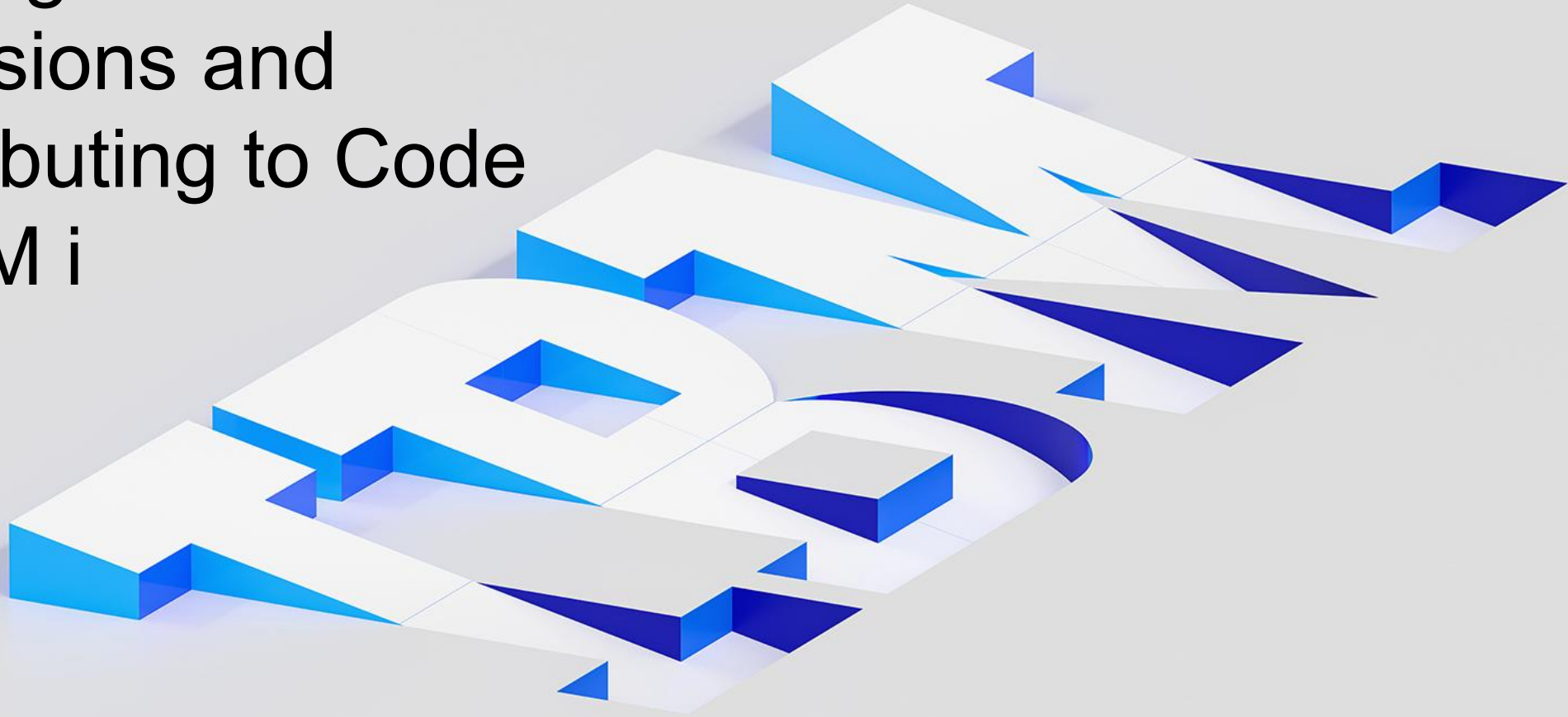


Building VS Code Extensions and Contributing to Code for IBM i

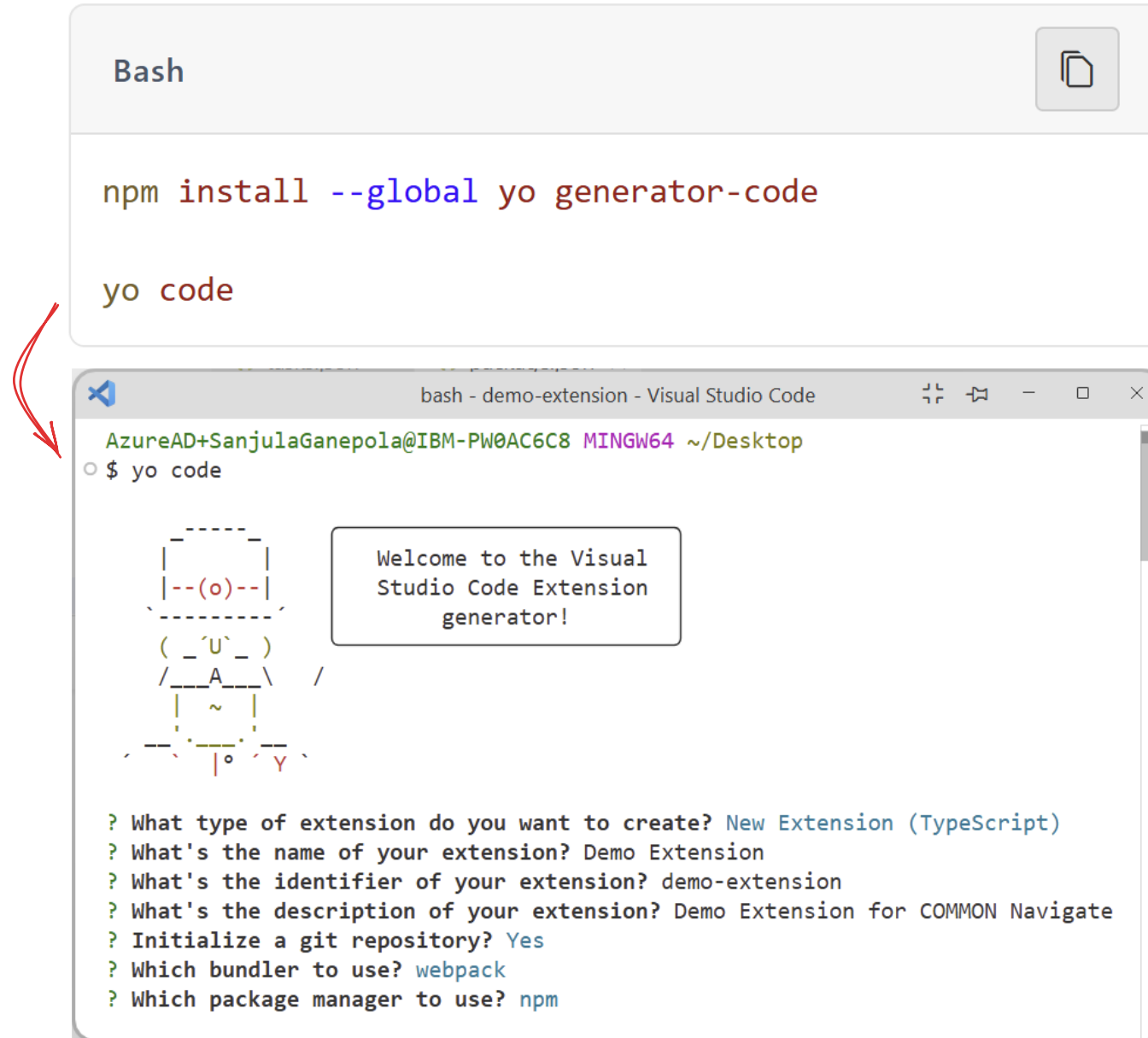


Agenda

- Building VS Code Extensions
 - Extension anatomy
 - Extension manifest
 - VS Code API
- 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

npm install --global yo generator-code

yo code
```

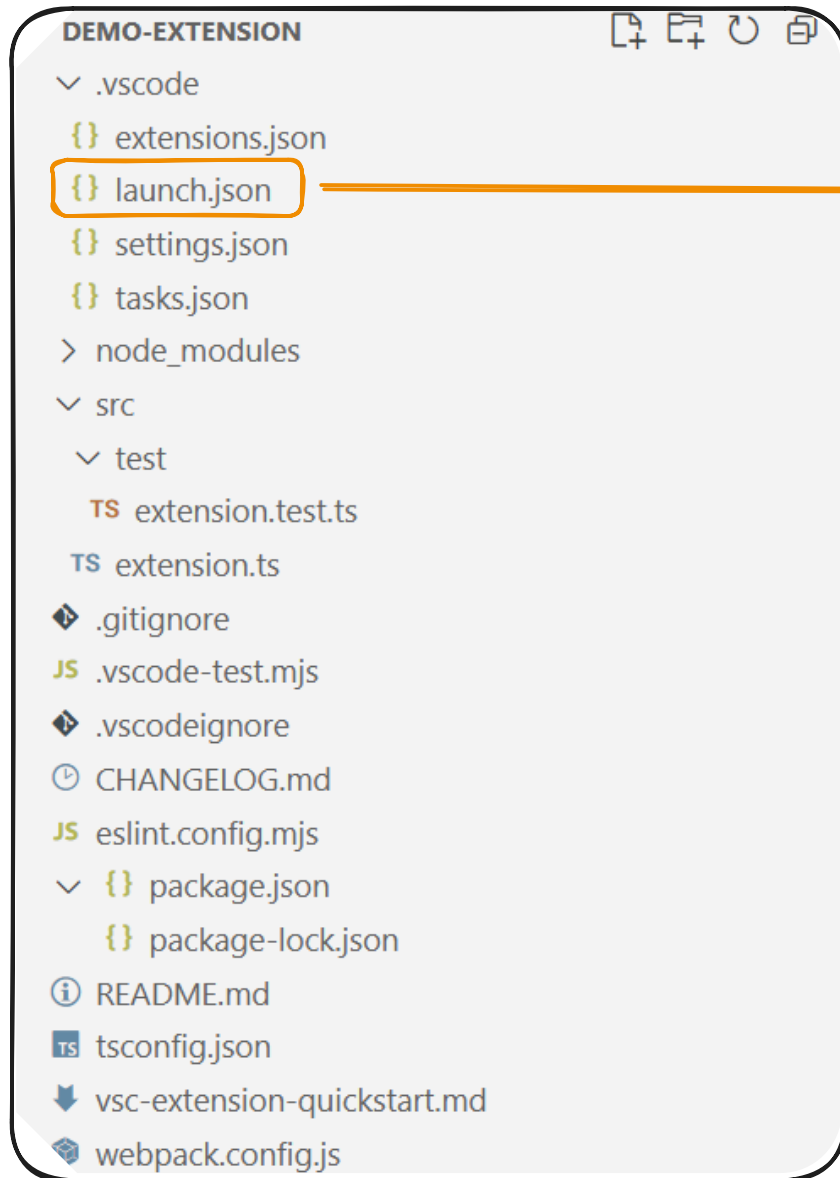
Visual Studio Code window: bash - demo-extension - Visual Studio Code

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

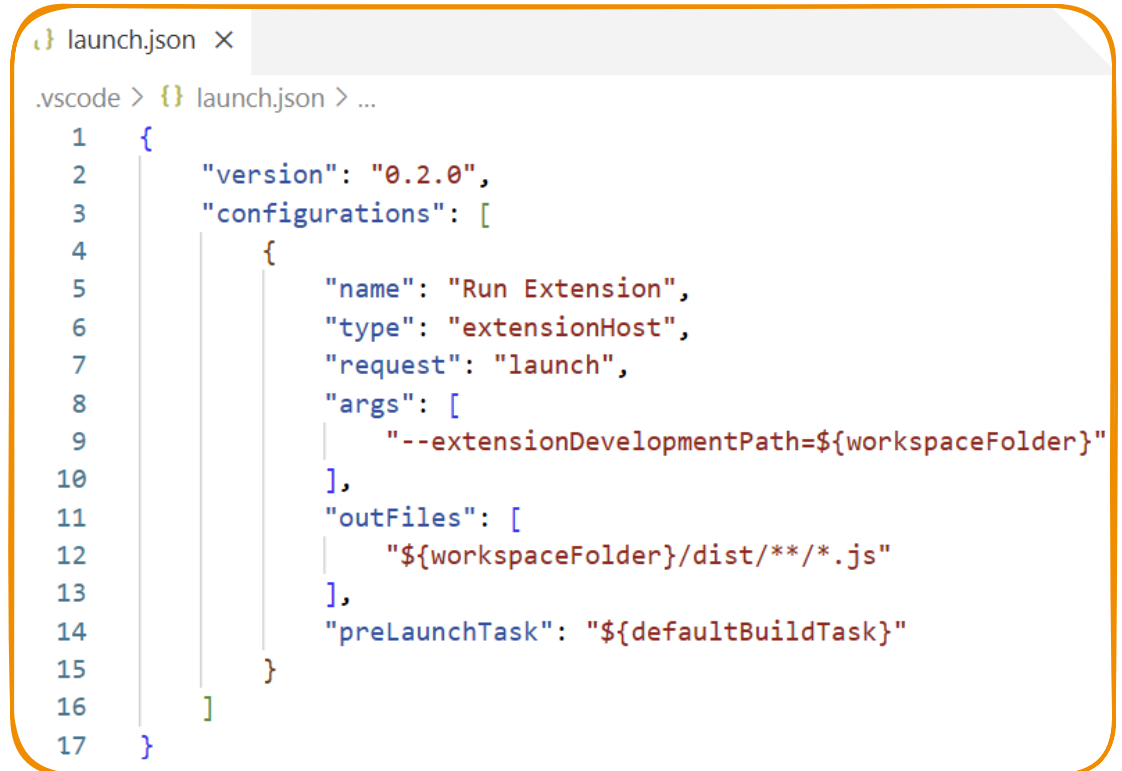
Welcome to the Visual Studio Code Extension generator!

? 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
```

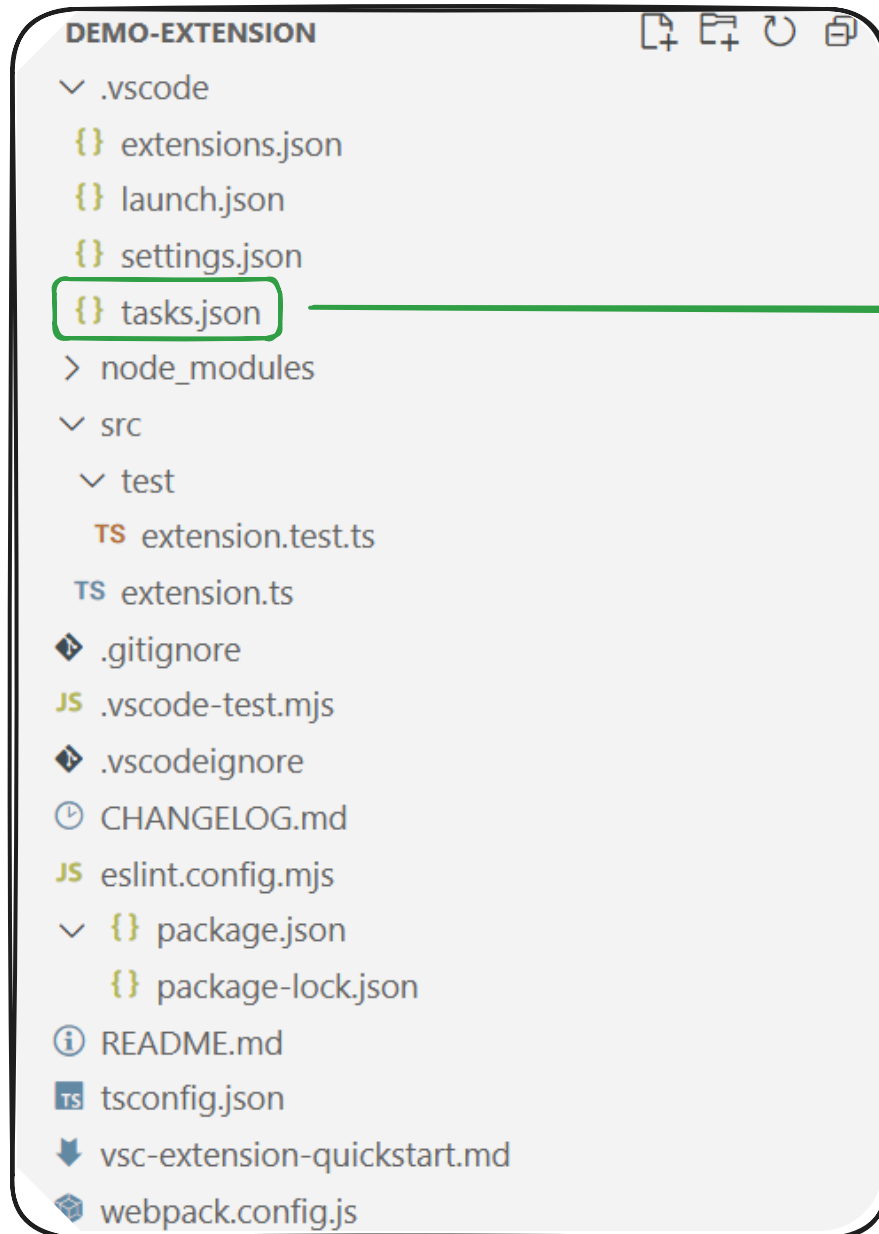
File Structure – *launch.json*



➤ Config for launching and debugging the extension



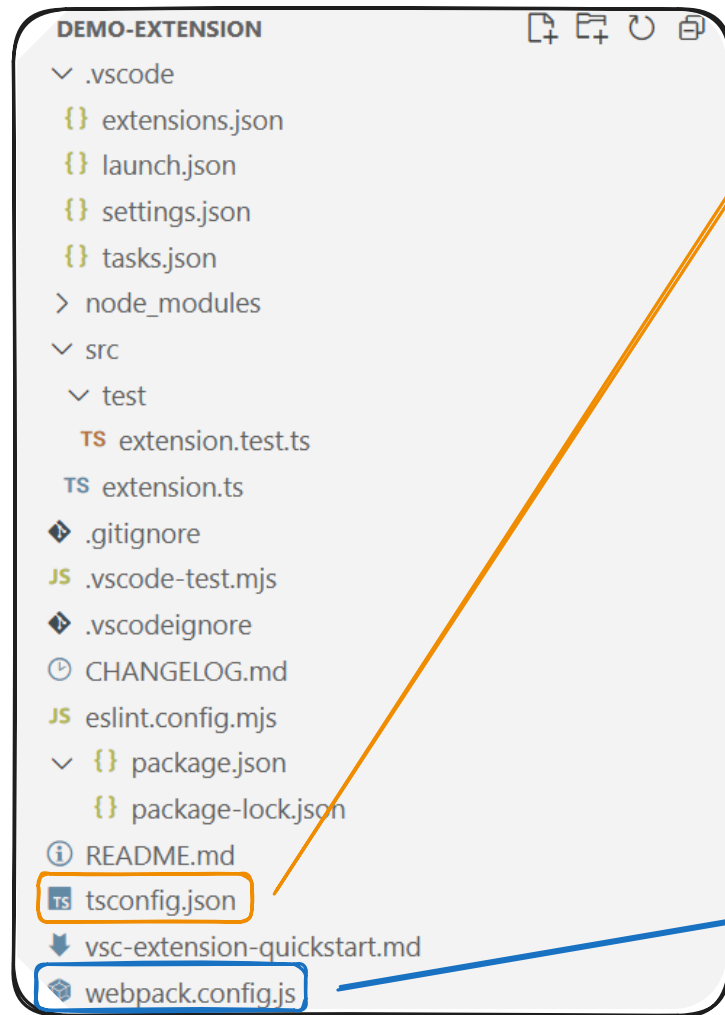
File Structure – *task.json*



Config for build task that compiles Typescript

```
.vscode > { } tasks.json > [ ] tasks > { } 1
1 // See https://go.microsoft.com/fwlink/?LinkId=733558
2 // for the documentation about the tasks.json format
3 {
4     "version": "2.0.0",
5     "tasks": [
6         {
7             "type": "npm",
8             "script": "watch",
9             "problemMatcher": "$ts-webpack-watch",
10            "isBackground": true,
11            "presentation": {
12                "reveal": "never",
13                "group": "watchers"
14            },
15            "group": {
16                "kind": "build",
17                "isDefault": true
18            }
19        },
20    ]
21 }
```

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



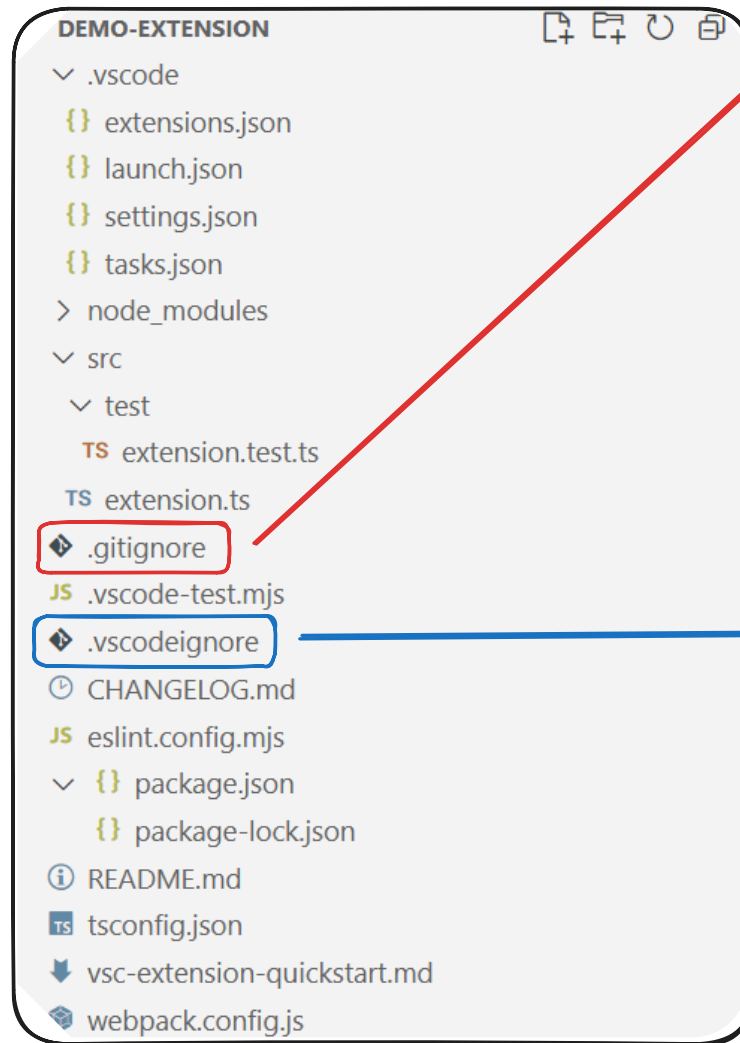
TypeScript configuration

```
tsconfig.json X
tsconfig.json > ...
1  {
2    "compilerOptions": {
3      "module": "Node16",
4      "target": "ES2022",
5      "lib": [
6        "ES2022"
7      ],
8      "sourceMap": true,
9      "rootDir": "src",
10     "strict": true
11   }
12 }
```

Webpack configuration

```
webpack.config.js X
webpack.config.js > extensionConfig > module > rules
10  /** @type WebpackConfig */
11  const extensionConfig = {
12    target: 'node',
13    mode: 'none',
14
15    entry: './src/extension.ts',
16    output: {
17      path: path.resolve(__dirname, 'dist'),
18      filename: 'extension.js',
19      libraryTarget: 'commonjs2'
20    },
21    externals: {
22      vscode: 'commonjs vscode'
23    },
24    resolve: {
25      extensions: ['.ts', '.js']
26    },
27  }
```

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

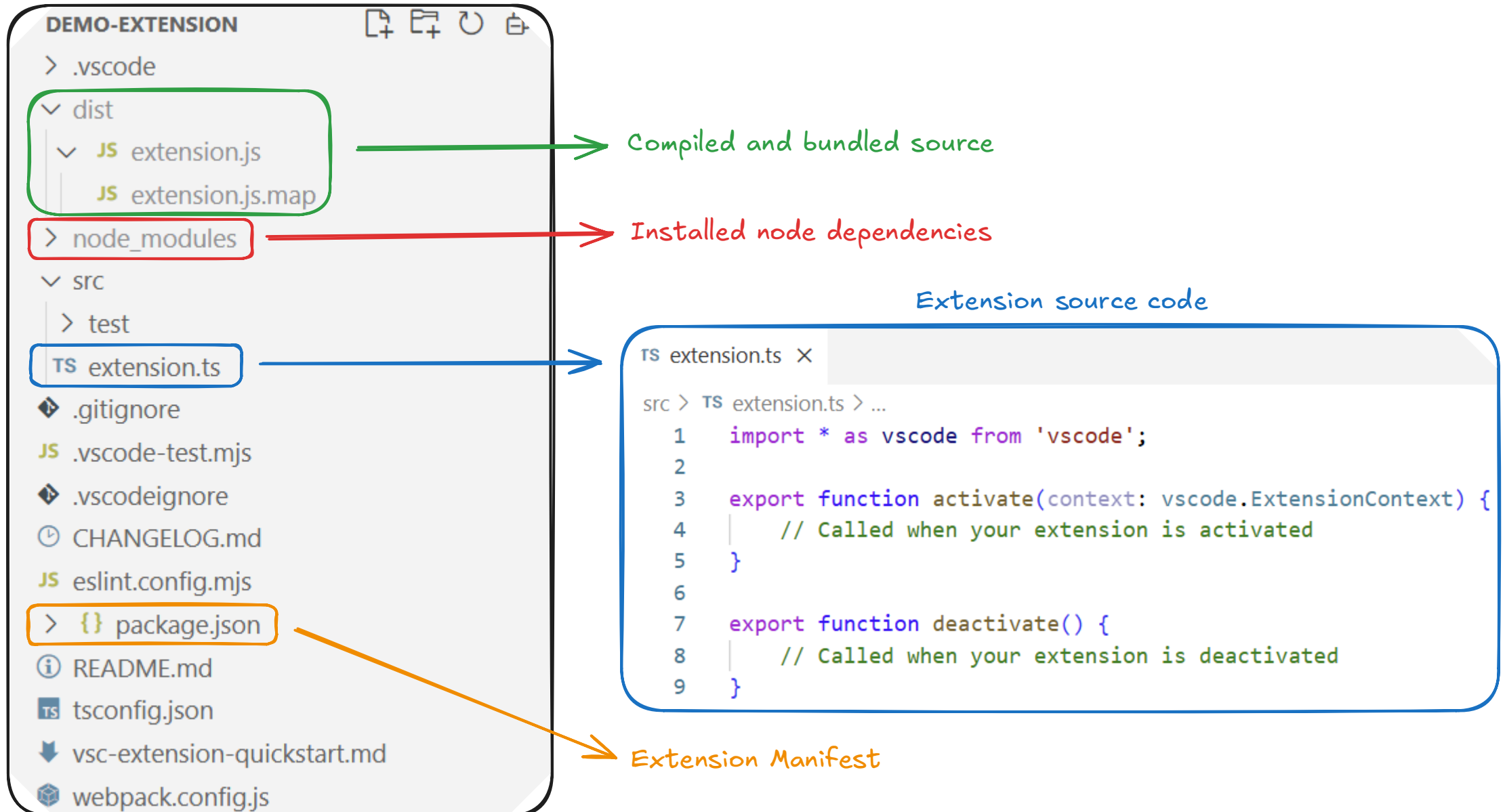


Ignore build output and node_modules Prevent files from being included in your extension's package.

```
◆ .gitignore X
◆ .gitignore
1 out
2 dist
3 node_modules
4 .vscode-test/
5 *.vsix
```

```
◆ .vscodeignore X
◆ .vscodeignore
1 .vscode/**
2 .vscode-test/**
3 out/**
4 node_modules/**
5 src/**
6 .gitignore
7 .yarnrc
8 webpack.config.js
9 vsc-extension-quickstart.md
10 **/tsconfig.json
11 **/eslint.config.mjs
12 **/*.map
13 **/*.ts
14 **/.vscode-test.*
```


File Structure – Source code, extension manifest, and more



File Structure – package.json

```
{ } package.json X
{ } package.json > { } scripts
1  {
2    "name": "demo-extension",
3    "displayName": "Demo Extension",
4    "description": "Demo extension for COMMON Navigate",
5    "version": "0.0.1",
6    "engines": {
7      "vscode": "^1.103.0"
8    },
9    "categories": [
10     "Other"
11   ],
12   "activationEvents": [],
13   "main": "./dist/extension.js",
```

"onStartupFinished"
is commonly used

↓
contributes

```
22   "scripts": {
23     "vscode:prepublish": "npm run package",
24     "compile": "webpack",
25     "watch": "webpack --watch",
26     "package": "webpack --mode production --devtool hidden-source-map",
27     "compile-tests": "tsc -p . --outDir out",
28     "watch-tests": "tsc -p . -w --outDir out",
29     "pretest": "npm run compile-tests && npm run compile && npm run lint",
30     "lint": "eslint src",
31     "test": "vscode-test"
32   },
33   "dependencies": {},
34   "devDependencies": {
35     "@types/vscode": "^1.103.0",
36     "@types/mocha": "^10.0.10",
37     "@types/node": "22.x",
38     "@typescript-eslint/eslint-plugin": "^8.39.0",
39     "@typescript-eslint/parser": "^8.39.0",
40     "eslint": "^9.32.0",
41     "typescript": "^5.9.2",
42     "ts-loader": "^9.5.2",
43     "webpack": "^5.101.0",
44     "webpack-cli": "^6.0.1",
45     "@vscode/test-cli": "^0.0.11",
46     "@vscode/test-electron": "^2.5.2"
47   }
48 }
```

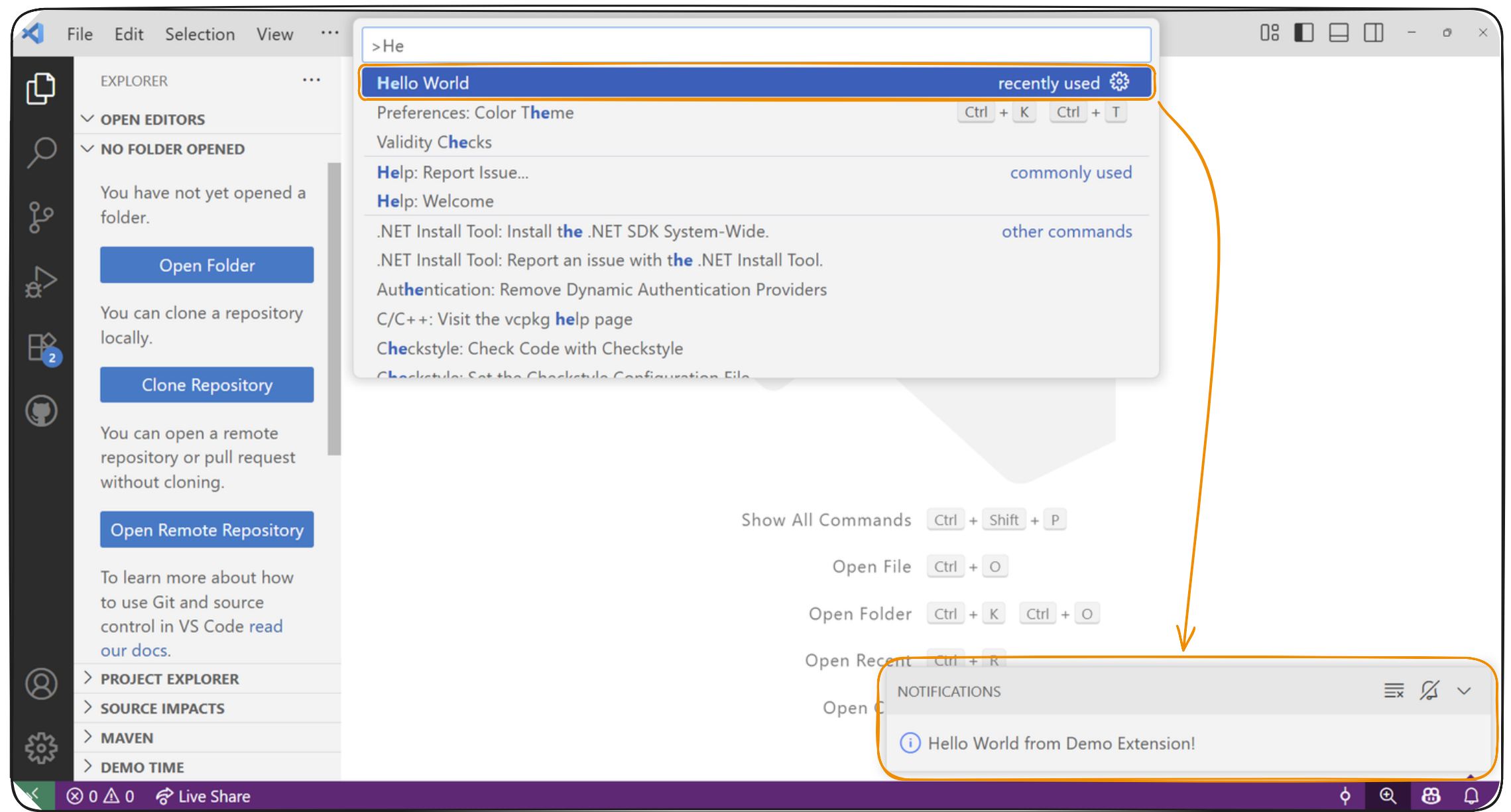
Extension Manifest - Contributions

```
14   "contributes": {  
15     "commands": [  
16       {  
17         "command": "demo-extension.helloWorld",  
18         "title": "Hello World"  
19       }  
20     ]  
21   },
```

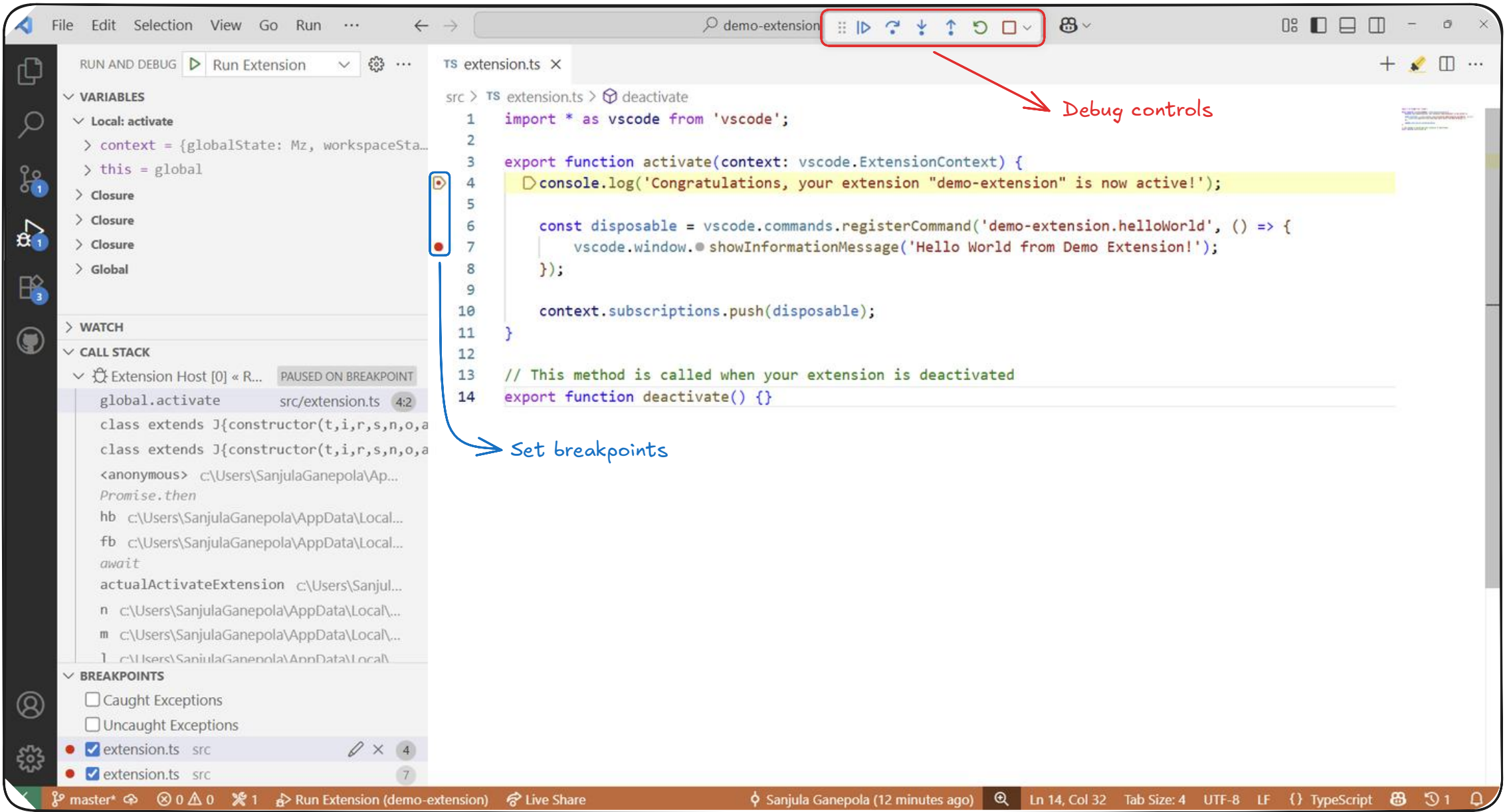


```
TS extension.ts ×  
src > TS extension.ts > ...  
1  import * as vscode from 'vscode';  
2  
3  export function activate(context: vscode.ExtensionContext) {  
4    console.log('Congratulations, your extension "demo-extension" is now active!');  
5  
6    const disposable = vscode.commands.registerCommand('demo-extension.helloWorld', () => {  
7      vscode.window.showInformationMessage('Hello World from Demo Extension!');  
8    });  
9  
10   context.subscriptions.push(disposable);  
11 }  
12  
13 // This method is called when your extension is deactivated  
14 export function deactivate() {}
```

Let's run our extension



Let's debug our extension



Contribution Points - Keybindings

```
{ } package.json 9 X
{ } package.json > { } contributes > [ ] keybindings > { } 2
43   "contributes": {
1448     "keybindings": [
1449       {
1450         "command": "vscode-db2i.runEditorStatement",
1451         "key": "ctrl+r",
1452         "mac": "cmd+r",
1453         "when": "editorLangId == sql && resourceExtname != .inb"
1454       },
1455       {
1456         "command": "vscode-db2i.runEditorStatement.inView",
1457         "key": "ctrl+alt+r",
1458         "mac": "cmd+ctrl+r",
1459         "when": "editorLangId == sql"
1460       },

```

Keyboard Shortcuts X

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 X

package.json > {} contributes > {} configuration > {} properties > {} code-for-ibmi.ObjectBrowser.showNam

41 "contributes": {
55 "configuration": {
57 "properties": {
422 "code-for-ibmi.autoOpenFile": {
423 "type": "boolean",
424 "default": true,
425 "description": "When enabled, files will automatically be
opened when copied or moved/renamed."
426 },

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 – View Containers

package.json 9+, M

×

package.json > {} contributes

41 "contributes": {

1785 "viewsContainers": {

1786 "activitybar": [

1787 {

1788 "id": "ibmi-explorer",

1789 "title": "IBM i",

1790 "icon": "\$(server-environment)"

1791 },



Contribution Points – Views

```
package.json 9+, M X
{} package.json > {} contributes
41      "contributes": {
1799      "views": {
1800      "ibmi-explorer": [
1810      ],
1817      {
1818      "id": "libraryListView",
1819      "name": "User Library List",
1820      "when": "code-for-ibmi:connected && code-for-ibmi:libraryListDisabled !== true"
1821      },

```

✓ USER LIBRARY LIST

+

📄

↺

⋮

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

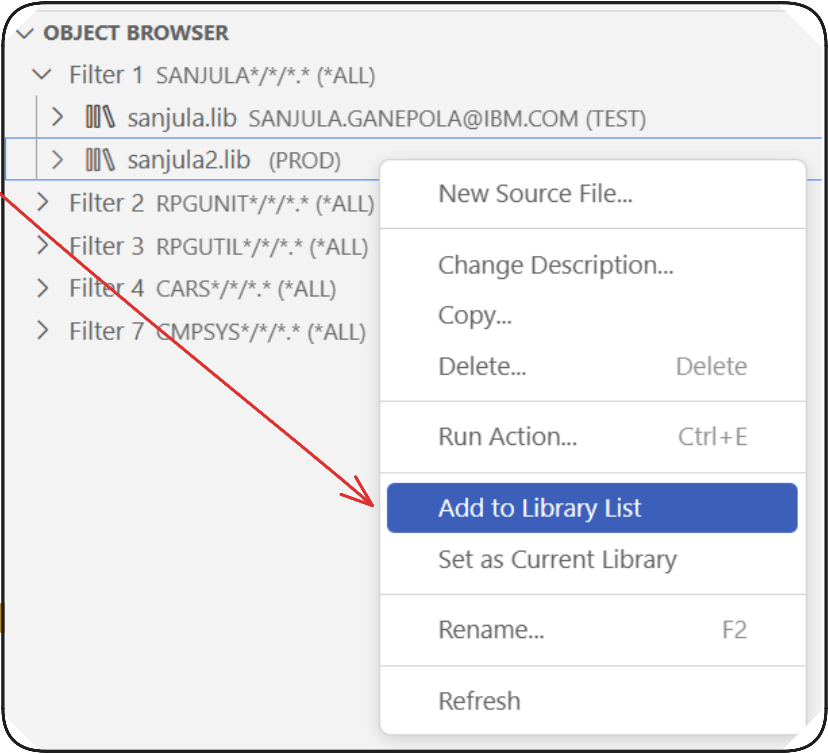
```
{} package.json 9+, M X
{} package.json > {} contributes > {} menus > [ ] view/title > {} 7 > [abc] group
41      "contributes": {
1891      "menus": {
2365      "view/title": [
2381      {
2382          "command": "code-for-ibmi.connectToPrevious",
2383          "group": "navigation@4",
2384          "when": "view == connectionBrowser && code-for-ibmi:hasPreviousConnection"
2385      },
```



Contribution Points – View item context menu

```
{} package.json 9+, M ×
{} package.json > {} contributes > {} menus > [ ] view/item/context > {} 40
41      "contributes": {
1891      "menus": {
2534      "view/item/context": [
2725      {
2726      "command": "code-for-ibmi.addToLibraryList",
2727      "when": "!code-for-ibmi:libraryListDisabled && view == objectBrowser && viewItem =~ /library/",
2728      "group": "2_library@1"
2729      },

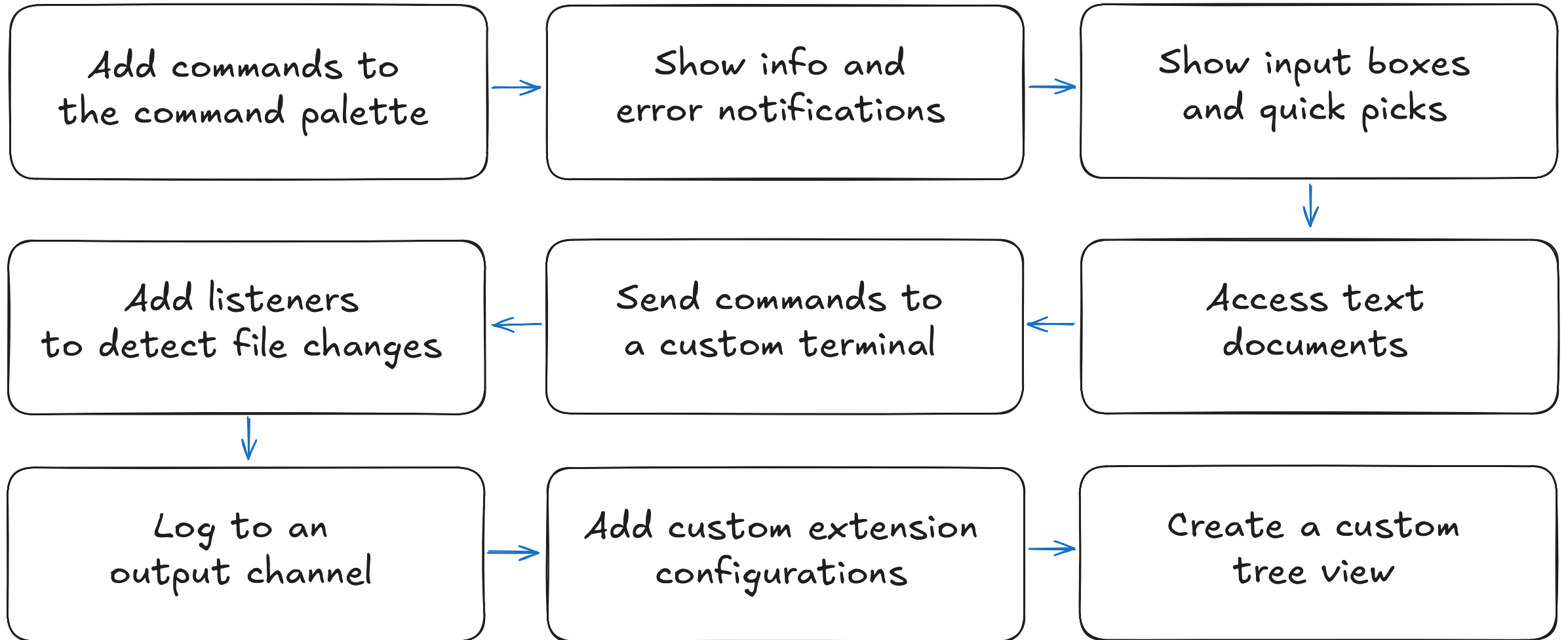
```



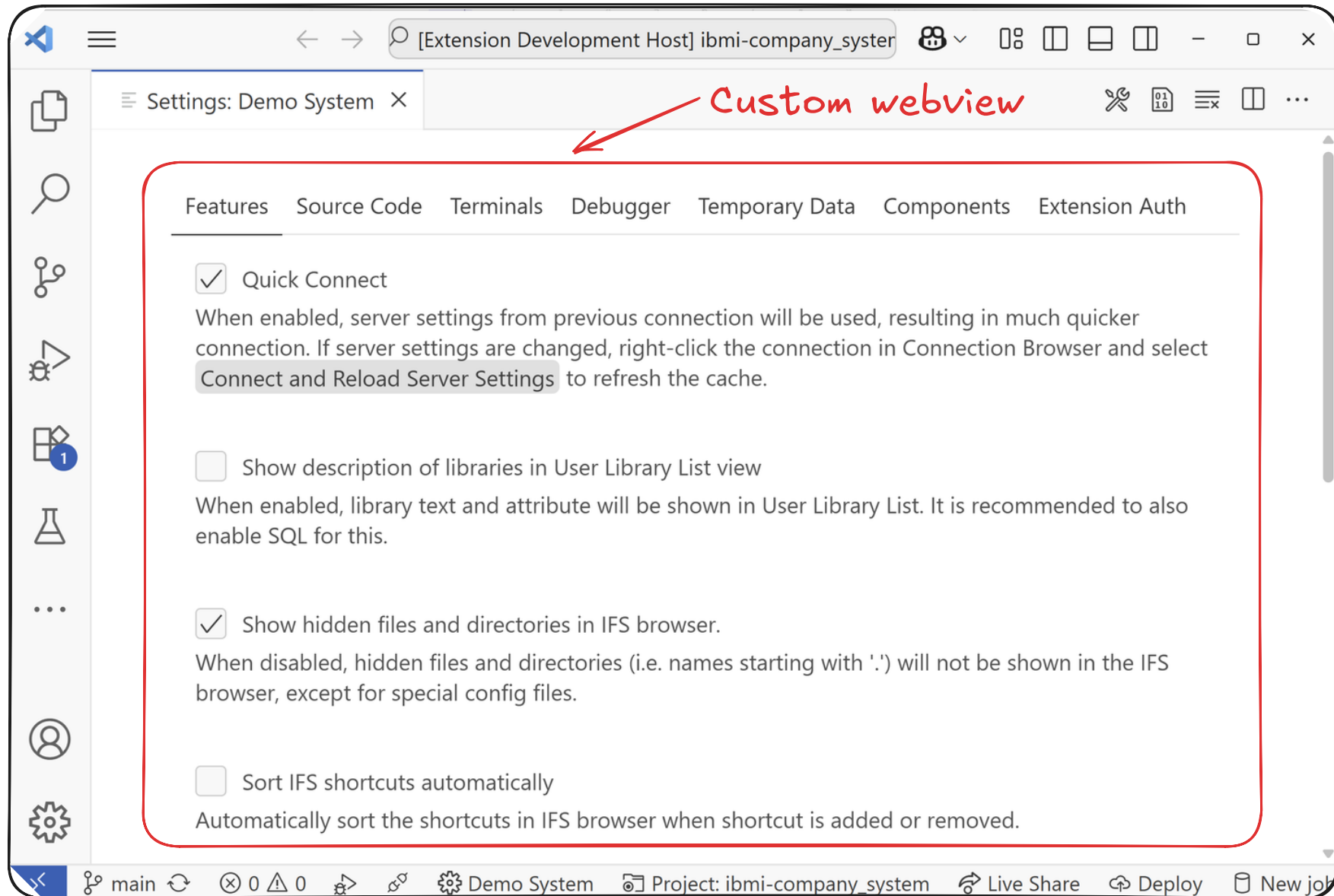
Contribution Points – Check the docs for more

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

Let's experiment with the VS Code API



VS Code API – Example webview in Code for IBM i



VS Code API – Example webview in Db2 for IBM i

VS Code Editor

1

SELECT * FROM SAMPLE.EMPLOYEE

PROBLEMS

DB2 FOR I

DEBUG CONSOLE

OUTPUT

TERMINAL

IBM I JOB LOG

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

main

0 0

Demo System

Project: ibmi-company_system

Live Share

Deploy

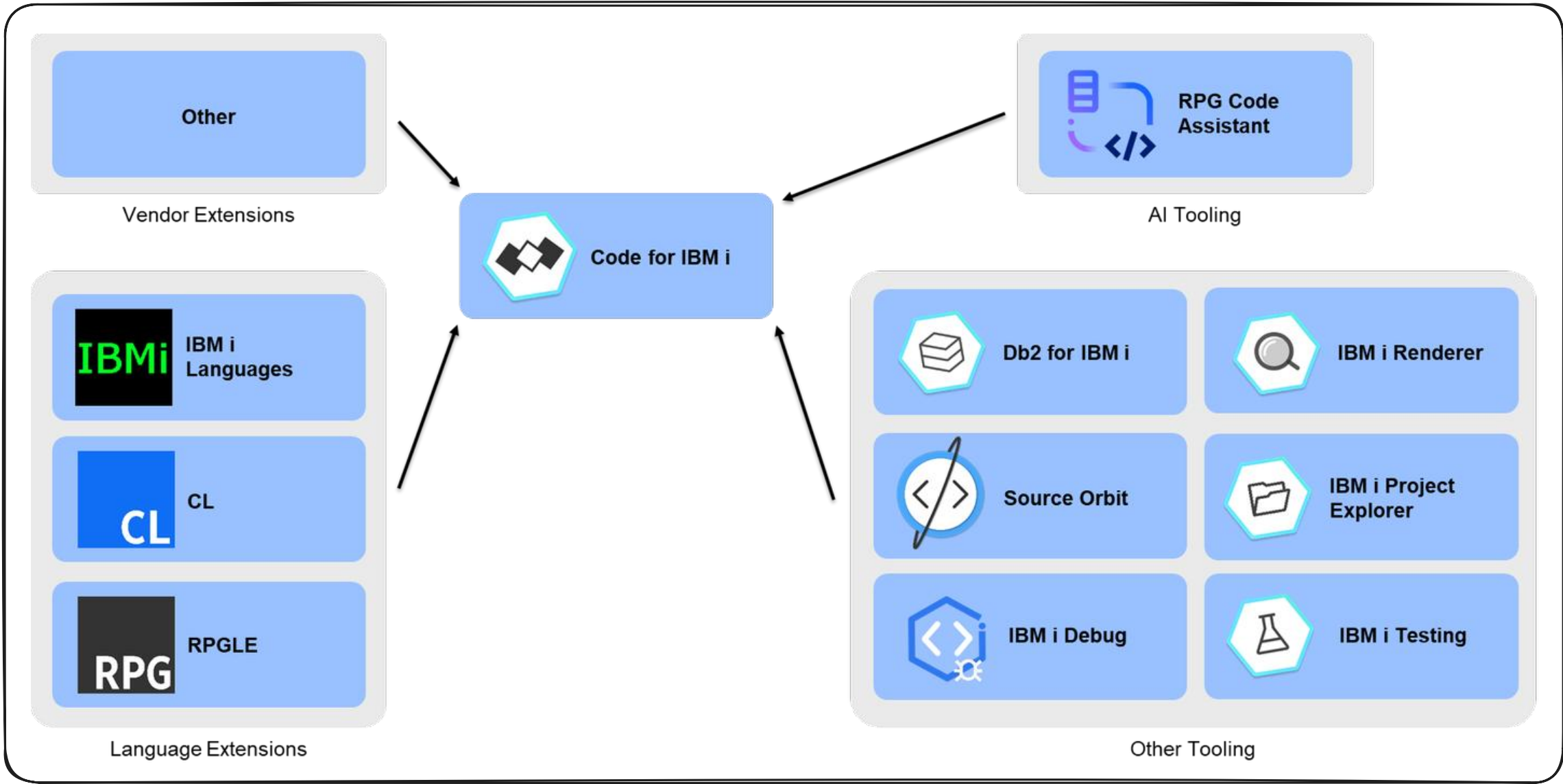
New

Custom webview



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

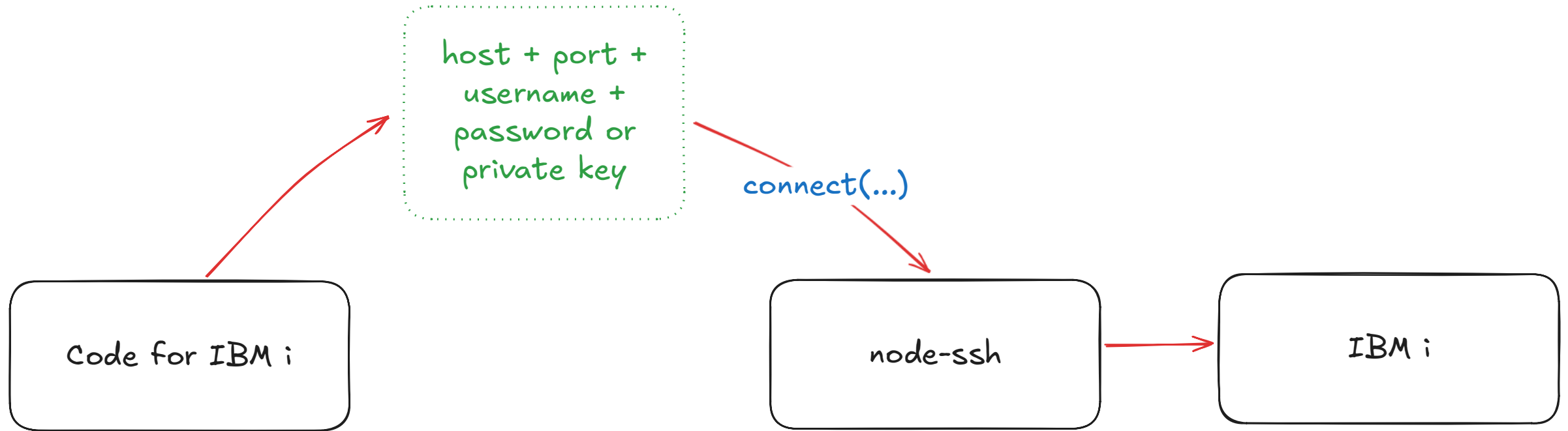
Community collaboration!

57 total contributors to Code for IBM !!!



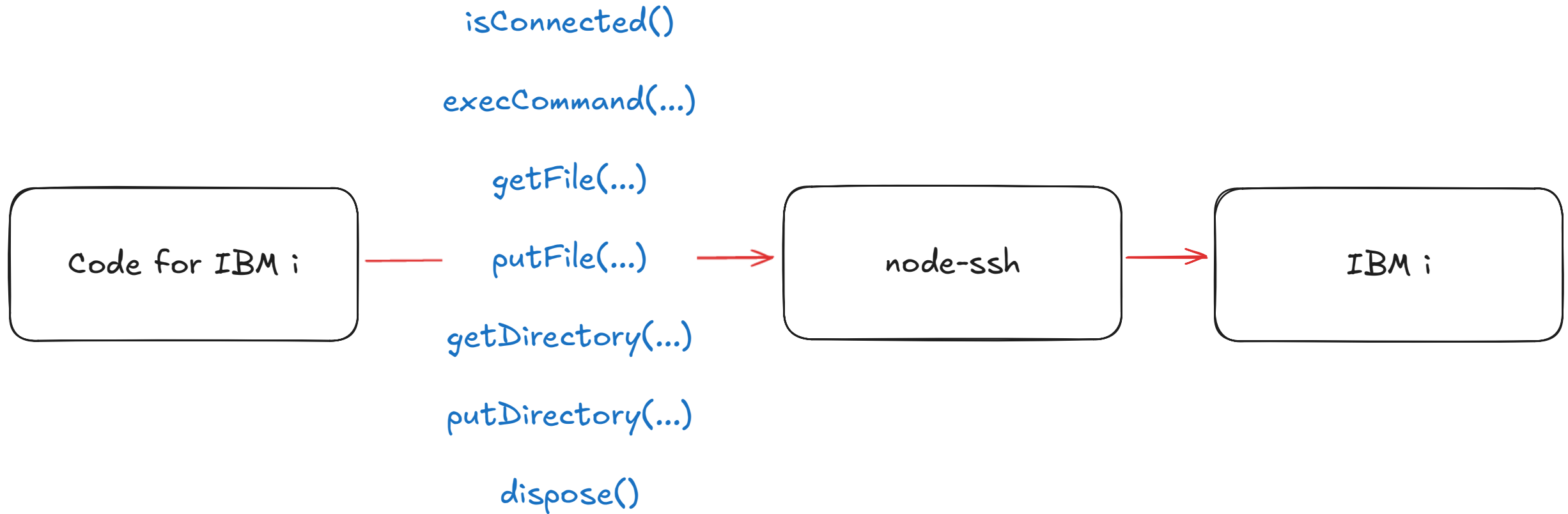
How does Code for IBM i connect to an IBM i?

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



How does it use Node-SSH?

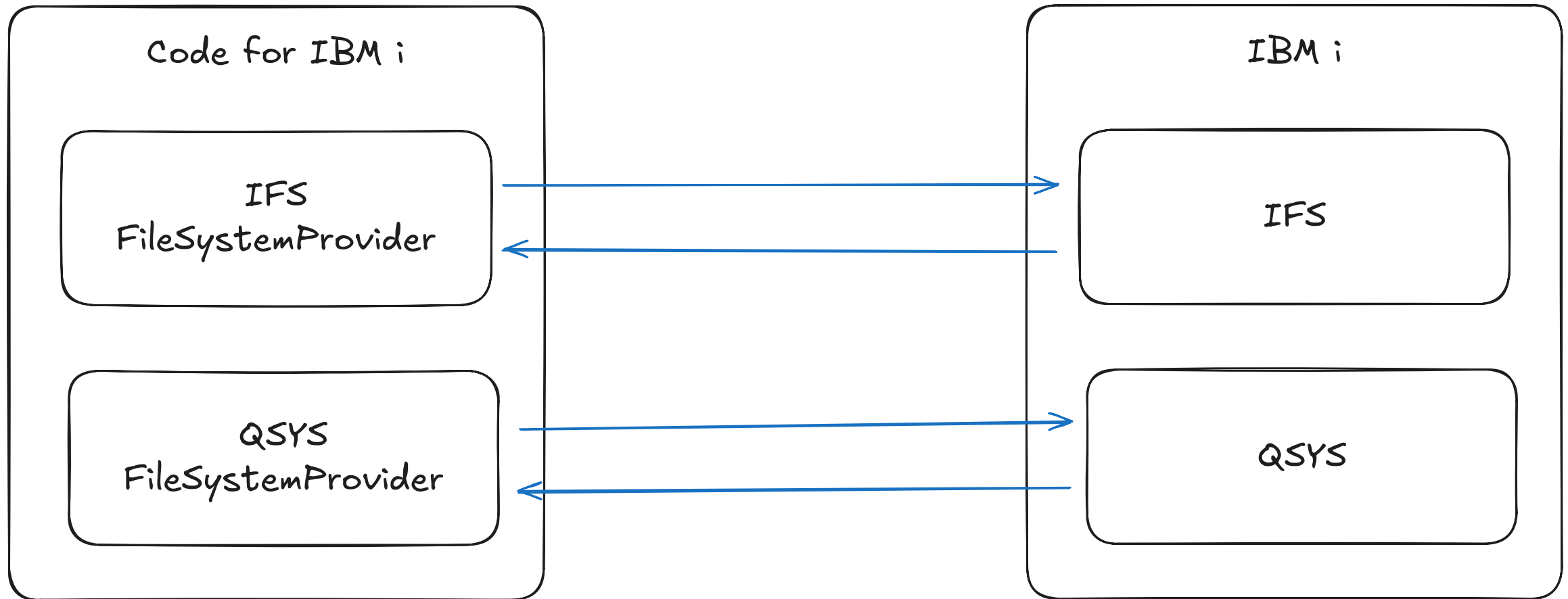
<https://github.com/codefori/vscode-ibmi/blob/master/src/api/IBMi.ts#L317>



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 X
C: > Users > SanjulaGanepola > AppData > Roaming > Code > User > settings.json > [ ] code-for-ibmi.connectionSettings
14  "code-for-ibmi.connectionSettings": [
15      {
16          "commandFor5250": [],
17          "ifsShortcuts": [
18              "/home/SANJULA"
19          ],
20          "autoSortIFSShortcuts": false,
21          "homeDirectory": "/home/SANJULA",
22          "tempLibrary": "ILEEDITOR",
23          "tempDir": "/tmp",
24          "currentLibrary": "SANJULA",
25          "sourceFileCCSID": "*FILE",
26          "autoConvertIFScsid": false,
27          "hideCompileErrors": [],
28          "enableSourceDates": false,
29          "sourceDateGutter": false,
30          "encodingFor5250": "default",
31          "terminalFor5250": "default",
32          "setDeviceNameFor5250": false,
33          "connectingStringFor5250": "+uninhibited localhost",
34          "autoSaveBeforeAction": false,
35          "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>

```
41  export async function activate(context: ExtensionContext): Promise<CodeForIBMi> {  
131      return {  
132          instance, customUI: () => new CustomUI(),  
133          deployTools: DeployTools,  
134          evfeventParser: parseErrors,  
135          tools: VscodeTools,  
136          componentRegistry: extensionComponentRegistry  
137      };  
138  }
```

Exported API in
Code for IBM i

RPGLE

CL

Db2 for IBM i

IBM i Testing

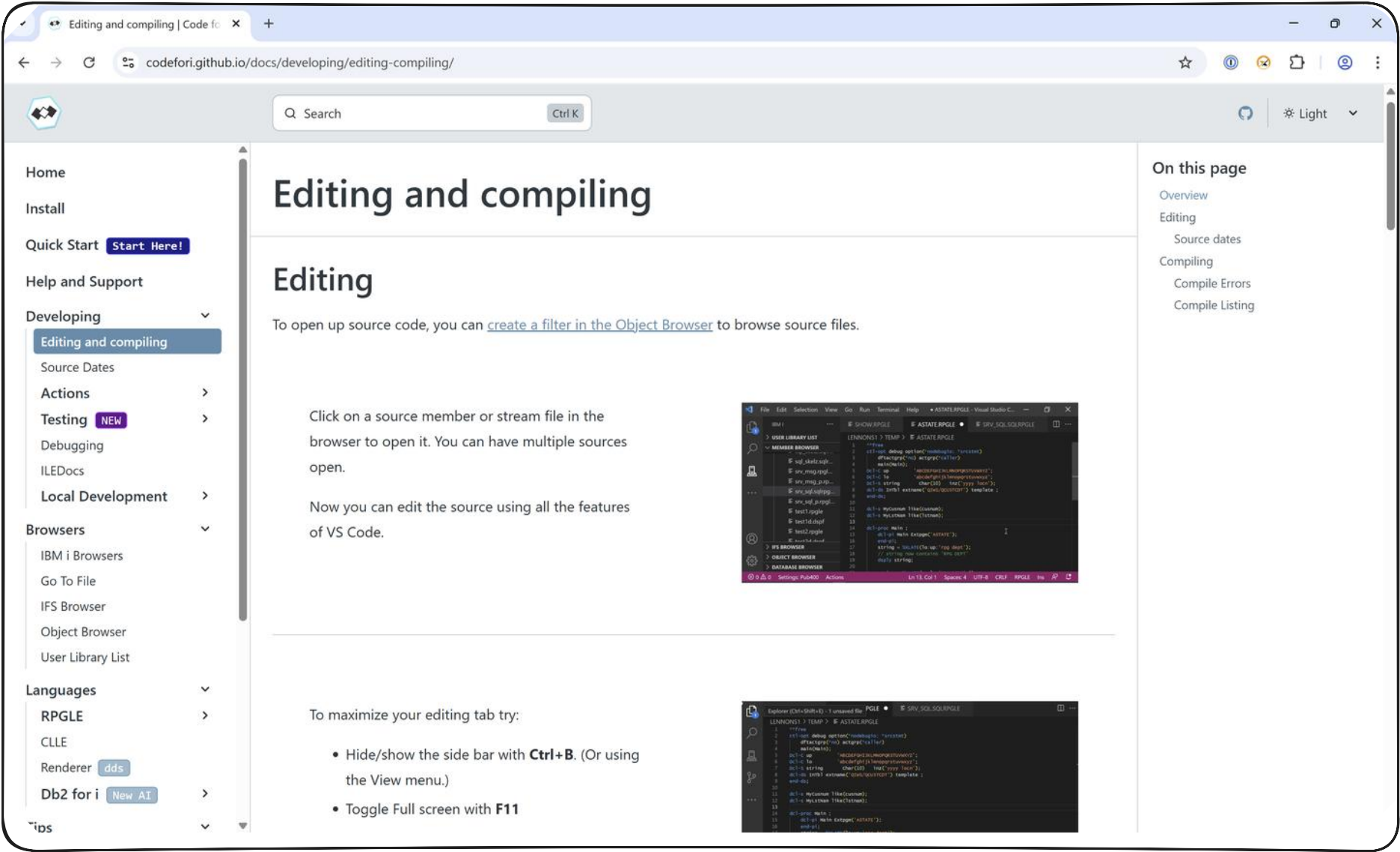
IBM i Project Explorer

```
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>



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.



COMMON

IBM i