

Create your own extension



And extend Code for IBM i



Code for IBM i

What's an extension? 🤔



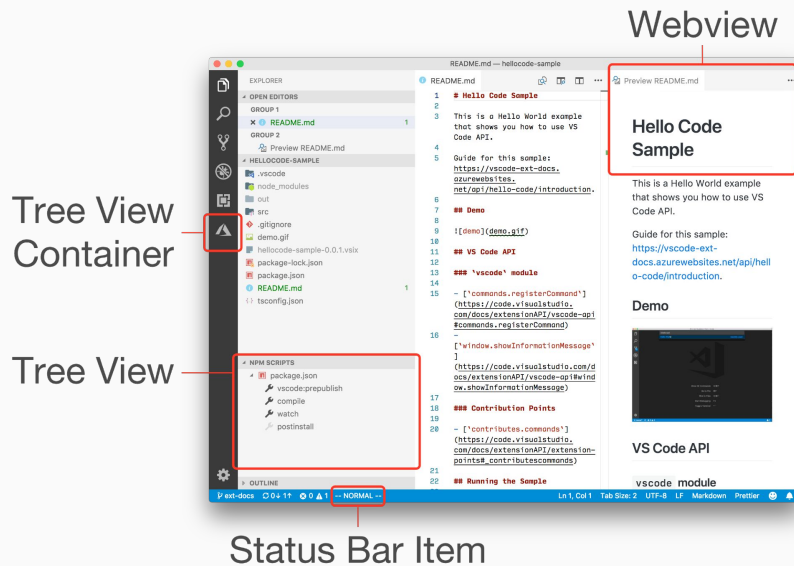
What can extensions do?

- Change the look of VS Code



What can extensions do?

- Change the look of VS Code
- Add custom UI elements



What can extensions do?

- Change the look of VS Code
- Add custom UI elements
- Add support for new programming languages



Code for IBM i

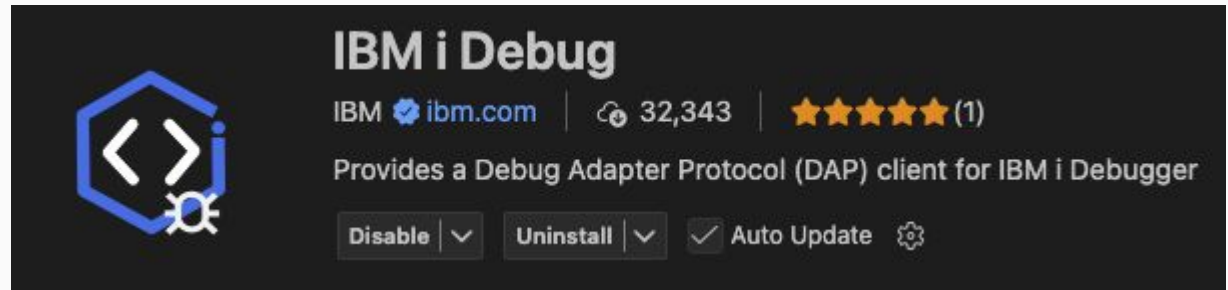





What can extensions do?

- Change the look of VS Code
- Add custom UI elements
- Add support for new programming languages
- Add support for debugging a specific runtime




Code for IBM i



IBM i Debug
IBM  ibm.com |  32,343 |  (1)

Provides a Debug Adapter Protocol (DAP) client for IBM i Debugger

Auto Update 

What can extensions do?

- Change the look of VS Code
- **Add custom UI elements <- our goal**
- Add support for new programming languages
- Add support for debugging a specific runtime



Coding Tools

- VS Code
- TypeScript
- Node.js
- git



Extension Anatomy

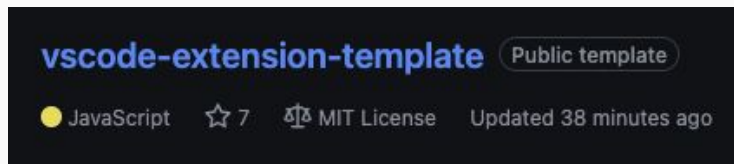


Project scaffolding

- Yeoman + generator-code
 - <https://www.npmjs.com/package/generator-code>
 - Basic project architecture (no linter, no l10n)
 - Generates any type of extension
- Seb's template
 - <https://github.com/sebjulliand/vscode-extension-template/tree/code4i>
 - code for i API extension dedicated branch
 - boilerplate
 - Ready for publication



Code for IBM i



File structure

- Required extensions → `{}` extensions.json
- Debug configuration → `{}` launch.json
- Build task → `{}` settings.json
- Internationalization → `{}` tasks.json
- Source code → `{}` bundle.l10n.json
 - Extension entry file → `TS` extension.ts
- Linter configuration → `.`eslintrc.json
- Files ignored by git → `.`gitignore
- npm configuration → `.`npmrc
- Files excluded from packaging → `.`vscodeignore
- Optional license → LICENSE
- Extension manifest → `{}` package-lock.json
- Extension nice description → `{}` package.json
- TypeScript configuration → `{}` package.nls.json
- Webpack configuration → `TS` tsconfig.json
- Webpack configuration → `.`webpack.config.js

```
└─ .vscode
   ├── {} extensions.json
   ├── {} launch.json
   ├── {} settings.json
   ├── {} tasks.json
   └─ l10n
      ├── {} bundle.l10n.json
      └─ src
         ├── TS extension.ts
         ├── .eslintrc.json
         ├── .gitignore
         ├── .npmrc
         ├── .vscodeignore
         ├── LICENSE
         ├── {} package-lock.json
         ├── {} package.json
         ├── {} package.nls.json
         ├── README.md
         ├── TS tsconfig.json
         └─ .webpack.config.js
```



Extension Manifest

- Extension's identity card
 - name / publisher
 - display name
 - version
 - extension dependencies
 - license
 - activation
 - entry point
 - VS Code version
 - contributions
 - scripts
 - node dependencies
- Supports content assist



Code for IBM i

```
{
  "name": "vscode-extension-template",
  "publisher": "sebjulliand",
  "displayName": "VSCode Extension Template",
  "description": "This is my template. Powered by Code for IBM i.",
  "version": "0.0.1",
  "engines": {
    "vscode": "^1.93.0"
  },
  "extensionDependencies": [
    "HalcyonTechLtd.code-for-ibmi"
  ],
  "license": "MIT",
  "categories": [
    "Other"
  ],
  "activationEvents": [
    "onStartupFinished"
  ],
  "main": "./dist/extension.js",
  "l10n": "./l10n",
  "contributes": {
    "commands": [
      {
        "command": "template.helloWorld",
        "title": "%hello.world%"
      }
    ]
  },
  "scripts": {
    "nls": "npx @vscode/l10n-dev export --outDir ./l10n ./src",
    "build": "webpack --mode development",
    "watch": "webpack --mode development --watch",
    "vscode:prepublish": "webpack --mode production --devtool hidden-source-map",
    "package": "npx @vscode/vsce package --pre-release"
  },
  "devDependencies": {
    "@halcyontech/vscode-ibmi-types": "^2.18.0",
    "@types/node": "^22",
    "@types/vscode": "1.104",
    "@typescript-eslint/eslint-plugin": "6.21.0",
    "@typescript-eslint/parser": "6.21.0",
    "esbuild-loader": "4.3.0",
    "typescript": "5.9.2",
    "webpack": "5.101.3",
    "webpack-cli": "6.0.1"
  }
}
```

Extension Manifest: Contribution Points

- The many functionalities the extension can...extend
- JSON declarations
- 30+ existing points
- <https://code.visualstudio.com/api/references/contribution-points>

```

  ✓ ( ) contributes
    > ( ) taskDefinitions
    > ( ) jsonValidation
    > ( ) configuration
    > ( ) commands
    > ( ) customEditors
    > ( ) keybindings
    > ( ) viewsContainers
    > ( ) views
    > ( ) submenus
    > ( ) menus
    > ( ) viewsWelcome

```



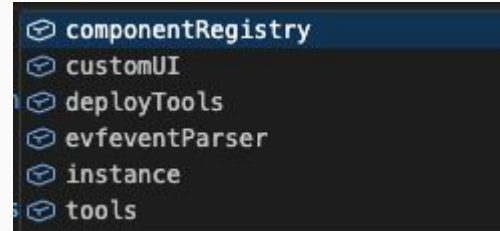
Extension Entry Point

- activate()
 - the only reference to the extension's context
 - register commands, views, etc
 - load the 'subscriptions' array
- deactivate()
 - clean everything before you go !



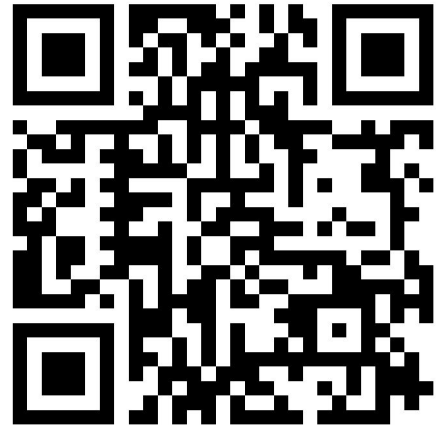
Code for IBM i API - Exported

- instance: gives access to current connection, config, storage,...
- tools: IBM i related general purpose functions
- deployTools: deploy local workspace
- evfeventParser(): parse lines of EVFEVENT
- customUI(): easy UI !
 - custom webviews
 - custom editors (soon)
- componentRegistry: connection based components
 - e.g.: install / update remote tools



Sample extension extending Code for i

- <https://github.com/sebjulliand/BYOE>
- Simple features
 - Run CL command
 - Open a spooled file
 - Job browser



What Can an Extension Become?



Code for IBM i



Carbon 400

Why Carbon/400 Exists



Code for IBM i



Carbon 400

From a Rust protocol to a modern IBM i toolchain



Code for IBM i



Carbon 400

**Enough slides
Let's dive in**



Code for IBM i



Carbon 400

Any questions?

Don't be shy!



Code for IBM i



Thank you for your attention!

 *Now go out there and create something beautiful* 

Seb Julliard

i and me - Training & solutions
R&D Tech Lead

sjulliard@mlitters.com

Carbon/400

Authors:
Ben Yeris Benouar
Théo Stocchetti



www.carbon400.com



Code for IBM i

Thank you !

Seb Julliard

sebjulliard@gmail.com



Code for IBM i

Meet me for Coffee