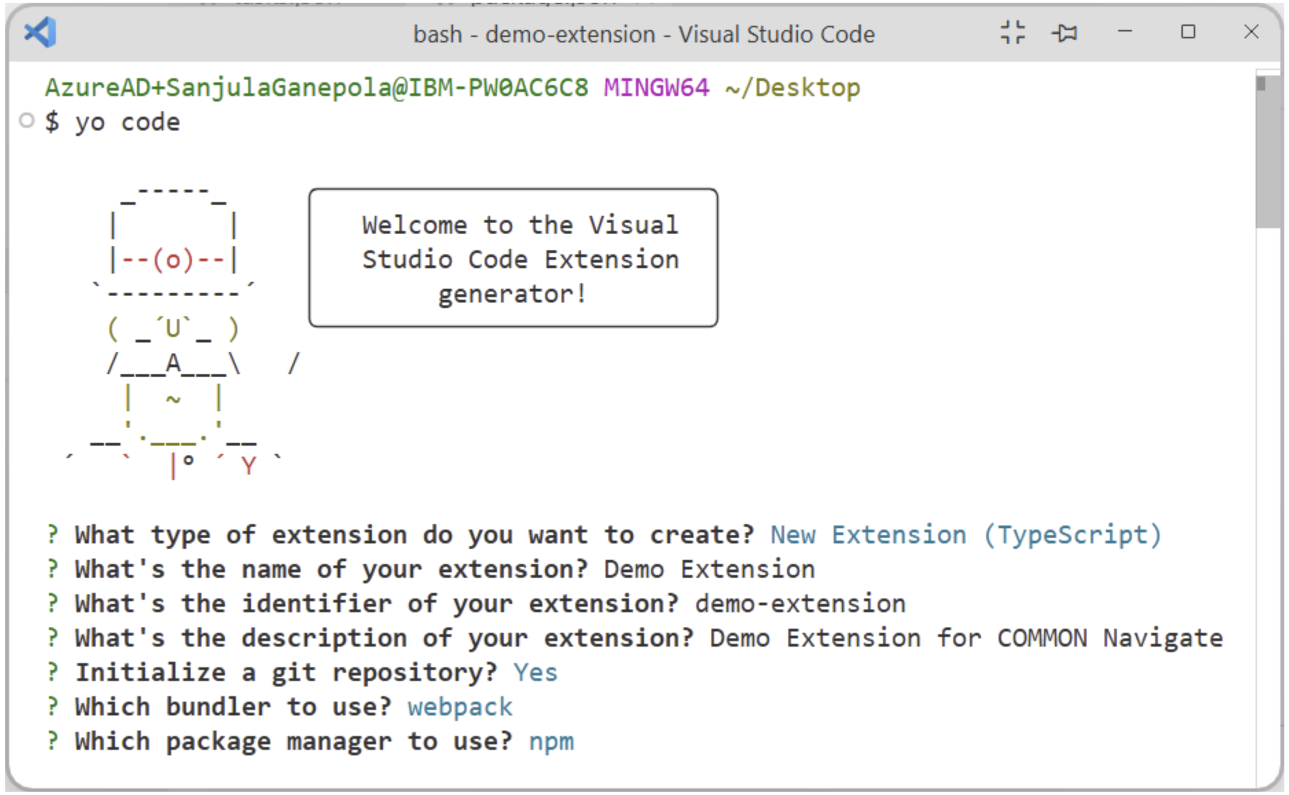
Building VS Code Extensions

Let’s create a new extension



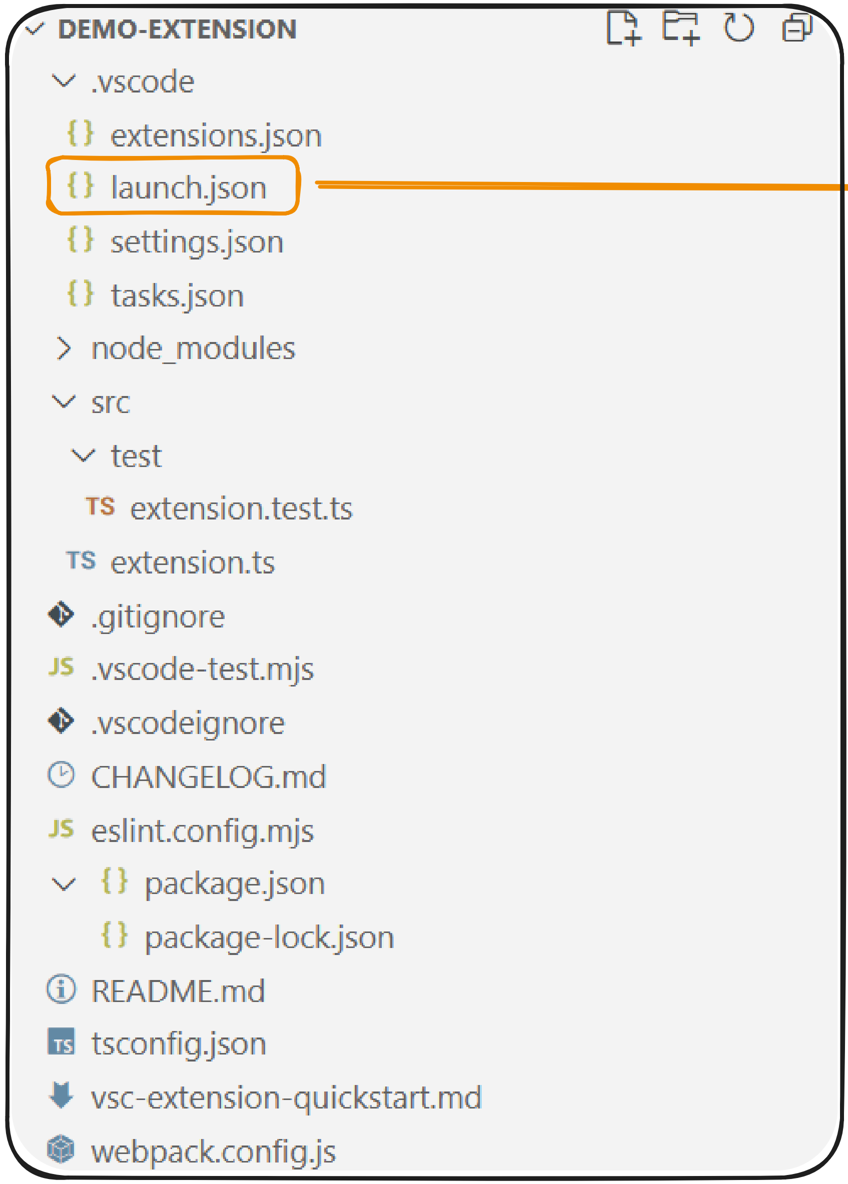
Bash

```bash
npm install --global yo generator-code

yo code
```



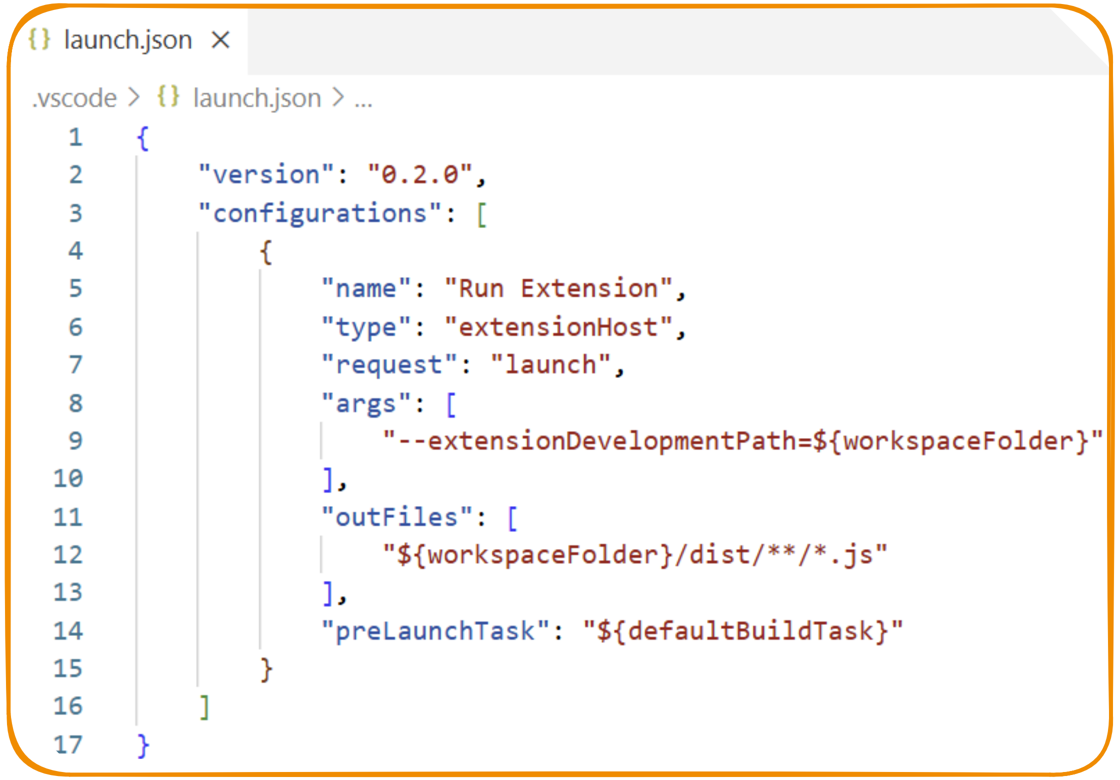
```
AzureAD+SanjulaGanepola@IBM-PW0AC6C8 MINGW64 ~/Desktop
$ yo code

     _-----_     ╭──────────────────────────╮
    |       |    │  Welcome to the Visual   │
    |--(o)--|    │  Studio Code Extension   │
   `---------´   │        generator!        │
    ( _´U`_ )    ╰──────────────────────────╯
    /___A___\   /
     |  ~  |
   __'.___.'__
 ´   `  |° ´ Y `

? What type of extension do you want to create? New Extension (TypeScript)
? What's the name of your extension? Demo Extension
? What's the identifier of your extension? demo-extension
? What's the description of your extension? Demo Extension for COMMON Navigate
? Initialize a git repository? Yes
? Which bundler to use? webpack
? Which package manager to use? npm
```

File Structure – *launch.json*



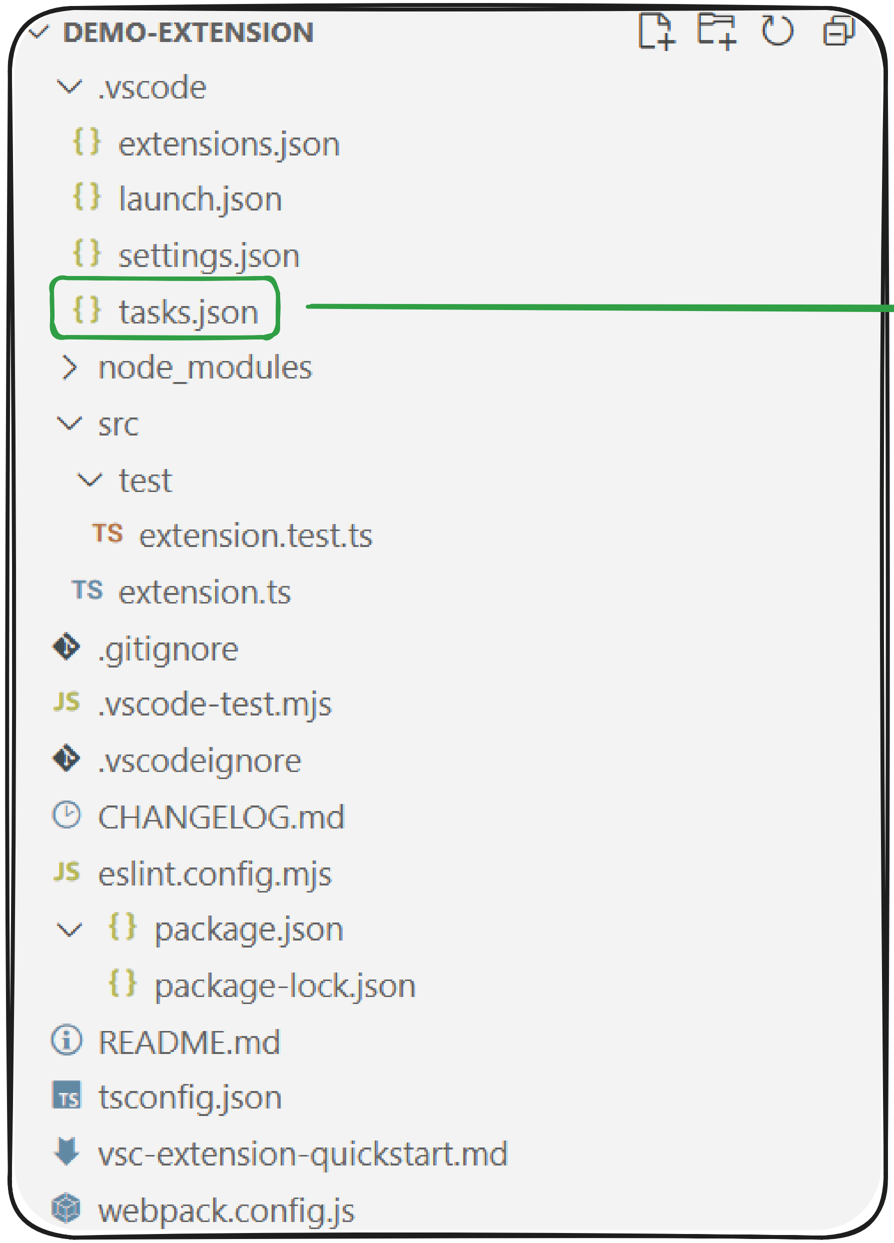
```
DEMO-EXTENSION
  .vscode
    extensions.json
    launch.json
    settings.json
    tasks.json
  node_modules
  src
    test
      extension.test.ts
    extension.ts
  .gitignore
  .vscode-test.mjs
  .vscodeignore
  CHANGELOG.md
  eslint.config.mjs
  package.json
    package-lock.json
  README.md
  tsconfig.json
  vsc-extension-quickstart.md
  webpack.config.js
```

Config for launching and debugging the extension



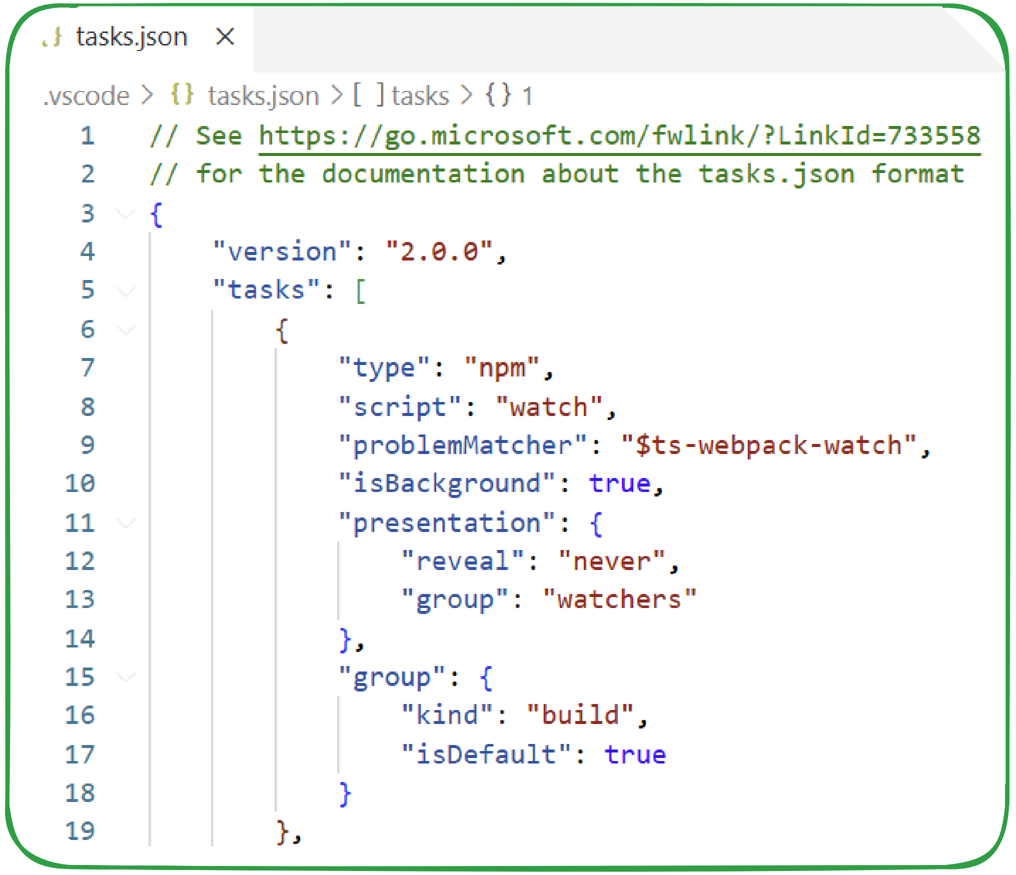
```json
{
    "version": "0.2.0",
    "configurations": [
        {
            "name": "Run Extension",
            "type": "extensionHost",
            "request": "launch",
            "args": [
                "--extensionDevelopmentPath=${workspaceFolder}"
            ],
            "outFiles": [
                "${workspaceFolder}/dist/**/*.js"
            ],
            "preLaunchTask": "${defaultBuildTask}"
        }
    ]
}
```

File Structure – *task.json*



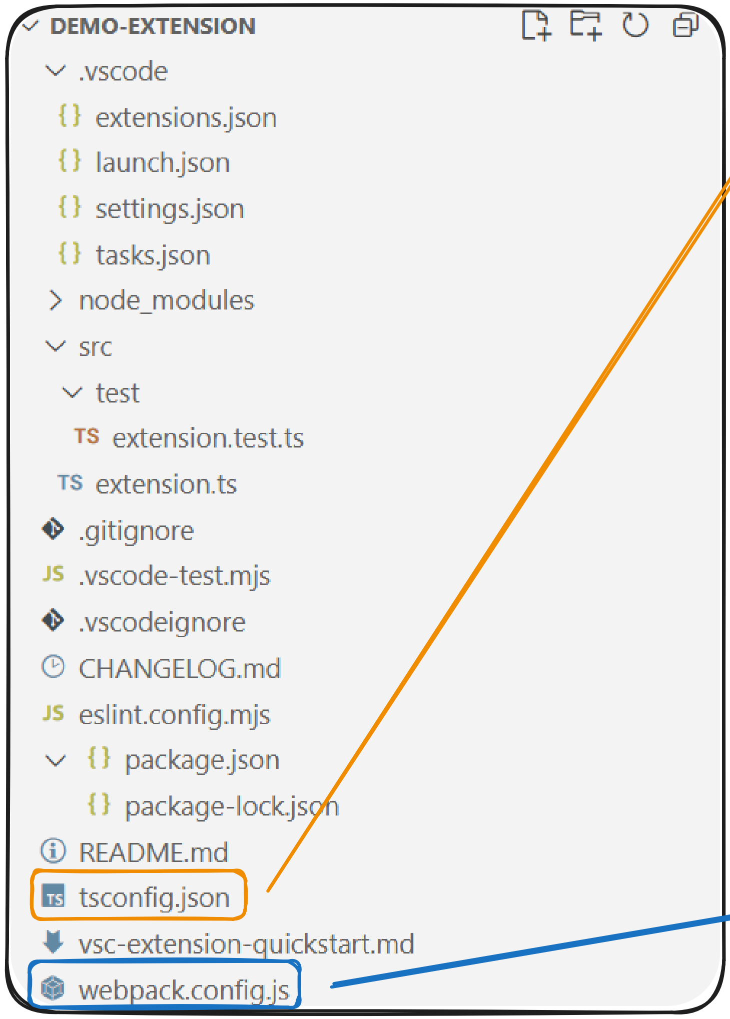
```
DEMO-EXTENSION
  .vscode
    extensions.json
    launch.json
    settings.json
    tasks.json
  node_modules
  src
    test
      extension.test.ts
    extension.ts
  .gitignore
  .vscode-test.mjs
  .vscodeignore
  CHANGELOG.md
  eslint.config.mjs
  package.json
    package-lock.json
  README.md
  tsconfig.json
  vsc-extension-quickstart.md
  webpack.config.js
```

Config for build task that compiles Typescript



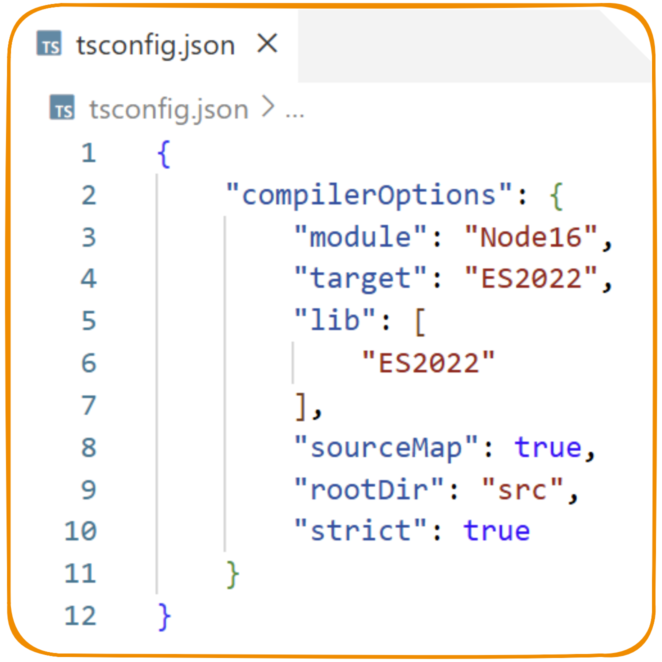
```
// See https://go.microsoft.com/fwlink/?LinkId=733558
// for the documentation about the tasks.json format
{
    "version": "2.0.0",
    "tasks": [
        {
            "type": "npm",
            "script": "watch",
            "problemMatcher": "$ts-webpack-watch",
            "isBackground": true,
            "presentation": {
                "reveal": "never",
                "group": "watchers"
            },
            "group": {
                "kind": "build",
                "isDefault": true
            }
        },
```

File Structure – *tsconfig.json* and *webpack.config.js*



```
DEMO-EXTENSION
  .vscode
    extensions.json
    launch.json
    settings.json
    tasks.json
  node_modules
  src
    test
      extension.test.ts
    extension.ts
  .gitignore
  .vscode-test.mjs
  .vscodeignore
  CHANGELOG.md
  eslint.config.mjs
  package.json
    package-lock.json
  README.md
  tsconfig.json
  vsc-extension-quickstart.md
  webpack.config.js
```

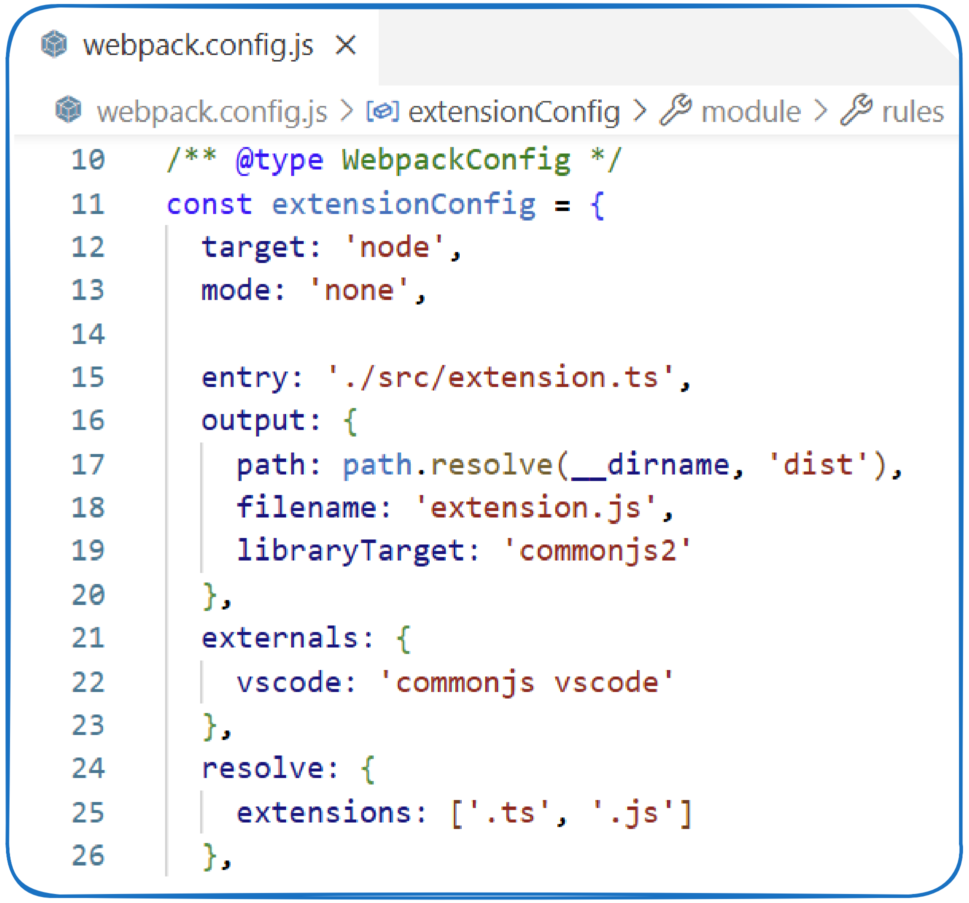
TypeScript configuration



tsconfig.json > ...

```json
{
    "compilerOptions": {
        "module": "Node16",
        "target": "ES2022",
        "lib": [
            "ES2022"
        ],
        "sourceMap": true,
        "rootDir": "src",
        "strict": true
    }
}
```

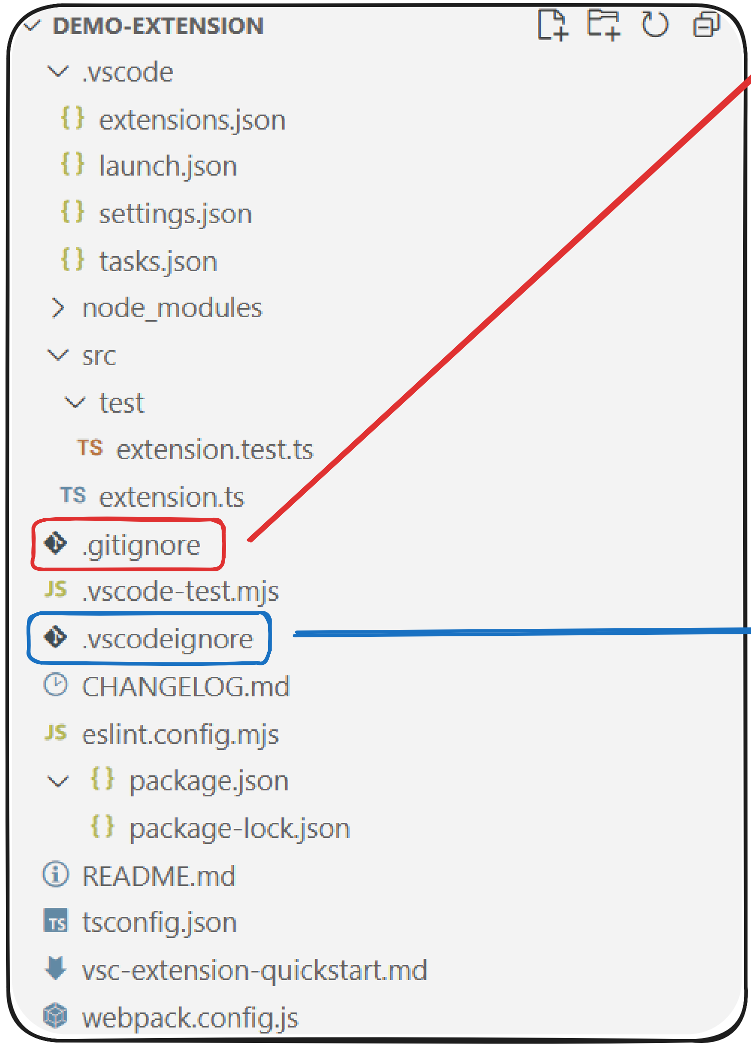
Webpack configuration



webpack.config.js > extensionConfig > module > rules

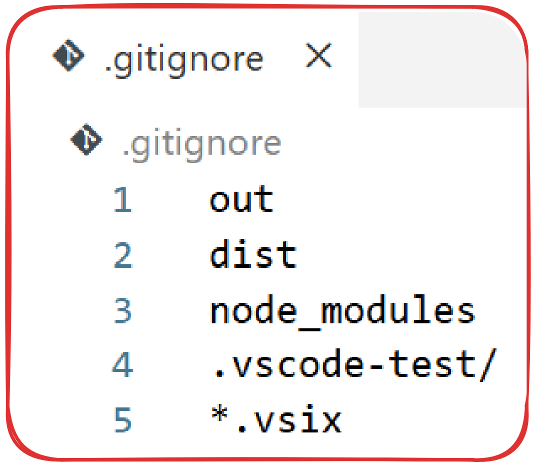
```js
/** @type WebpackConfig */
const extensionConfig = {
  target: 'node',
  mode: 'none',

  entry: './src/extension.ts',
  output: {
    path: path.resolve(__dirname, 'dist'),
    filename: 'extension.js',
    libraryTarget: 'commonjs2'
  },
  externals: {
    vscode: 'commonjs vscode'
  },
  resolve: {
    extensions: ['.ts', '.js']
  },
```

File Structure – *.gitignore* and *.vscodeignore*



```
DEMO-EXTENSION
  .vscode
    extensions.json
    launch.json
    settings.json
    tasks.json
  node_modules
  src
    test
      extension.test.ts
    extension.ts
  .gitignore
  .vscode-test.mjs
  .vscodeignore
  CHANGELOG.md
  eslint.config.mjs
  package.json
    package-lock.json
  README.md
  tsconfig.json
  vsc-extension-quickstart.md
  webpack.config.js
```

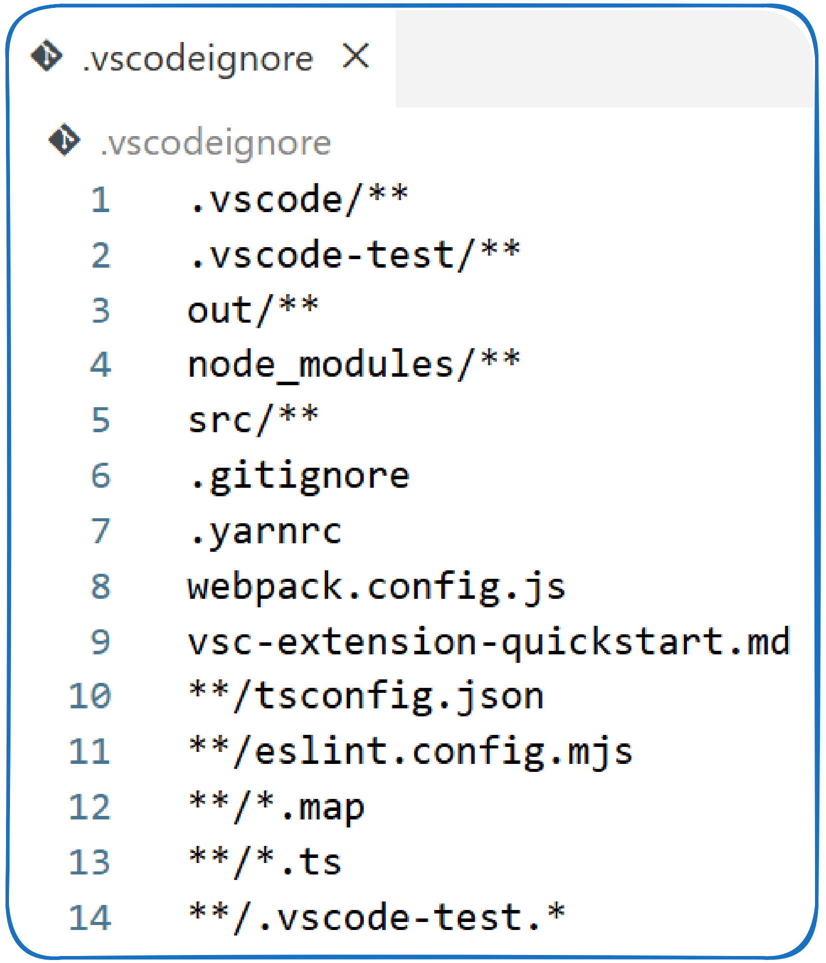
Ignore build output and node_modules



.gitignore

```
out
dist
node_modules
.vscode-test/
*.vsix
```

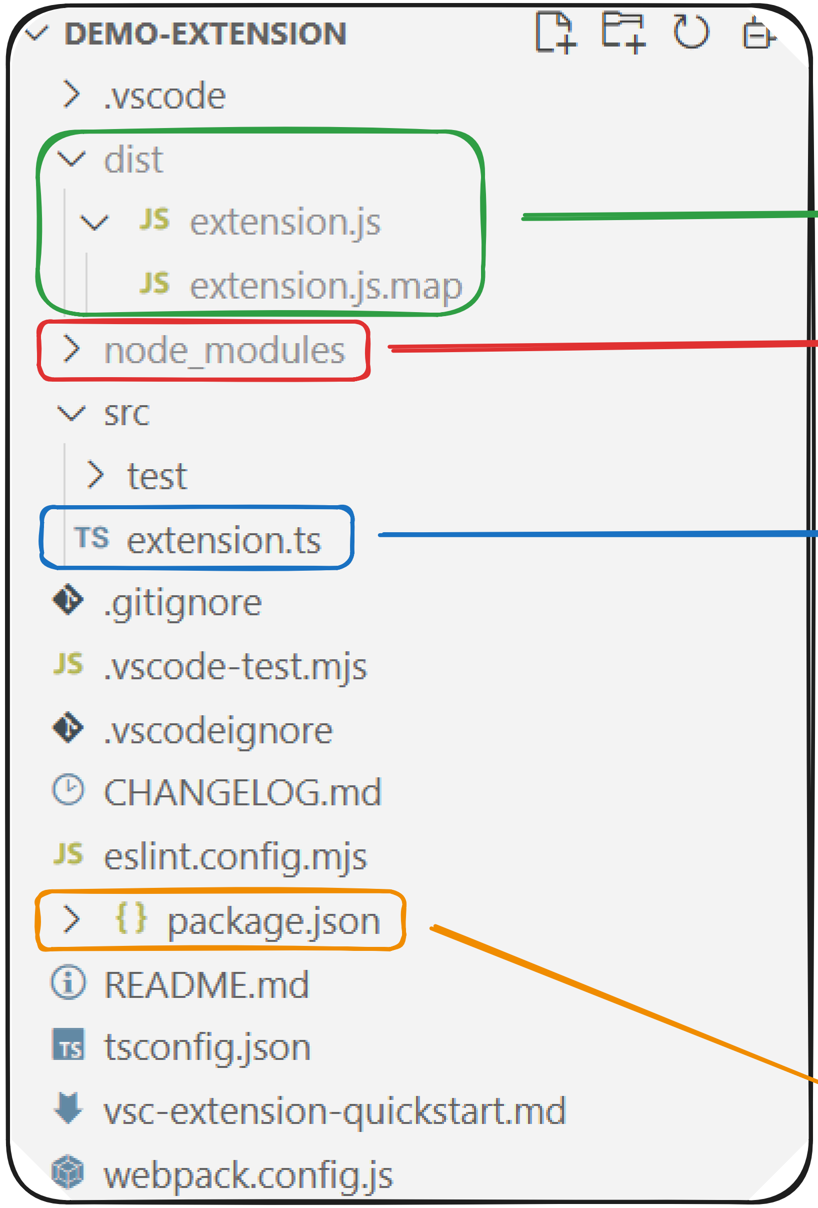
Prevent files from being included in your extension's package.



.vscodeignore

```
.vscode/**
.vscode-test/**
out/**
node_modules/**
src/**
.gitignore
.yarnrc
webpack.config.js
vsc-extension-quickstart.md
**/tsconfig.json
**/eslint.config.mjs
**/*.map
**/*.ts
**/.vscode-test.*
```

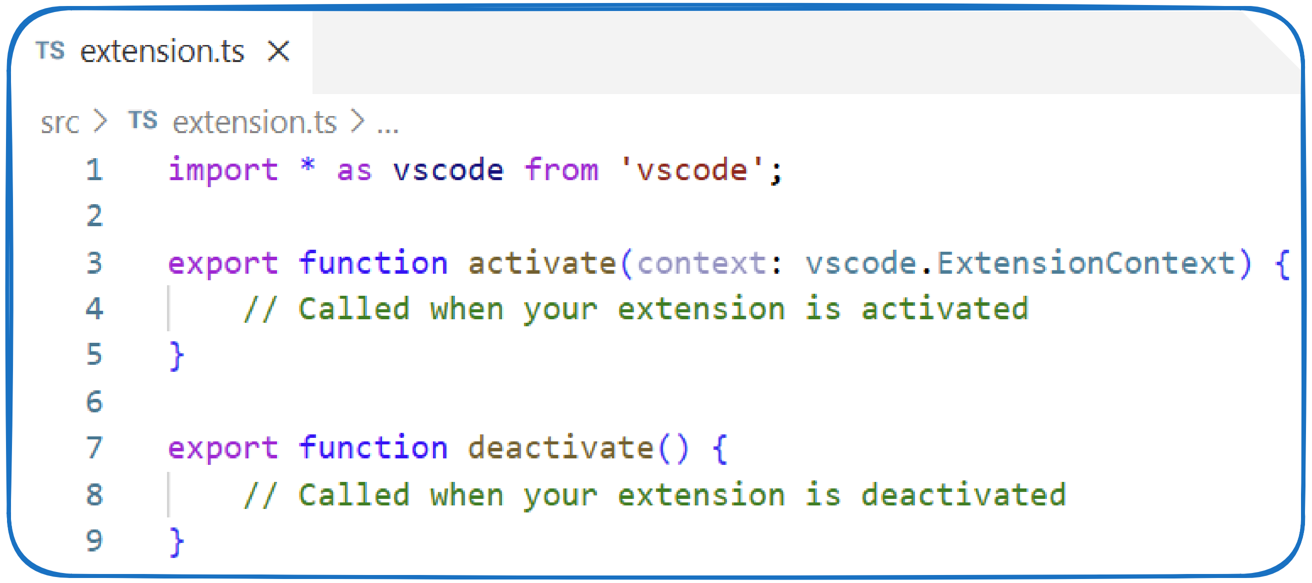
File Structure – Source code, extension manifest, and more



Compiled and bundled source

Installed node dependencies

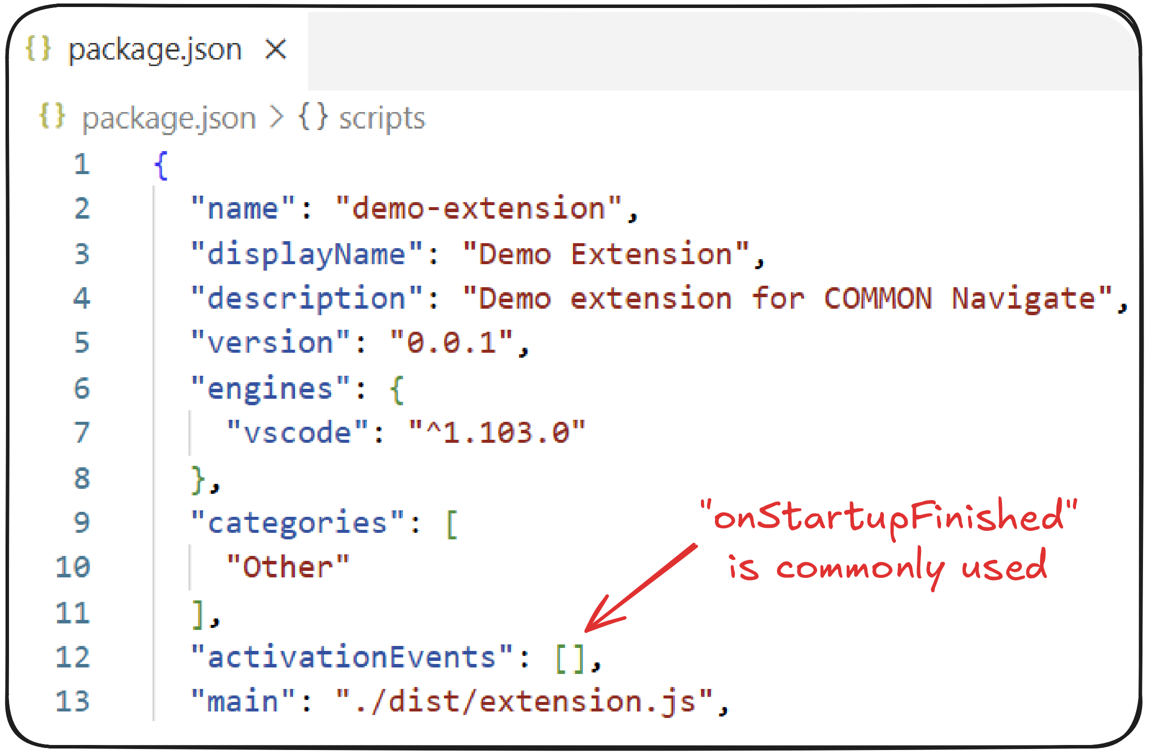
Extension source code



```
import * as vscode from 'vscode';

export function activate(context: vscode.ExtensionContext) {
    // Called when your extension is activated
}

export function deactivate() {
    // Called when your extension is deactivated
}
```

Extension Manifest

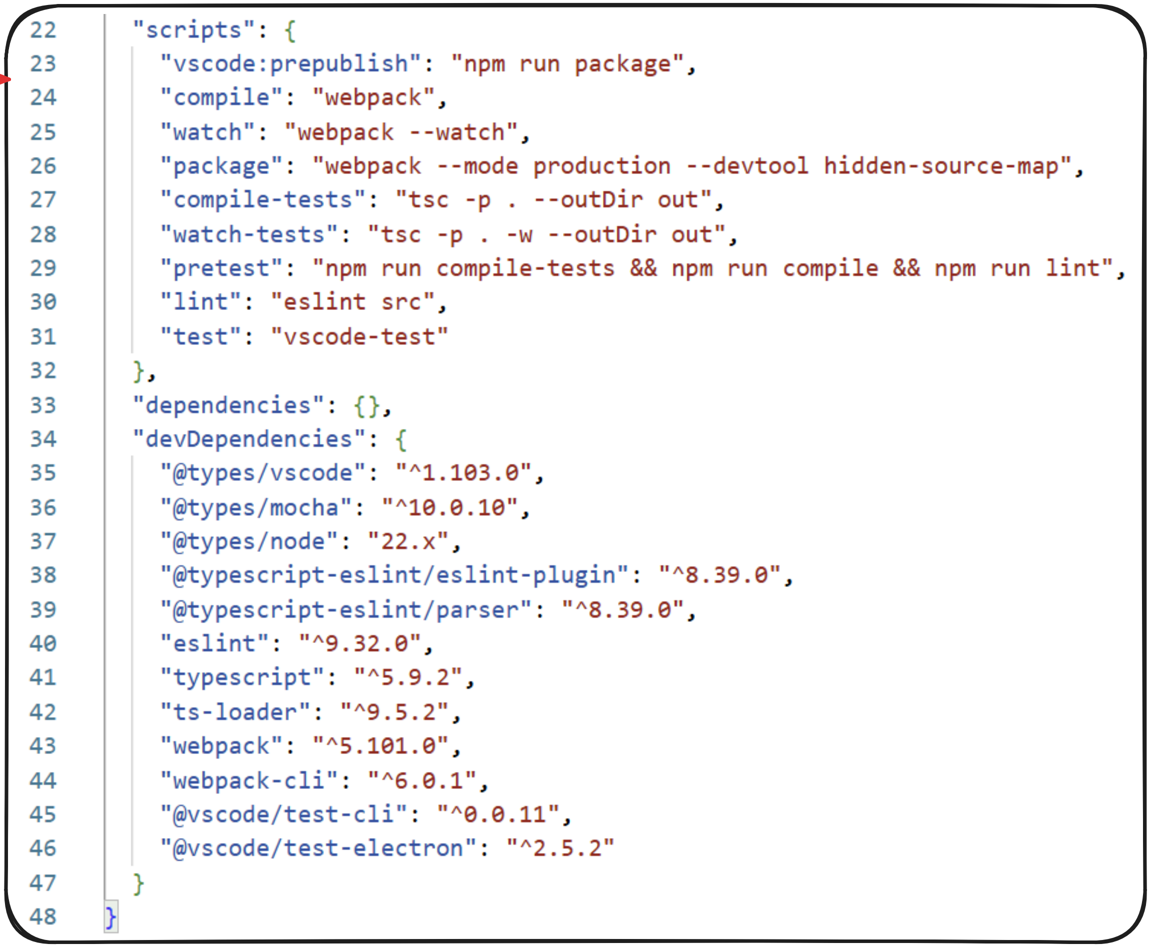
File Structure – package.json



```json
{
  "name": "demo-extension",
  "displayName": "Demo Extension",
  "description": "Demo extension for COMMON Navigate",
  "version": "0.0.1",
  "engines": {
    "vscode": "^1.103.0"
  },
  "categories": [
    "Other"
  ],
  "activationEvents": [],
  "main": "./dist/extension.js",
```

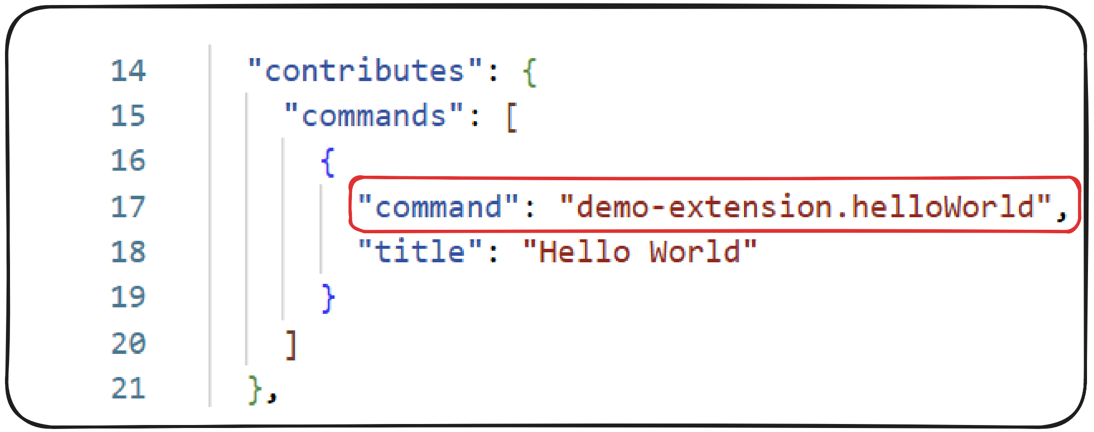
"onStartupFinished" is commonly used

contributes

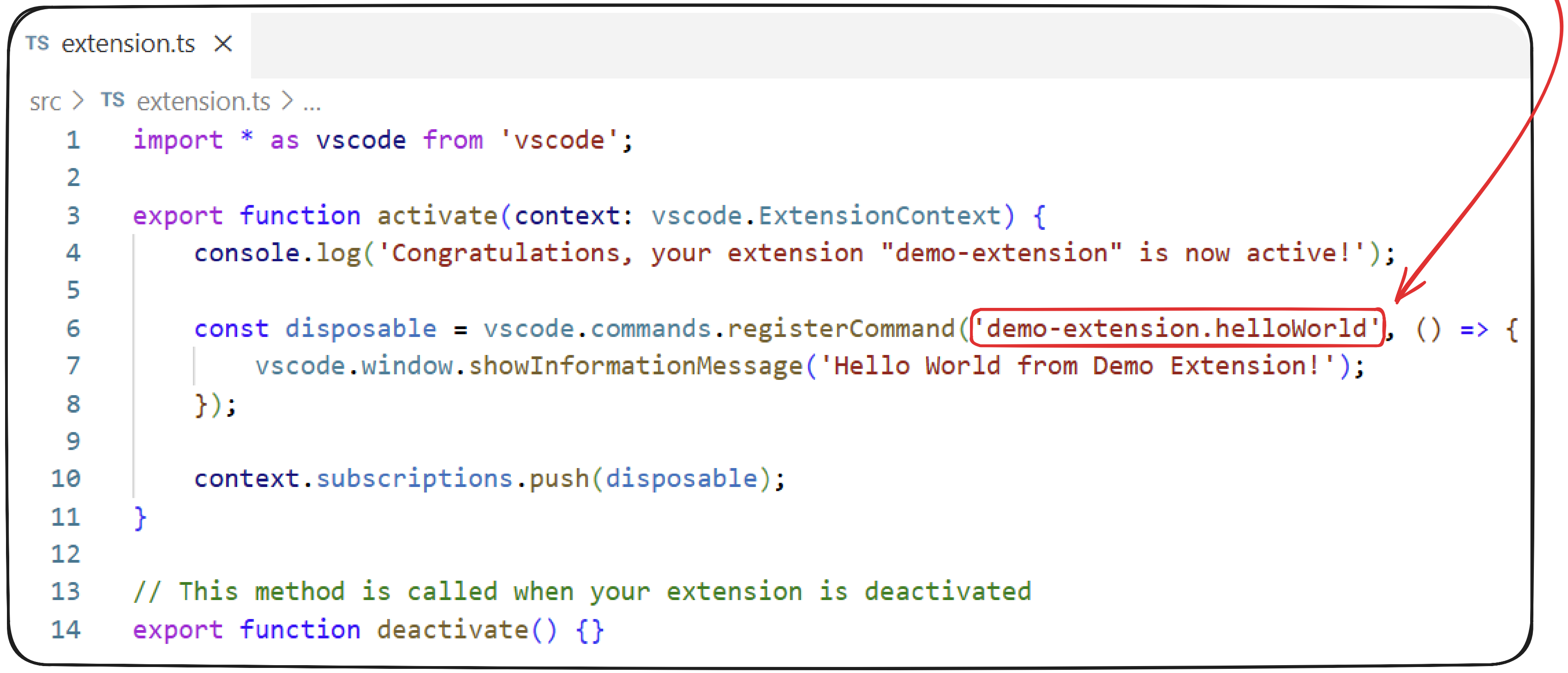


```json
  "scripts": {
    "vscode:prepublish": "npm run package",
    "compile": "webpack",
    "watch": "webpack --watch",
    "package": "webpack --mode production --devtool hidden-source-map",
    "compile-tests": "tsc -p . --outDir out",
    "watch-tests": "tsc -p . -w --outDir out",
    "pretest": "npm run compile-tests && npm run compile && npm run lint",
    "lint": "eslint src",
    "test": "vscode-test"
  },
  "dependencies": {},
  "devDependencies": {
    "@types/vscode": "^1.103.0",
    "@types/mocha": "^10.0.10",
    "@types/node": "22.x",
    "@typescript-eslint/eslint-plugin": "^8.39.0",
    "@typescript-eslint/parser": "^8.39.0",
    "eslint": "^9.32.0",
    "typescript": "^5.9.2",
    "ts-loader": "^9.5.2",
    "webpack": "^5.101.0",
    "webpack-cli": "^6.0.1",
    "@vscode/test-cli": "^0.0.11",
    "@vscode/test-electron": "^2.5.2"
  }
}
```

Extension Manifest - Contributions

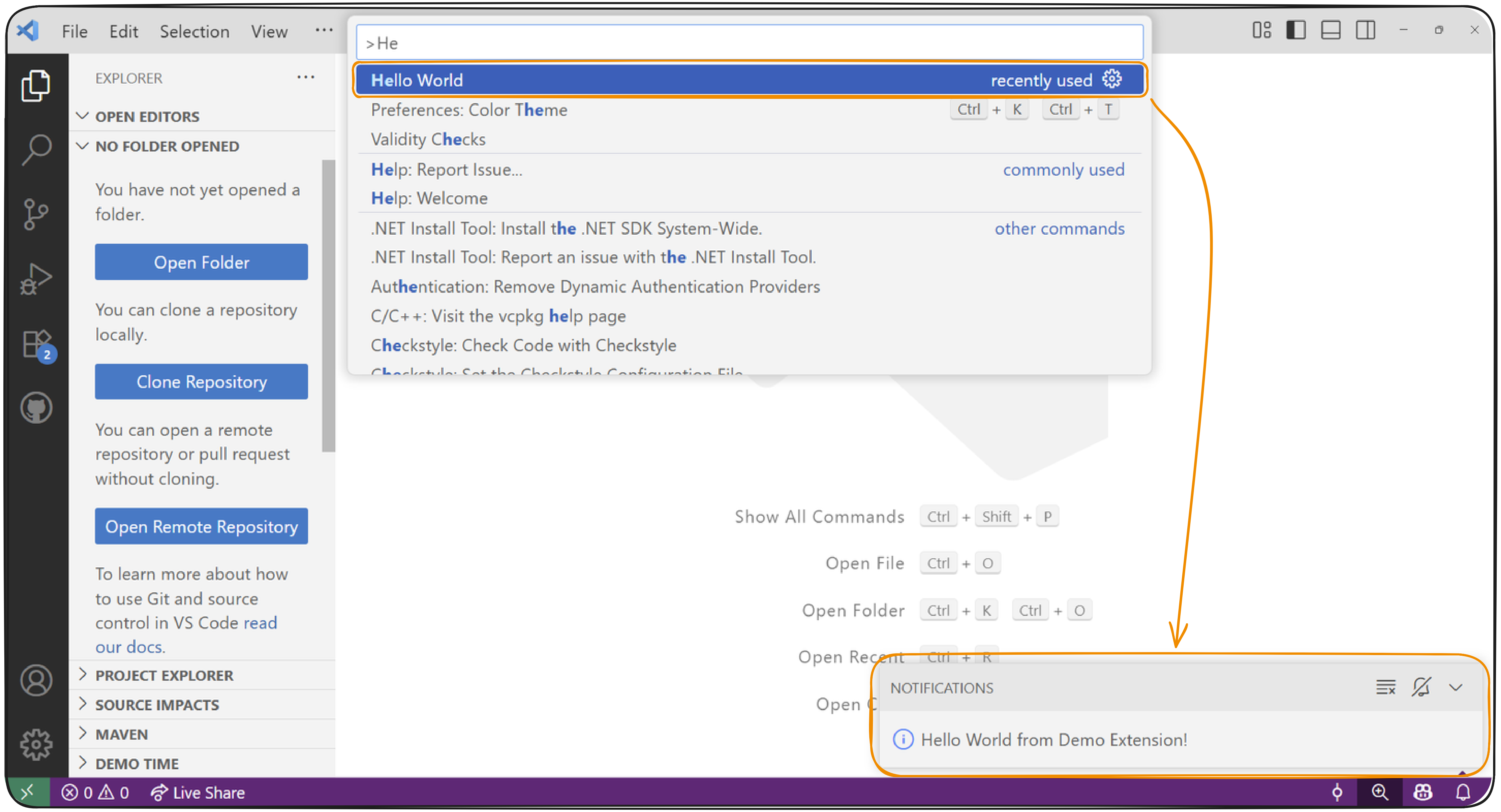


```
"contributes": {
  "commands": [
    {
      "command": "demo-extension.helloWorld",
      "title": "Hello World"
    }
  ]
},
```



```
import * as vscode from 'vscode';

export function activate(context: vscode.ExtensionContext) {
    console.log('Congratulations, your extension "demo-extension" is now active!');

    const disposable = vscode.commands.registerCommand('demo-extension.helloWorld', () => {
        vscode.window.showInformationMessage('Hello World from Demo Extension!');
    });

    context.subscriptions.push(disposable);
}

// This method is called when your extension is deactivated
export function deactivate() {}
```

Let’s run our extension



You have not yet opened a folder.

You can clone a repository locally.

You can open a remote repository or pull request without cloning.

To learn more about how to use Git and source control in VS Code read our docs.

>He

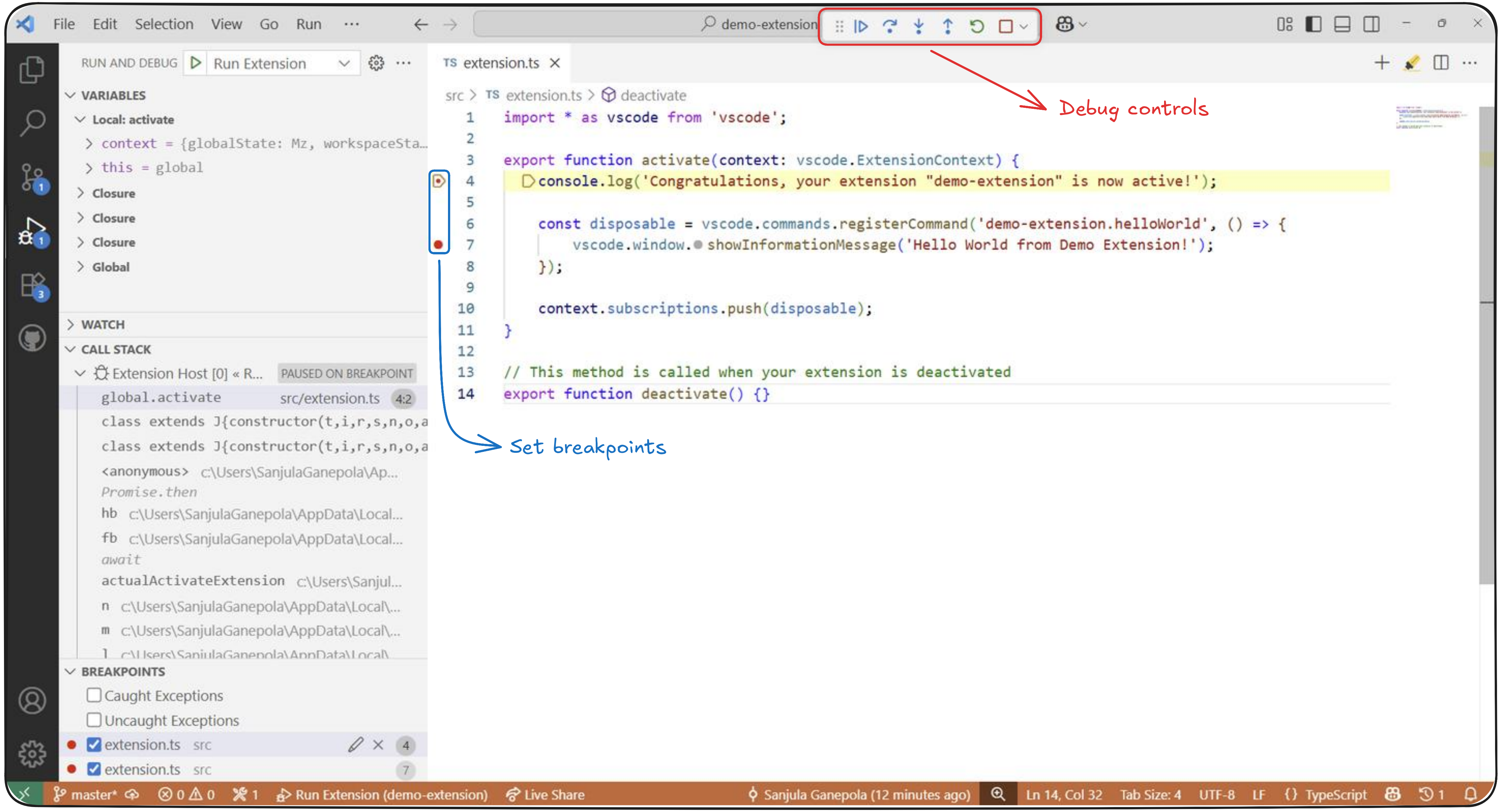
Hello World recently used
Preferences: Color Theme Ctrl + K Ctrl + T
Validity Checks
Help: Report Issue... commonly used
Help: Welcome
.NET Install Tool: Install the .NET SDK System-Wide. other commands
.NET Install Tool: Report an issue with the .NET Install Tool.
Authentication: Remove Dynamic Authentication Providers
C/C++: Visit the vcpkg help page
Checkstyle: Check Code with Checkstyle
Checkstyle: Set the Checkstyle Configuration File

Show All Commands Ctrl + Shift + P
Open File Ctrl + O
Open Folder Ctrl + K Ctrl + O
Open Recent Ctrl + R

NOTIFICATIONS

Hello World from Demo Extension!

Let’s debug our extension



Debug controls

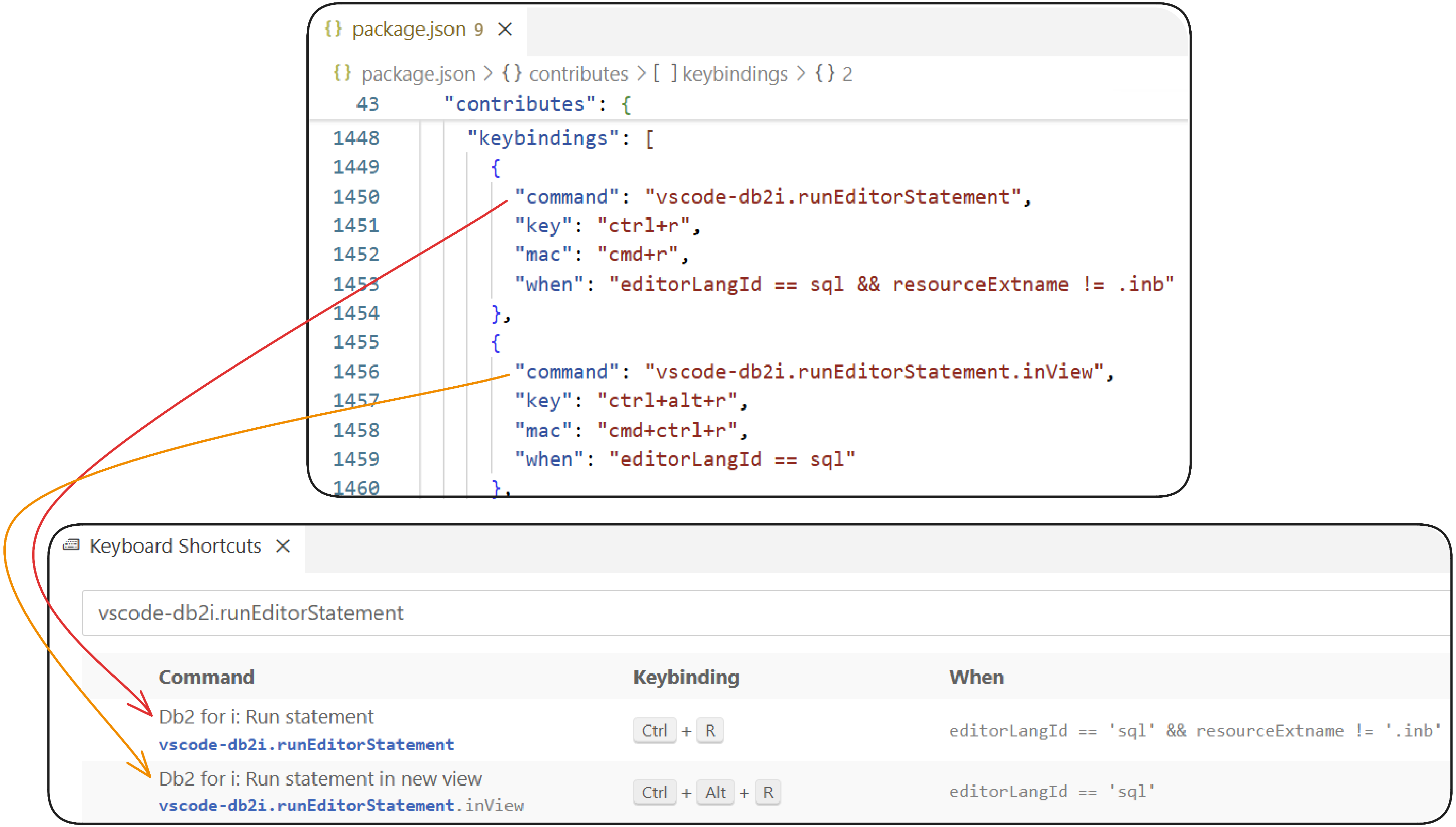
Set breakpoints

```
import * as vscode from 'vscode';

export function activate(context: vscode.ExtensionContext) {
    console.log('Congratulations, your extension "demo-extension" is now active!');

    const disposable = vscode.commands.registerCommand('demo-extension.helloWorld', () => {
        vscode.window.showInformationMessage('Hello World from Demo Extension!');
    });

    context.subscriptions.push(disposable);
}

// This method is called when your extension is deactivated
export function deactivate() {}
```

```
VARIABLES
  Local: activate
    > context = {globalState: Mz, workspaceSta...
    > this = global
  > Closure
  > Closure
  > Closure
  > Global

> WATCH

CALL STACK
  Extension Host [0] « R...          PAUSED ON BREAKPOINT
    global.activate                  src/extension.ts  4:2
    class extends J{constructor(t,i,r,s,n,o,a
    class extends J{constructor(t,i,r,s,n,o,a
    <anonymous>  c:\Users\SanjulaGanepola\Ap...
    Promise.then
    hb  c:\Users\SanjulaGanepola\AppData\Local...
    fb  c:\Users\SanjulaGanepola\AppData\Local...
    await
    actualActivateExtension  c:\Users\Sanjul...
    n  c:\Users\SanjulaGanepola\AppData\Local\...
    m  c:\Users\SanjulaGanepola\AppData\Local\...
    l  c:\Users\SanjulaGanepola\AppData\Local\...

BREAKPOINTS
  □ Caught Exceptions
  □ Uncaught Exceptions
  ● ☑ extension.ts  src   4
  ● ☑ extension.ts  src   7
```

```
master*   ⊗ 0 ⚠ 0   1   Run Extension (demo-extension)   Live Share   Sanjula Ganepola (12 minutes ago)   Ln 14, Col 32   Tab Size: 4   UTF-8   LF   {} TypeScript   1
```

Contribution Points - Keybindings

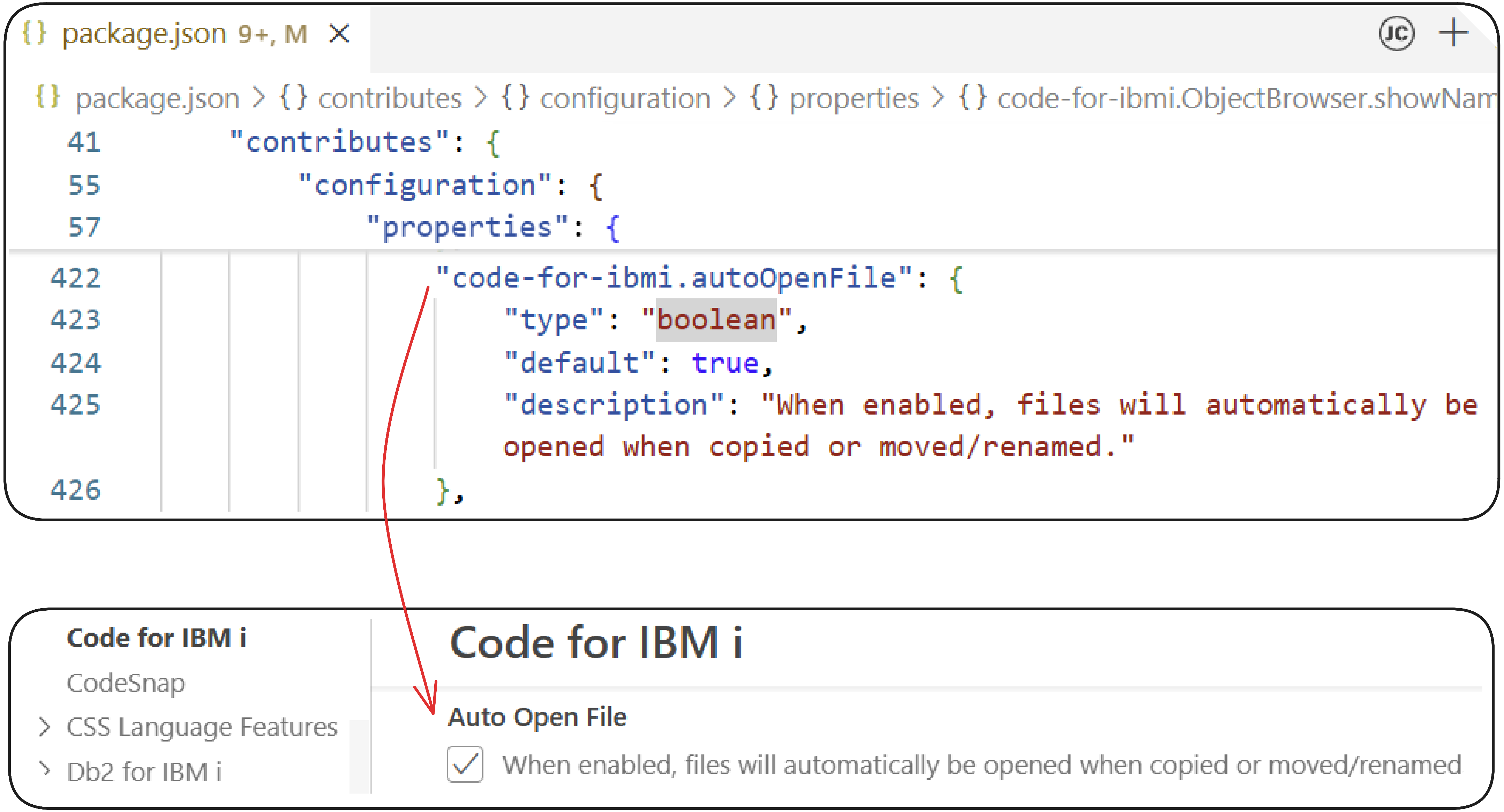


```
    "contributes": {
      "keybindings": [
        {
          "command": "vscode-db2i.runEditorStatement",
          "key": "ctrl+r",
          "mac": "cmd+r",
          "when": "editorLangId == sql && resourceExtname != .inb"
        },
        {
          "command": "vscode-db2i.runEditorStatement.inView",
          "key": "ctrl+alt+r",
          "mac": "cmd+ctrl+r",
          "when": "editorLangId == sql"
        },
```

vscode-db2i.runEditorStatement

Command	Keybinding	When
Db2 for i: Run statement vscode-db2i.runEditorStatement	Ctrl + R	editorLangId == 'sql' && resourceExtname != '.inb'
Db2 for i: Run statement in new view vscode-db2i.runEditorStatement.inView	Ctrl + Alt + R	editorLangId == 'sql'

Contribution Points - Configurations



{} package.json 9+, M
{} package.json > {} contributes > {} configuration > {} properties > {} code-for-ibmi.ObjectBrowser.showNam

```json
"contributes": {
    "configuration": {
        "properties": {
            "code-for-ibmi.autoOpenFile": {
                "type": "boolean",
                "default": true,
                "description": "When enabled, files will automatically be opened when copied or moved/renamed."
            },
```

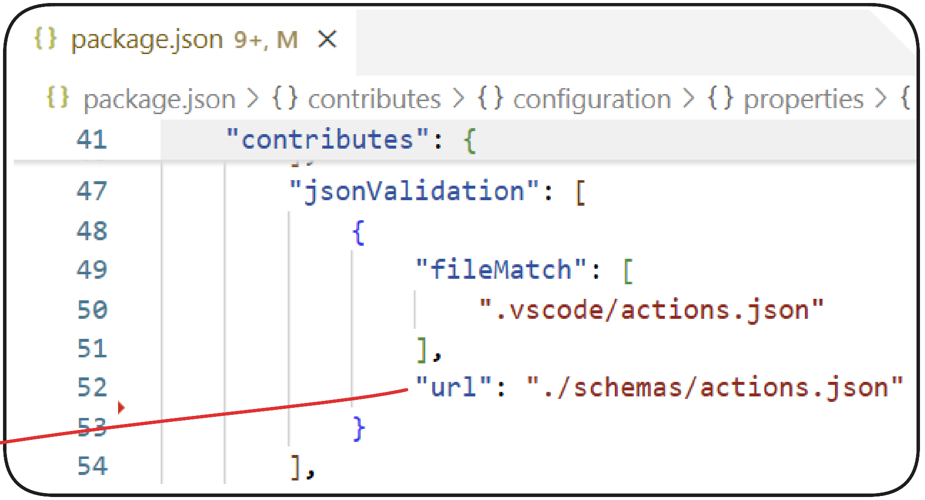
Code for IBM i
CodeSnap
CSS Language Features
Db2 for IBM i

Code for IBM i

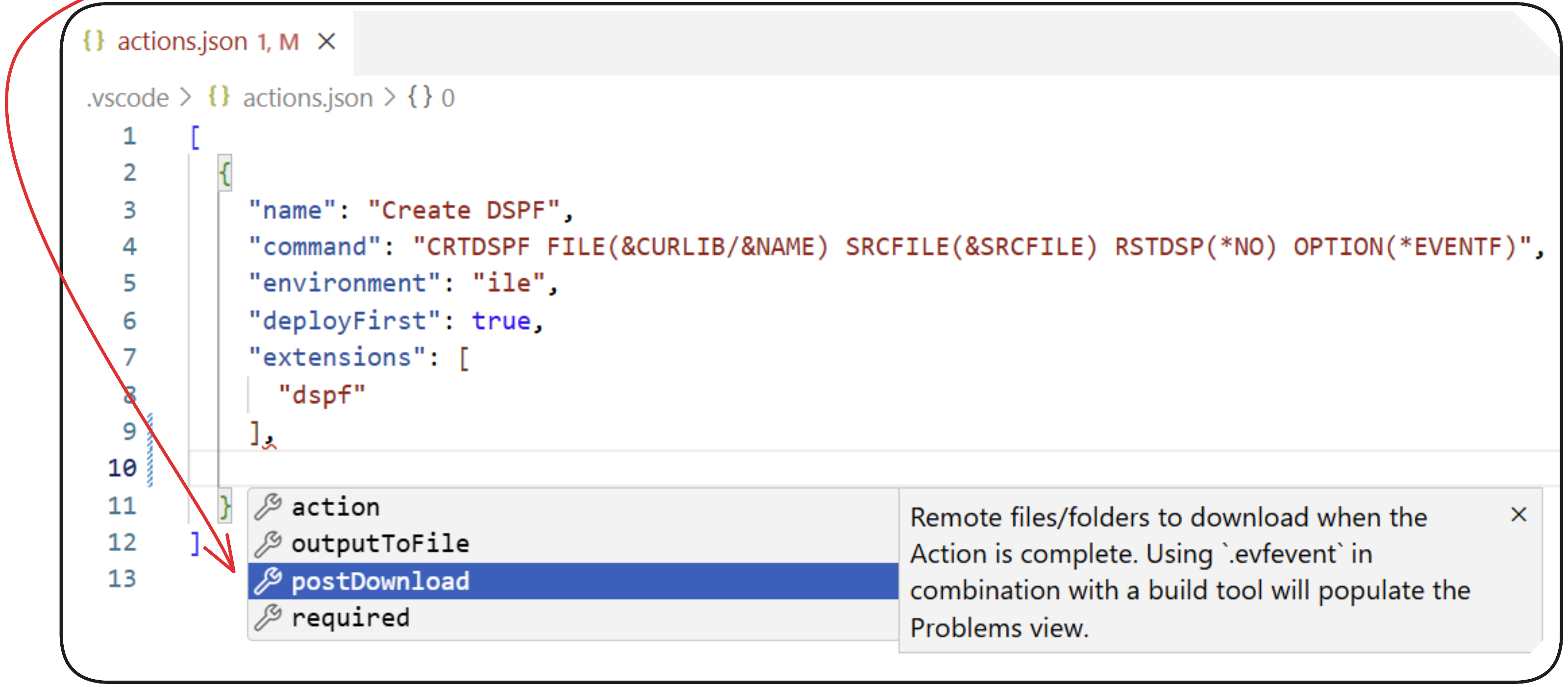
Auto Open File

☑ When enabled, files will automatically be opened when copied or moved/renamed

Contribution Points – JSON Validation


{} package.json 9+, M ×
{} package.json > {} contributes > {} configuration > {} properties > {

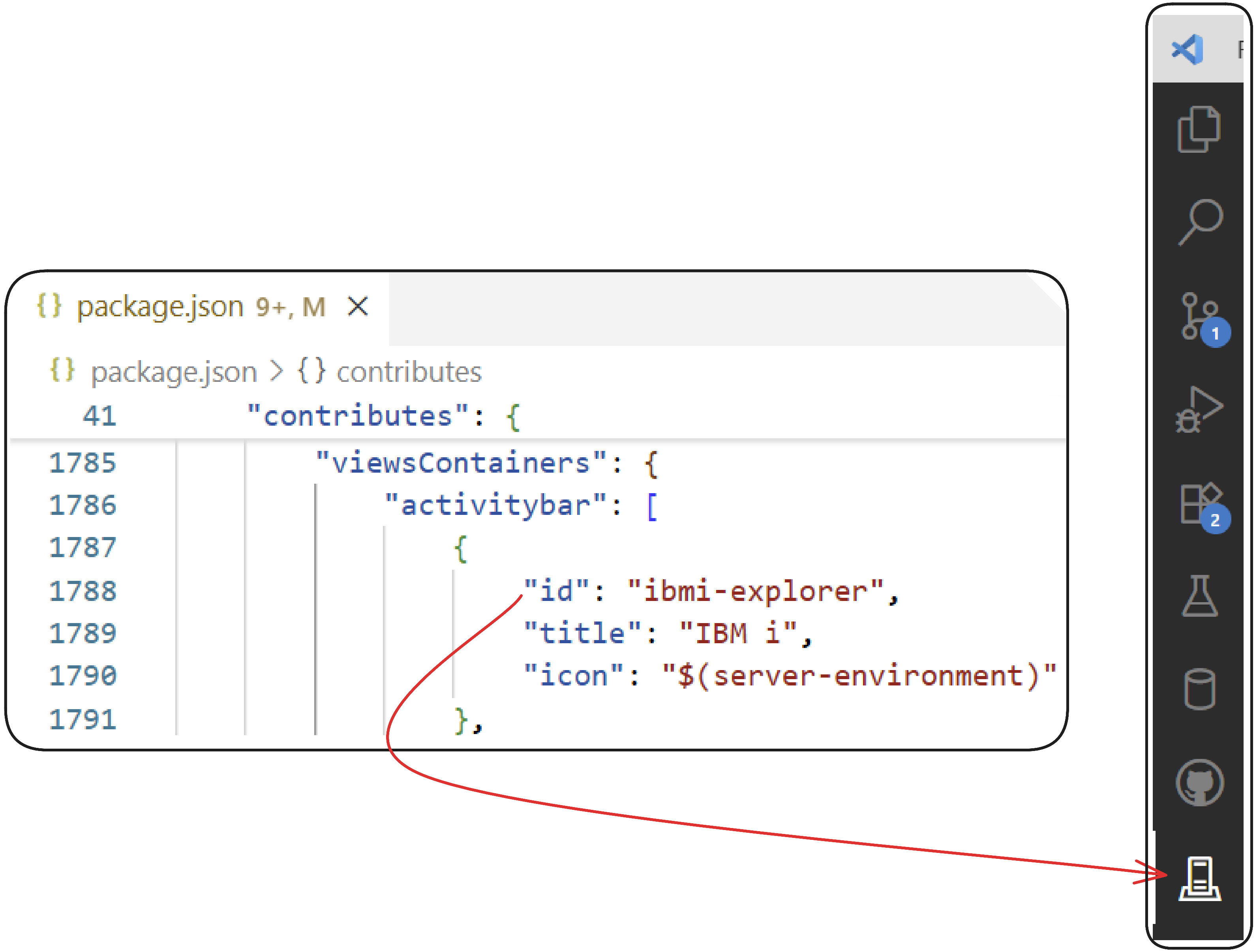
```
    "contributes": {
        "jsonValidation": [
            {
                "fileMatch": [
                    ".vscode/actions.json"
                ],
                "url": "./schemas/actions.json"
            }
        ],
```



```
[
  {
    "name": "Create DSPF",
    "command": "CRTDSPF FILE(&CURLIB/&NAME) SRCFILE(&SRCFILE) RSTDSP(*NO) OPTION(*EVENTF)",
    "environment": "ile",
    "deployFirst": true,
    "extensions": [
      "dspf"
    ],

  }
]
```

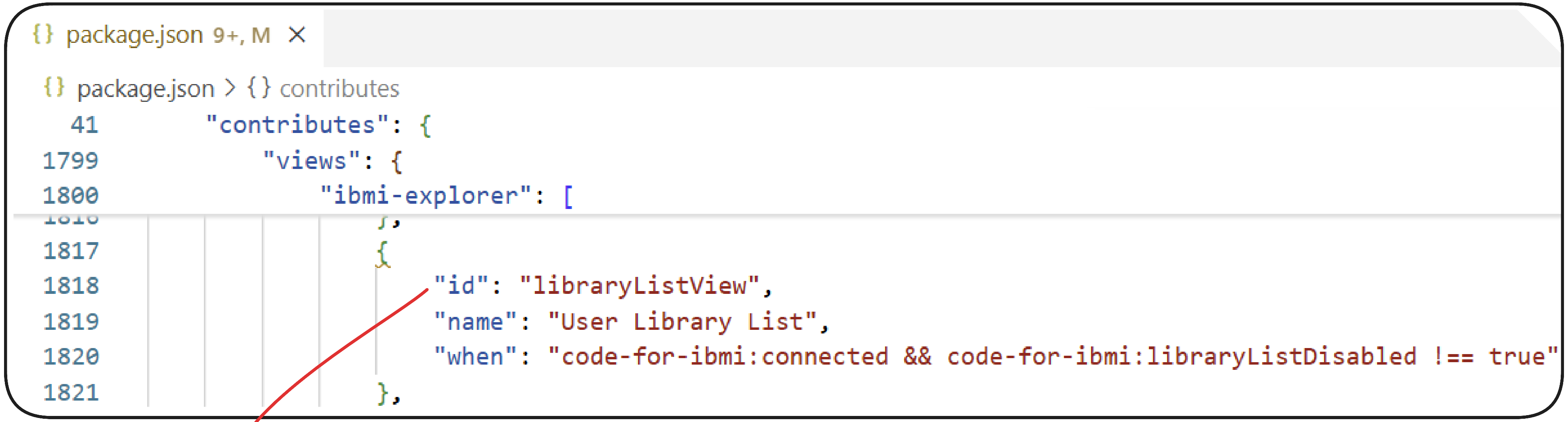
Remote files/folders to download when the Action is complete. Using `.evfevent` in combination with a build tool will populate the Problems view.

Contribution Points – View Containers

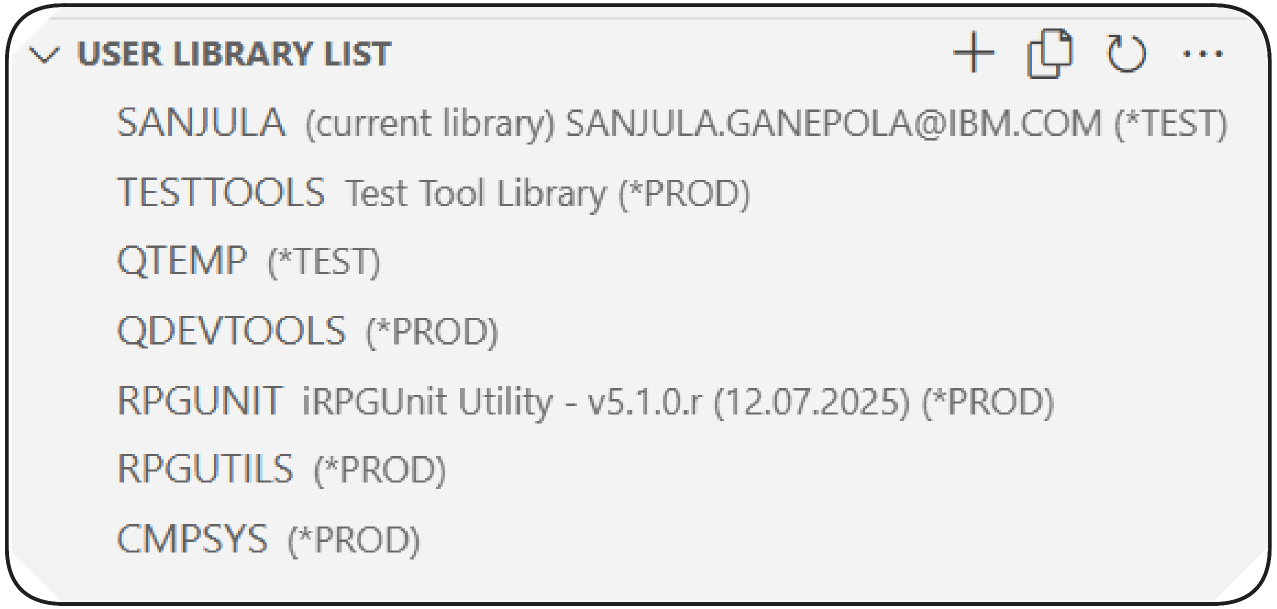


```
        "contributes": {
            "viewsContainers": {
                "activitybar": [
                    {
                        "id": "ibmi-explorer",
                        "title": "IBM i",
                        "icon": "$(server-environment)"
                    },
```

Contribution Points – Views



```
    "contributes": {
        "views": {
            "ibmi-explorer": [
                },
                {
                    "id": "libraryListView",
                    "name": "User Library List",
                    "when": "code-for-ibmi:connected && code-for-ibmi:libraryListDisabled !== true"
                },
```



SANJULA (current library) SANJULA.GANEPOLA@IBM.COM (*TEST)

TESTTOOLS Test Tool Library (*PROD)

QTEMP (*TEST)

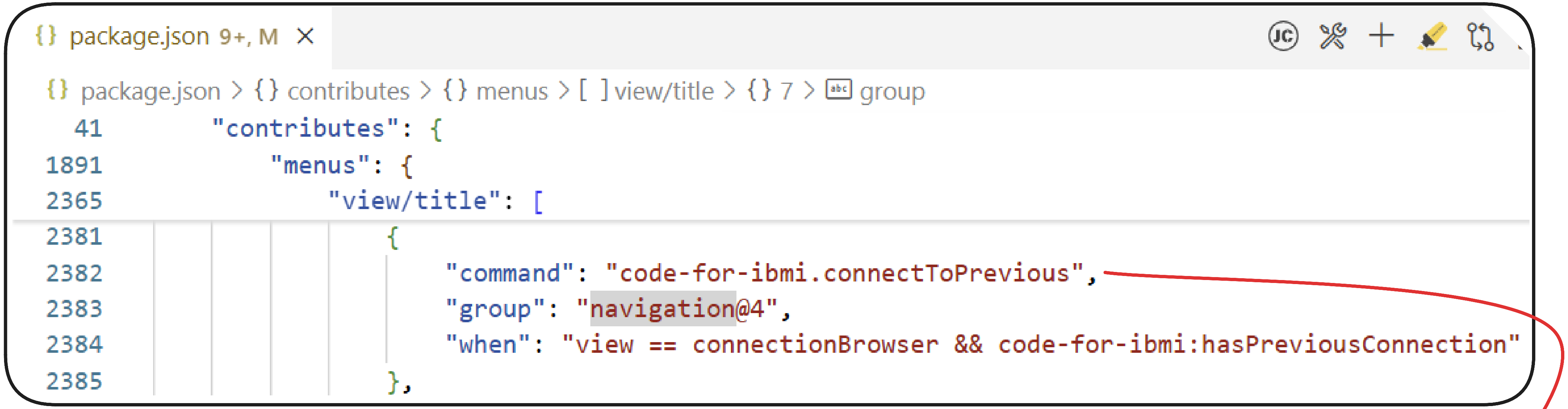
QDEVTOOLS (*PROD)

RPGUNIT iRPGUnit Utility - v5.1.0.r (12.07.2025) (*PROD)

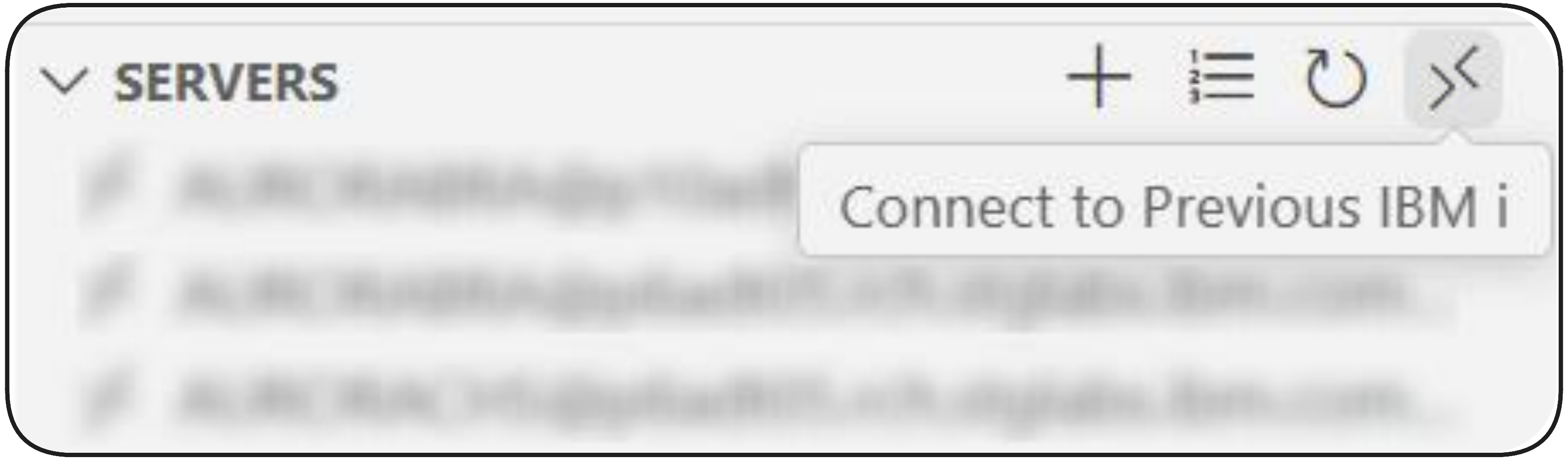
RPGUTILS (*PROD)

CMPSYS (*PROD)

Contribution Points – View title menu

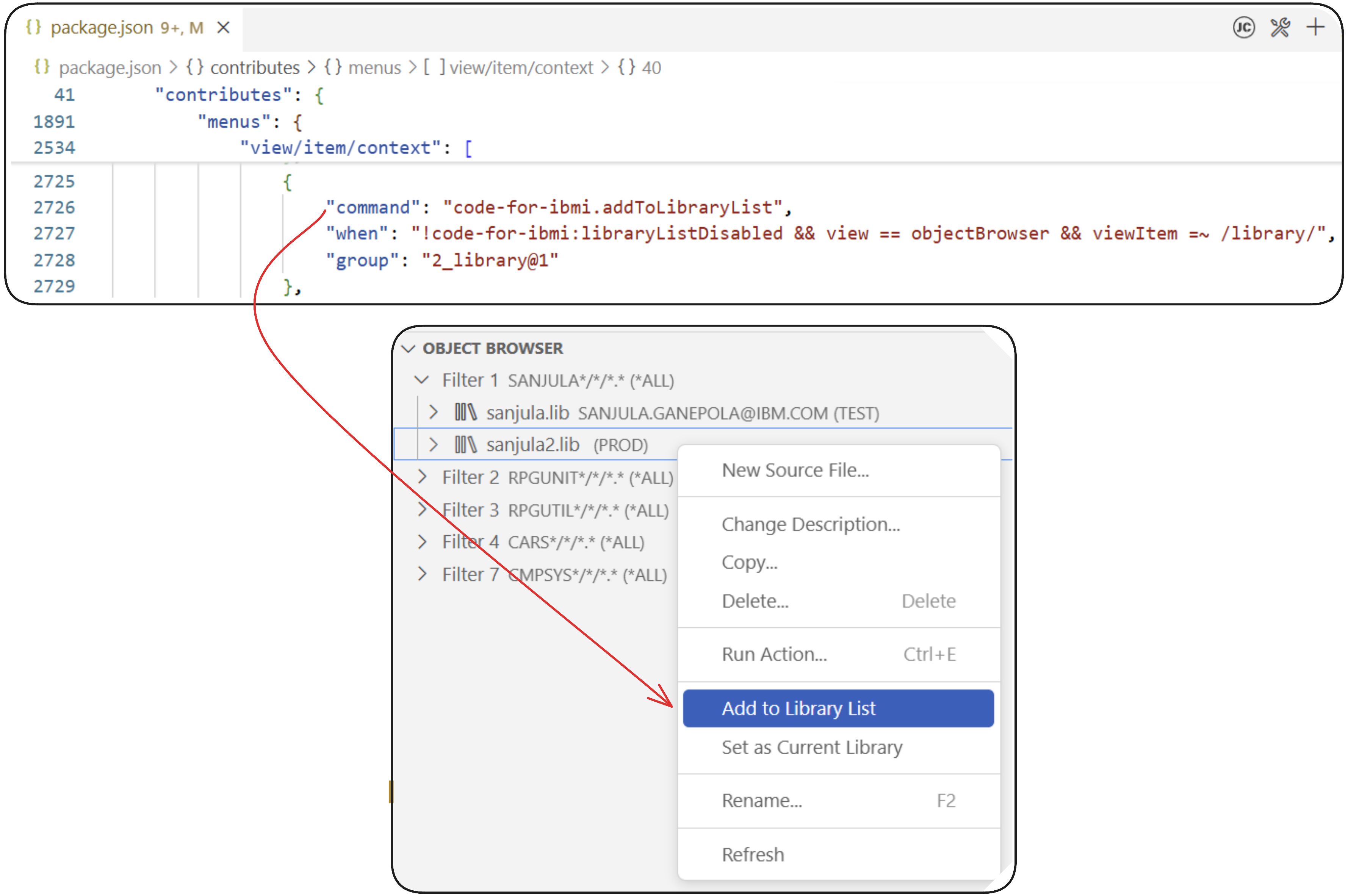

{} package.json 9+, M ×
{} package.json > {} contributes > {} menus > [ ] view/title > {} 7 > group

```
    "contributes": {
        "menus": {
            "view/title": [
                {
                    "command": "code-for-ibmi.connectToPrevious",
                    "group": "navigation@4",
                    "when": "view == connectionBrowser && code-for-ibmi:hasPreviousConnection"
                },
```



Connect to Previous IBM i

Contribution Points – View item context menu

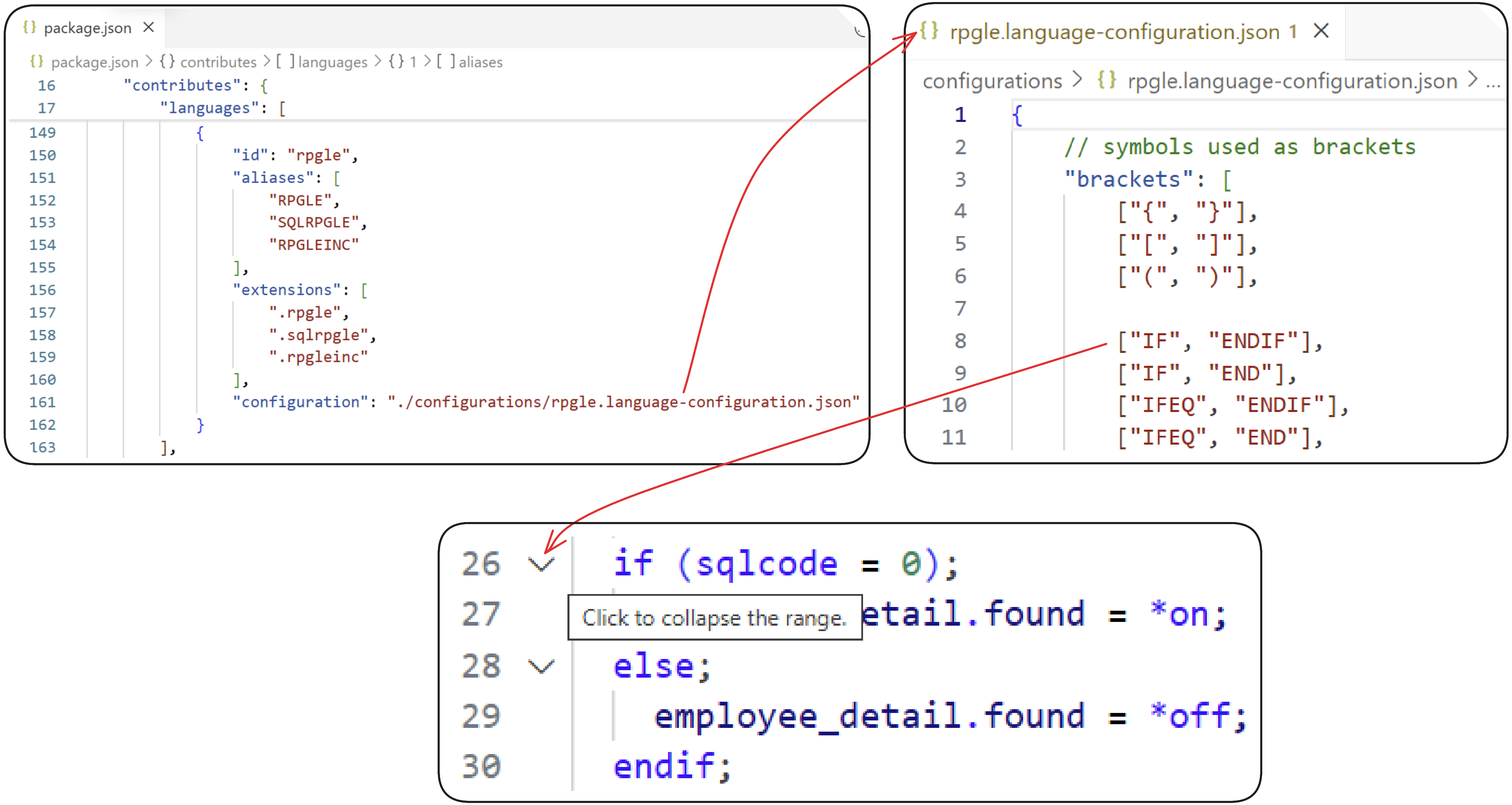


{} package.json 9+, M ×
{} package.json > {} contributes > {} menus > [ ] view/item/context > {} 40

```
    "contributes": {
        "menus": {
            "view/item/context": [
                {
                    "command": "code-for-ibmi.addToLibraryList",
                    "when": "!code-for-ibmi:libraryListDisabled && view == objectBrowser && viewItem =~ /library/",
                    "group": "2_library@1"
                },
```

OBJECT BROWSER
Filter 1 SANJULA*/*/*.* (*ALL)
sanjula.lib SANJULA.GANEPOLA@IBM.COM (TEST)
sanjula2.lib (PROD)
Filter 2 RPGUNIT*/*/*.* (*ALL)
Filter 3 RPGUTIL*/*/*.* (*ALL)
Filter 4 CARS*/*/*.* (*ALL)
Filter 7 CMPSYS*/*/*.* (*ALL)
New Source File...
Change Description...
Copy...
Delete... Delete
Run Action... Ctrl+E
Add to Library List
Set as Current Library
Rename... F2
Refresh

Contribution Points – Languages



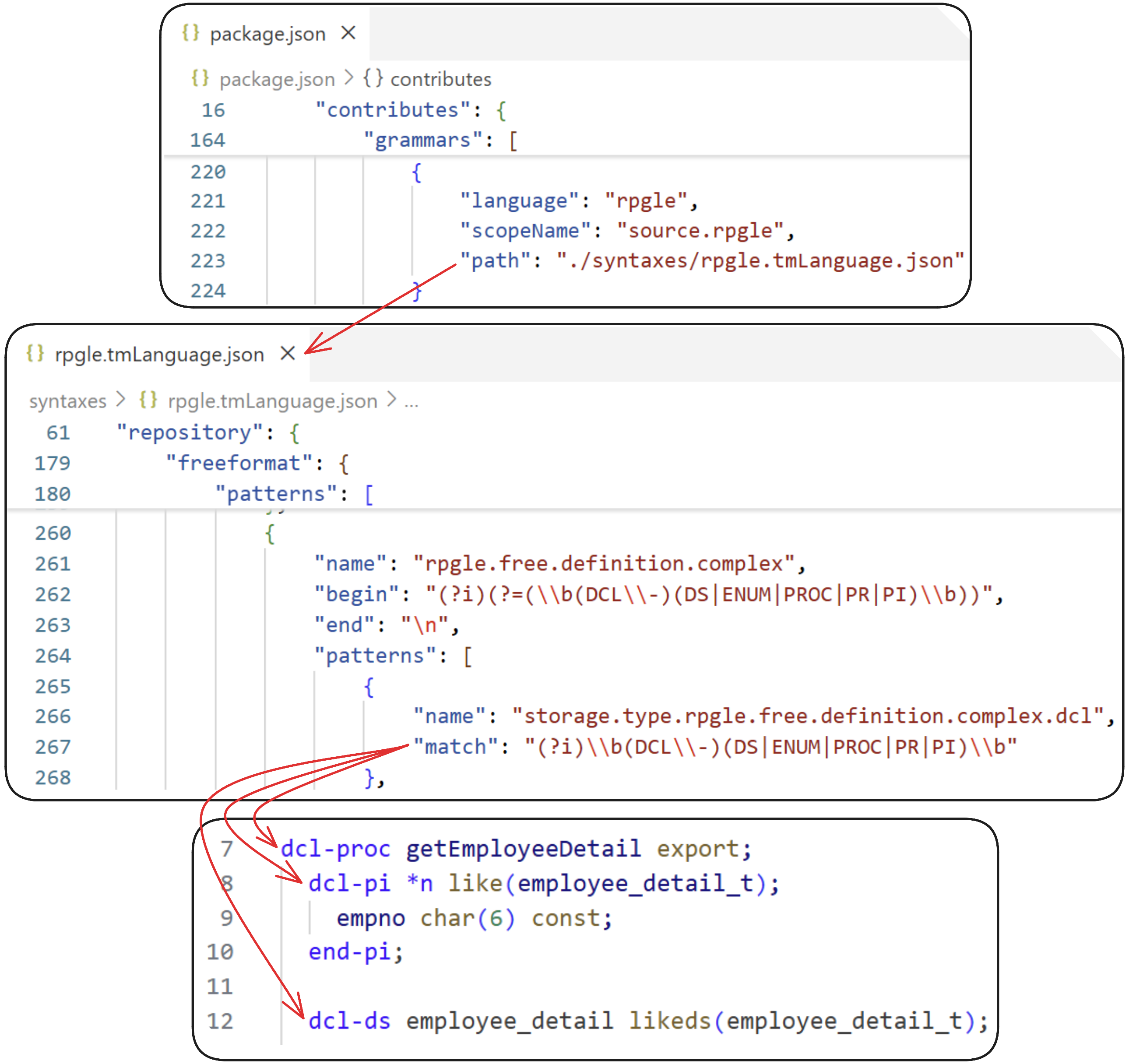
package.json ×
package.json > contributes > languages > 1 > aliases
rpgle.language-configuration.json 1 ×
configurations > rpgle.language-configuration.json > ...
Click to collapse the range.

```
    "contributes": {
        "languages": [
            {
                "id": "rpgle",
                "aliases": [
                    "RPGLE",
                    "SQLRPGLE",
                    "RPGLEINC"
                ],
                "extensions": [
                    ".rpgle",
                    ".sqlrpgle",
                    ".rpgleinc"
                ],
                "configuration": "./configurations/rpgle.language-configuration.json"
            }
        ],
```

```
{
    // symbols used as brackets
    "brackets": [
        ["{", "}"],
        ["[", "]"],
        ["(", ")"],

        ["IF", "ENDIF"],
        ["IF", "END"],
        ["IFEQ", "ENDIF"],
        ["IFEQ", "END"],
```

```
if (sqlcode = 0);
                 etail.found = *on;
else;
  employee_detail.found = *off;
endif;
```

Contribution Points – Grammers



package.json ×
package.json > contributes

```
"contributes": {
    "grammars": [
        {
            "language": "rpgle",
            "scopeName": "source.rpgle",
            "path": "./syntaxes/rpgle.tmLanguage.json"
        }
```

rpgle.tmLanguage.json ×
syntaxes > rpgle.tmLanguage.json > ...

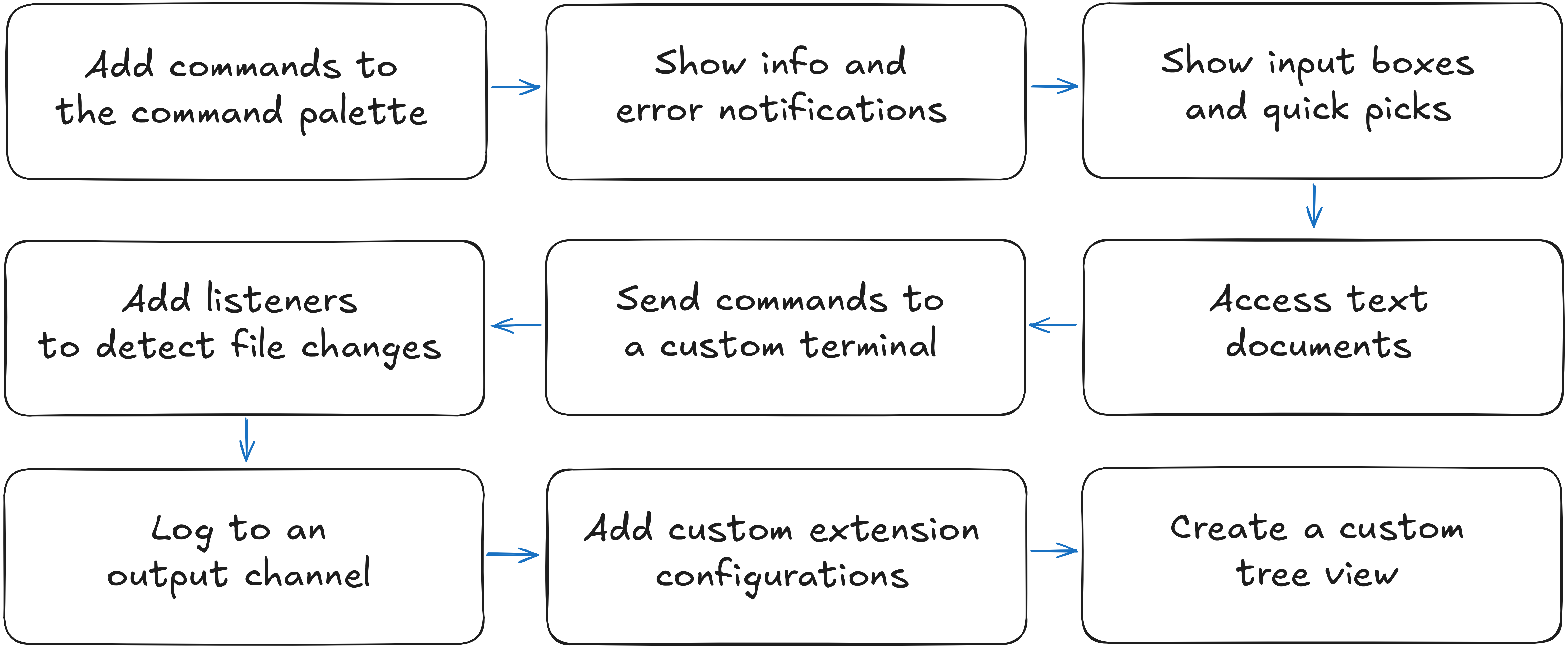
```
"repository": {
    "freeformat": {
        "patterns": [
            {
                "name": "rpgle.free.definition.complex",
                "begin": "(?i)(?=(\\b(DCL\\-)(DS|ENUM|PROC|PR|PI)\\b))",
                "end": "\n",
                "patterns": [
                    {
                        "name": "storage.type.rpgle.free.definition.complex.dcl",
                        "match": "(?i)\\b(DCL\\-)(DS|ENUM|PROC|PR|PI)\\b"
                    },
```

```
dcl-proc getEmployeeDetail export;
  dcl-pi *n like(employee_detail_t);
    empno char(6) const;
  end-pi;

  dcl-ds employee_detail likeds(employee_detail_t);
```

Contribution Points – Check the docs for more

https://code.visualstudio.com/api/references/contribution-points

Let’s experiment with the VS Code API



Add commands to the command palette

Show info and error notifications

Show input boxes and quick picks

Access text documents

Send commands to a custom terminal

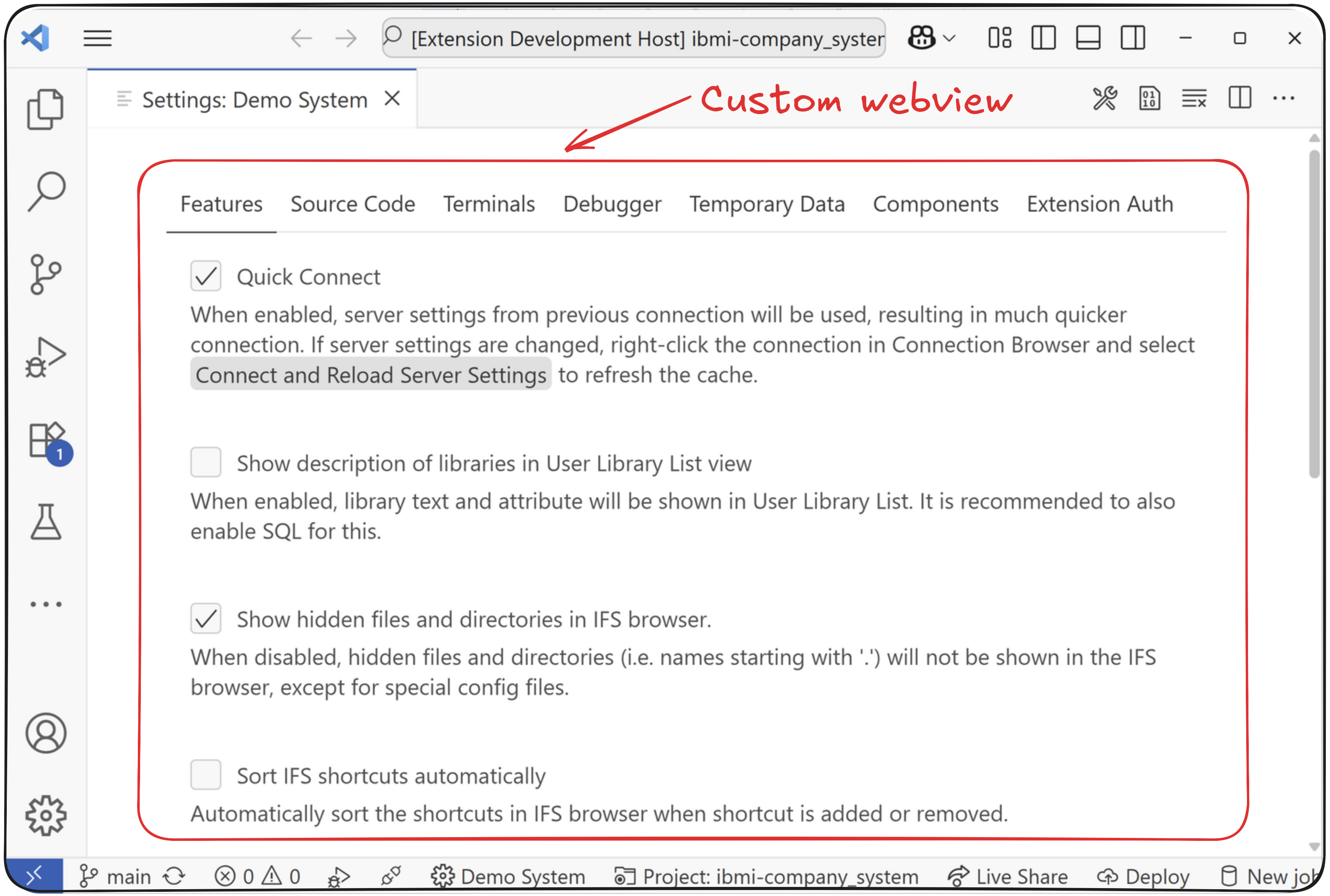
Add listeners to detect file changes

Log to an output channel

Add custom extension configurations

Create a custom tree view

VS Code API – Example webview in Code for IBM i



☑ Quick Connect

When enabled, server settings from previous connection will be used, resulting in much quicker connection. If server settings are changed, right-click the connection in Connection Browser and select `Connect and Reload Server Settings` to refresh the cache.

□ Show description of libraries in User Library List view

When enabled, library text and attribute will be shown in User Library List. It is recommended to also enable SQL for this.

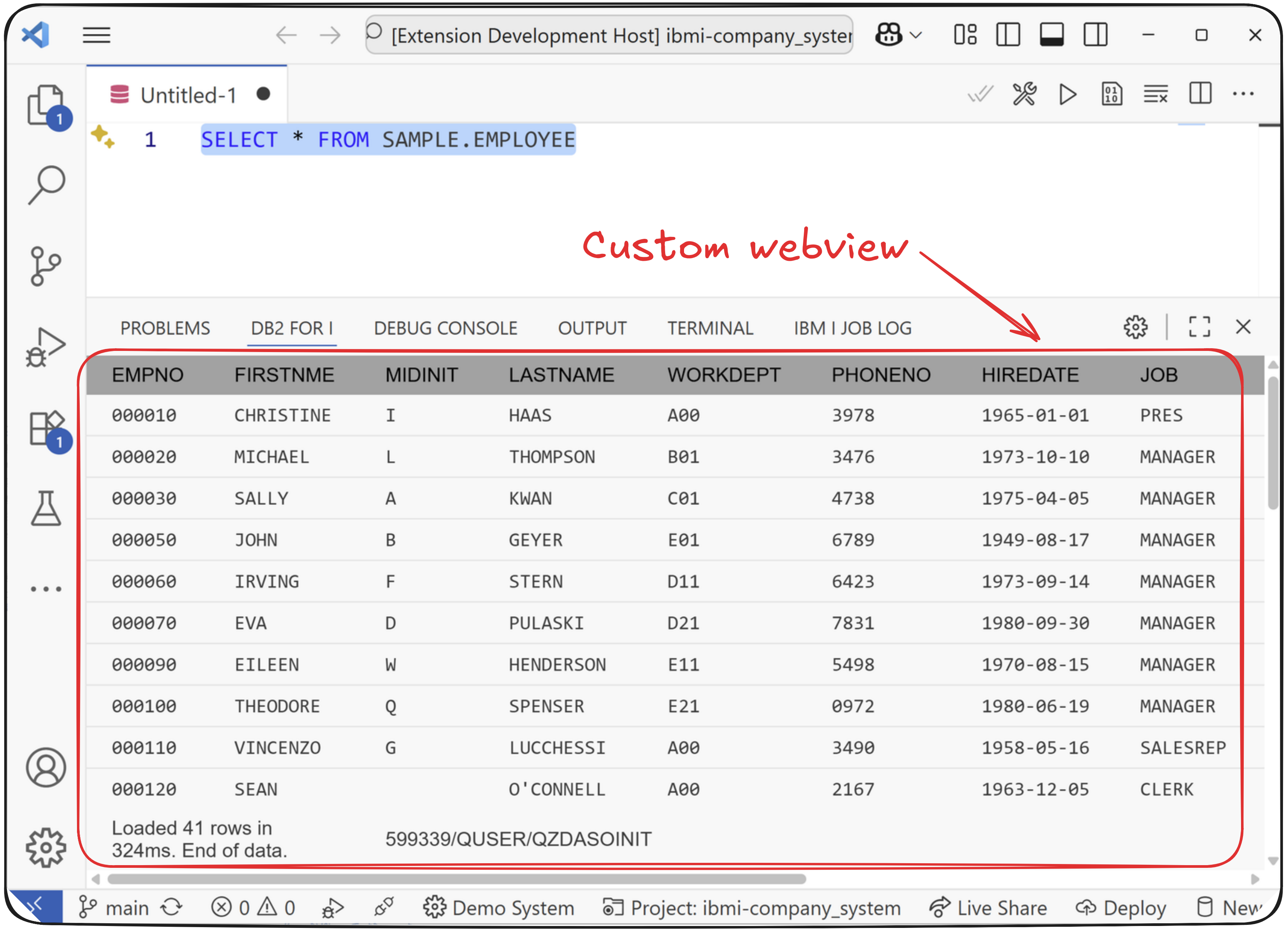
☑ Show hidden files and directories in IFS browser.

When disabled, hidden files and directories (i.e. names starting with '.') will not be shown in the IFS browser, except for special config files.

□ Sort IFS shortcuts automatically

Automatically sort the shortcuts in IFS browser when shortcut is added or removed.

VS Code API – Example webview in Db2 for IBM i



```
SELECT * FROM SAMPLE.EMPLOYEE
```

Custom webview

EMPNO	FIRSTNME	MIDINIT	LASTNAME	WORKDEPT	PHONENO	HIREDATE	JOB
000010	CHRISTINE	I	HAAS	A00	3978	1965-01-01	PRES
000020	MICHAEL	L	THOMPSON	B01	3476	1973-10-10	MANAGER
000030	SALLY	A	KWAN	C01	4738	1975-04-05	MANAGER
000050	JOHN	B	GEYER	E01	6789	1949-08-17	MANAGER
000060	IRVING	F	STERN	D11	6423	1973-09-14	MANAGER
000070	EVA	D	PULASKI	D21	7831	1980-09-30	MANAGER
000090	EILEEN	W	HENDERSON	E11	5498	1970-08-15	MANAGER
000100	THEODORE	Q	SPENSER	E21	0972	1980-06-19	MANAGER
000110	VINCENZO	G	LUCCHESSI	A00	3490	1958-05-16	SALESREP
000120	SEAN		O'CONNELL	A00	2167	1963-12-05	CLERK

Loaded 41 rows in 324ms. End of data. 599339/QUSER/QZDASOINIT

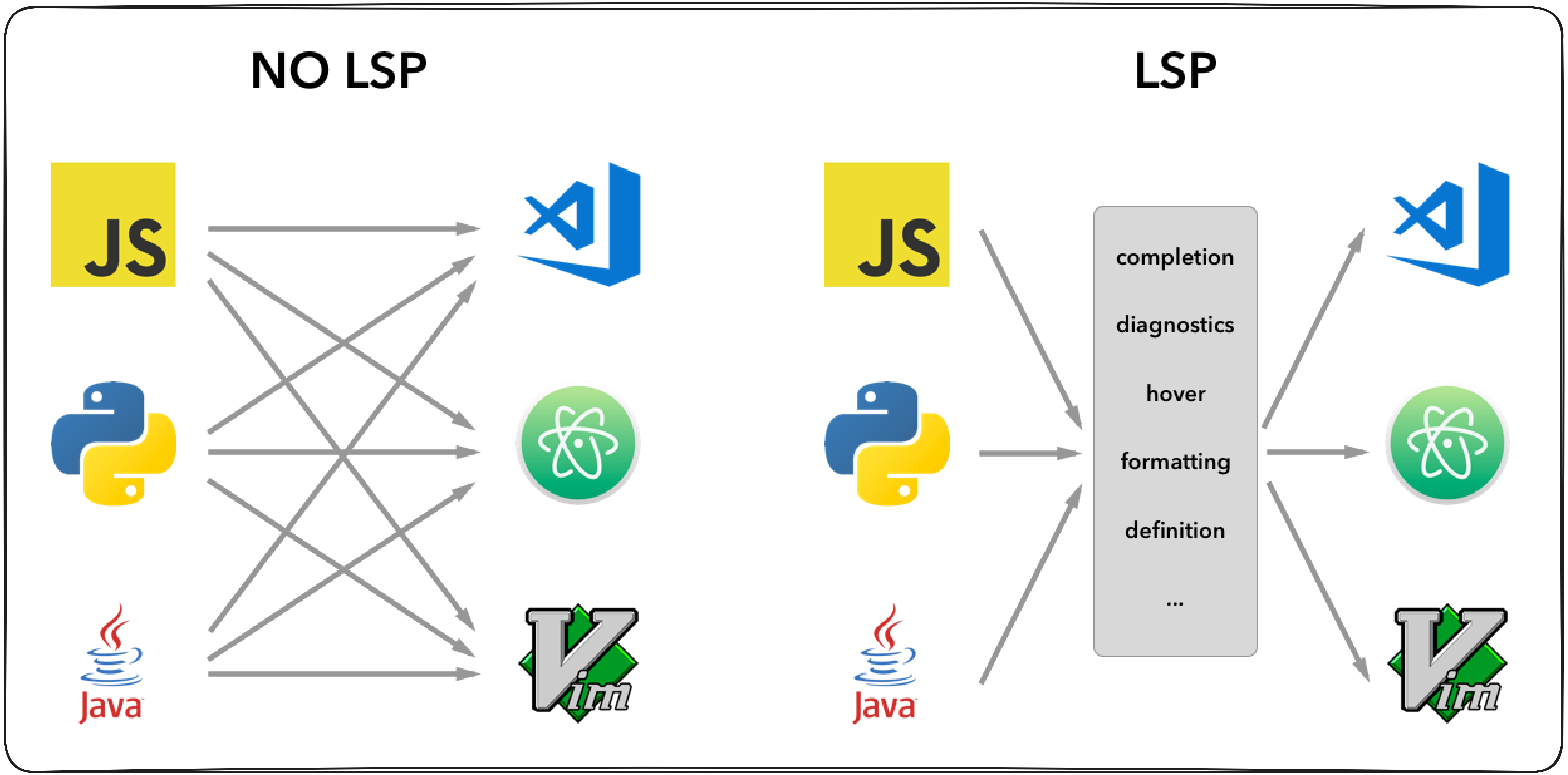
Language Extensions – 3 core problems

Language servers are usually implemented in their native programming languages, but VS Code has a Node.js runtime.

Language features can be resource intensive (building a Abstract Syntax Tree, performing static analysis, etc.). Need VS Code's performance to be unaffected.

Integrating multiple language toolings with multiple code editors could involve significant effort.

Language Extensions – Benefit of using an LSP



completion

diagnostics

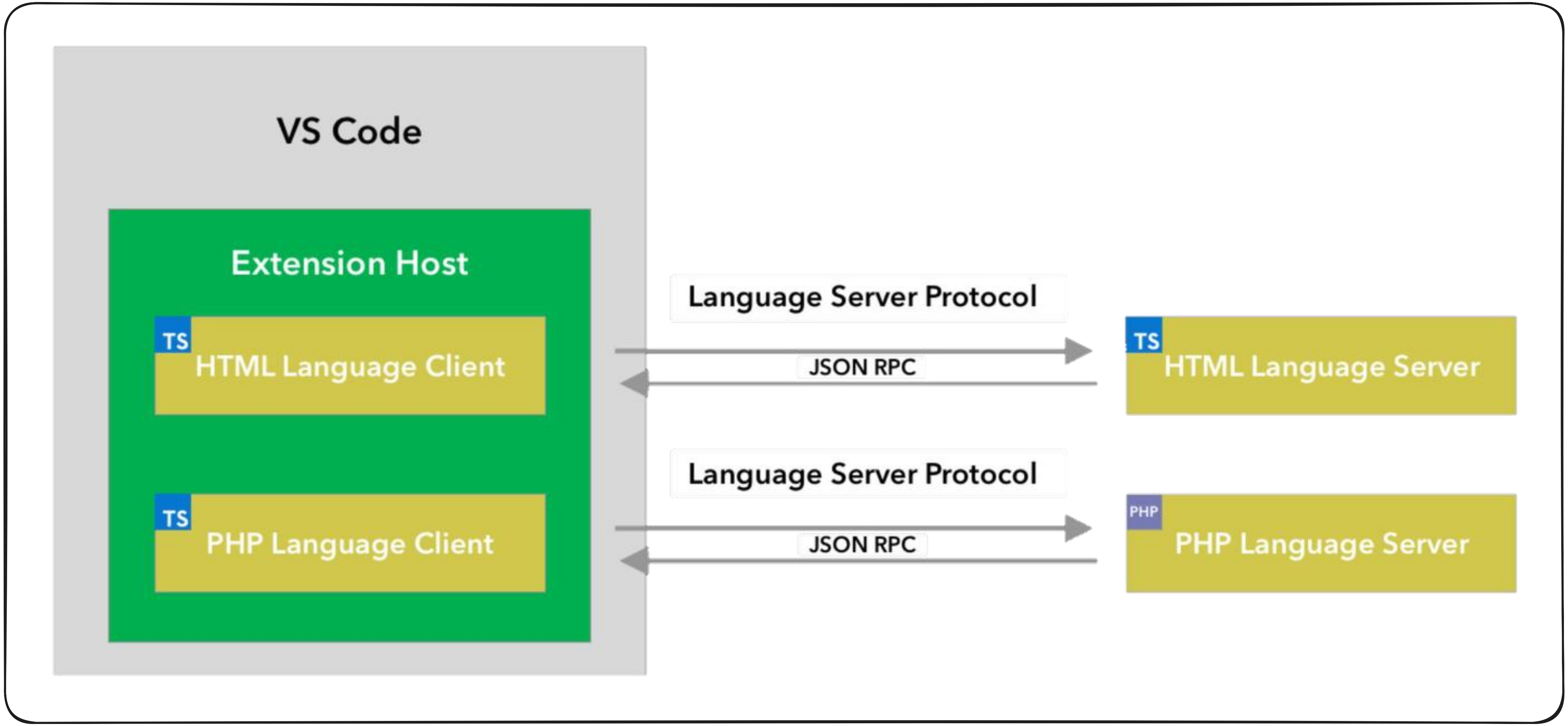
hover

formatting

definition

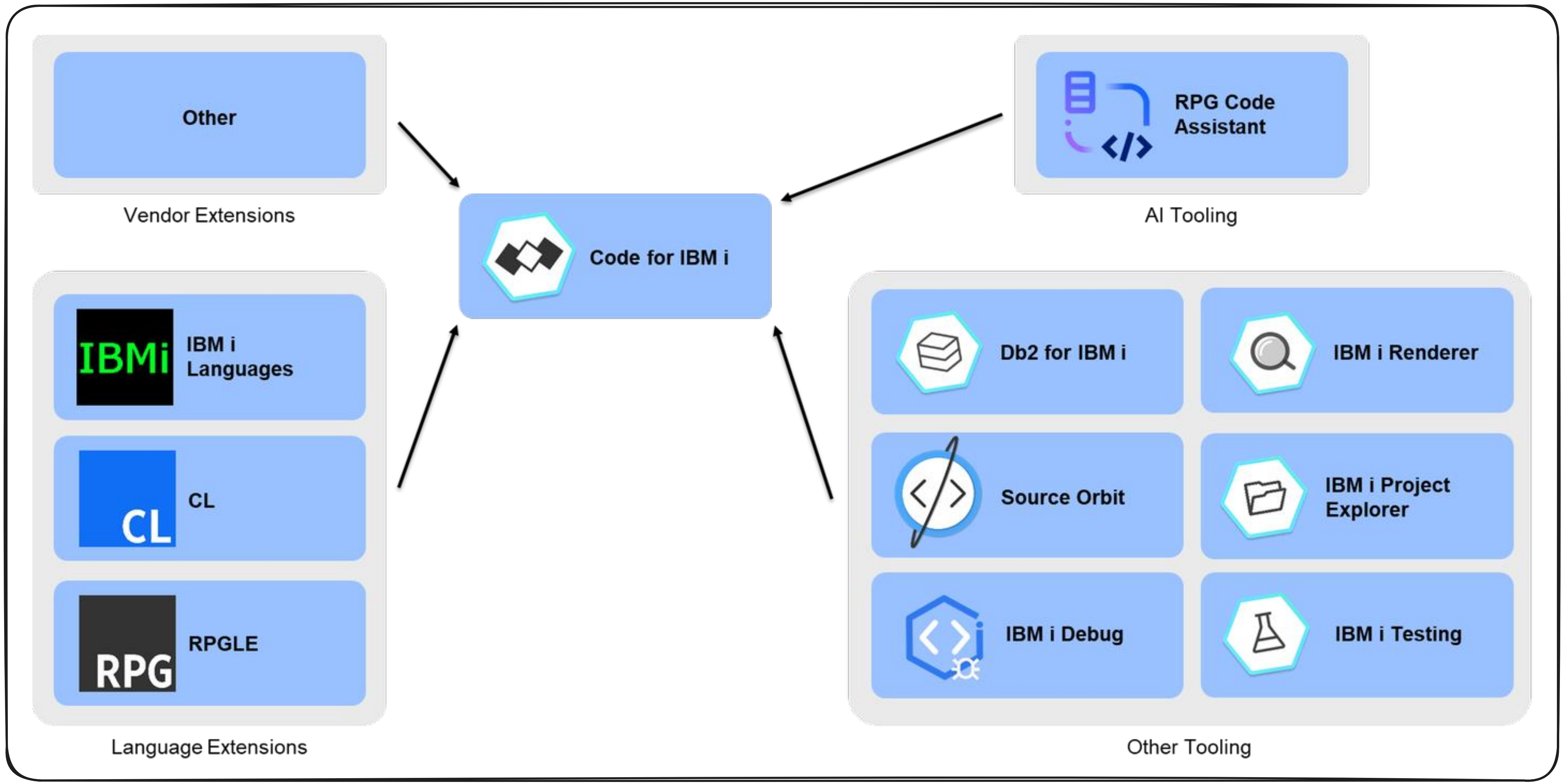
…

Language Extensions – How does an LSP work?



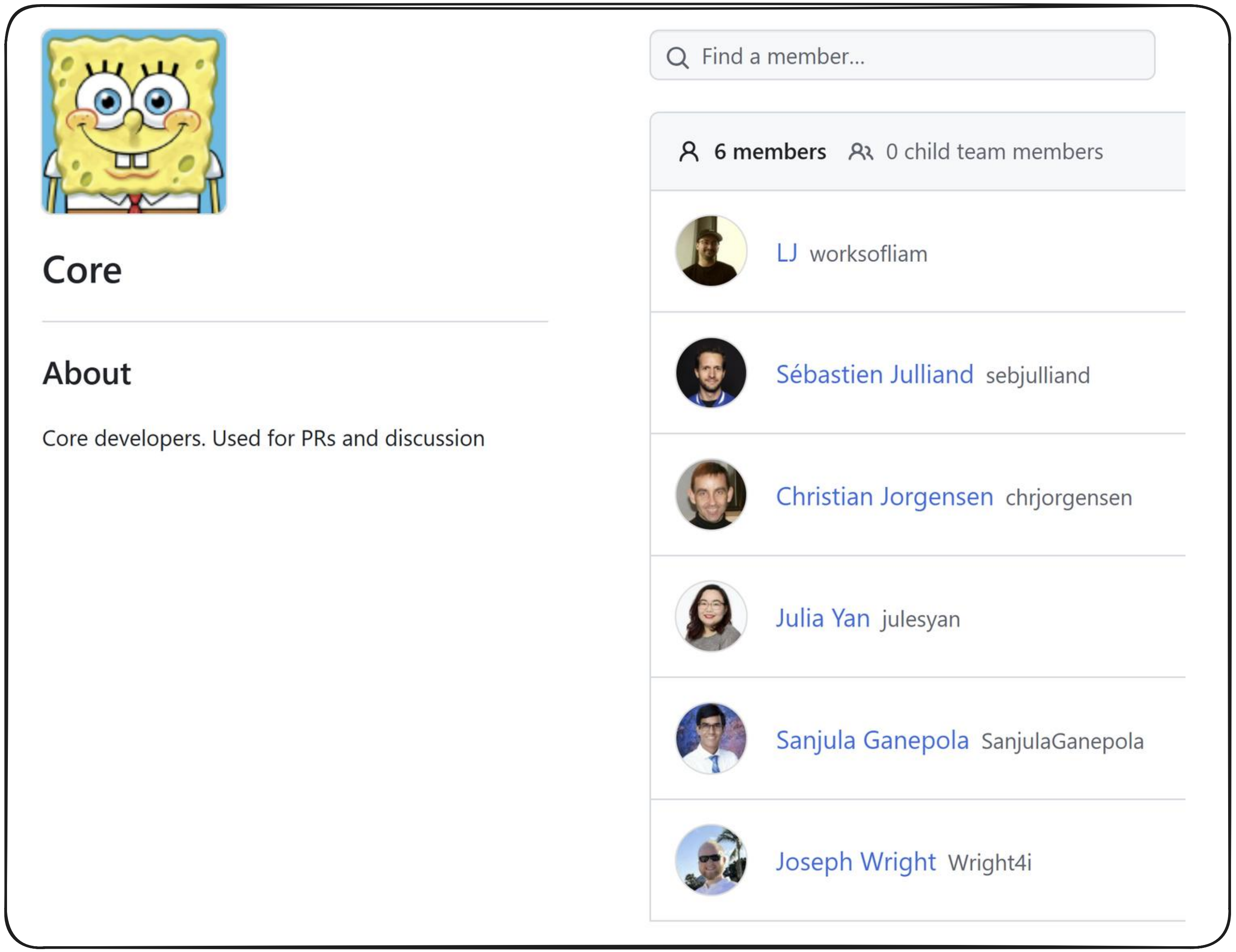
Contributing to Code for IBM i

Family of extensions



The core team

We help steer the ship!



Core

About

Core developers. Used for PRs and discussion

Find a member...

6 members 0 child team members

LJ worksofliam

Sébastien Julliand sebjulliand

Christian Jorgensen chrjorgensen

Julia Yan julesyan

Sanjula Ganepola SanjulaGanepola

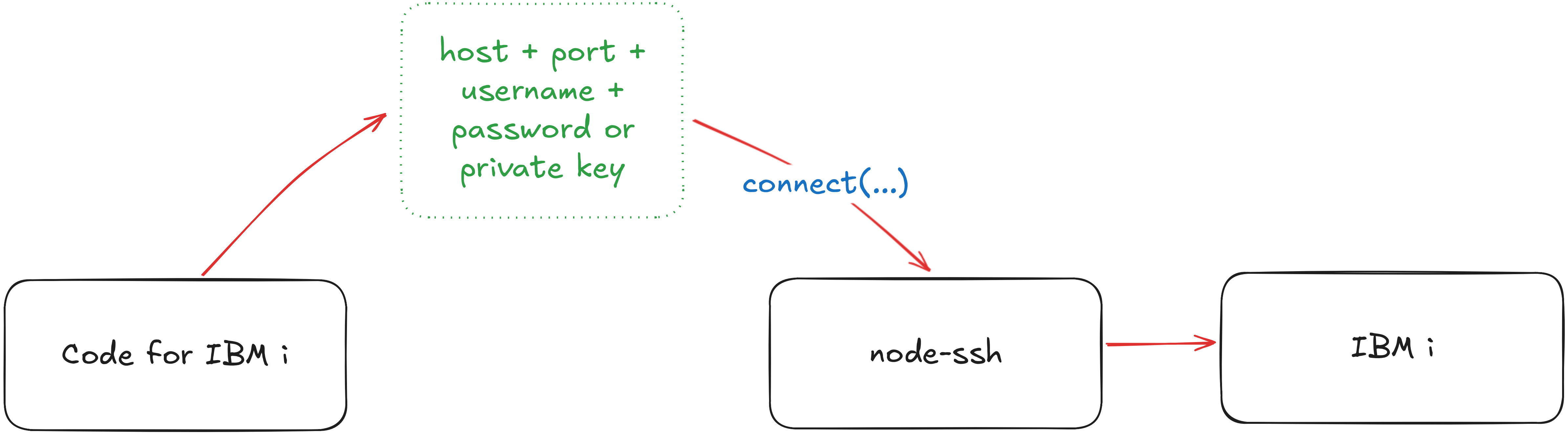
Joseph Wright Wright4i

Community collaboration!

57 total contributors to Code for IBM i!!

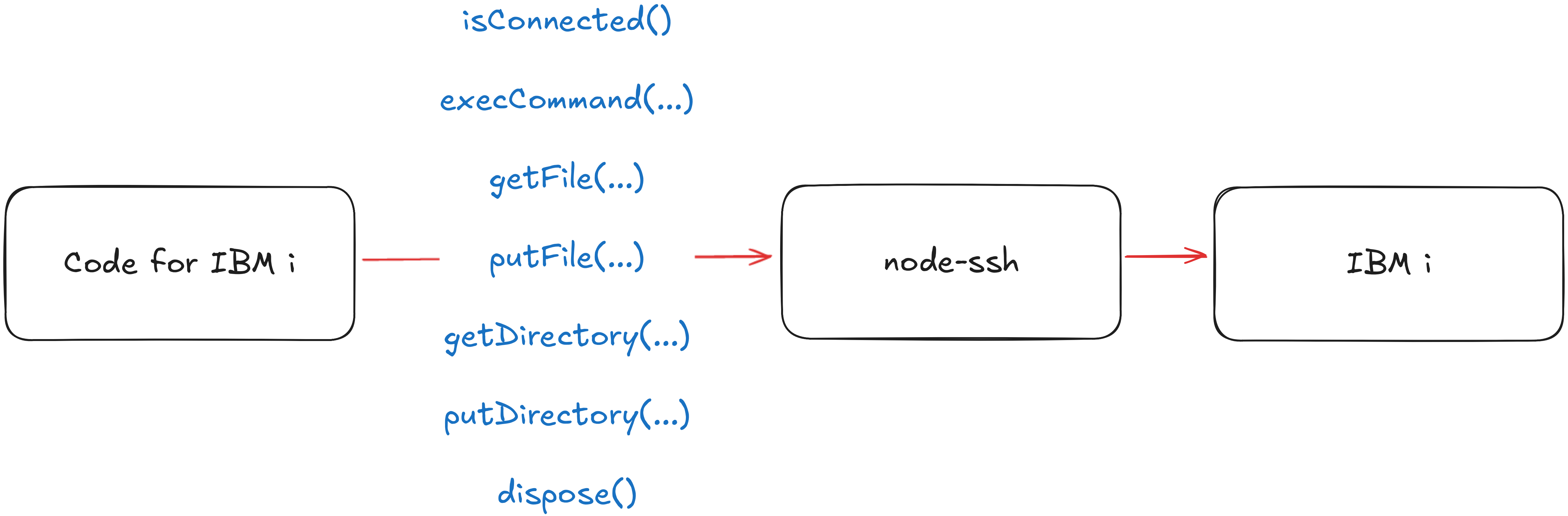
How does Code for IBM i connect to an IBM i?

https://github.com/codefori/vscode-ibmi/blob/master/src/api/IBMi.ts#L254



How does it use Node-SSH?

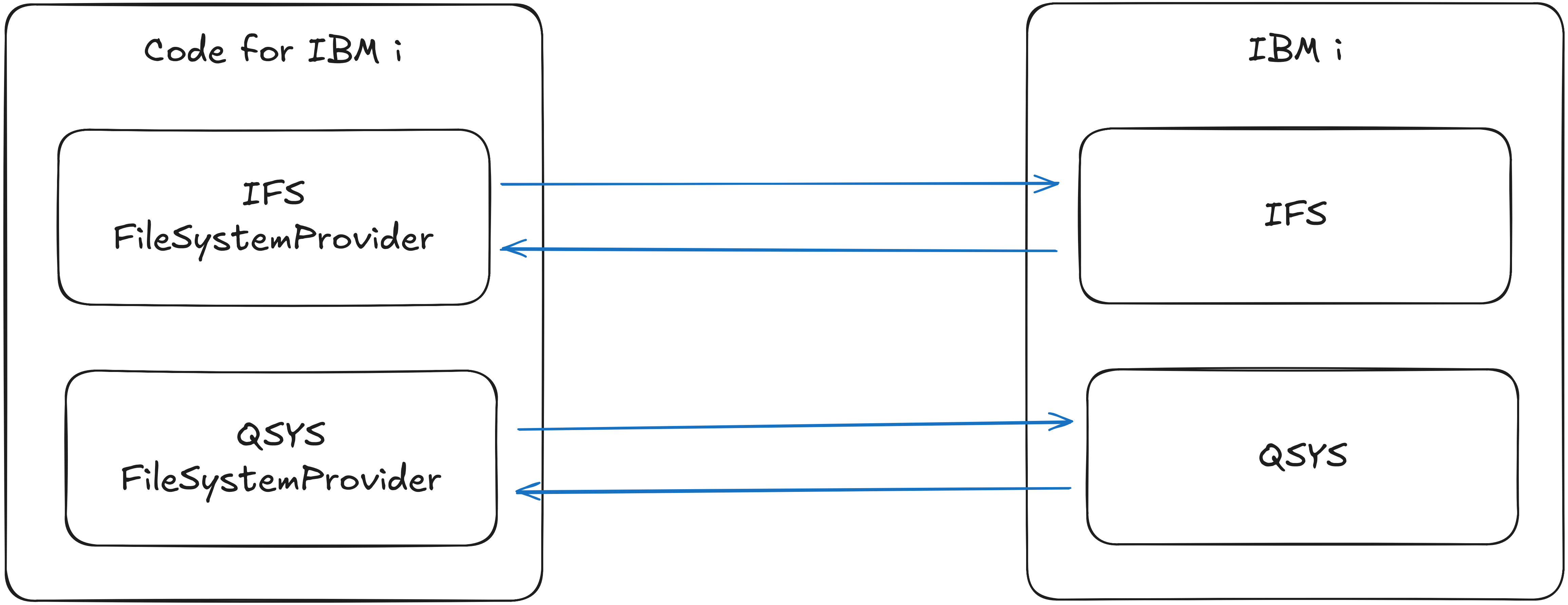
https://github.com/codefori/vscode-ibmi/blob/master/src/api/IBMi.ts#L317



File system providers

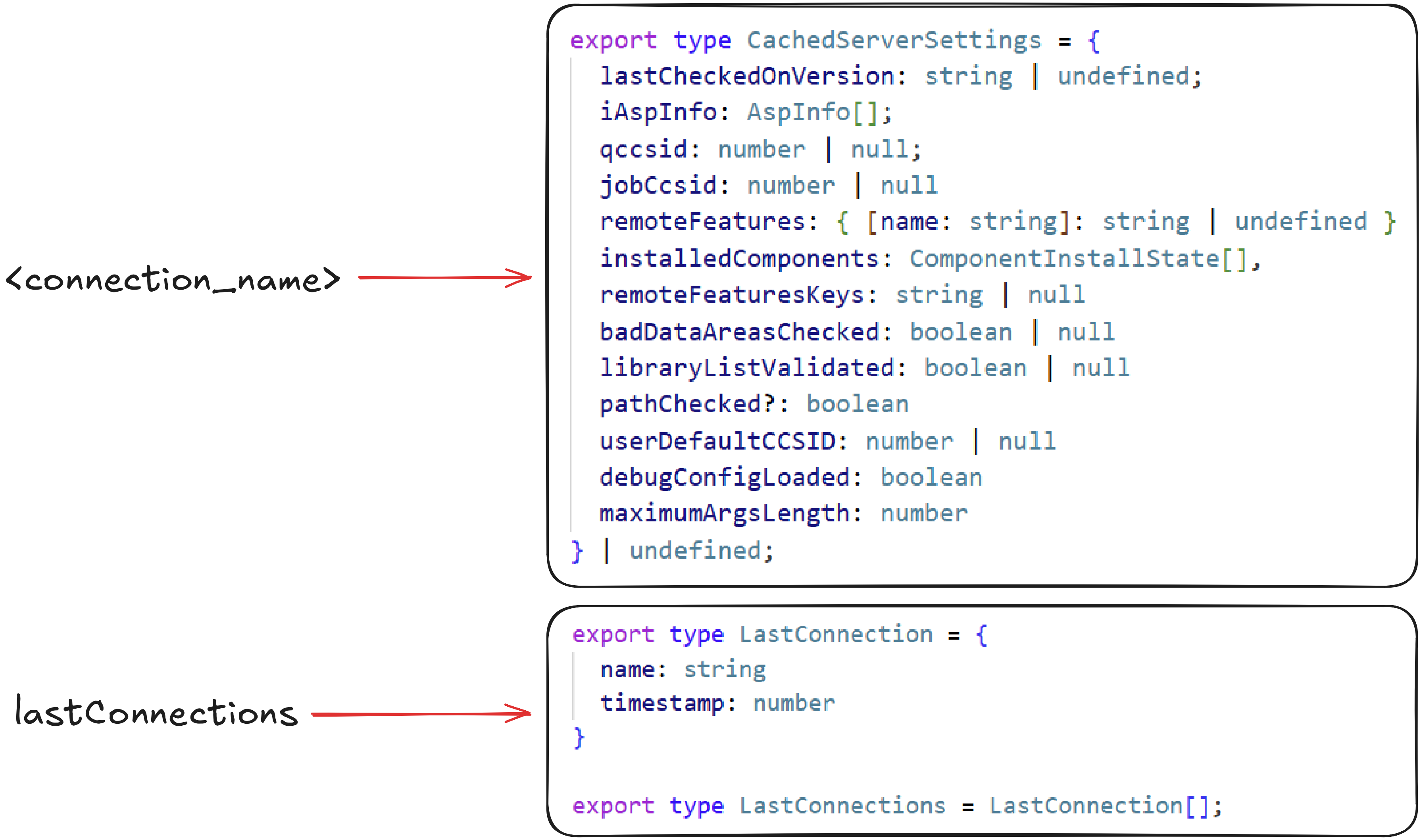
https://github.com/codefori/vscode-ibmi/blob/master/src/filesystems/ifsFs.ts

https://github.com/codefori/vscode-ibmi/blob/master/src/filesystems/qsys/QSysFs.ts



Internal storage

Internal Storage - Key values stored in extension's globalState



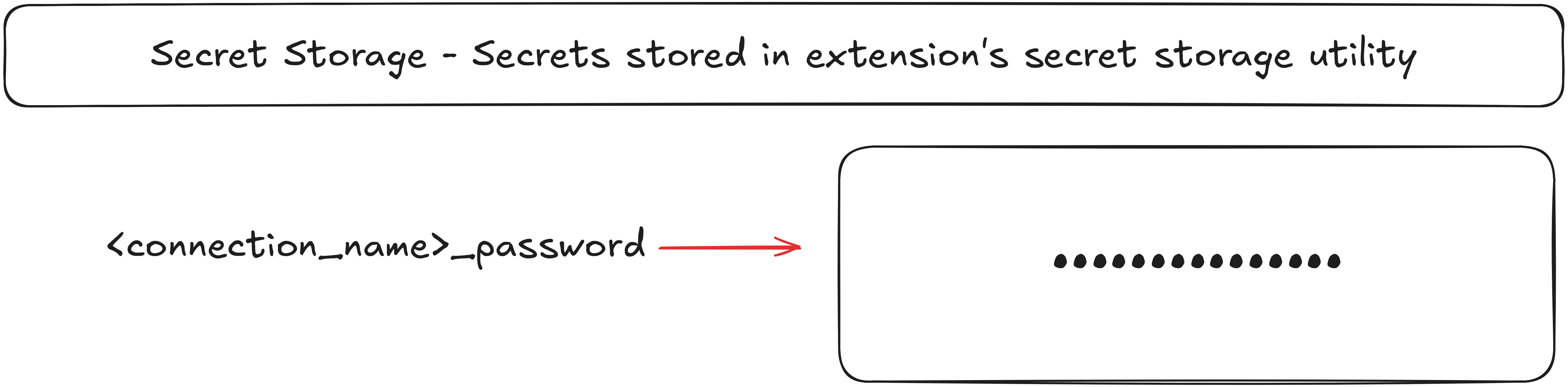
<connection_name>

```
export type CachedServerSettings = {
  lastCheckedOnVersion: string | undefined;
  iAspInfo: AspInfo[];
  qccsid: number | null;
  jobCcsid: number | null
  remoteFeatures: { [name: string]: string | undefined }
  installedComponents: ComponentInstallState[],
  remoteFeaturesKeys: string | null
  badDataAreasChecked: boolean | null
  libraryListValidated: boolean | null
  pathChecked?: boolean
  userDefaultCCSID: number | null
  debugConfigLoaded: boolean
  maximumArgsLength: number
} | undefined;
```

lastConnections

```
export type LastConnection = {
  name: string
  timestamp: number
}

export type LastConnections = LastConnection[];
```

Secret storage



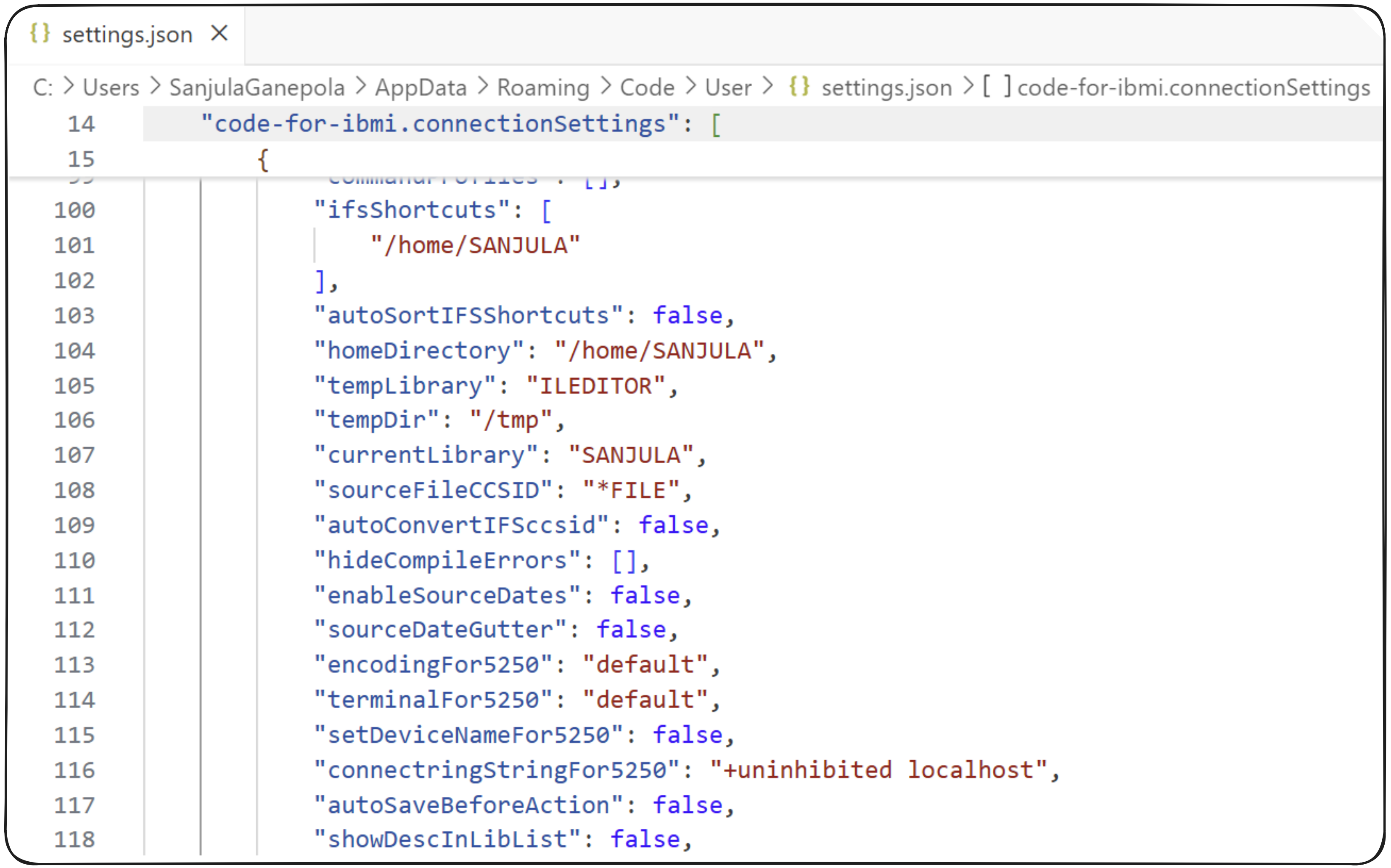
Secret Storage - Secrets stored in extension's secret storage utility

<connection_name>_password

•••••••••••••••

Connection settings

Connection Settings - Configurations for different connections


{} settings.json ×
C: > Users > SanjulaGanepola > AppData > Roaming > Code > User > {} settings.json > [ ] code-for-ibmi.connectionSettings

```
    "code-for-ibmi.connectionSettings": [
        {
            "commandProfiles": [],
            "ifsShortcuts": [
                "/home/SANJULA"
            ],
            "autoSortIFSShortcuts": false,
            "homeDirectory": "/home/SANJULA",
            "tempLibrary": "ILEDITOR",
            "tempDir": "/tmp",
            "currentLibrary": "SANJULA",
            "sourceFileCCSID": "*FILE",
            "autoConvertIFSccsid": false,
            "hideCompileErrors": [],
            "enableSourceDates": false,
            "sourceDateGutter": false,
            "encodingFor5250": "default",
            "terminalFor5250": "default",
            "setDeviceNameFor5250": false,
            "connectringStringFor5250": "+uninhibited localhost",
            "autoSaveBeforeAction": false,
            "showDescInLibList": false,
```

Tree views

https://github.com/codefori/vscode-ibmi/blob/master/src/ui/views

Servers

Object Browser

IFS Browser

User Library List

Profiles

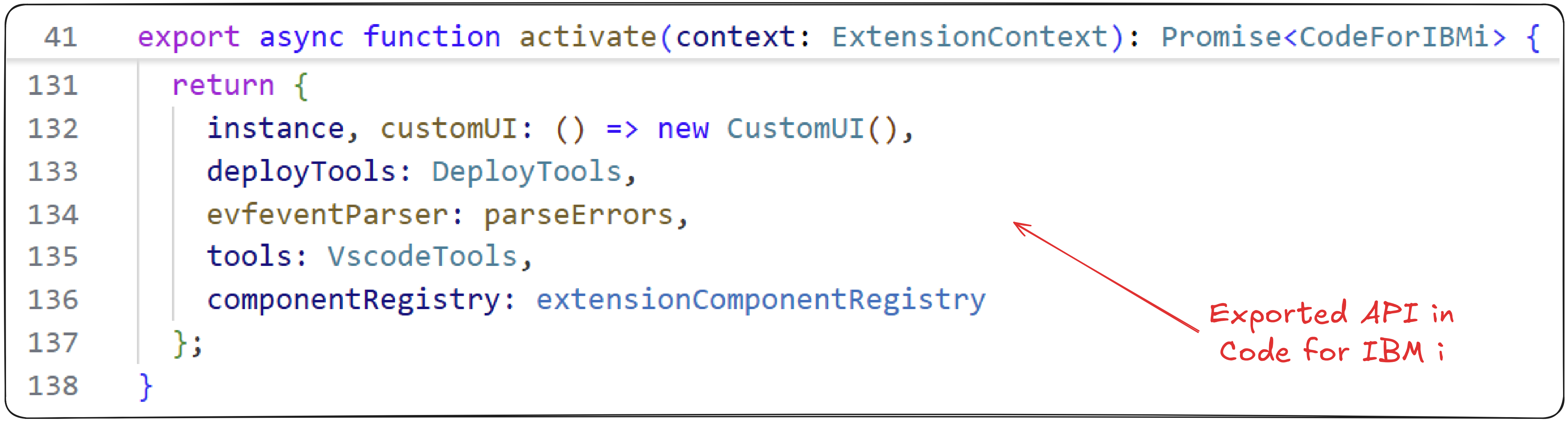
Results

Help and Support

IBM i debugger

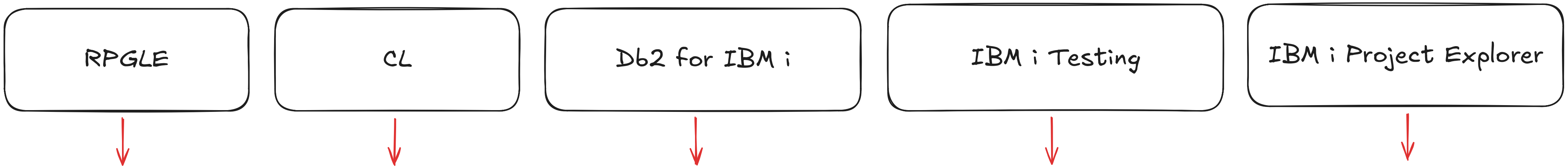
Exported API

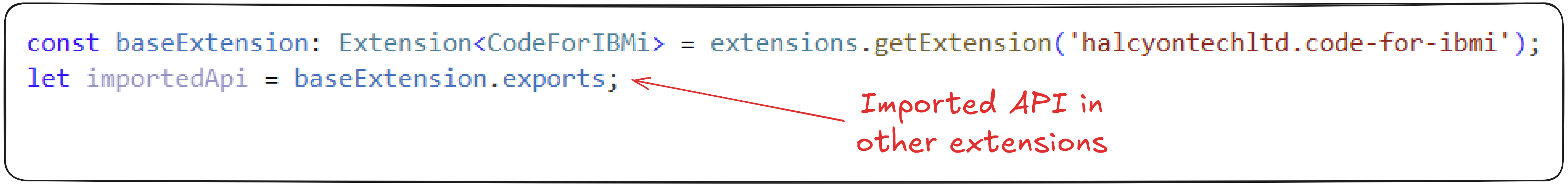
https://github.com/IBM/vscode-ibmi-testing/blob/main/src/extension.ts#L22



```typescript
export async function activate(context: ExtensionContext): Promise<CodeForIBMi> {
  return {
    instance, customUI: () => new CustomUI(),
    deployTools: DeployTools,
    evfeventParser: parseErrors,
    tools: VscodeTools,
    componentRegistry: extensionComponentRegistry
  };
}
```

Exported API in Code for IBM i



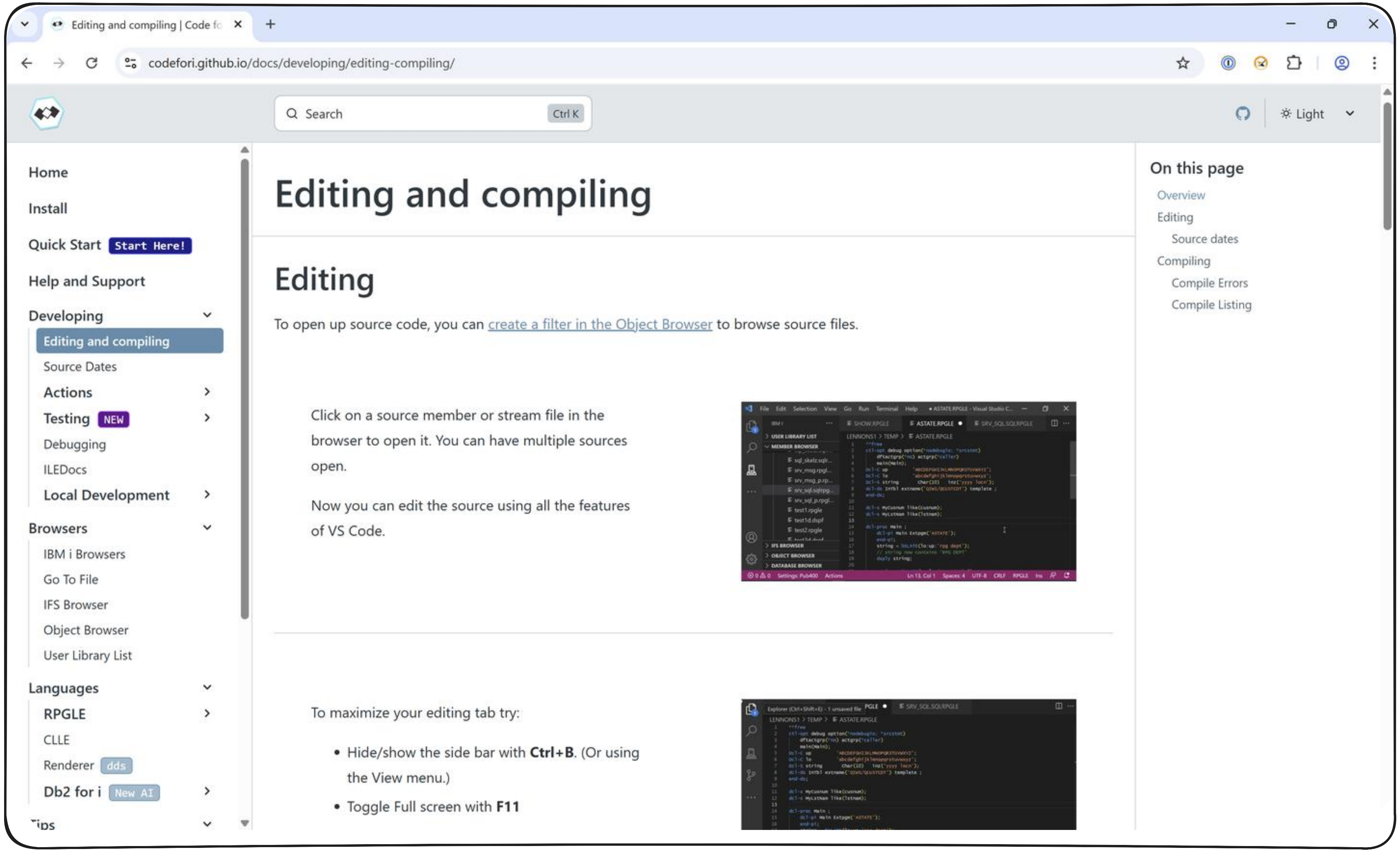


```typescript
const baseExtension: Extension<CodeForIBMi> = extensions.getExtension('halcyontechltd.code-for-ibmi');
let importedApi = baseExtension.exports;
```

Imported API in other extensions

Documentation

https://github.com/codefori/docs



Editing and compiling | Code fc
codefori.github.io/docs/developing/editing-compiling/
Search
Ctrl K
Light

Home

Install

Quick Start Start Here!

Help and Support

Developing

- Editing and compiling
- Source Dates
- Actions
- Testing NEW
- Debugging
- ILEDocs
- Local Development

Browsers

- IBM i Browsers
- Go To File
- IFS Browser
- Object Browser
- User Library List

Languages

- RPGLE
- CLLE
- Renderer dds
- Db2 for i New AI

Tips

Editing and compiling

Editing

To open up source code, you can create a filter in the Object Browser to browse source files.

Click on a source member or stream file in the browser to open it. You can have multiple sources open.

Now you can edit the source using all the features of VS Code.

To maximize your editing tab try:

- Hide/show the side bar with **Ctrl+B**. (Or using the View menu.)
- Toggle Full screen with **F11**

On this page

- Overview
- Editing
  - Source dates
- Compiling
  - Compile Errors
  - Compile Listing

Any Questions?

Important Links

VS Code

API	https://code.visualstudio.com/api/references/vscode-api
Extension Guides	https://code.visualstudio.com/api/extension-guides/overview
Language Extensions	https://code.visualstudio.com/api/language-extensions/overview
Sample Extensions	https://github.com/microsoft/vscode-extension-samples/tree/main

Code for IBM i

API	https://codefori.github.io/docs/dev/api/
GitHub Repository	https://github.com/codefori/vscode-ibmi

Sanjula Ganepola - Building VS Code Extensions and Contributing to Code for IBM i

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