Consultingwerk
software architecture and development

VS Code Extension Development for Dummies

Janine Keser, Consultingwerk
janine.keser@consultingwerk.de

Janine Keser

Joined Consultingwerk 2018

Experience in OpenEdge, Progress ABL/4GL, JavaScript, TypeScript, HTML, CSS, and database design

Experienced in tools like VS Code, Windsurf, Jira, and Jenkins, on Windows and Linux

Consultingwerk
software architecture and development

Modernization in Focus

Modernization of Legacy OpenEdge Applications

Deep Technical Expertise

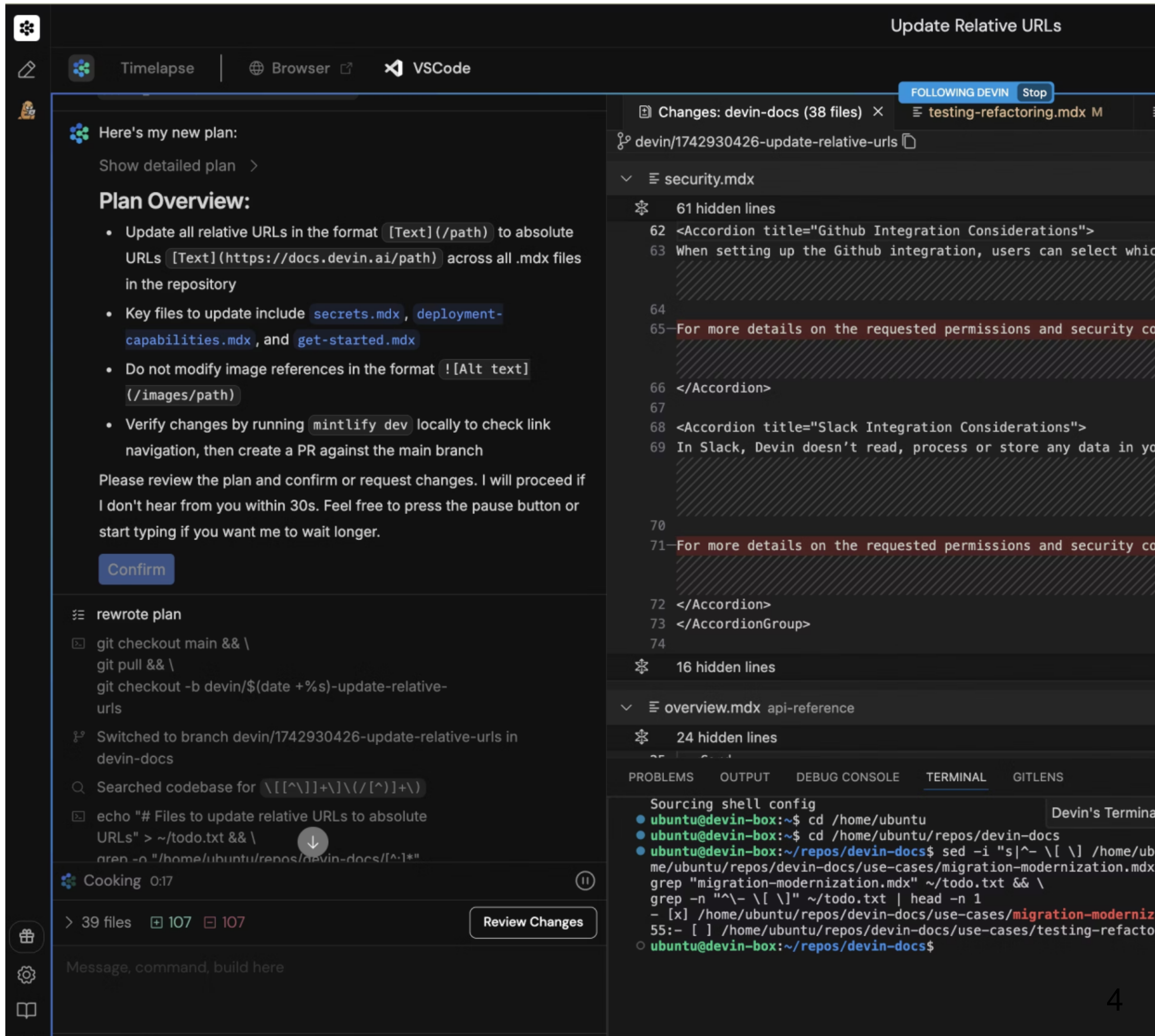
Global IT Partner with Local Presence

More than Consulting – We Deliver Tools & Solutions

VS Code Extension Development for Dummies

Visual Studio Code Overview

- Built using **Electron** – runs on **Windows, macOS, and Linux**
- Is a **browser-based version**
- **Desktop app** uses:
  - **Chromium** (for the user interface)
  - **Node.js** (for backend functionality)
- **Web version** runs in the **browser sandbox**
  - Offers **different capabilities** than the desktop version


Update Relative URLs
Timelapse
Browser
VSCode
FOLLOWING DEVIN
Stop
Changes: devin-docs (38 files)
testing-refactoring.mdx M
devin/1742930426-update-relative-urls

Here's my new plan:

Show detailed plan

Plan Overview:

- Update all relative URLs in the format `[Text](/path)` to absolute URLs `[Text](https://docs.devin.ai/path)` across all .mdx files in the repository
- Key files to update include `secrets.mdx`, `deployment-capabilities.mdx`, and `get-started.mdx`
- Do not modify image references in the format `![Alt text](/images/path)`
- Verify changes by running `mintlify dev` locally to check link navigation, then create a PR against the main branch

Please review the plan and confirm or request changes. I will proceed if I don't hear from you within 30s. Feel free to press the pause button or start typing if you want me to wait longer.

Confirm

rewrote plan

```
git checkout main && \
git pull && \
git checkout -b devin/$(date +%s)-update-relative-
urls
```

Switched to branch devin/1742930426-update-relative-urls in devin-docs

Searched codebase for `\[[^\]]+\]\(/[^)]+\)`

```
echo "# Files to update relative URLs to absolute
URLs" > ~/todo.txt && \
grep -o "/home/ubuntu/repos/devin-docs/[^:]*"
```

Cooking 0:17

39 files 107 107 Review Changes

Message, command, build here

security.mdx

61 hidden lines

```
<Accordion title="Github Integration Considerations">
When setting up the Github integration, users can select whic

For more details on the requested permissions and security co

</Accordion>

<Accordion title="Slack Integration Considerations">
In Slack, Devin doesn’t read, process or store any data in yo

For more details on the requested permissions and security co

</Accordion>
</AccordionGroup>
```

16 hidden lines

overview.mdx api-reference

24 hidden lines

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL GITLENS

Devin's Termina

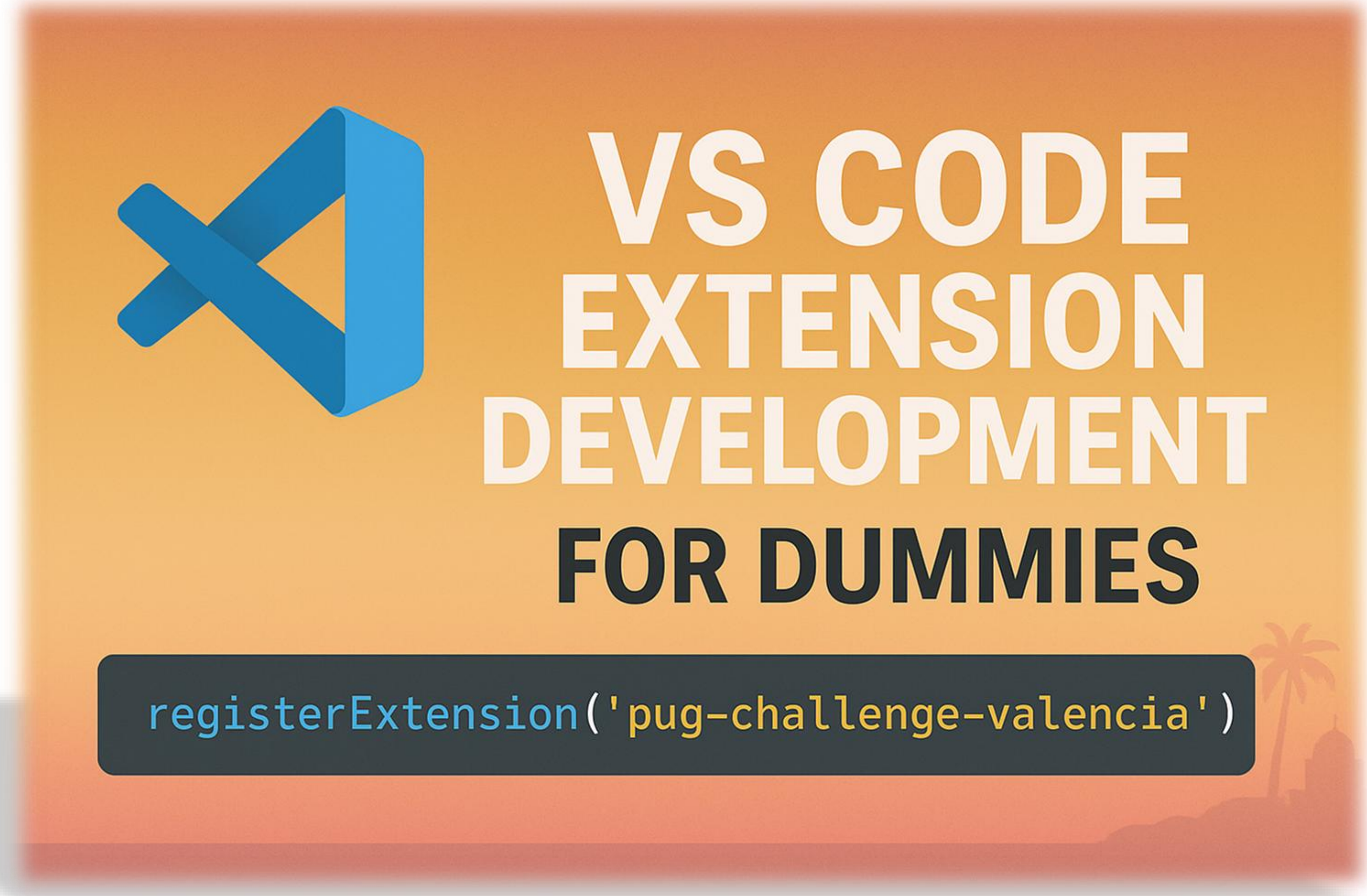
```
Sourcing shell config
ubuntu@devin-box:~$ cd /home/ubuntu
ubuntu@devin-box:~$ cd /home/ubuntu/repos/devin-docs
ubuntu@devin-box:~/repos/devin-docs$ sed -i "s|^- \[ \] /home/ub
me/ubuntu/repos/devin-docs/use-cases/migration-modernization.mdx
grep "migration-modernization.mdx" ~/todo.txt && \
grep -n "^\- \[ \]" ~/todo.txt | head -n 1
- [x] /home/ubuntu/repos/devin-docs/use-cases/migration-moderniz
55:- [ ] /home/ubuntu/repos/devin-docs/use-cases/testing-refacto
ubuntu@devin-box:~/repos/devin-docs$
```

Consultingwerk
software architecture and development

VS Code Extension Development for Dummies

Why Extensions Matter

- Extensions are an **essential part of Visual Studio Code**
- You can also **create your own custom extensions**
- This presentation will show you **how to build a VS Code extension**
  - **Quickly and easily**
  - **No prior knowledge required**



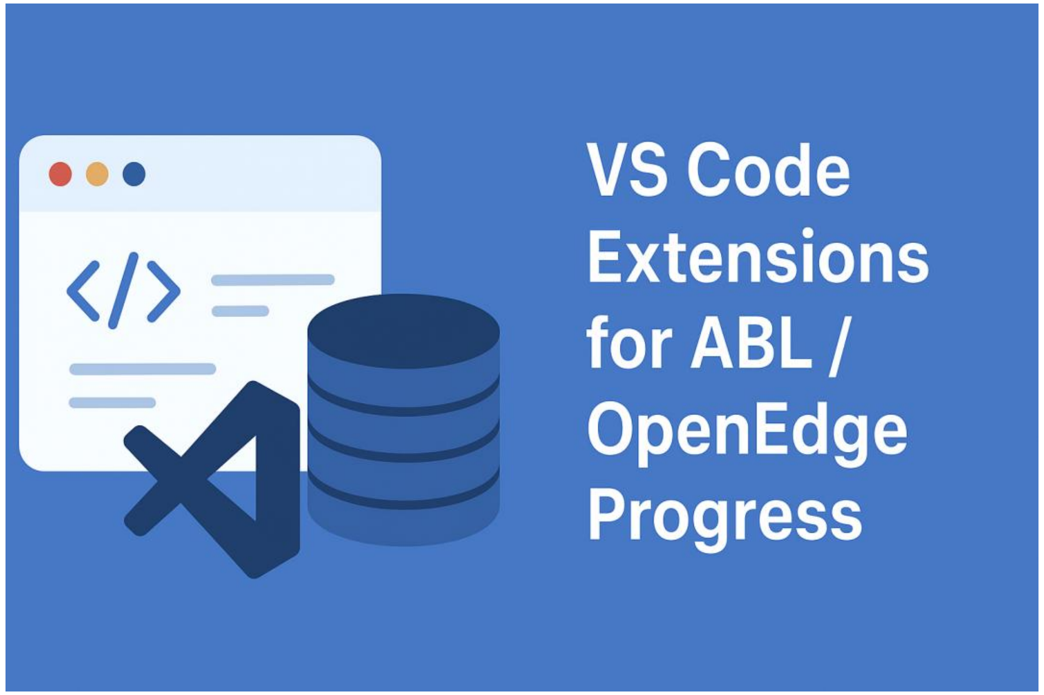
```
registerExtension('pug-challenge-valencia')
```

Consultingwerk
software architecture and development

VS Code Extension Development for Dummies

VS Code Extensions for ABL / OpenEdge Progress

- **ABLUnit Test Runner (kherring):** Runs ABLUnit tests directly in VS Code.
- **OpenEdge ABL (riversidesoftware):** Syntax highlighting & IntelliSense for ABL code.
- **CABL / SonarLint (riversidesoftware):** Real-time code quality checking for ABL.
- **ProBro (balticamadeus):** OpenEdge database browser for easier data analysis.
- **OpenEdge ABL Formatter (balticamadeus):** Automatically formats ABL code.



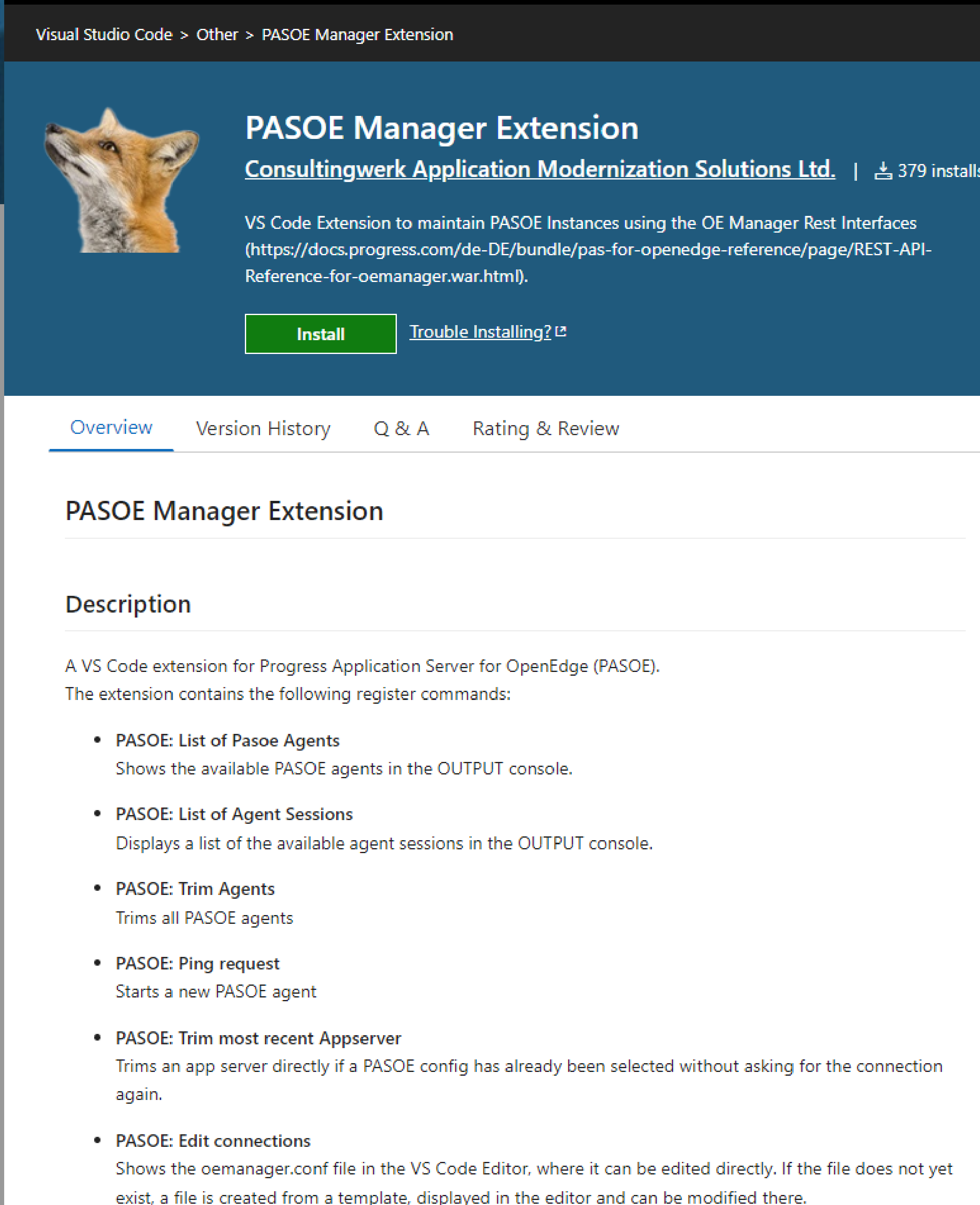
Agenda

- Sample PASOE Manager Extension
- VS Code Marketplace and alternatives
- VS Code as an IDE
- Microsoft SDK for VS Code Extensions
- Building an Extension with Visual Studio Code
- Programming languages and file types used in a VS Code extension
- Event Callbacks
- Implementing the PASOE Manager Extension
- Package and publish your extension
- Important Common Capabilities
- Questions

Sample PASOE Manager Extension

My First Self-Created Extension

- A **useful VS Code extension** we use regularly in our daily work
- Designed for **Progress Application Server for OpenEdge (PASOE)**
- **Monitor/Manage PASOE instances directly from VS Code**
- Provides several **registered commands**



PASOE Manager Extension

Consultingwerk Application Modernization Solutions Ltd. | 379 install

VS Code Extension to maintain PASOE Instances using the OE Manager Rest Interfaces (https://docs.progress.com/de-DE/bundle/pas-for-openedge-reference/page/REST-API-Reference-for-oemanager.war.html).

PASOE Manager Extension

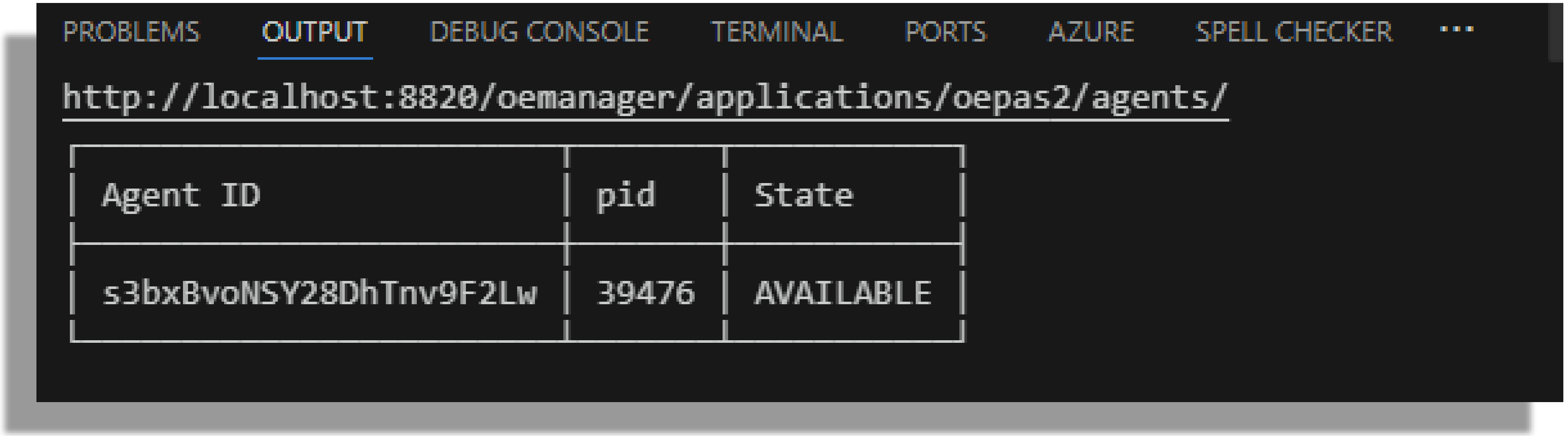
Description

A VS Code extension for Progress Application Server for OpenEdge (PASOE).
The extension contains the following register commands:

- PASOE: List of Pasoe Agents
  Shows the available PASOE agents in the OUTPUT console.
- PASOE: List of Agent Sessions
  Displays a list of the available agent sessions in the OUTPUT console.
- PASOE: Trim Agents
  Trims all PASOE agents
- PASOE: Ping request
  Starts a new PASOE agent
- PASOE: Trim most recent Appserver
  Trims an app server directly if a PASOE config has already been selected without asking for the connection again.
- PASOE: Edit connections
  Shows the oemanager.conf file in the VS Code Editor, where it can be edited directly. If the file does not yet exist, a file is created from a template, displayed in the editor and can be modified there.

Sample PASOE Manager Extension

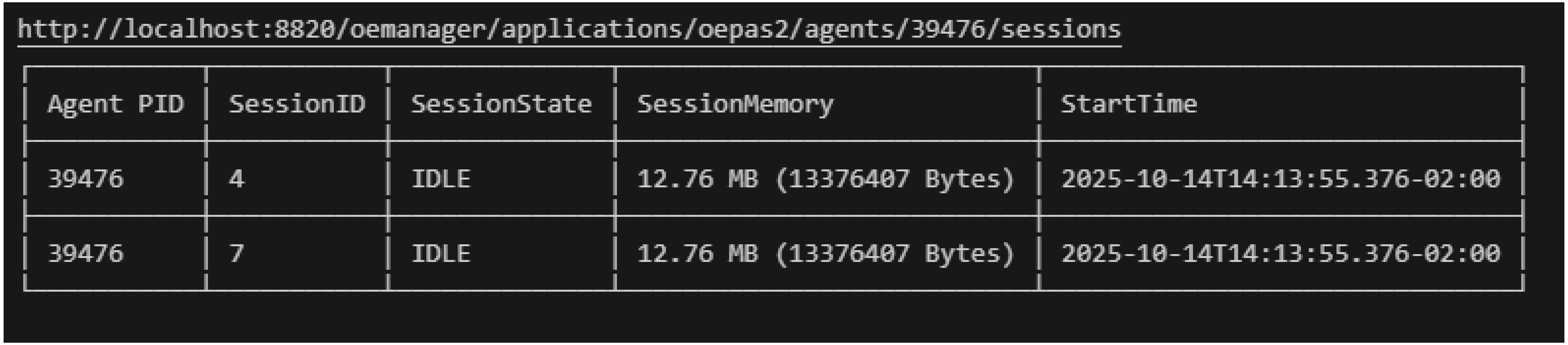
PASOE: List of PASOE Agents



```
http://localhost:8820/oemanager/applications/oepas2/agents/

 ┌──────────────────────────┬───────┬────────────┐
 │ Agent ID                 │ pid   │ State      │
 ├──────────────────────────┼───────┼────────────┤
 │ s3bxBvoNSY28DhTnv9F2Lw   │ 39476 │ AVAILABLE  │
 └──────────────────────────┴───────┴────────────┘
```

Sample PASOE Manager Extension

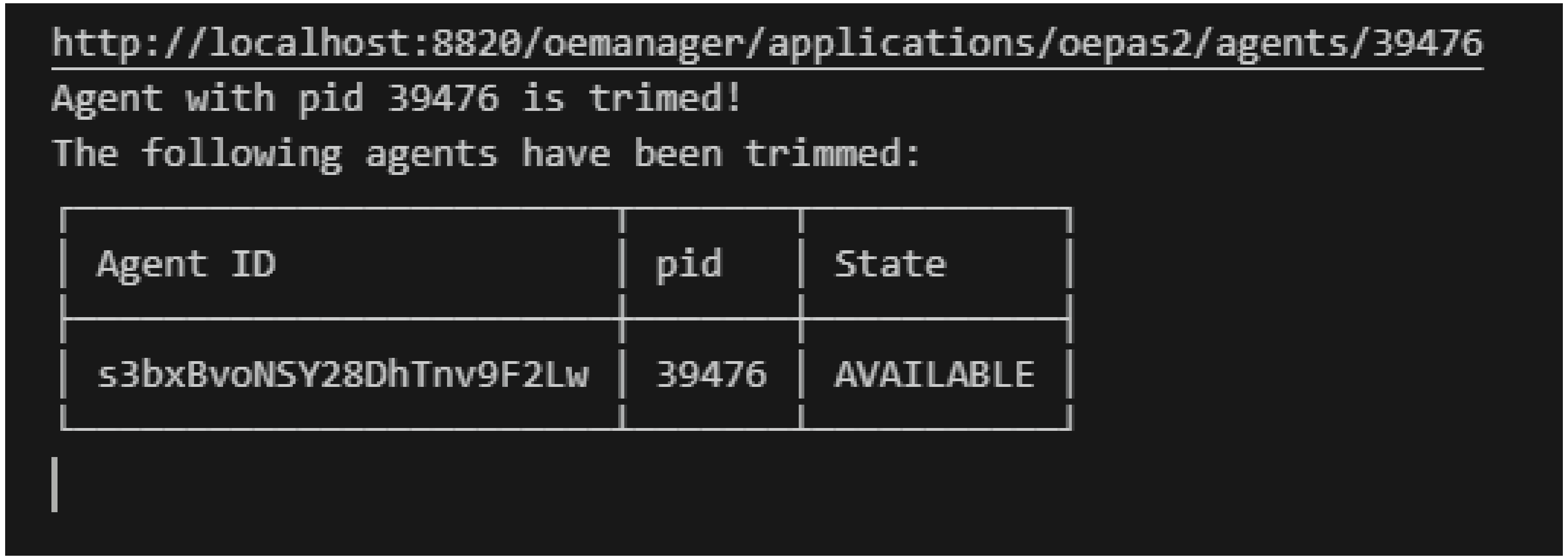
PASOE: List of Agent Sessions



```
http://localhost:8820/oemanager/applications/oepas2/agents/39476/sessions

| Agent PID | SessionID | SessionState | SessionMemory               | StartTime                     |
|-----------|-----------|--------------|-----------------------------|-------------------------------|
| 39476     | 4         | IDLE         | 12.76 MB (13376407 Bytes)   | 2025-10-14T14:13:55.376-02:00 |
| 39476     | 7         | IDLE         | 12.76 MB (13376407 Bytes)   | 2025-10-14T14:13:55.376-02:00 |
```

Sample PASOE Manager Extension

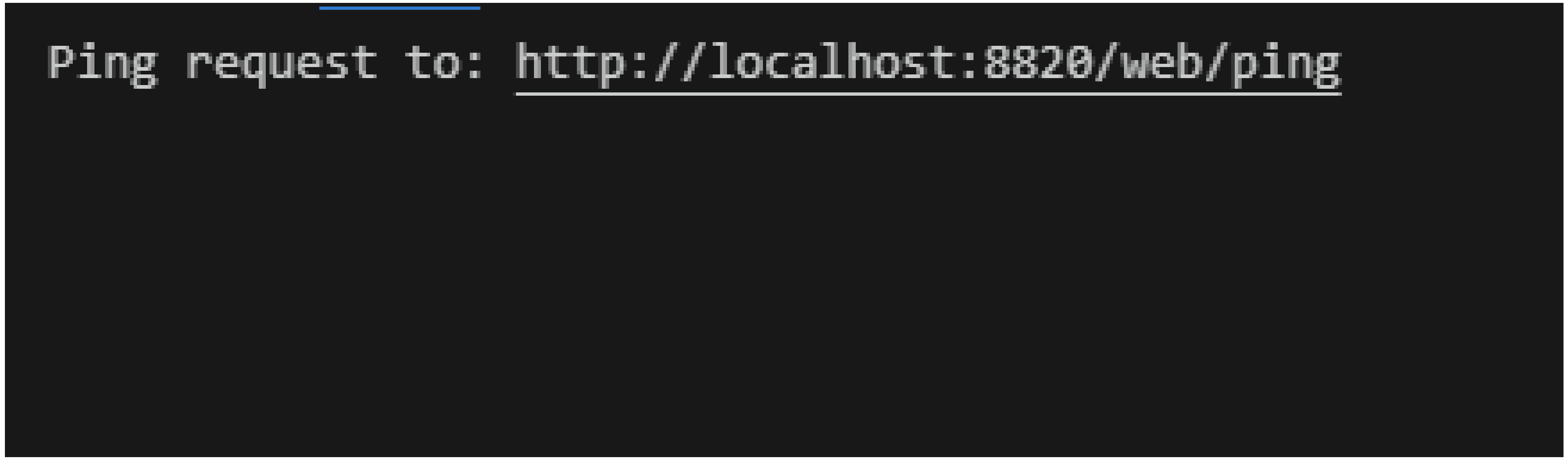
PASOE: Trim Agents



```
http://localhost:8820/oemanager/applications/oepas2/agents/39476
Agent with pid 39476 is trimed!
The following agents have been trimmed:

 ┌─────────────────────────┬───────┬───────────┐
 │ Agent ID                │ pid   │ State     │
 ├─────────────────────────┼───────┼───────────┤
 │ s3bxBvoNSY28DhTnv9F2Lw  │ 39476 │ AVAILABLE │
 └─────────────────────────┴───────┴───────────┘

|
```

Sample PASOE Manager Extension

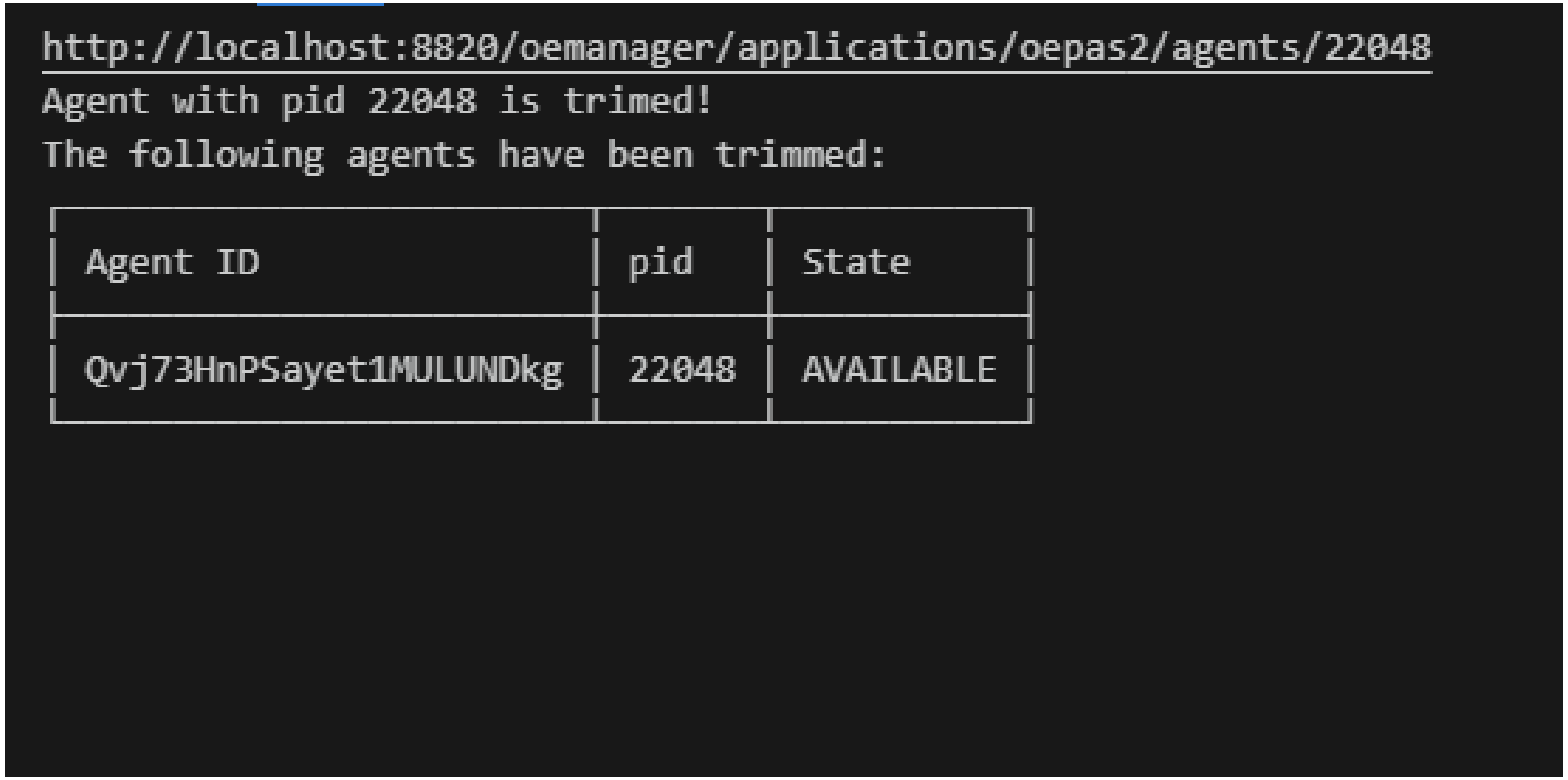
PASOE: Ping Request



```
Ping request to: http://localhost:8820/web/ping
```

Sample PASOE Manager Extension

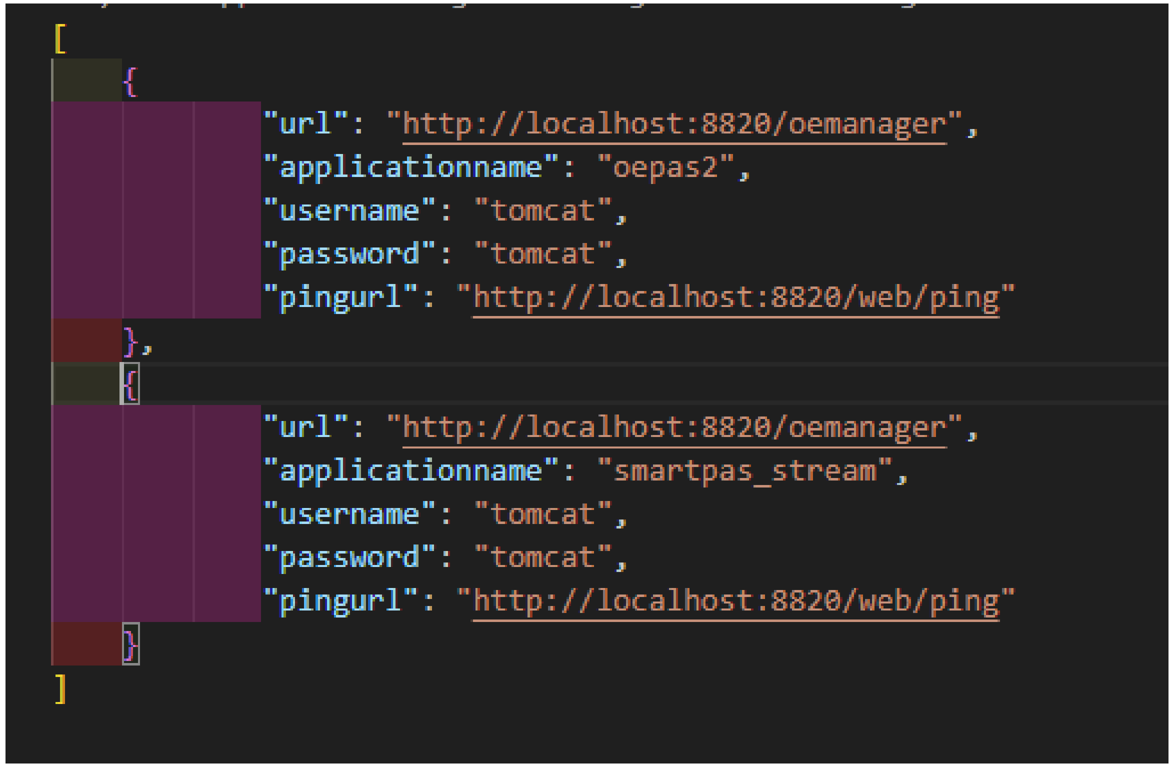
PASOE: Trim Most Recent AppServer



```
http://localhost:8820/oemanager/applications/oepas2/agents/22048
Agent with pid 22048 is trimed!
The following agents have been trimmed:

┌────────────────────────┬───────┬───────────┐
│ Agent ID               │ pid   │ State     │
├────────────────────────┼───────┼───────────┤
│ Qvj73HnPSayet1MULUNDkg │ 22048 │ AVAILABLE │
└────────────────────────┴───────┴───────────┘
```

Sample PASOE Manager Extension

PASOE: Edit Connections



```
[
    {
            "url": "http://localhost:8820/oemanager",
            "applicationname": "oepas2",
            "username": "tomcat",
            "password": "tomcat",
            "pingurl": "http://localhost:8820/web/ping"
    },
    {
            "url": "http://localhost:8820/oemanager",
            "applicationname": "smartpas_stream",
            "username": "tomcat",
            "password": "tomcat",
            "pingurl": "http://localhost:8820/web/ping"
    }
]
```

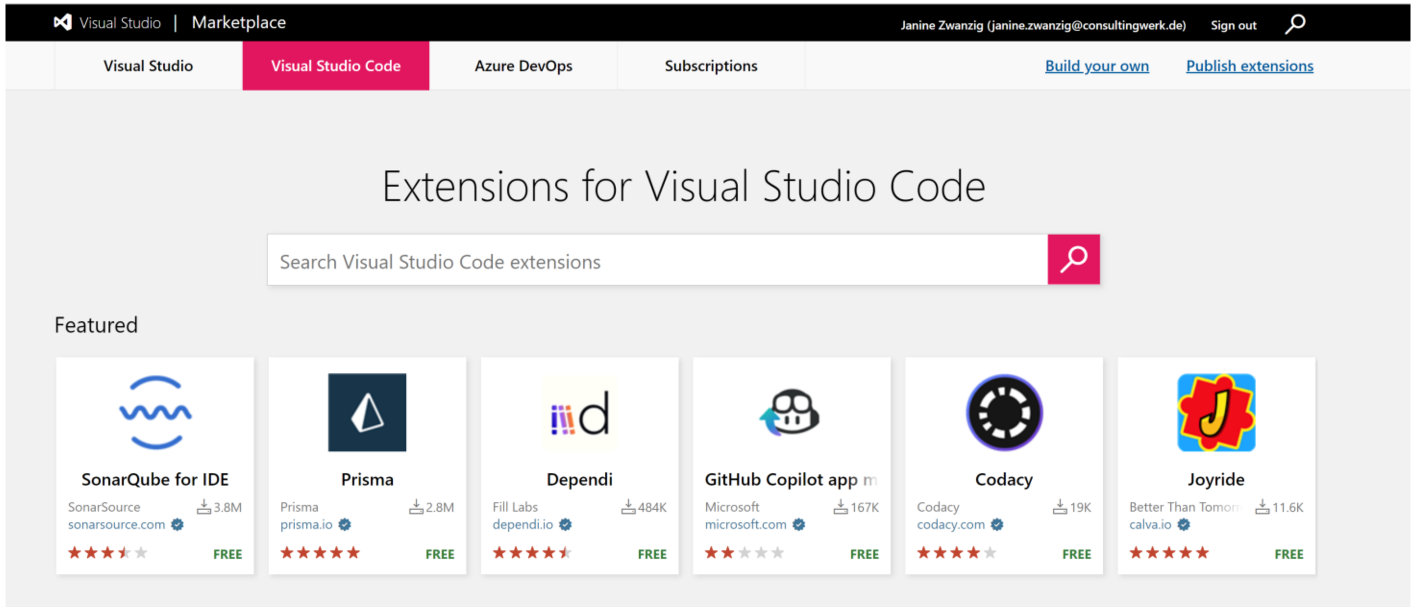
Agenda

- Sample PASOE Manager Extension
- VS Code Marketplace and alternatives
- VS Code as an IDE
- Microsoft SDK for VS Code Extensions
- Building an Extension with Visual Studio Code
- Programming languages and file types used in a VS Code extension
- Event Callbacks
- Implementing the PASOE Manager Extension
- Package and publish your extension
- Important Common Capabilities
- Questions

VS Code Marketplace and alternatives

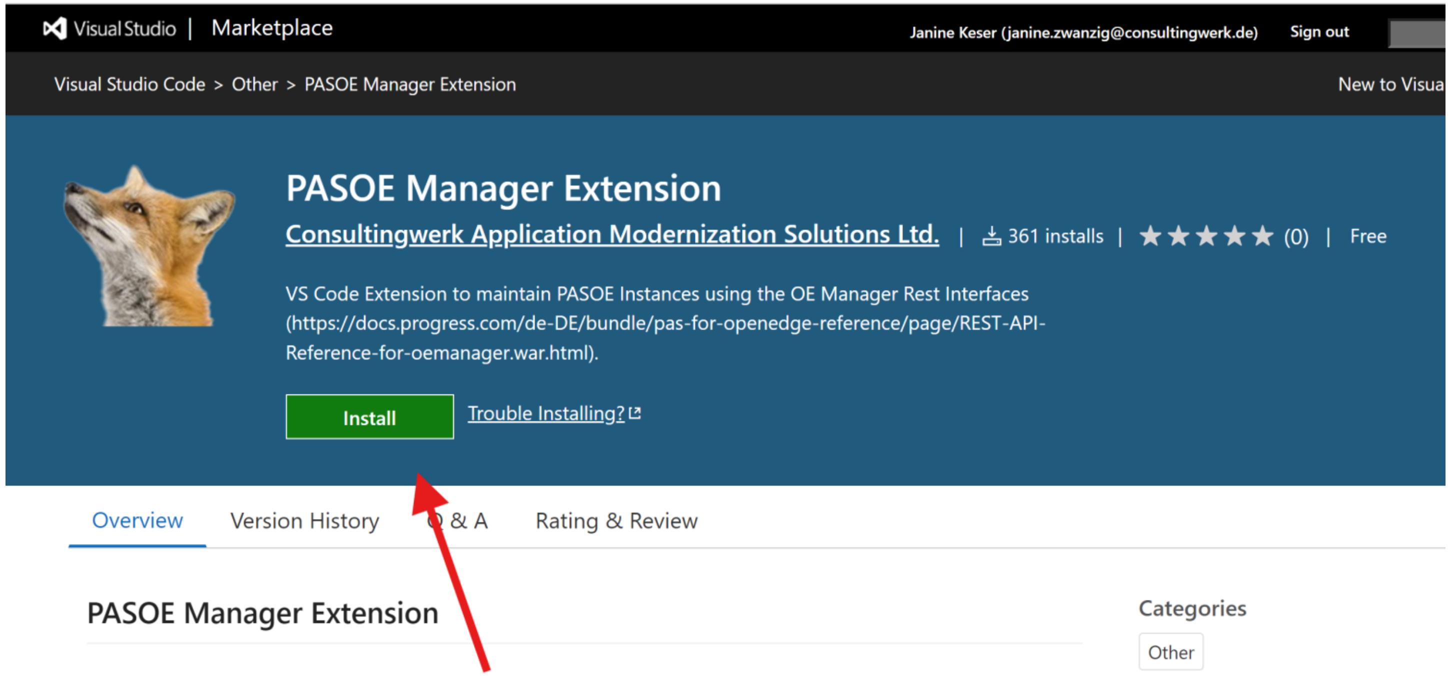
https://marketplace.visualstudio.com/

- **VS Code Marketplace** – Online platform for VS Code extensions
- Discover, install, and manage extensions
- Enhance coding, debugging, and workflows
- Makes VS Code flexible and powerful


Visual Studio | Marketplace
Janine Zwanzig (janine.zwanzig@consultingwerk.de)
Sign out
Visual Studio
Visual Studio Code
Azure DevOps
Subscriptions
Build your own
Publish extensions
Extensions for Visual Studio Code
Search Visual Studio Code extensions
Featured
SonarQube for IDE
SonarSource 3.8M sonarsource.com FREE
Prisma
Prisma 2.8M prisma.io FREE
Dependi
Fill Labs 484K dependi.io FREE
GitHub Copilot app m
Microsoft 167K microsoft.com FREE
Codacy
Codacy 19K codacy.com FREE
Joyride
Better Than Tomor 11.6K calva.io FREE

VS Code Marketplace and alternatives

- Browse, install, and manage extensions directly from the Marketplace
- Use the **Search** field to find the PASOE Manager extension
- Install an extension quickly with the **Install** button


Visual Studio | Marketplace
Janine Keser (janine.zwanzig@consultingwerk.de)
Sign out
Visual Studio Code > Other > PASOE Manager Extension
New to Visua
Install
Trouble Installing?
Overview
Version History
Q & A
Rating & Review
Categories
Other

PASOE Manager Extension

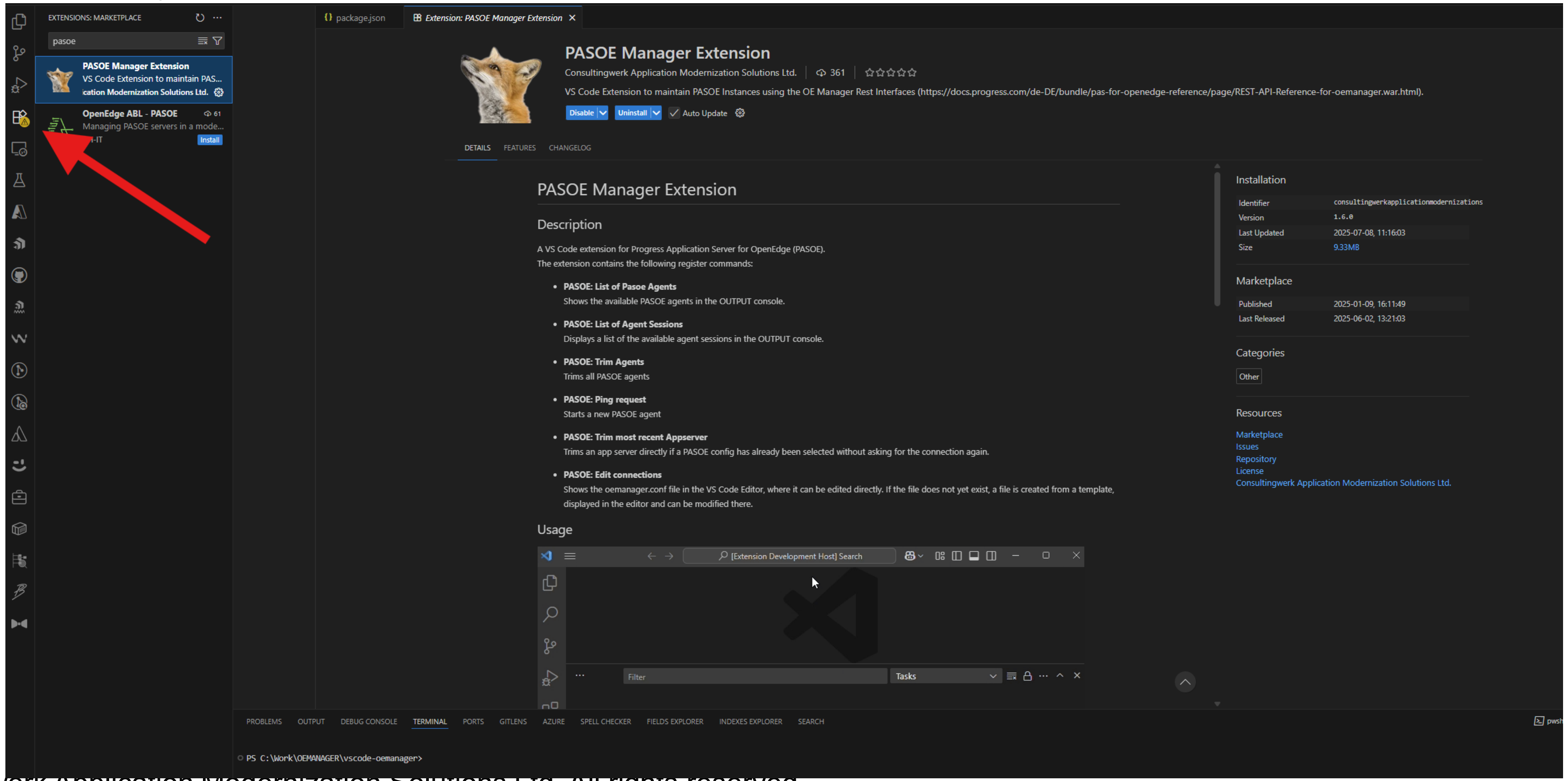
Consultingwerk Application Modernization Solutions Ltd. | 361 installs | ★★★★★ (0) | Free

VS Code Extension to maintain PASOE Instances using the OE Manager Rest Interfaces (https://docs.progress.com/de-DE/bundle/pas-for-openedge-reference/page/REST-API-Reference-for-oemanager.war.html).

PASOE Manager Extension

VS Code Marketplace and alternatives

Alternatively, you can also find and manage the extensions directly via the Marketplace Icon in the Activity Bar from VS Code.



PASOE Manager Extension
VS Code Extension to maintain PAS...
ication Modernization Solutions Ltd.

OpenEdge ABL - PASOE 61
Managing PASOE servers in a mode...
-IT

PASOE Manager Extension
Consultingwerk Application Modernization Solutions Ltd. | 361 | ☆☆☆☆☆
VS Code Extension to maintain PASOE Instances using the OE Manager Rest Interfaces (https://docs.progress.com/de-DE/bundle/pas-for-openedge-reference/page/REST-API-Reference-for-oemanager.war.html).

PASOE Manager Extension

Description

A VS Code extension for Progress Application Server for OpenEdge (PASOE).
The extension contains the following register commands:

- **PASOE: List of Pasoe Agents**
  Shows the available PASOE agents in the OUTPUT console.
- **PASOE: List of Agent Sessions**
  Displays a list of the available agent sessions in the OUTPUT console.
- **PASOE: Trim Agents**
  Trims all PASOE agents
- **PASOE: Ping request**
  Starts a new PASOE agent
- **PASOE: Trim most recent Appserver**
  Trims an app server directly if a PASOE config has already been selected without asking for the connection again.
- **PASOE: Edit connections**
  Shows the oemanager.conf file in the VS Code Editor, where it can be edited directly. If the file does not yet exist, a file is created from a template, displayed in the editor and can be modified there.

Usage

Installation

Identifier	consultingwerkapplicationmodernizations
Version	1.6.0
Last Updated	2025-07-08, 11:16:03
Size	9.33MB

Marketplace

Published	2025-01-09, 16:11:49
Last Released	2025-06-02, 13:21:03

Categories

Other

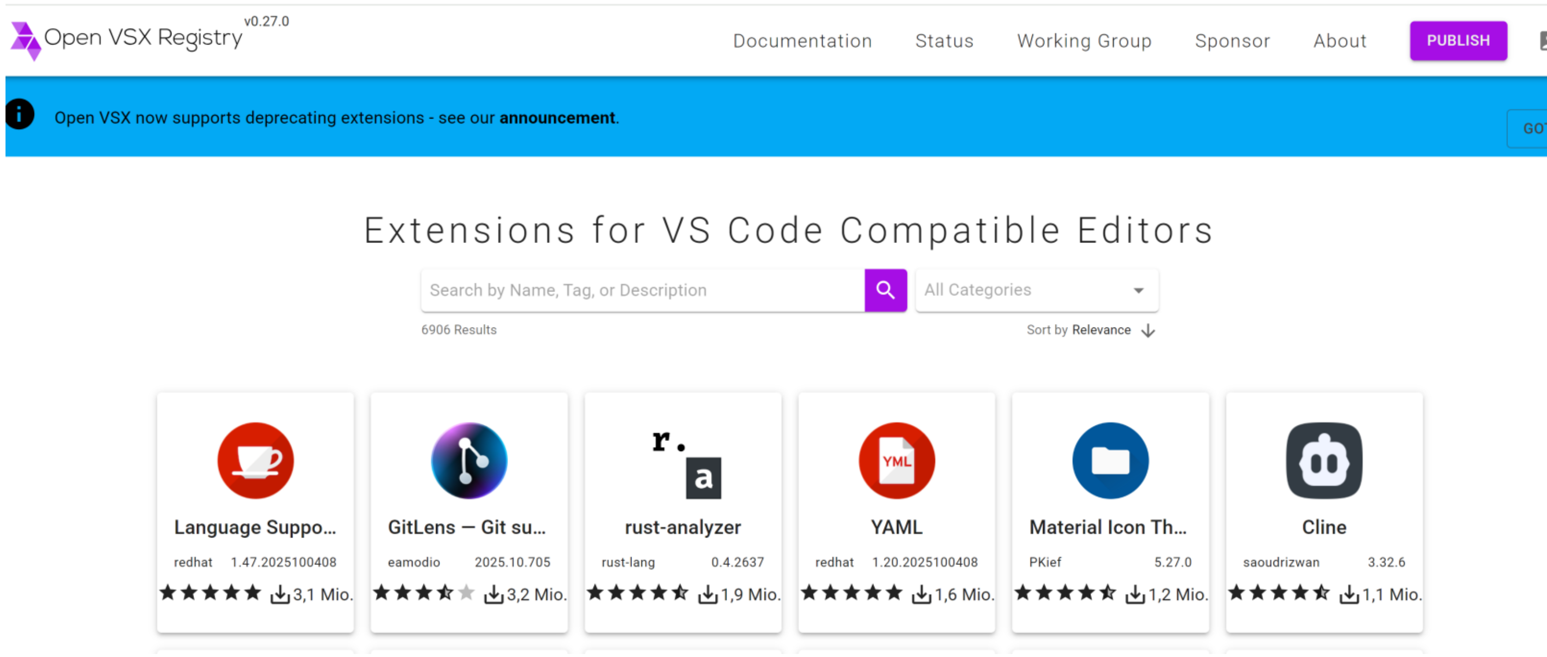
Resources

Marketplace
Issues
Repository
License
Consultingwerk Application Modernization Solutions Ltd.

```
PS C:\Work\OEMANAGER\vscode-oemanager>
```

VS Code Marketplace and alternatives

- **Open VSX Registry** – a good alternative to the VS Code Marketplace: https://open-vsx.org/
- Most extensions available in the Marketplace can also be found here
- Usage and management is very similar to the Marketplace
- Extensions can also be used in VS Code forks like **Windsurf or Cursor**.



Open VSX now supports deprecating extensions - see our **announcement**.

Extensions for VS Code Compatible Editors

6906 Results

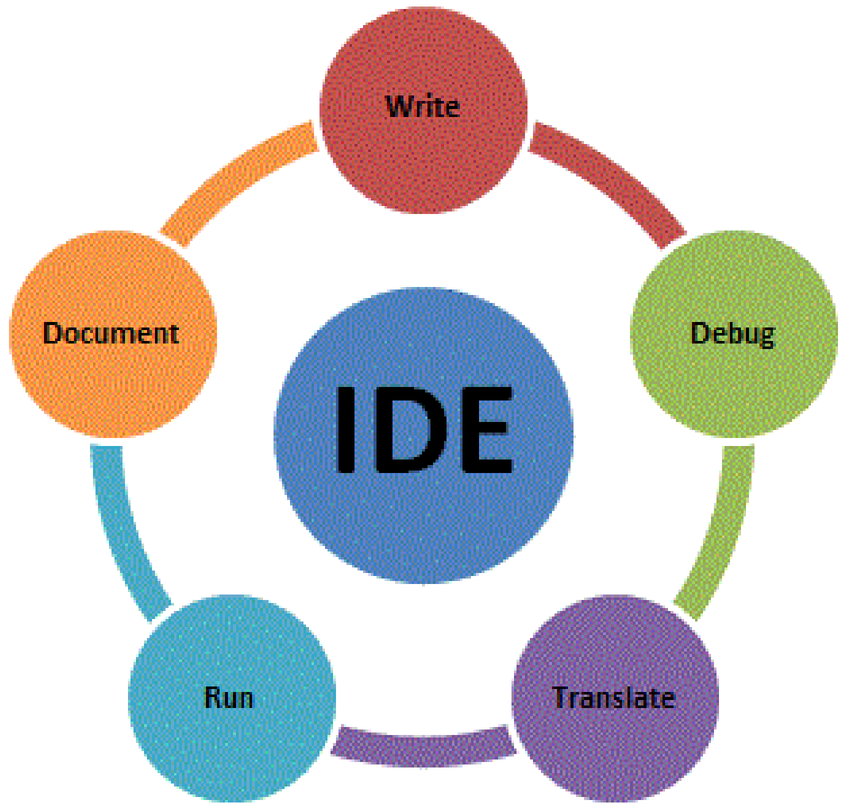
Sort by Relevance

Language Suppo...	redhat	1.47.2025100408	3,1 Mio.
GitLens — Git su...	eamodio	2025.10.705	3,2 Mio.
rust-analyzer	rust-lang	0.4.2637	1,9 Mio.
YAML	redhat	1.20.2025100408	1,6 Mio.
Material Icon Th...	PKief	5.27.0	1,2 Mio.
Cline	saoudrizwan	3.32.6	1,1 Mio.

Agenda

- Sample PASOE Manager Extension
- VS Code Marketplace and alternatives
- VS Code as an IDE
- Microsoft SDK for VS Code Extensions
- Building an Extension with Visual Studio Code
- Programming languages and file types used in a VS Code extension
- Event Callbacks
- Implementing the PASOE Manager Extension
- Package and publish your extension
- Important Common Capabilities
- Questions

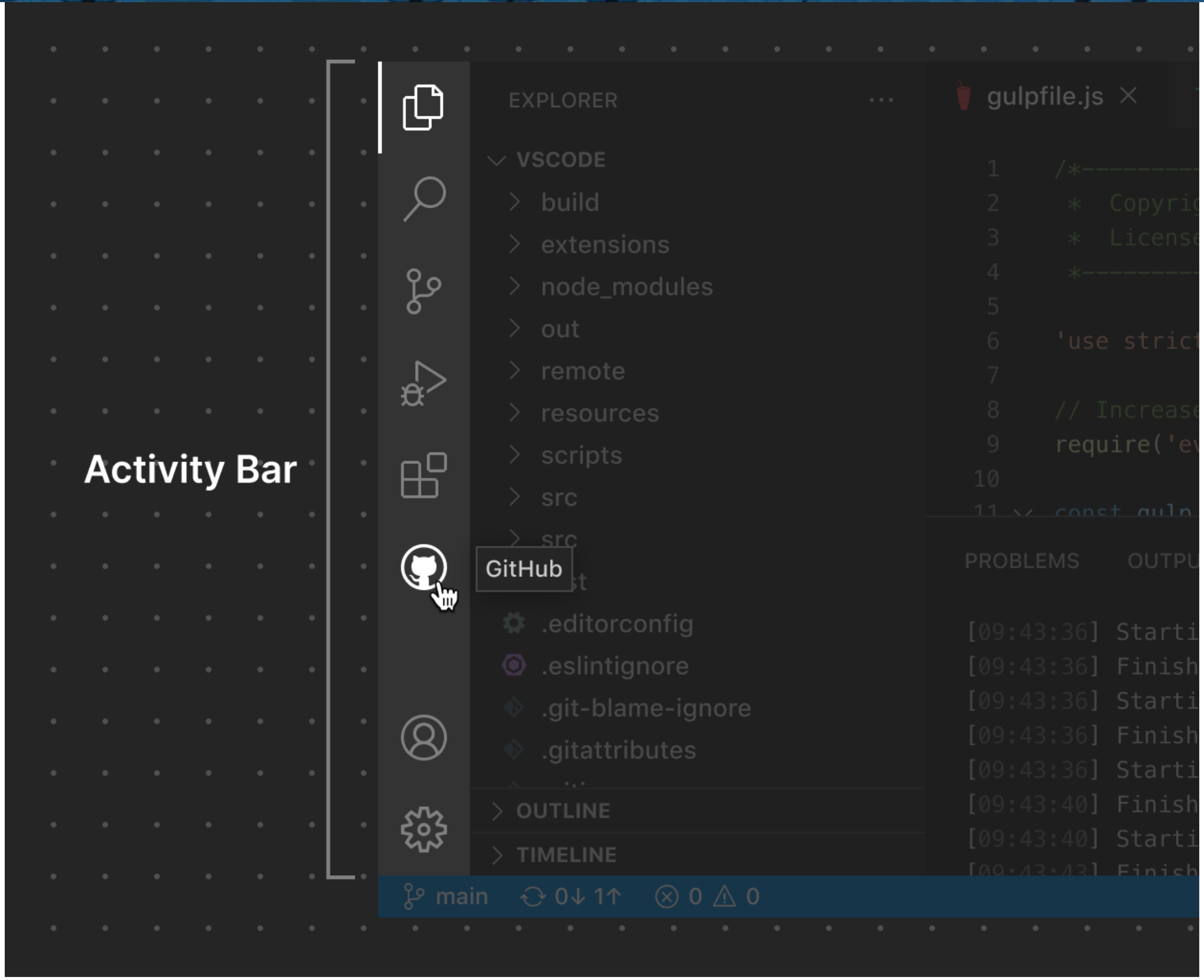
VS Code as an IDE



- **Visual Studio Code** – modern, streamlined IDE with a clear user interface
- IDE = Integrated Development Environment – software that bundles programming tools
- VS Code offers:
  - Code Editor – syntax highlighting, autocomplete, code navigation
  - Debugger – can be used directly in the editor
  - Extensions & Tools – extensions for different languages and workflows
- Everything in one interface – no separate tools required

VS Code as an IDE

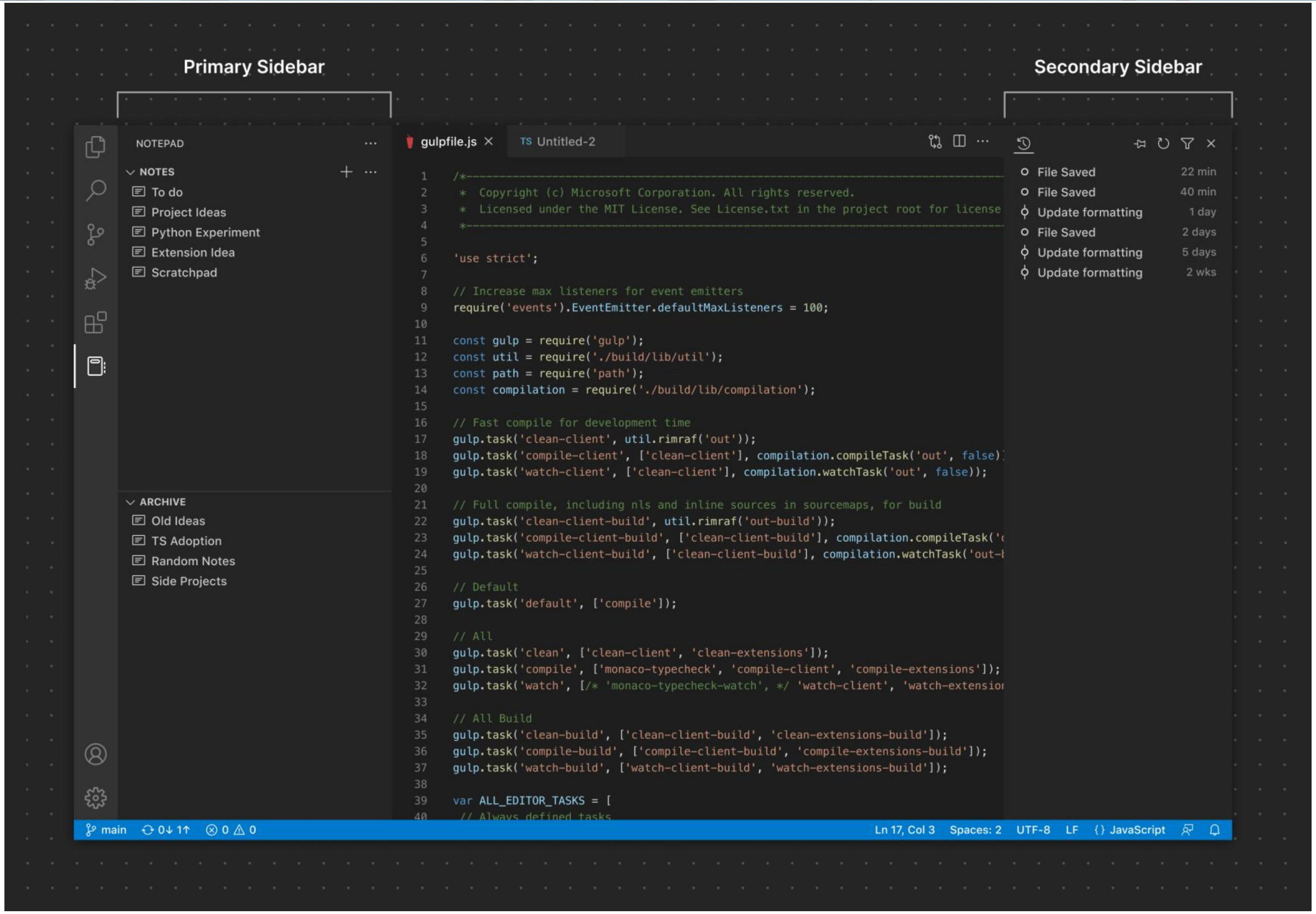
- **Activity Bar** – located on the left, a core navigation surface in VS Code
- Extensions can add **View Containers** to the Activity Bar, appearing as **Activity Bar Items**



VS Code as an IDE

Primary Sidebar – displays main content based on the selected area (e.g., project structure)

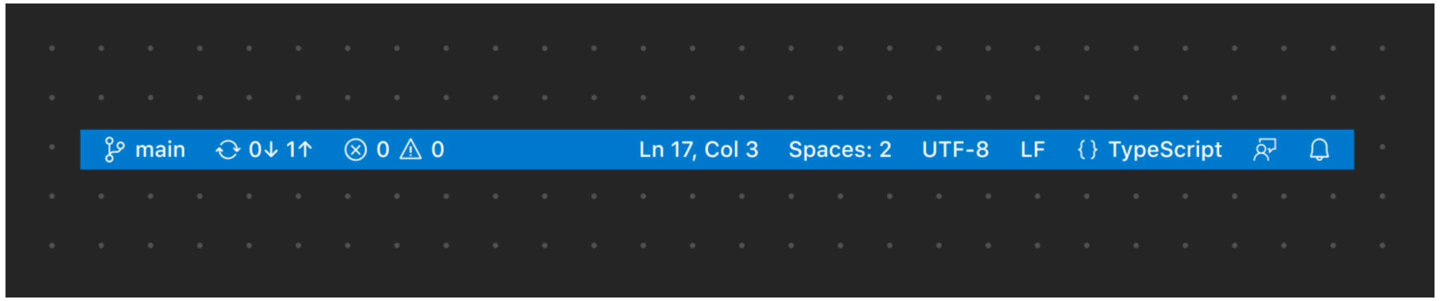
Secondary Sidebar – shows additional views side by side with the primary content



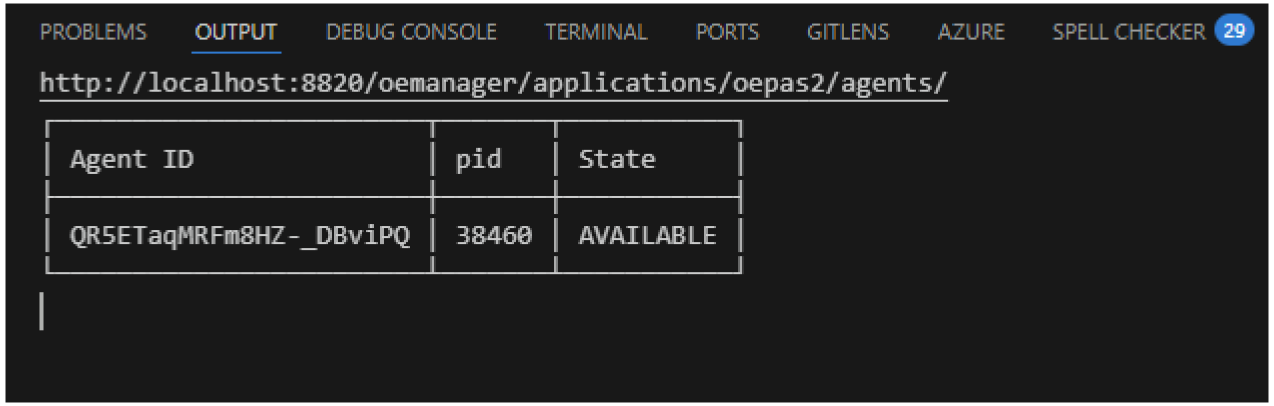
```
/*---------------------------------------------------------------------------------------------
 *  Copyright (c) Microsoft Corporation. All rights reserved.
 *  Licensed under the MIT License. See License.txt in the project root for license
 *--------------------------------------------------------------------------------------------

'use strict';

// Increase max listeners for event emitters
require('events').EventEmitter.defaultMaxListeners = 100;

const gulp = require('gulp');
const util = require('./build/lib/util');
const path = require('path');
const compilation = require('./build/lib/compilation');

// Fast compile for development time
gulp.task('clean-client', util.rimraf('out'));
gulp.task('compile-client', ['clean-client'], compilation.compileTask('out', false)
gulp.task('watch-client', ['clean-client'], compilation.watchTask('out', false));

// Full compile, including nls and inline sources in sourcemaps, for build
gulp.task('clean-client-build', util.rimraf('out-build'));
gulp.task('compile-client-build', ['clean-client-build'], compilation.compileTask('
gulp.task('watch-client-build', ['clean-client-build'], compilation.watchTask('out-

// Default
gulp.task('default', ['compile']);

// All
gulp.task('clean', ['clean-client', 'clean-extensions']);
gulp.task('compile', ['monaco-typecheck', 'compile-client', 'compile-extensions']);
gulp.task('watch', [/* 'monaco-typecheck-watch', */ 'watch-client', 'watch-extensio

// All Build
gulp.task('clean-build', ['clean-client-build', 'clean-extensions-build']);
gulp.task('compile-build', ['compile-client-build', 'compile-extensions-build']);
gulp.task('watch-build', ['watch-client-build', 'watch-extensions-build']);

var ALL_EDITOR_TASKS = [
  // Always defined tasks
```

VS Code as an IDE

- Status Bar (bottom) – shows important information: Git branch, errors, current language settin
- Extensions can add their own indicators (e.g., Atlassian extension shows current JIRA issue)



- Output Panel – displays program and process output

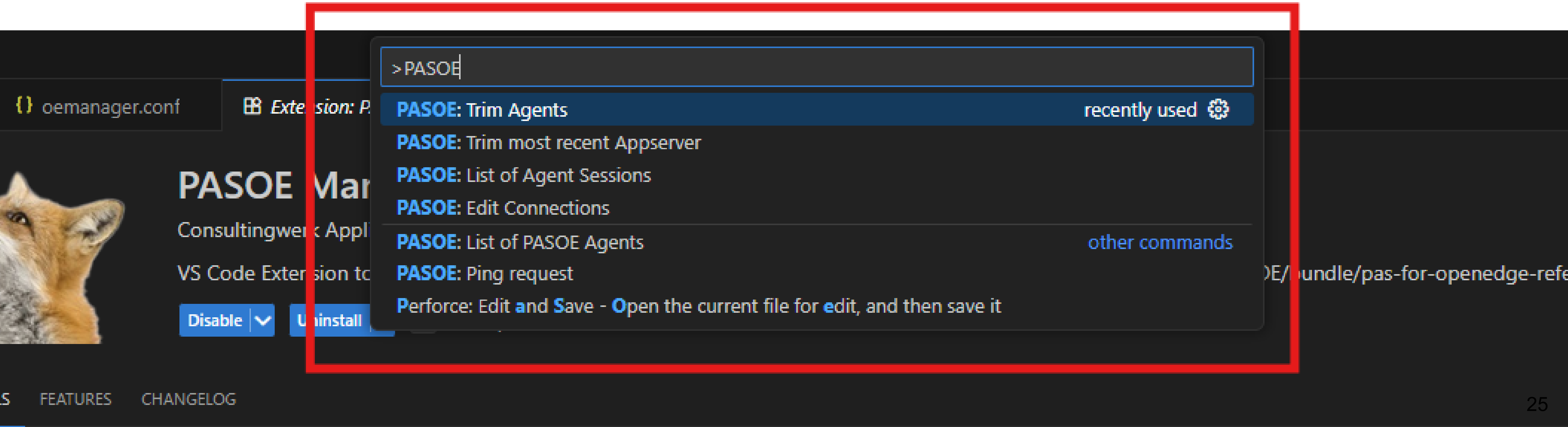


```
http://localhost:8820/oemanager/applications/oepas2/agents/
┌────────────────────────┬───────┬───────────┐
│ Agent ID               │ pid   │ State     │
├────────────────────────┼───────┼───────────┤
│ QR5ETaqMRFm8HZ-_DBviPQ │ 38460 │ AVAILABLE │
└────────────────────────┴───────┴───────────┘
```

VS Code as an IDE

Another key tool is the Command Palette:

- Accessible via Ctrl + Shift + P
- Which can be used to quickly execute almost any function in VS Code.



```
>PASOE
PASOE: Trim Agents                                   recently used
PASOE: Trim most recent Appserver
PASOE: List of Agent Sessions
PASOE: Edit Connections
PASOE: List of PASOE Agents                          other commands
PASOE: Ping request
Perforce: Edit and Save - Open the current file for edit, and then save it
```

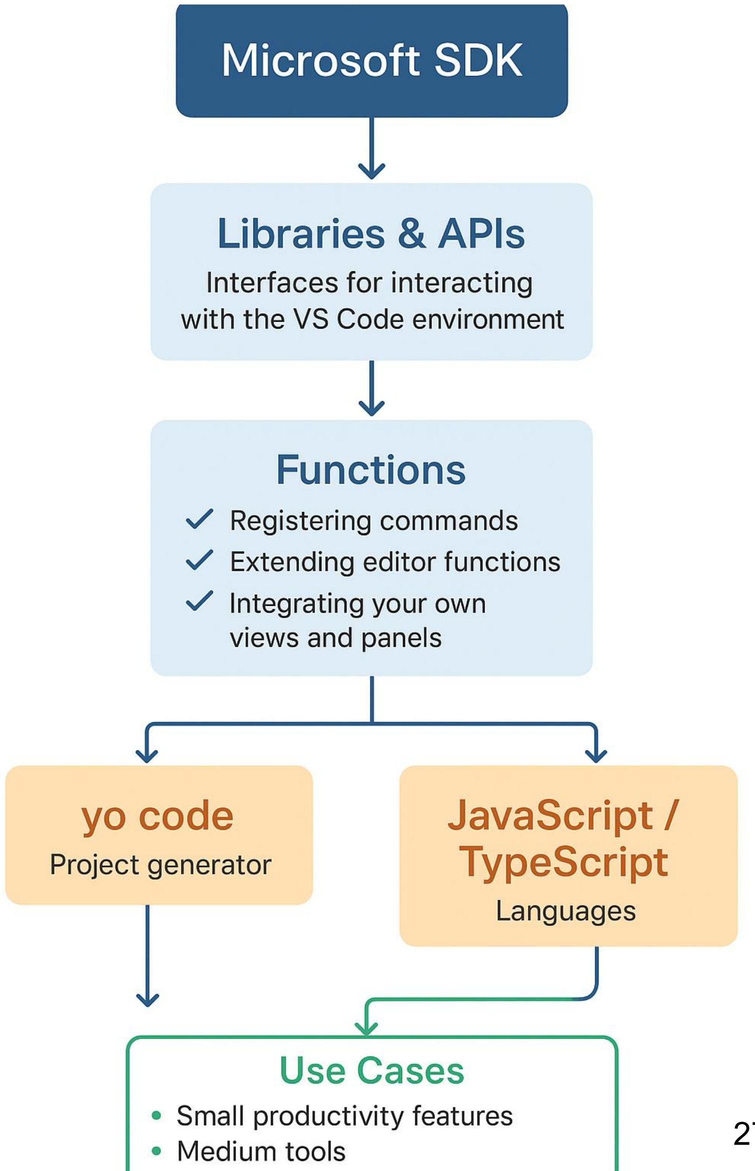
Consultingwerk
software architecture and development

Agenda

- Sample PASOE Manager Extension
- VS Code Marketplace and alternatives
- VS Code as an IDE
- Microsoft SDK for VS Code Extensions
- Building an Extension with Visual Studio Code
- Programming languages and file types used in a VS Code extension
- Event Callbacks
- Implementing the PASOE Manager Extension
- Package and publish your extension
- Important Common Capabilities
- Questions

Microsoft SDK for VS Code Extensions

- Provided by Microsoft for building VS Code extensions
- Libraries & APIs to interact with the VS Code environment
- Enables
  - Registering commands
  - Editor extensions
  - Custom views & panels
- Includes **‘yo code’** tool to scaffold new projects
- Built with **JavaScript** or **TypeScript**
- For simple add-ons or advanced developer tools


Microsoft SDK
Libraries & APIs
Interfaces for interacting with the VS Code environment
Functions
✓ Registering commands
✓ Extending editor functions
✓ Integrating your own views and panels
yo code
Project generator
JavaScript / TypeScript
Languages
Use Cases
• Small productivity features
• Medium tools

Agenda

- Sample PASOE Manager Extension
- VS Code Marketplace and alternatives
- VS Code as an IDE
- Microsoft SDK for VS Code Extensions
- Building an Extension with Visual Studio Code
- Programming languages and file types used in a VS Code extension
- Event Callbacks
- Implementing the PASOE Manager Extension
- Package and publish your extension
- Important Common Capabilities
- Questions

Consultingwerk
software architecture and development

Building an Extension with Visual Studio Code “Your first Extension”

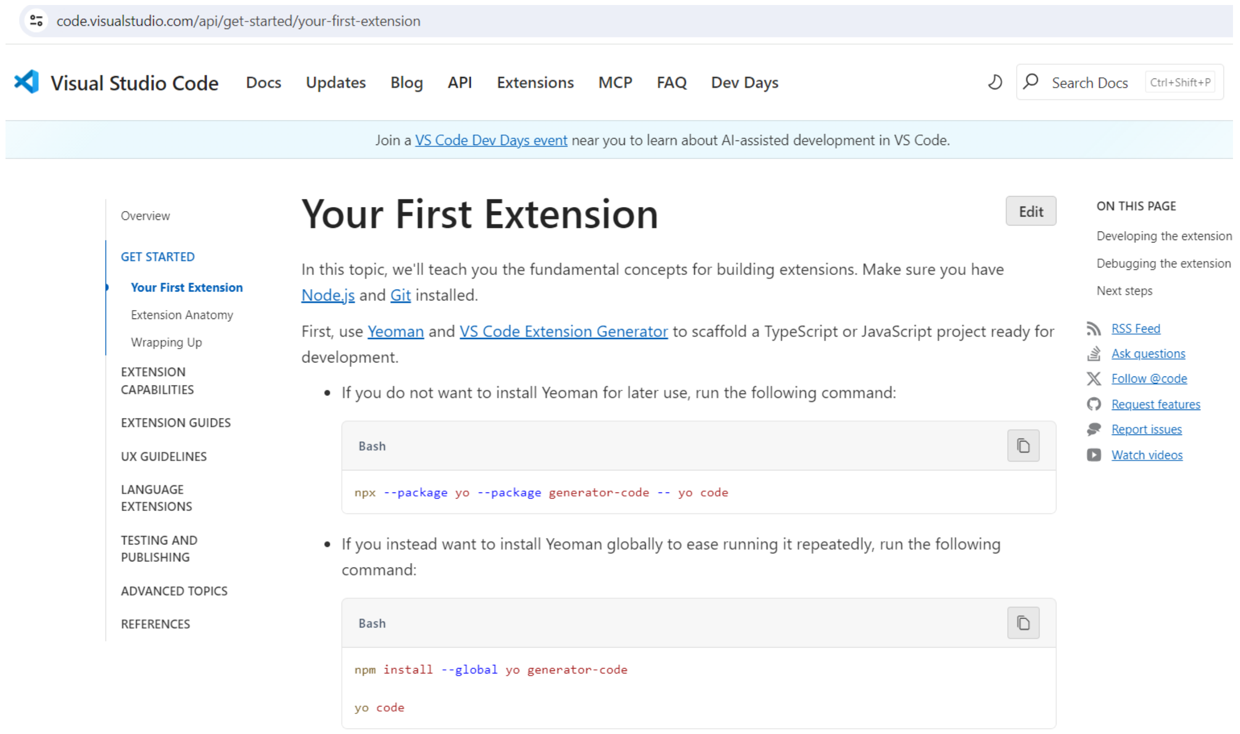
Building Your First Extension – step-by-step guide available at:

https://code.visualstudio.com/api/get-started/your-first-extension

Easiest way to create an extension:

- Open command prompt and create a project directory
- Run command:

```
npx --package yo --package generator-code -- yo code
```



code.visualstudio.com/api/get-started/your-first-extension

Join a VS Code Dev Days event near you to learn about AI-assisted development in VS Code.

Your First Extension

In this topic, we'll teach you the fundamental concepts for building extensions. Make sure you have Node.js and Git installed.

First, use Yeoman and VS Code Extension Generator to scaffold a TypeScript or JavaScript project ready for development.

- If you do not want to install Yeoman for later use, run the following command:

Bash
```
npx --package yo --package generator-code -- yo code
```

- If you instead want to install Yeoman globally to ease running it repeatedly, run the following command:

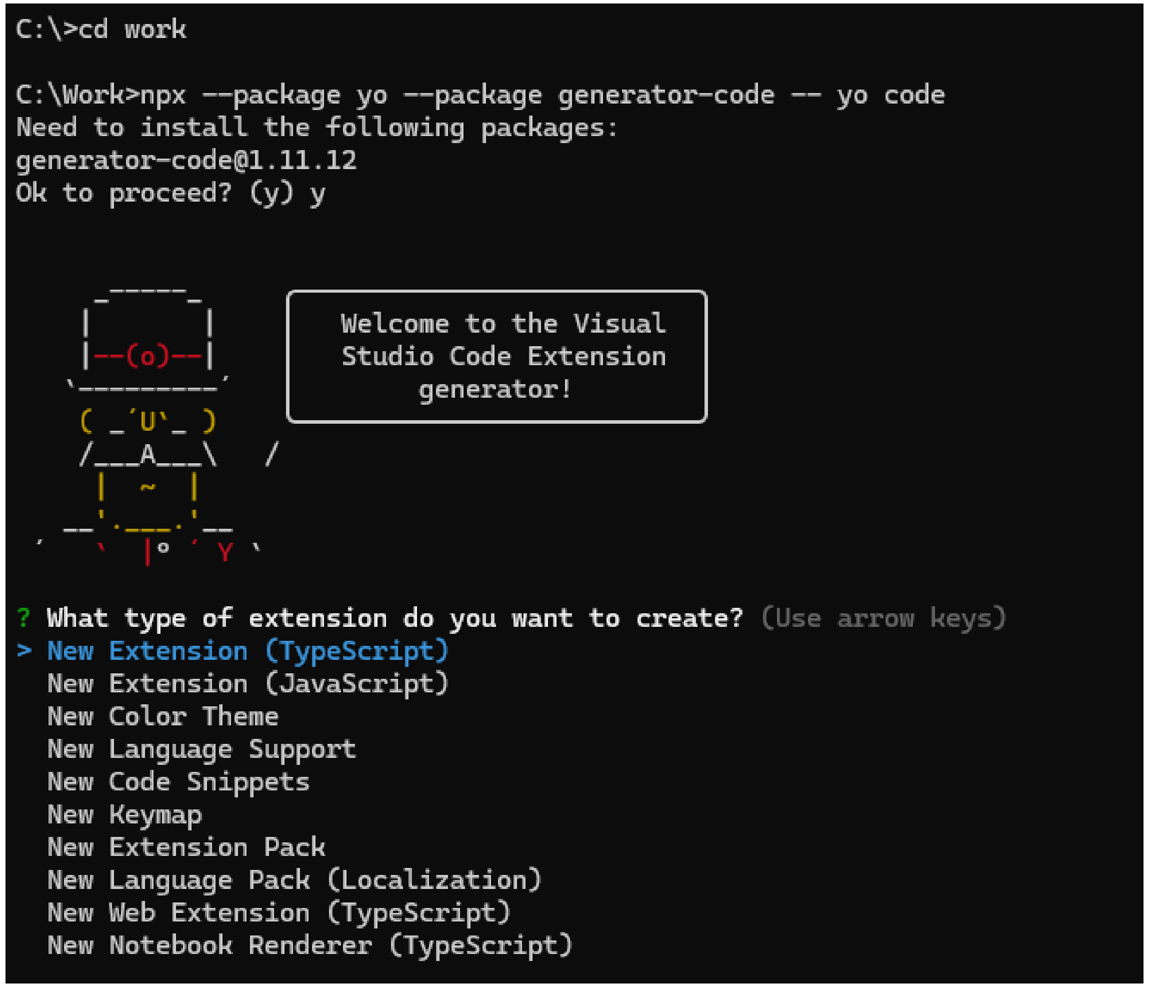
Bash
```
npm install --global yo generator-code

yo code
```

Consultingwerk
software architecture and development

Building an Extension with Visual Studio Code “Your first Extension”

The VS Code Extension generator appears

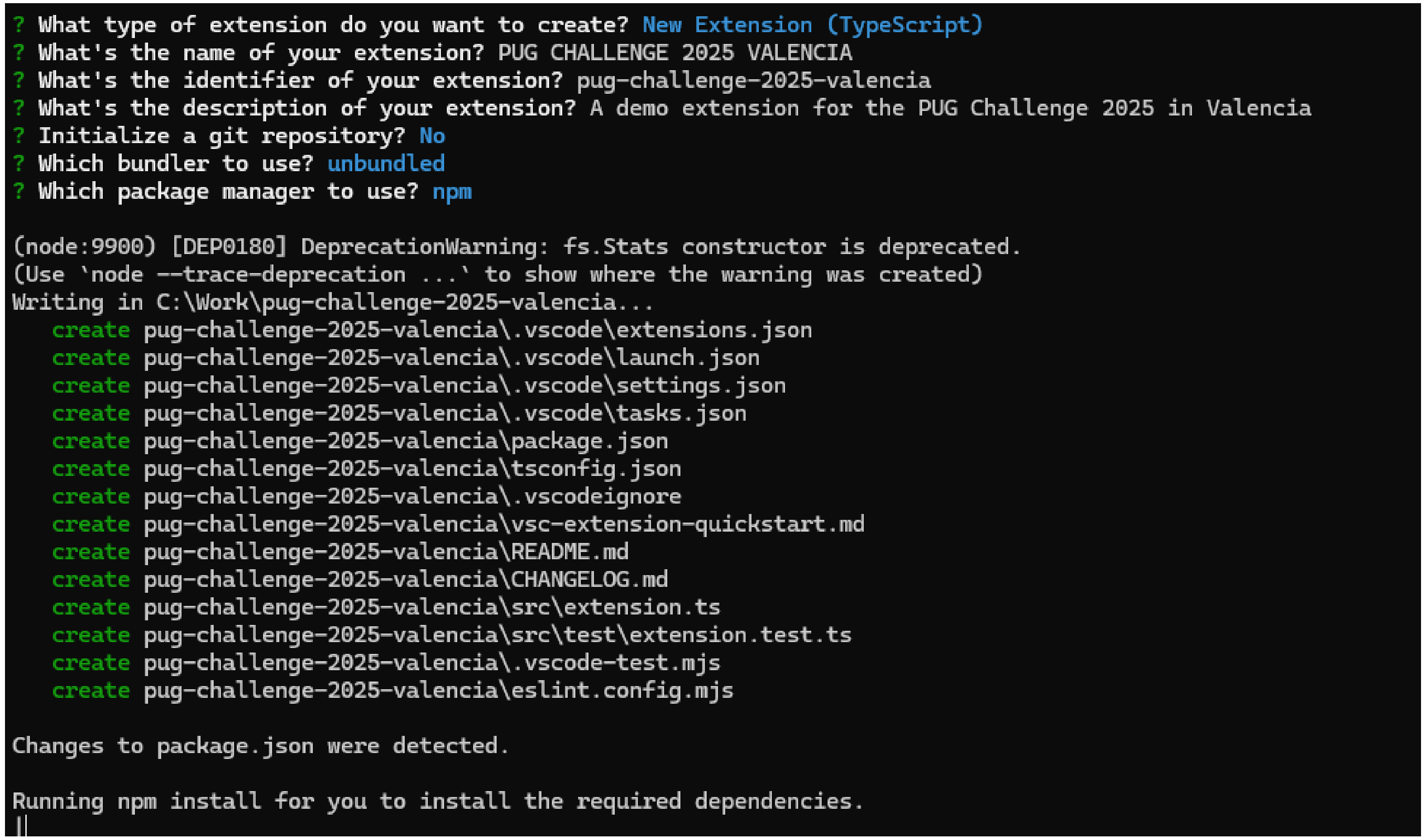
Select “New Extension” →



```
C:\>cd work

C:\Work>npx --package yo --package generator-code -- yo code
Need to install the following packages:
generator-code@1.11.12
Ok to proceed? (y) y


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

? What type of extension do you want to create? (Use arrow keys)
> New Extension (TypeScript)
  New Extension (JavaScript)
  New Color Theme
  New Language Support
  New Code Snippets
  New Keymap
  New Extension Pack
  New Language Pack (Localization)
  New Web Extension (TypeScript)
  New Notebook Renderer (TypeScript)
```

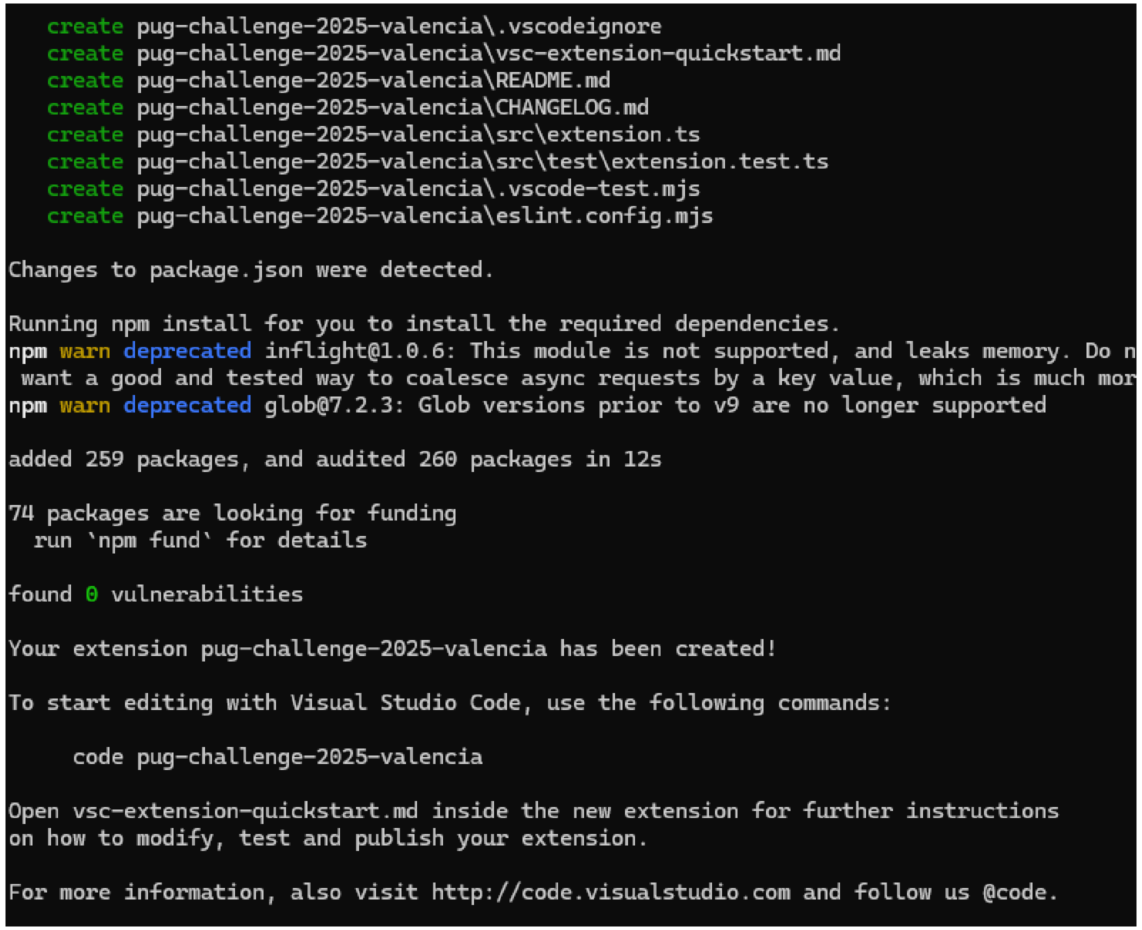
Building an Extension with Visual Studio Code "Your first Extension"

- Type a name
- The identifier is already proposed
- Description
- Initialize a git repo
- Bundler
- Package manager



```
? What type of extension do you want to create? New Extension (TypeScript)
? What's the name of your extension? PUG CHALLENGE 2025 VALENCIA
? What's the identifier of your extension? pug-challenge-2025-valencia
? What's the description of your extension? A demo extension for the PUG Challenge 2025 in Valencia
? Initialize a git repository? No
? Which bundler to use? unbundled
? Which package manager to use? npm

(node:9900) [DEP0180] DeprecationWarning: fs.Stats constructor is deprecated.
(Use `node --trace-deprecation ...` to show where the warning was created)
Writing in C:\Work\pug-challenge-2025-valencia...
   create pug-challenge-2025-valencia\.vscode\extensions.json
   create pug-challenge-2025-valencia\.vscode\launch.json
   create pug-challenge-2025-valencia\.vscode\settings.json
   create pug-challenge-2025-valencia\.vscode\tasks.json
   create pug-challenge-2025-valencia\package.json
   create pug-challenge-2025-valencia\tsconfig.json
   create pug-challenge-2025-valencia\.vscodeignore
   create pug-challenge-2025-valencia\vsc-extension-quickstart.md
   create pug-challenge-2025-valencia\README.md
   create pug-challenge-2025-valencia\CHANGELOG.md
   create pug-challenge-2025-valencia\src\extension.ts
   create pug-challenge-2025-valencia\src\test\extension.test.ts
   create pug-challenge-2025-valencia\.vscode-test.mjs
   create pug-challenge-2025-valencia\eslint.config.mjs

Changes to package.json were detected.

Running npm install for you to install the required dependencies.
```

Building an Extension with Visual Studio Code “Your first Extension”

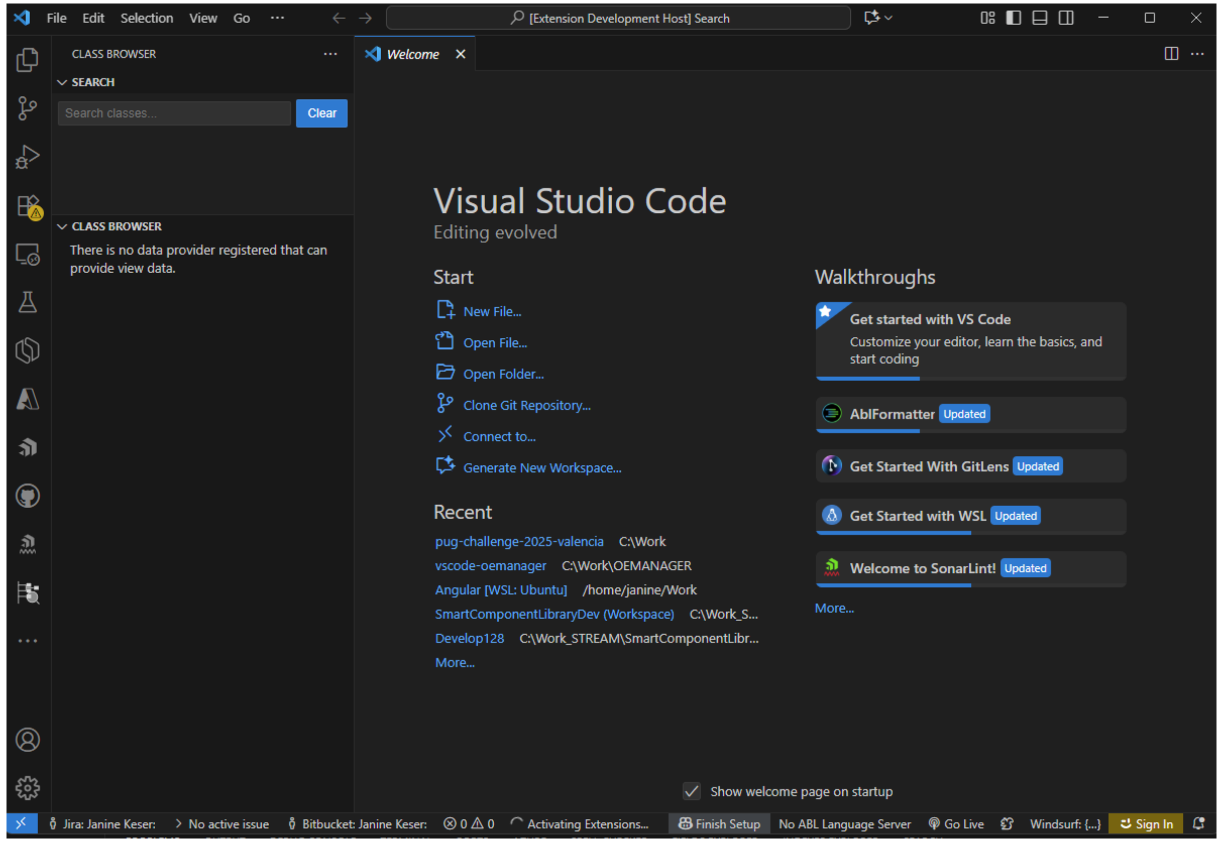


```
   create pug-challenge-2025-valencia\.vscodeignore
   create pug-challenge-2025-valencia\vsc-extension-quickstart.md
   create pug-challenge-2025-valencia\README.md
   create pug-challenge-2025-valencia\CHANGELOG.md
   create pug-challenge-2025-valencia\src\extension.ts
   create pug-challenge-2025-valencia\src\test\extension.test.ts
   create pug-challenge-2025-valencia\.vscode-test.mjs
   create pug-challenge-2025-valencia\eslint.config.mjs

Changes to package.json were detected.

Running npm install for you to install the required dependencies.
npm warn deprecated inflight@1.0.6: This module is not supported, and leaks memory. Do n
 want a good and tested way to coalesce async requests by a key value, which is much mor
npm warn deprecated glob@7.2.3: Glob versions prior to v9 are no longer supported

added 259 packages, and audited 260 packages in 12s

74 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities

Your extension pug-challenge-2025-valencia has been created!

To start editing with Visual Studio Code, use the following commands:

     code pug-challenge-2025-valencia

Open vsc-extension-quickstart.md inside the new extension for further instructions
on how to modify, test and publish your extension.

For more information, also visit http://code.visualstudio.com and follow us @code.
```

- After Installation finished
- Select “Open with `code`
- Press enter

Building an Extension with Visual Studio Code “Your first Extension”

- Open **src/extension.ts** in the editor
- Press **F5** or run **Debug: Start Debugging** from the Command Palette (**Ctrl + Shift + P**)
- Extension is compiled and runs in a new **Extension Development Host** window

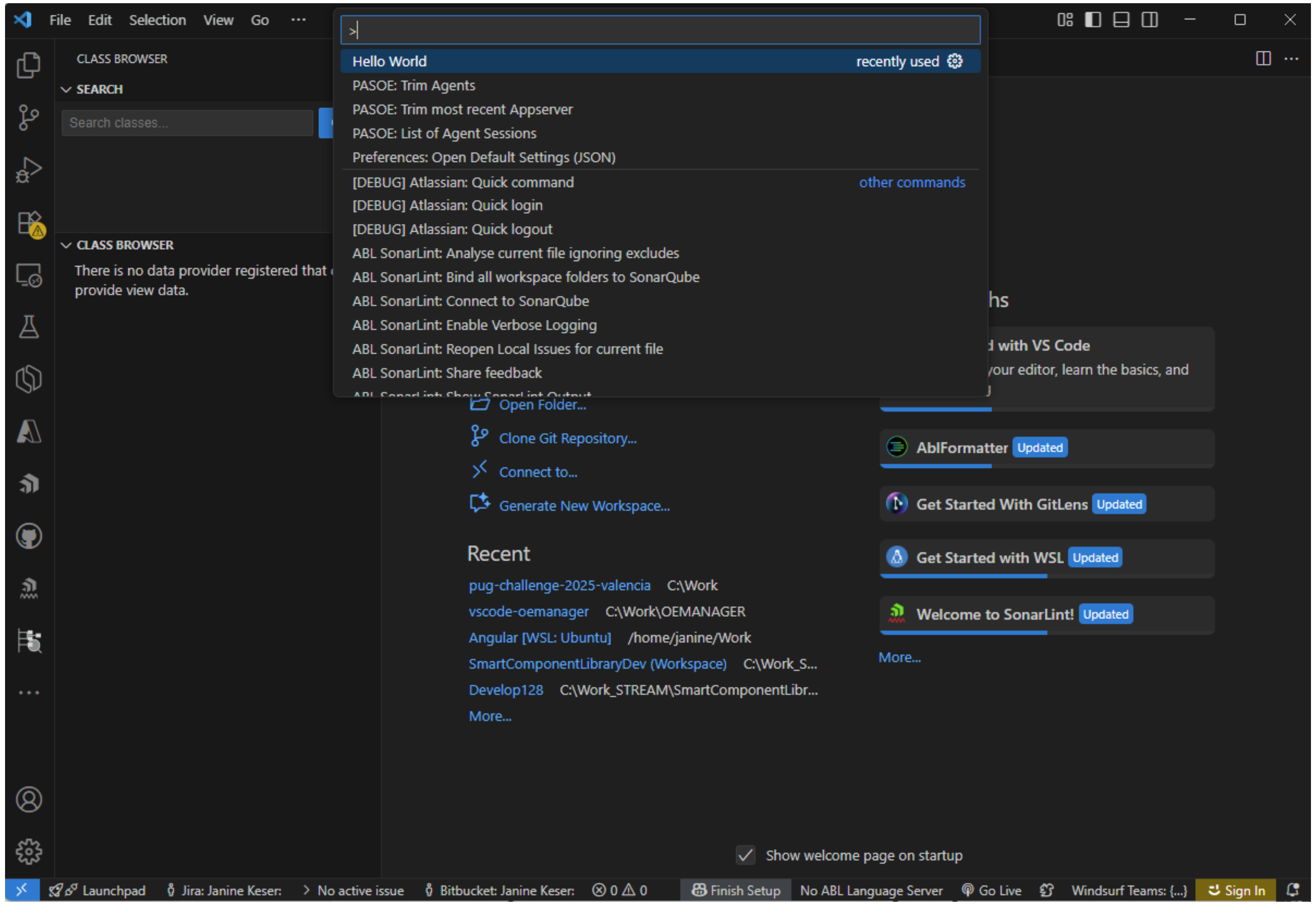


There is no data provider registered that can provide view data.

Customize your editor, learn the basics, and start coding

Building an Extension with Visual Studio Code “Your first Extension”

- Run the Hello World command from the Command Palette (Ctrl+Shift+P) in the new window.
- This is the register command for your extension.


File Edit Selection View Go ···

>|

Hello World recently used
PASOE: Trim Agents
PASOE: Trim most recent Appserver
PASOE: List of Agent Sessions
Preferences: Open Default Settings (JSON)
[DEBUG] Atlassian: Quick command other commands
[DEBUG] Atlassian: Quick login
[DEBUG] Atlassian: Quick logout
ABL SonarLint: Analyse current file ignoring excludes
ABL SonarLint: Bind all workspace folders to SonarQube
ABL SonarLint: Connect to SonarQube
ABL SonarLint: Enable Verbose Logging
ABL SonarLint: Reopen Local Issues for current file
ABL SonarLint: Share feedback

CLASS BROWSER
SEARCH
Search classes...
CLASS BROWSER
There is no data provider registered that provide view data.

Open Folder...
Clone Git Repository...
Connect to...
Generate New Workspace...

Recent
pug-challenge-2025-valencia C:\Work
vscode-oemanager C:\Work\OEMANAGER
Angular [WSL: Ubuntu] /home/janine/Work
SmartComponentLibraryDev (Workspace) C:\Work_S...
Develop128 C:\Work_STREAM\SmartComponentLibr...
More...

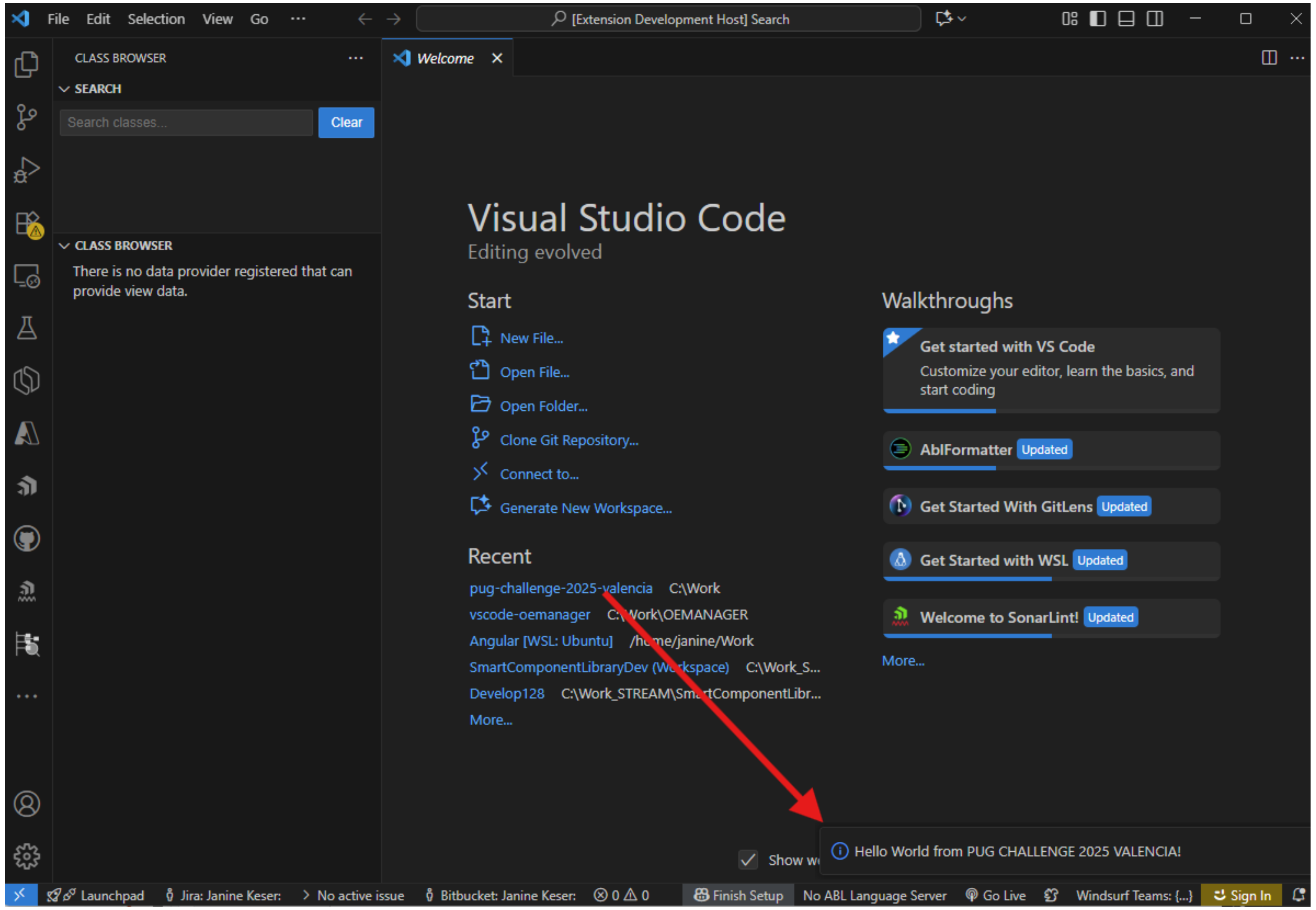
with VS Code
your editor, learn the basics, and
AblFormatter Updated
Get Started With GitLens Updated
Get Started with WSL Updated
Welcome to SonarLint! Updated
More...

Show welcome page on startup

Launchpad Jira: Janine Keser: No active issue Bitbucket: Janine Keser: 0 0 Finish Setup No ABL Language Server Go Live Windsurf Teams: {...} Sign In

Building an Extension with Visual Studio Code “Your first Extension”

- The registered command currently displays an information message in the bottom-right corner
- Uses: vscode.window.showInformationMessage('Hello World from PUG CHALLENGE 2025 VALENCIA!‘);
- A simple and quick way to show a message in VS Code



File Edit Selection View Go
[Extension Development Host] Search
CLASS BROWSER
SEARCH
Search classes...
Clear
CLASS BROWSER
Welcome
Visual Studio Code
Editing evolved
Start
New File...
Open File...
Open Folder...
Clone Git Repository...
Connect to...
Generate New Workspace...
Recent
pug-challenge-2025-valencia C:\Work
vscode-oemanager C:\Work\OEMANAGER
Angular [WSL: Ubuntu] /home/janine/Work
SmartComponentLibraryDev (Workspace) C:\Work_S...
Develop128 C:\Work_STREAM\SmartComponentLibr...
More...
Walkthroughs
Get started with VS Code
Customize your editor, learn the basics, and start coding
AblFormatter Updated
Get Started With GitLens Updated
Get Started with WSL Updated
Welcome to SonarLint! Updated
More...
Show w
Launchpad
Jira: Janine Keser:
No active issue
Bitbucket: Janine Keser:
0 0
Finish Setup
No ABL Language Server
Go Live
Windsurf Teams: {...}
Sign In

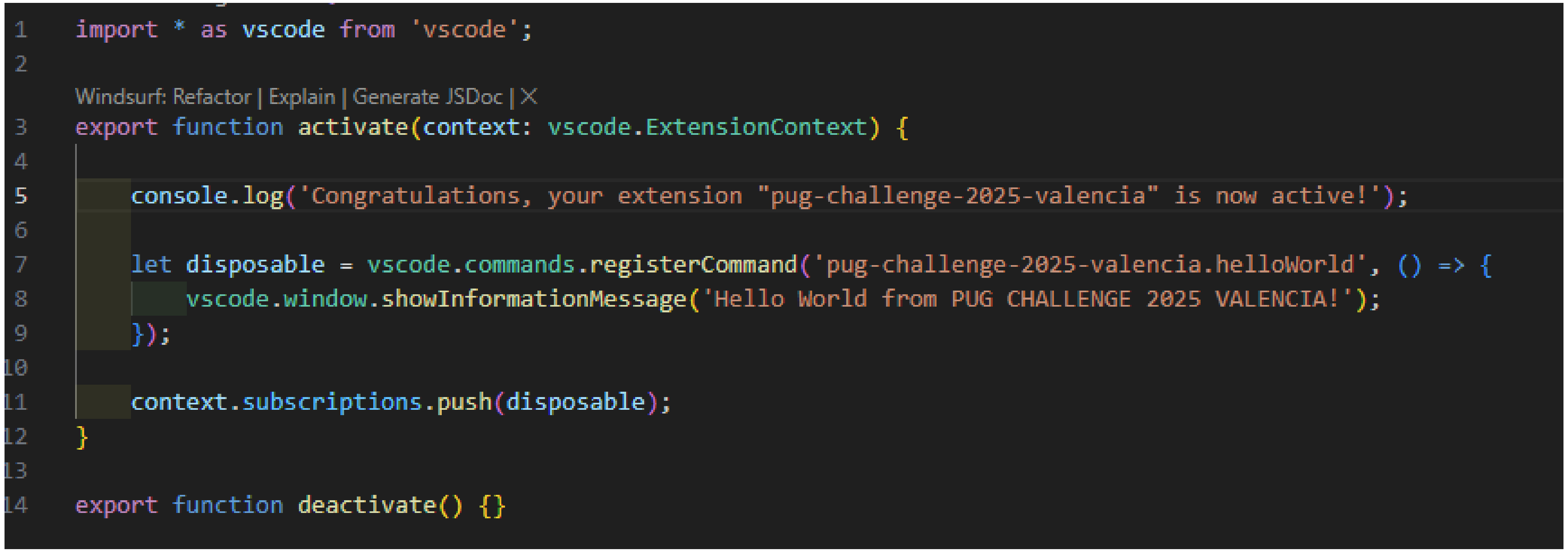
There is no data provider registered that can provide view data.

Hello World from PUG CHALLENGE 2025 VALENCIA!

Building an Extension with Visual Studio Code “Your first Extension”

Everything that has happened so far is shown in the extension.ts file.

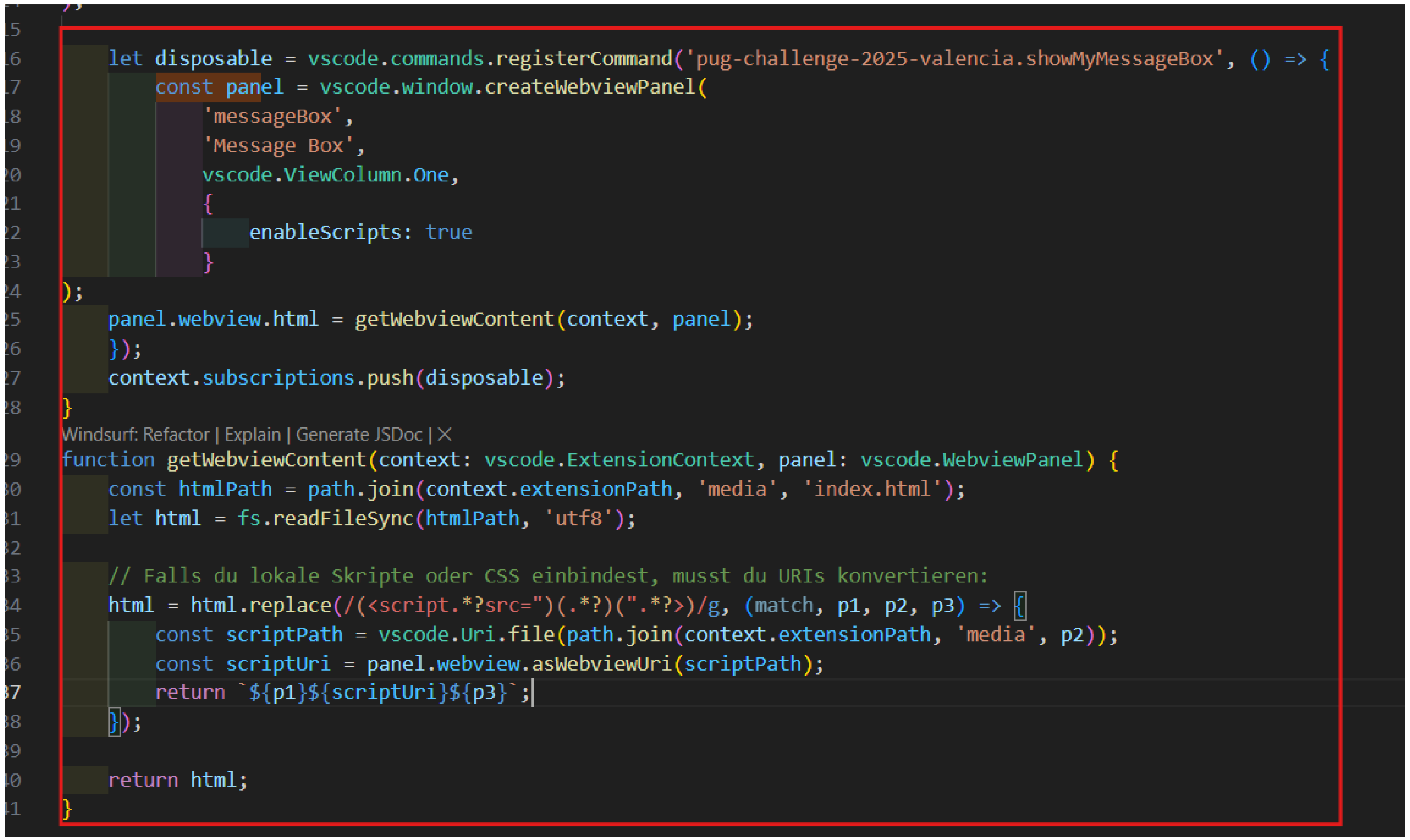
The code reflects the scaffold and the command we have registered.



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

Windsurf: Refactor | Explain | Generate JSDoc | ✕
export function activate(context: vscode.ExtensionContext) {

    console.log('Congratulations, your extension "pug-challenge-2025-valencia" is now active!');

    let disposable = vscode.commands.registerCommand('pug-challenge-2025-valencia.helloWorld', () => {
        vscode.window.showInformationMessage('Hello World from PUG CHALLENGE 2025 VALENCIA!');
    });

    context.subscriptions.push(disposable);
}

export function deactivate() {}
```

Building an Extension with Visual Studio Code “Your first Extension”

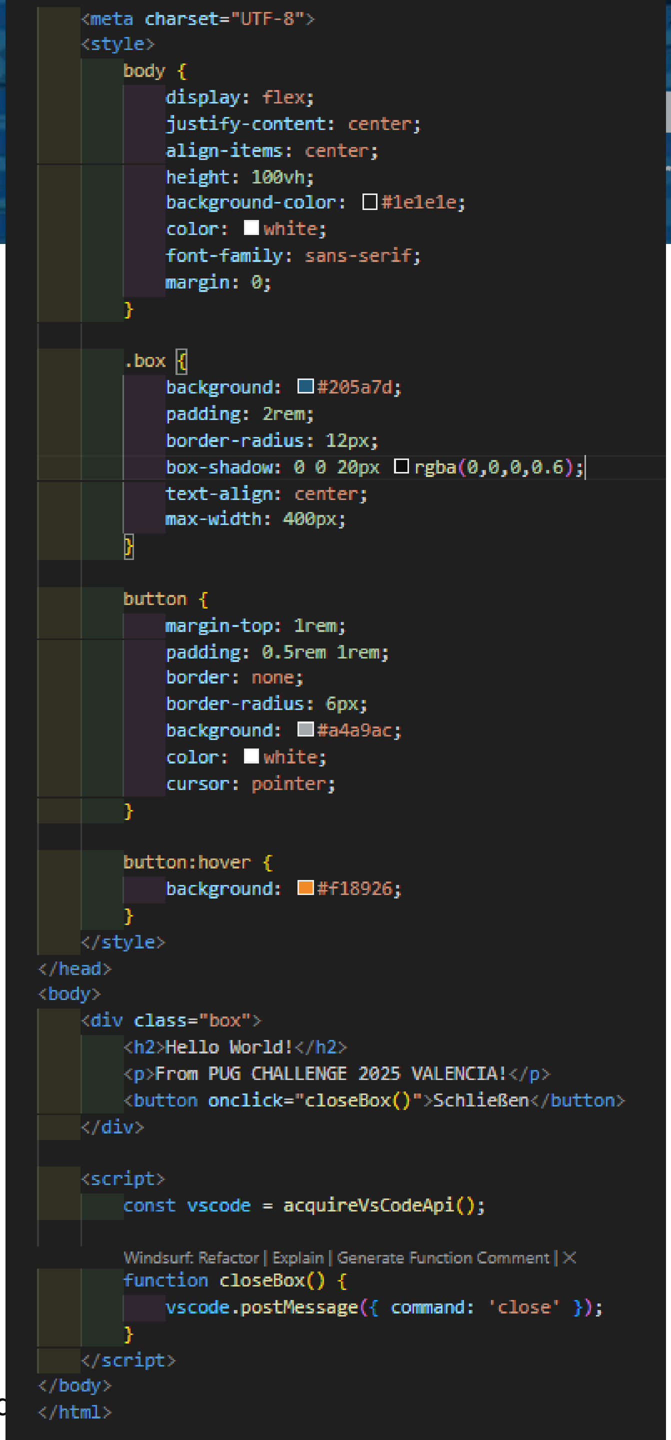
- Default information message has **limited customization** (display, position, size)
- If you want **more customization**, you can build your own views using small HTML files
- Simply **add another command** to your code to display the custom view



```
    let disposable = vscode.commands.registerCommand('pug-challenge-2025-valencia.showMyMessageBox', () => {
        const panel = vscode.window.createWebviewPanel(
            'messageBox',
            'Message Box',
            vscode.ViewColumn.One,
            {
                enableScripts: true
            }
);
    panel.webview.html = getWebviewContent(context, panel);
    });
    context.subscriptions.push(disposable);
}
Windsurf: Refactor | Explain | Generate JSDoc | ✕
function getWebviewContent(context: vscode.ExtensionContext, panel: vscode.WebviewPanel) {
    const htmlPath = path.join(context.extensionPath, 'media', 'index.html');
    let html = fs.readFileSync(htmlPath, 'utf8');

    // Falls du lokale Skripte oder CSS einbindest, musst du URIs konvertieren:
    html = html.replace(/(<script.*?src=")(.*?)(".*?>)/g, (match, p1, p2, p3) => {
        const scriptPath = vscode.Uri.file(path.join(context.extensionPath, 'media', p2));
        const scriptUri = panel.webview.asWebviewUri(scriptPath);
        return `${p1}${scriptUri}${p3}`;
    });

    return html;
}
```

Building an Extension with Visual Studio Code “Your first Extension”

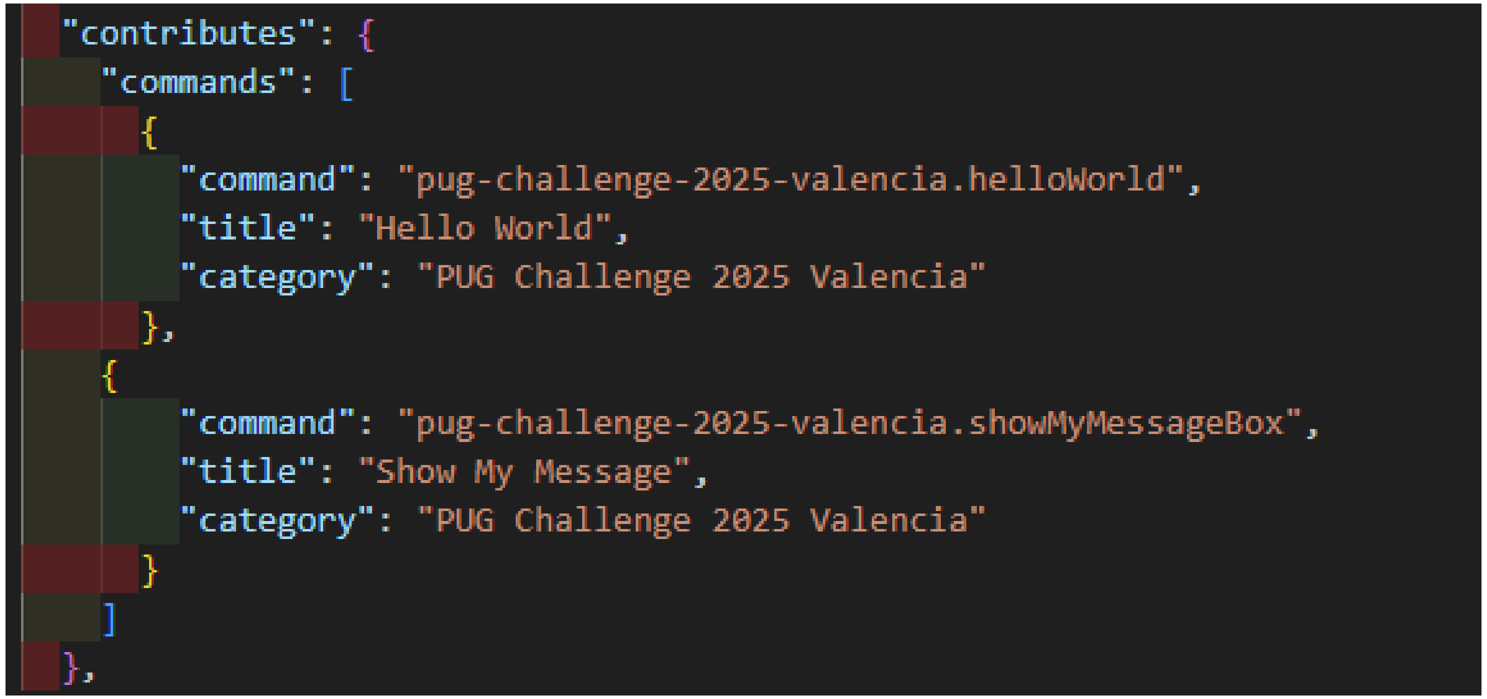
- **Create and implement an HTML file** for your custom view
- In my example it is called index.html



```
    <meta charset="UTF-8">
    <style>
        body {
            display: flex;
            justify-content: center;
            align-items: center;
            height: 100vh;
            background-color: #1e1e1e;
            color: white;
            font-family: sans-serif;
            margin: 0;
        }

        .box {
            background: #205a7d;
            padding: 2rem;
            border-radius: 12px;
            box-shadow: 0 0 20px rgba(0,0,0,0.6);
            text-align: center;
            max-width: 400px;
        }

        button {
            margin-top: 1rem;
            padding: 0.5rem 1rem;
            border: none;
            border-radius: 6px;
            background: #a4a9ac;
            color: white;
            cursor: pointer;
        }

        button:hover {
            background: #f18926;
        }
    </style>
</head>
<body>
    <div class="box">
        <h2>Hello World!</h2>
        <p>From PUG CHALLENGE 2025 VALENCIA!</p>
        <button onclick="closeBox()">Schließen</button>
    </div>

    <script>
        const vscode = acquireVsCodeApi();

        Windsurf: Refactor | Explain | Generate Function Comment | ×
        function closeBox() {
            vscode.postMessage({ command: 'close' });
        }
    </script>
</body>
</html>
```

Building an Extension with Visual Studio Code “Your first Extension”

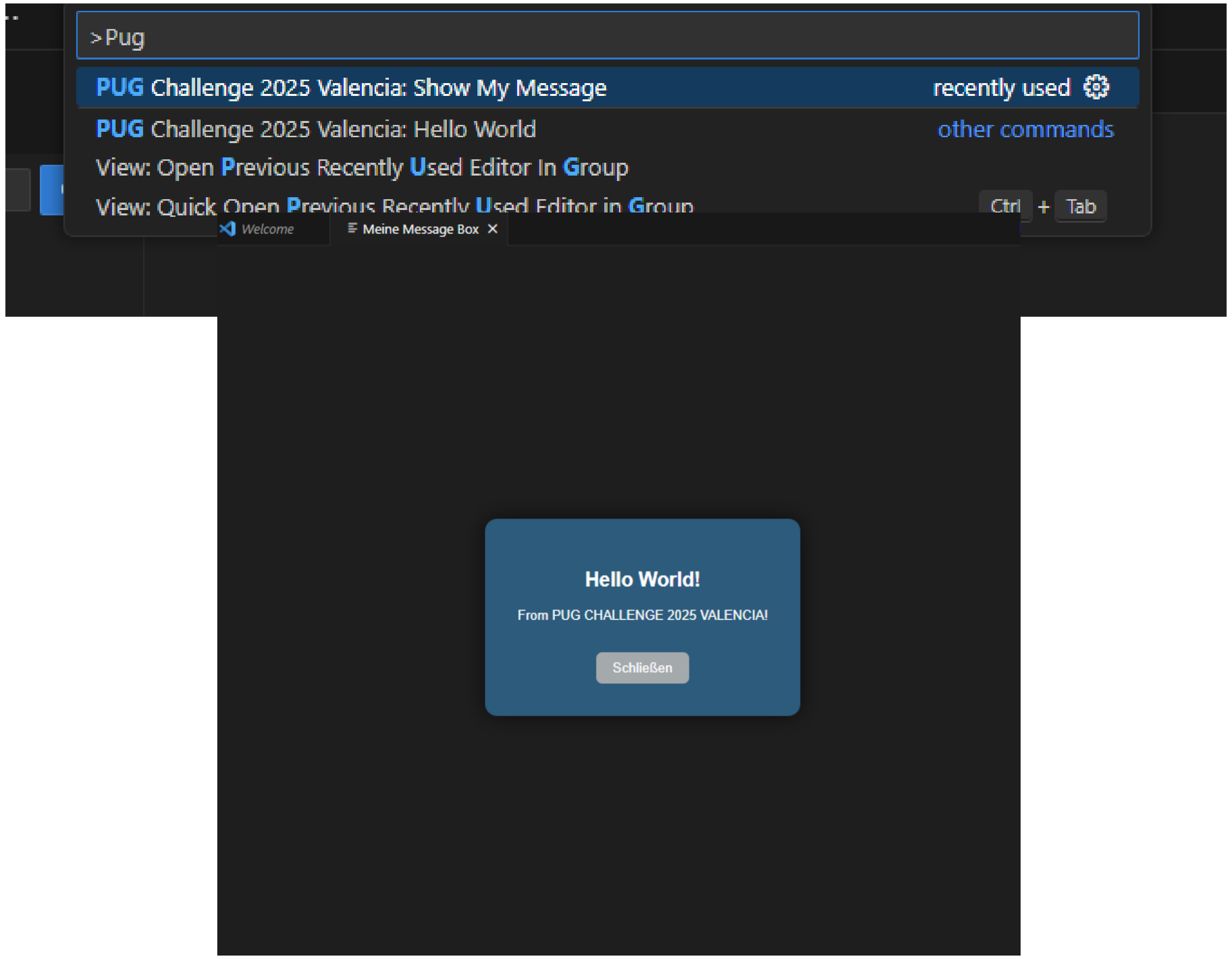
You also need to register the command in the package.json commands area.



```
"contributes": {
    "commands": [
        {
            "command": "pug-challenge-2025-valencia.helloWorld",
            "title": "Hello World",
            "category": "PUG Challenge 2025 Valencia"
        },
        {
            "command": "pug-challenge-2025-valencia.showMyMessageBox",
            "title": "Show My Message",
            "category": "PUG Challenge 2025 Valencia"
        }
    ]
},
```

Building an Extension with Visual Studio Code "Your first Extension"

- Press **F5** to start the extension
- The new command appears in the **Command Palette**
- Select **"Show My Message"** to display your custom message box



>Pug

PUG Challenge 2025 Valencia: Show My Message — recently used

PUG Challenge 2025 Valencia: Hello World — other commands

View: Open Previous Recently Used Editor In Group

View: Quick Open Previous Recently Used Editor in Group — Ctrl + Tab

Hello World!

From PUG CHALLENGE 2025 VALENCIA!

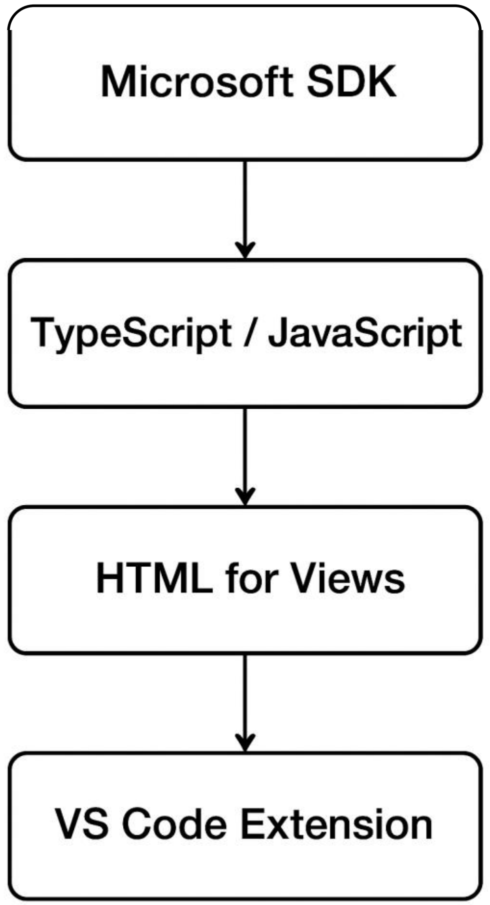
Agenda

- Sample PASOE Manager Extension
- VS Code Marketplace and alternatives
- VS Code as an IDE
- Microsoft SDK for VS Code Extensions
- Building an Extension with Visual Studio Code
- Programming languages and file types used in a VS Code extension
- Event Callbacks
- Implementing the PASOE Manager Extension
- Package and publish your extension
- Important Common Capabilities
- Questions

Programming languages and file types used in a VS Code extension

Extensions are built through interaction between different technologies:

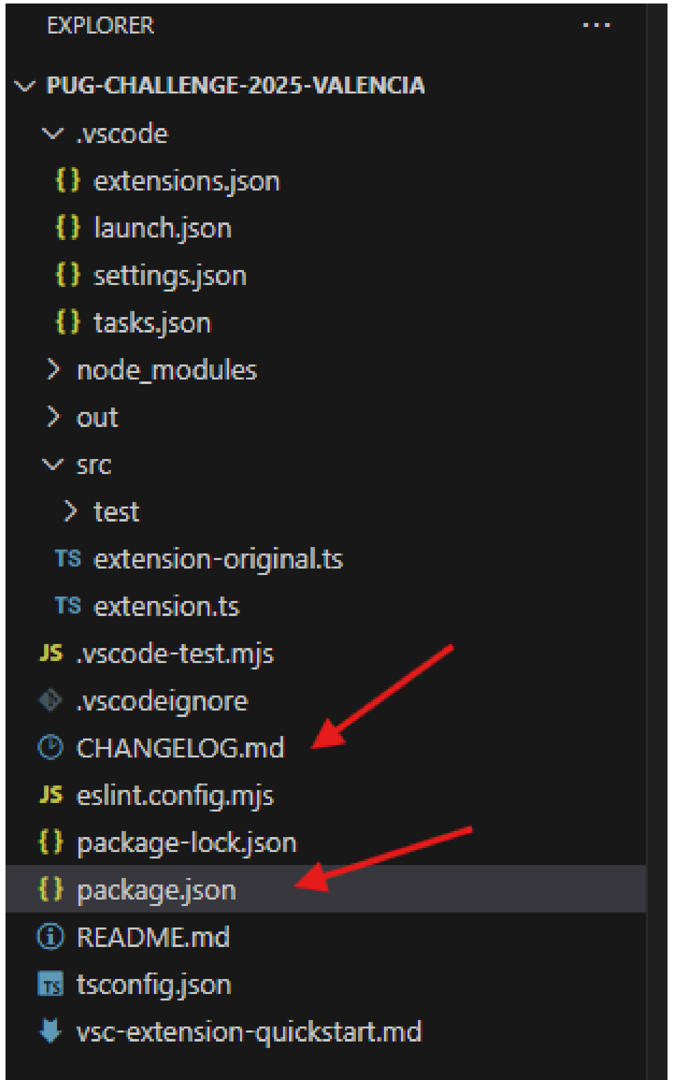
- Combine multiple file types and languages
- TypeScript / JavaScript – implement the core functionality and logic
- HTML – used for webviews or custom user interfaces within VS Code
- Together, they enable interactive and powerful extension features



Programming languages and file types used in a VS Code extension

Let’s take a look at the basic file structure of a VS Code extension:

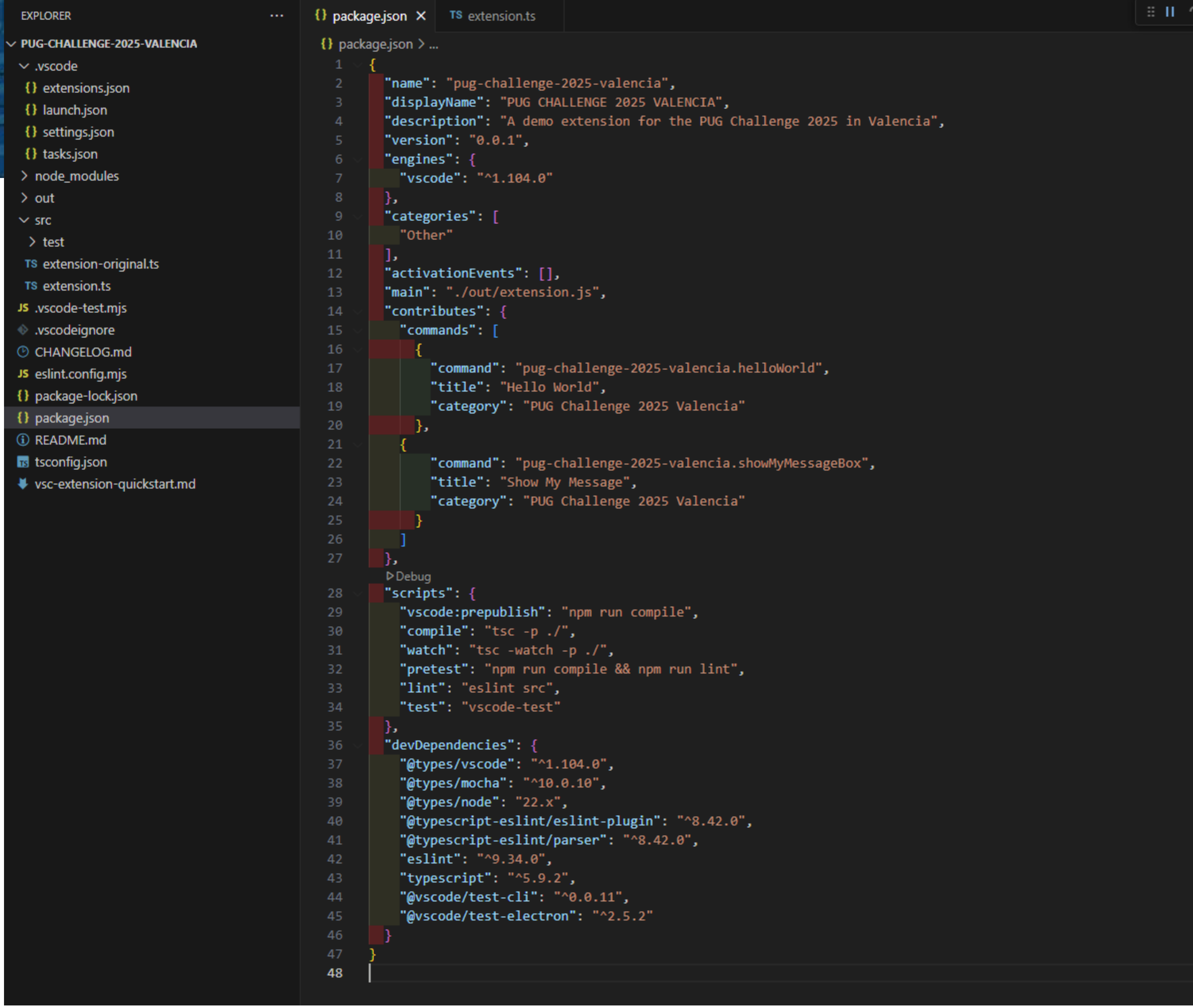
- Most files are TypeScript (.ts) – they are automatically compiled into JavaScript (.js) files in the “out” folder
- .json files – define structure and configuration (e.g., package.json with metadata and activation rules)
- .md files (Markdown) – used for documentation, such as README or help texts
- The interaction of these different technologies creates a functional and user-friendly extension



```
EXPLORER
∨ PUG-CHALLENGE-2025-VALENCIA
  ∨ .vscode
      extensions.json
      launch.json
      settings.json
      tasks.json
  > node_modules
  > out
  ∨ src
    > test
      extension-original.ts
      extension.ts
    .vscode-test.mjs
    .vscodeignore
    CHANGELOG.md
    eslint.config.mjs
    package-lock.json
    package.json
    README.md
    tsconfig.json
    vsc-extension-quickstart.md
```

The package.json

- Acts as the manifest of the extension
- Describes the project (metadata, name, version, etc.)
- Defines behavior (activation events, commands, contributions)



```
PUG-CHALLENGE-2025-VALENCIA
  .vscode
    extensions.json
    launch.json
    settings.json
    tasks.json
  node_modules
  out
  src
    test
    extension-original.ts
    extension.ts
  .vscode-test.mjs
  .vscodeignore
  CHANGELOG.md
  eslint.config.mjs
  package-lock.json
  package.json
  README.md
  tsconfig.json
  vsc-extension-quickstart.md
```

```
{
  "name": "pug-challenge-2025-valencia",
  "displayName": "PUG CHALLENGE 2025 VALENCIA",
  "description": "A demo extension for the PUG Challenge 2025 in Valencia",
  "version": "0.0.1",
  "engines": {
    "vscode": "^1.104.0"
  },
  "categories": [
    "Other"
  ],
  "activationEvents": [],
  "main": "./out/extension.js",
  "contributes": {
    "commands": [
      {
        "command": "pug-challenge-2025-valencia.helloWorld",
        "title": "Hello World",
        "category": "PUG Challenge 2025 Valencia"
      },
      {
        "command": "pug-challenge-2025-valencia.showMyMessageBox",
        "title": "Show My Message",
        "category": "PUG Challenge 2025 Valencia"
      }
    ]
  },
  "scripts": {
    "vscode:prepublish": "npm run compile",
    "compile": "tsc -p ./",
    "watch": "tsc -watch -p ./",
    "pretest": "npm run compile && npm run lint",
    "lint": "eslint src",
    "test": "vscode-test"
  },
  "devDependencies": {
    "@types/vscode": "^1.104.0",
    "@types/mocha": "^10.0.10",
    "@types/node": "22.x",
    "@typescript-eslint/eslint-plugin": "^8.42.0",
    "@typescript-eslint/parser": "^8.42.0",
    "eslint": "^9.34.0",
    "typescript": "^5.9.2",
    "@vscode/test-cli": "^0.0.11",
    "@vscode/test-electron": "^2.5.2"
  }
}
```

Agenda

- Sample PASOE Manager Extension
- VS Code Marketplace and alternatives
- VS Code as an IDE
- Microsoft SDK for VS Code Extensions
- Building an Extension with Visual Studio Code
- Programming languages and file types used in a VS Code extension
- Event Callbacks
- Implementing the PASOE Manager Extension
- Package and publish your extension
- Important Common Capabilities
- Questions

Consultingwerk
software architecture and development

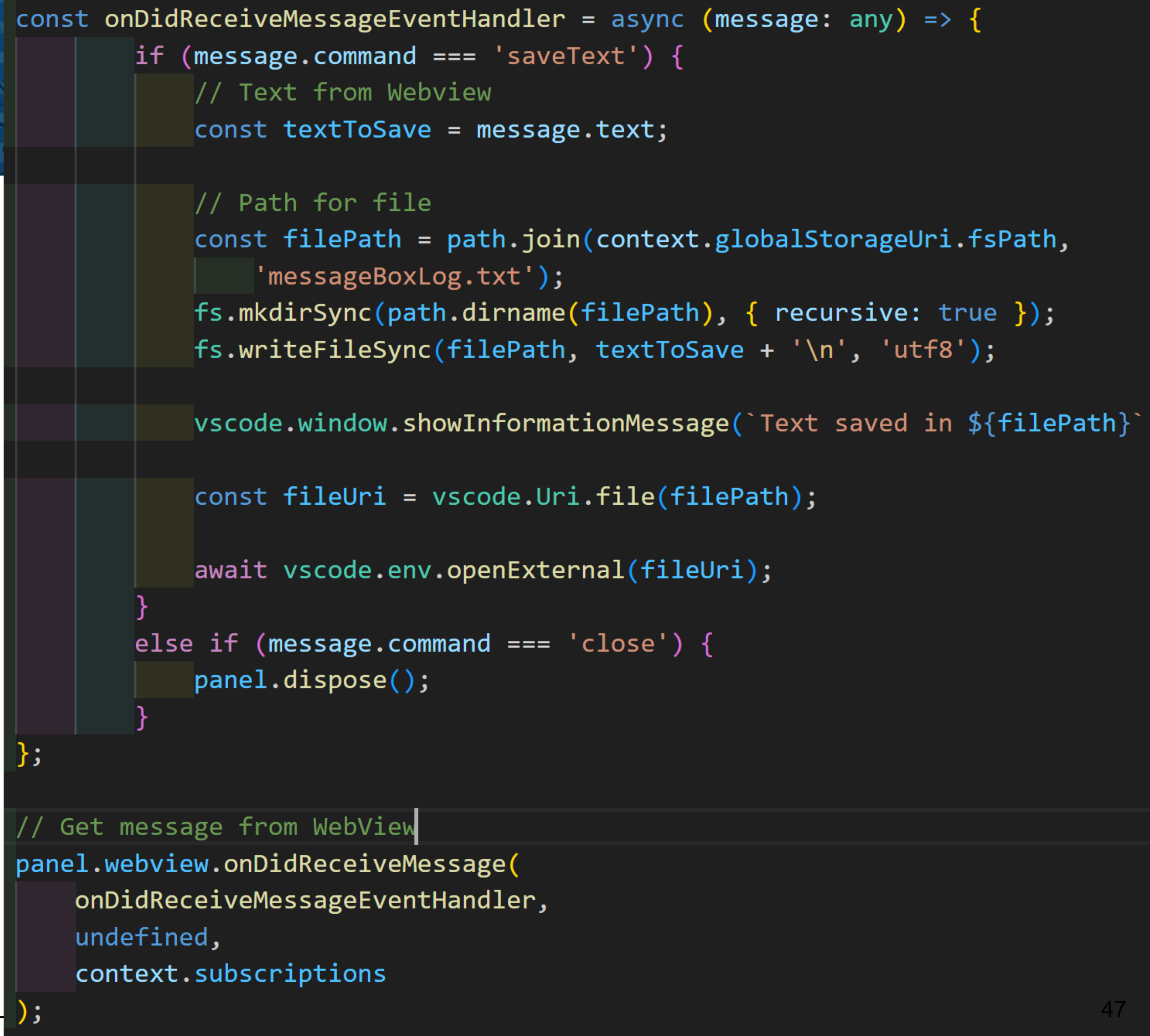
Event Callbacks

- Event callbacks are a central component of VS Code extensions
- Allow extensions to respond to specific events in the development environment
  - Opening or saving a file
  - Changing the active editor
  - Executing a command
- An event callback is a function automatically called when the event occurs
- Enables extensions to be interactive and dynamic
  - Run analysis when a file is saved
  - Display content when a new file is opened
- Makes VS Code flexibly extensible, integrating seamlessly into the developer’s workflow without manual intervention

Event Callbacks

A small example from our demo application:

- Customized message box created using a VS Code Webview
- Added a button: “Save in file”
- Clicking the button sends a message back to the extension code
- It acts as event handler
- Writes the text from the message box into a file

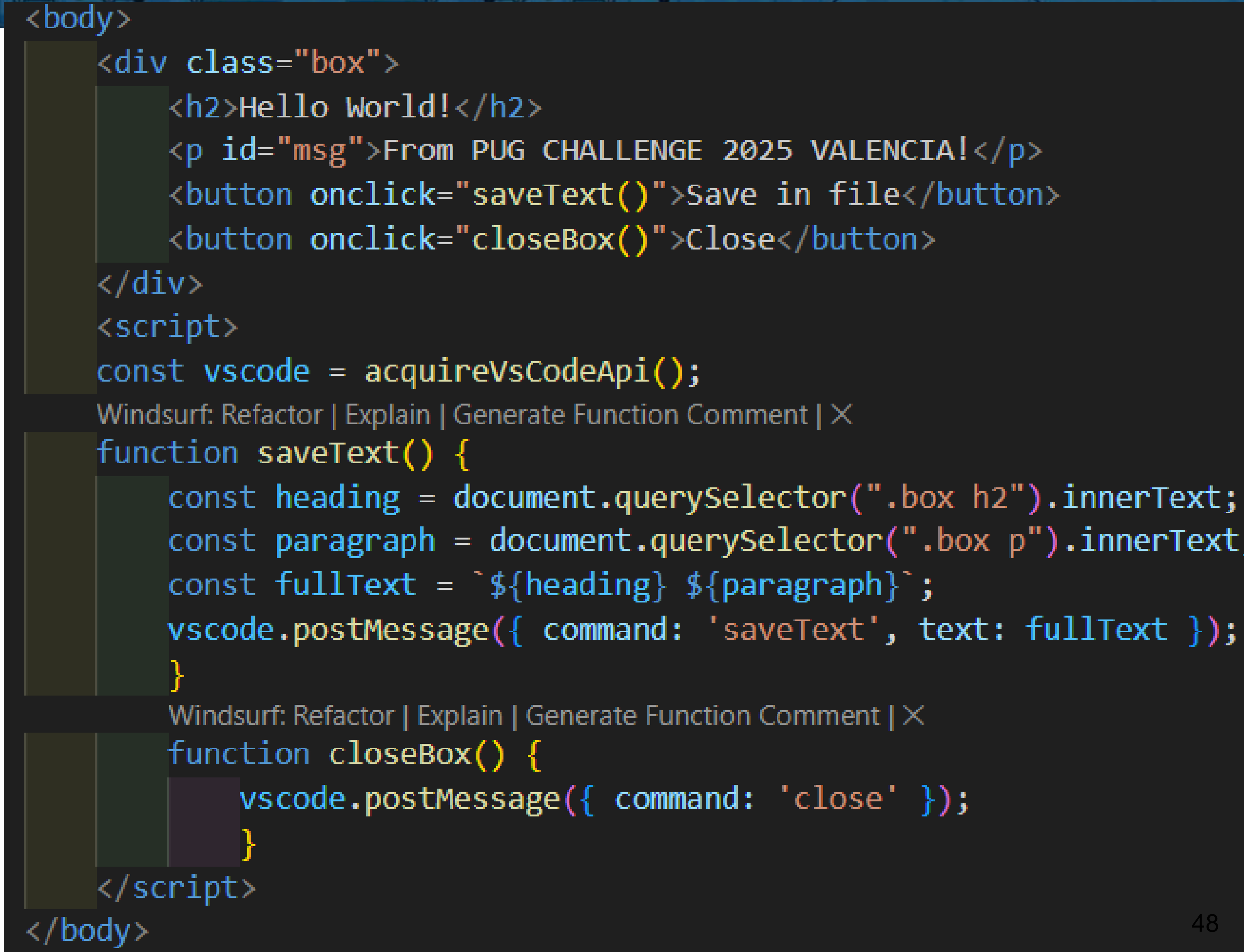


```
const onDidReceiveMessageEventHandler = async (message: any) => {
        if (message.command === 'saveText') {
            // Text from Webview
            const textToSave = message.text;

            // Path for file
            const filePath = path.join(context.globalStorageUri.fsPath,
                'messageBoxLog.txt');
            fs.mkdirSync(path.dirname(filePath), { recursive: true });
            fs.writeFileSync(filePath, textToSave + '\n', 'utf8');

            vscode.window.showInformationMessage(`Text saved in ${filePath}`

            const fileUri = vscode.Uri.file(filePath);

            await vscode.env.openExternal(fileUri);
        }
        else if (message.command === 'close') {
            panel.dispose();
        }
};

// Get message from WebView
panel.webview.onDidReceiveMessage(
    onDidReceiveMessageEventHandler,
    undefined,
    context.subscriptions
);
```

Event Callbacks

In the HTML file:

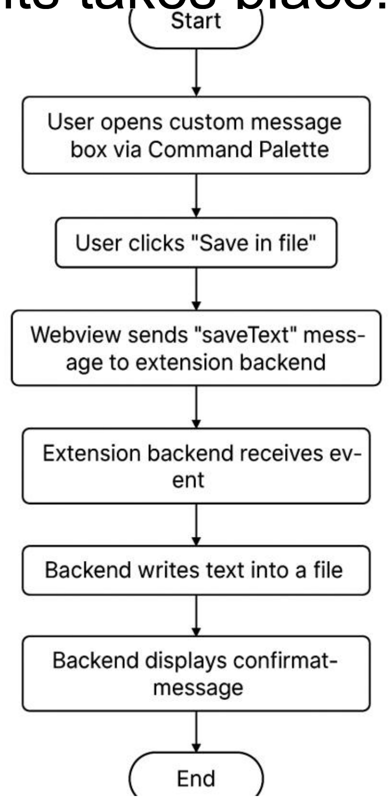
- Extend the custom message box with button
- Button calls a small function: saveText()
- Function collects text from <h2> and <p>
- Sends the text back via vscode.postMessage()
- HTML part actively triggers events in the extension

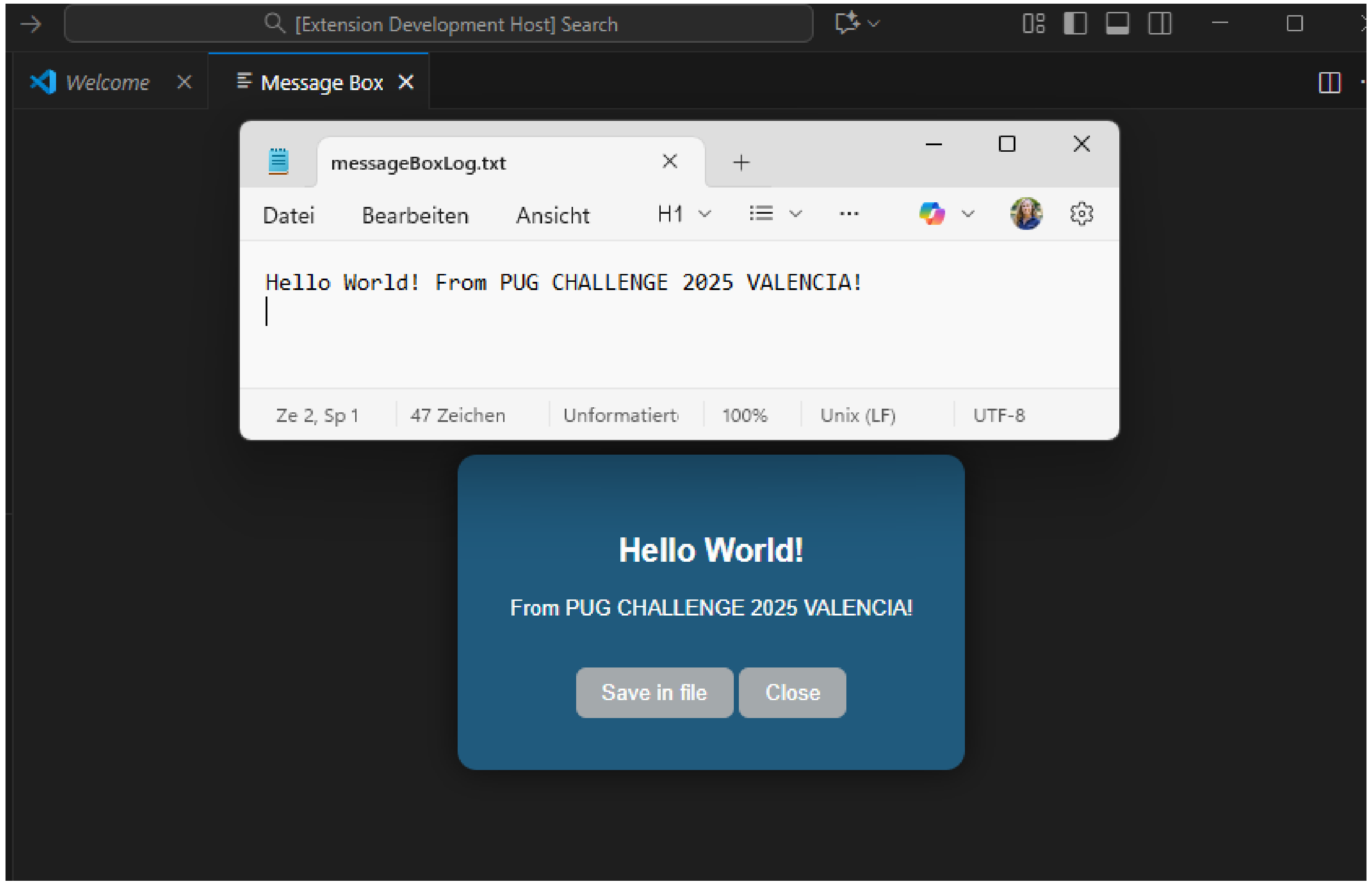


```
<body>
    <div class="box">
        <h2>Hello World!</h2>
        <p id="msg">From PUG CHALLENGE 2025 VALENCIA!</p>
        <button onclick="saveText()">Save in file</button>
        <button onclick="closeBox()">Close</button>
    </div>
    <script>
    const vscode = acquireVsCodeApi();
    Windsurf: Refactor | Explain | Generate Function Comment | ×
    function saveText() {
        const heading = document.querySelector(".box h2").innerText;
        const paragraph = document.querySelector(".box p").innerText
        const fullText = `${heading} ${paragraph}`;
        vscode.postMessage({ command: 'saveText', text: fullText });
        }
        Windsurf: Refactor | Explain | Generate Function Comment | ×
        function closeBox() {
            vscode.postMessage({ command: 'close' });
            }
    </script>
</body>
```

Event Callbacks

When we run the extension, the following sequence of events takes place:


Start
User opens custom message box via Command Palette
User clicks "Save in file"
Webview sends "saveText" mess-age to extension backend
Extension backend receives ev-ent
Backend writes text into a file
Backend displays confirmat-message
End

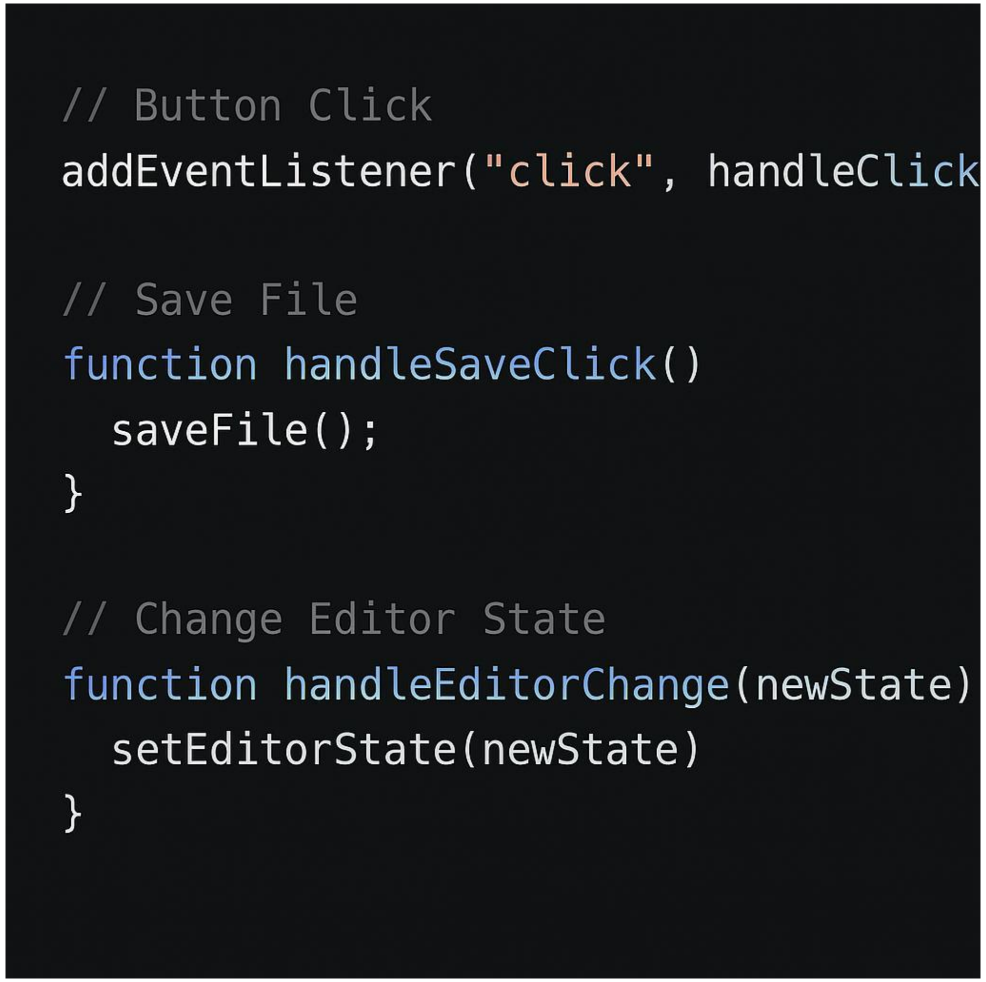


```
Hello World! From PUG CHALLENGE 2025 VALENCIA!
```

Hello World!

From PUG CHALLENGE 2025 VALENCIA!

Event Callbacks



```
// Button Click
addEventListener("click", handleClick

// Save File
function handleSaveClick()
  saveFile();
}

// Change Editor State
function handleEditorChange(newState)
  setEditorState(newState)
}
```

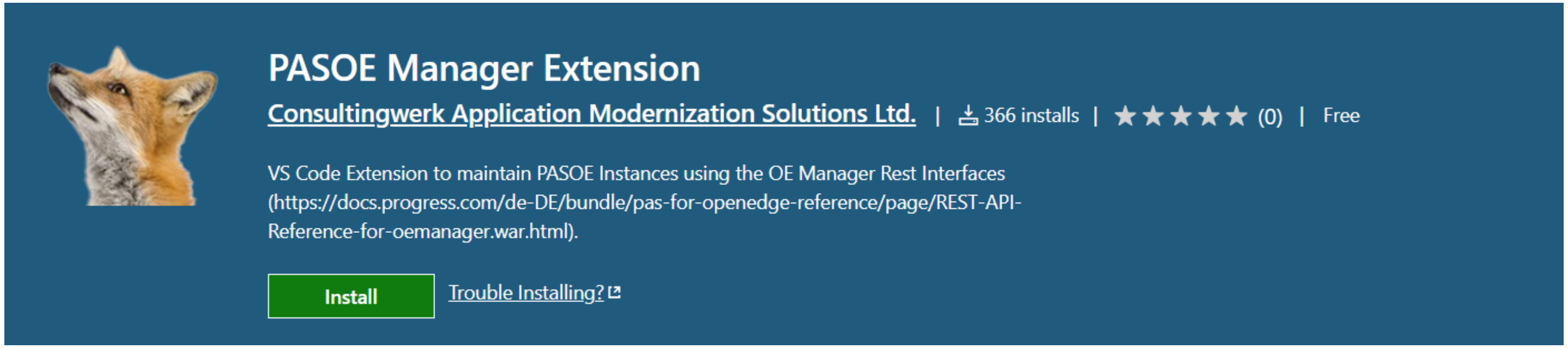
Why Event Callbacks are important:

- Central mechanism in VS Code extensions
- Enable the extension to react immediately to user interactions or system events
  - Examples: button clicks, saving a file, changing editor state
- Without callbacks, the extension wouldn’t know when relevant events occur
- Demo insight: the button itself is simple, real functionality comes from the callback connecting the Webview to the extension logic
- Makes extensions interactive, dynamic, and useful

Agenda

- Sample PASOE Manager Extension
- VS Code Marketplace and alternatives
- VS Code as an IDE
- Microsoft SDK for VS Code Extensions
- Building an Extension with Visual Studio Code
- Programming languages and file types used in a VS Code extension
- Event Callbacks
- Implementing the PASOE Manager Extension
- Package and publish your extension
- Important Common Capabilities
- Questions

OE Manager Web App of PASOE


PASOE Manager Extension
Consultingwerk Application Modernization Solutions Ltd. | 366 installs | ★★★★★ (0) | Free

VS Code Extension to maintain PASOE Instances using the OE Manager Rest Interfaces (https://docs.progress.com/de-DE/bundle/pas-for-openedge-reference/page/REST-API-Reference-for-oemanager.war.html).

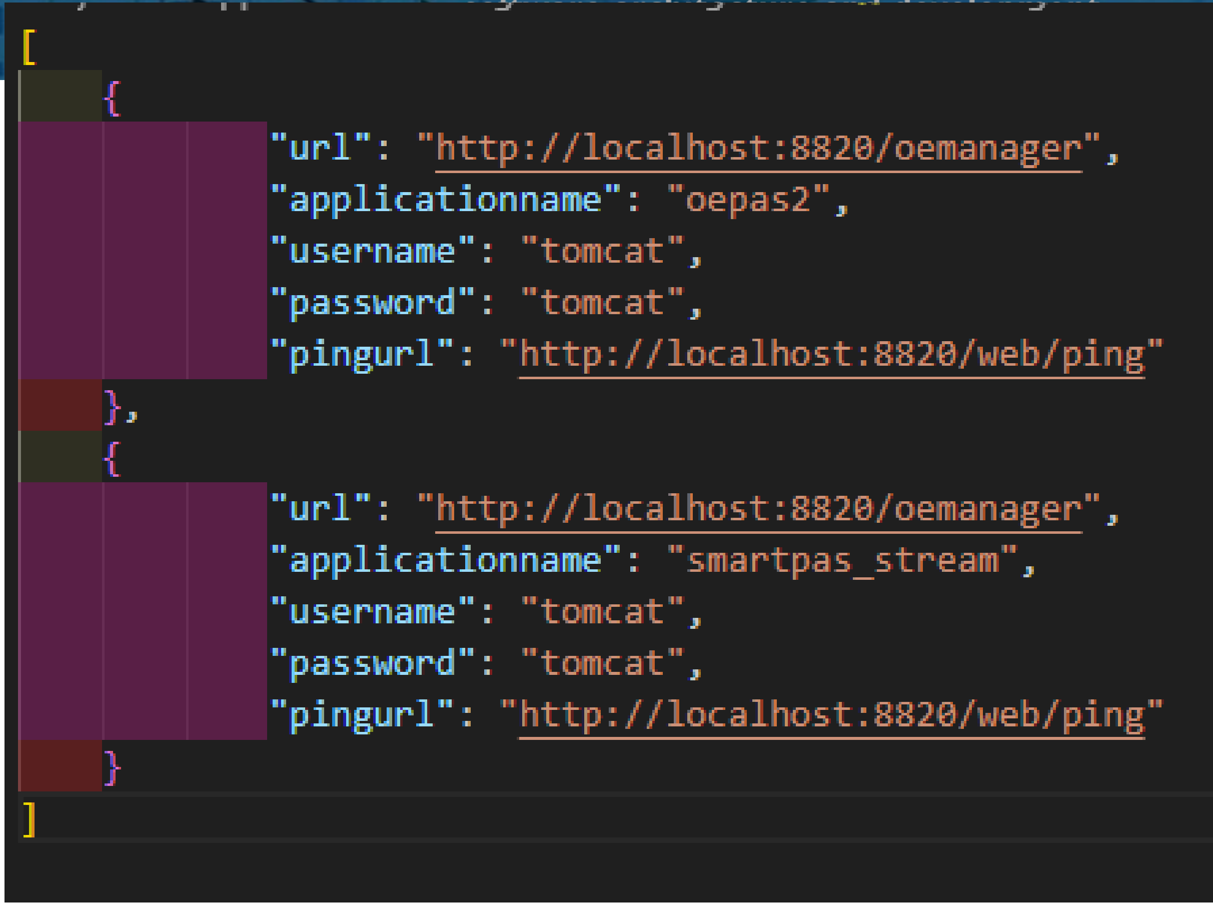
Install
Trouble Installing?

- PASOE Manager Extension links *VS Code* with the *OE Manager Web App*
- Brings monitoring and management features directly into the IDE
- OE Manager Web App offers real-time insights and tools to manage agents
- Extension integrates these features seamlessly into VS Code
- Enables efficient PASOE management without context switching
- Keeps developers focused on coding

OE Manager Web App of PASOE

Configuring PASOE in VS Code

- Workspace Settings:
  - Path: Extensions → OEMANAGER
- Local Config File (oemanager.conf):
  - Project-specific, stored locally
  - Supports single and multiple application configurations
- Edit Connections Command:
  - PASOE: Edit Connections opens the config file in VS Code
  - File is automatically created if it doesn’t exist



```
[
    {
        "url": "http://localhost:8820/oemanager",
        "applicationname": "oepas2",
        "username": "tomcat",
        "password": "tomcat",
        "pingurl": "http://localhost:8820/web/ping"
    },
    {
        "url": "http://localhost:8820/oemanager",
        "applicationname": "smartpas_stream",
        "username": "tomcat",
        "password": "tomcat",
        "pingurl": "http://localhost:8820/web/ping"
    }
]
```

OE Manager Web App of PASOE

Register command:

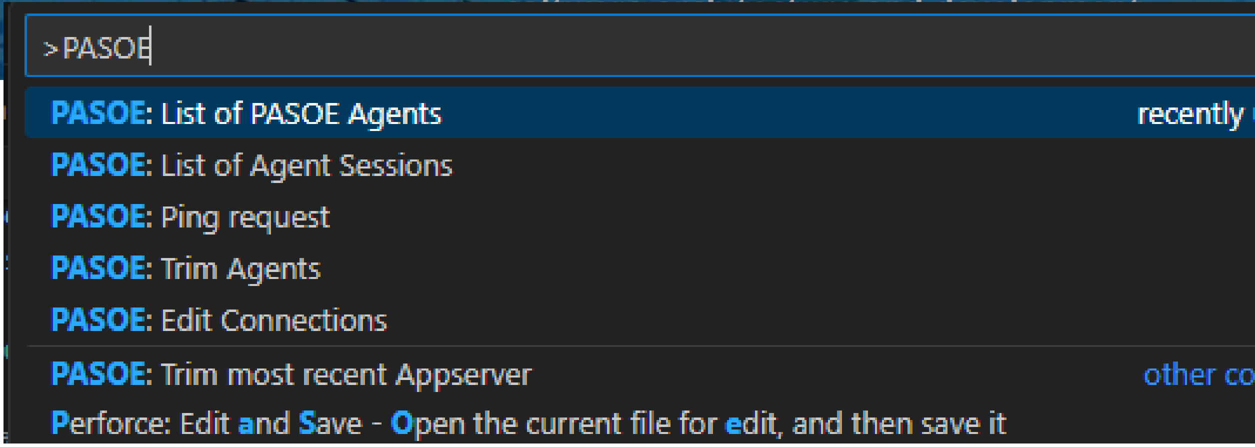
PASOE: List of Pasoe Agents

Purpose:

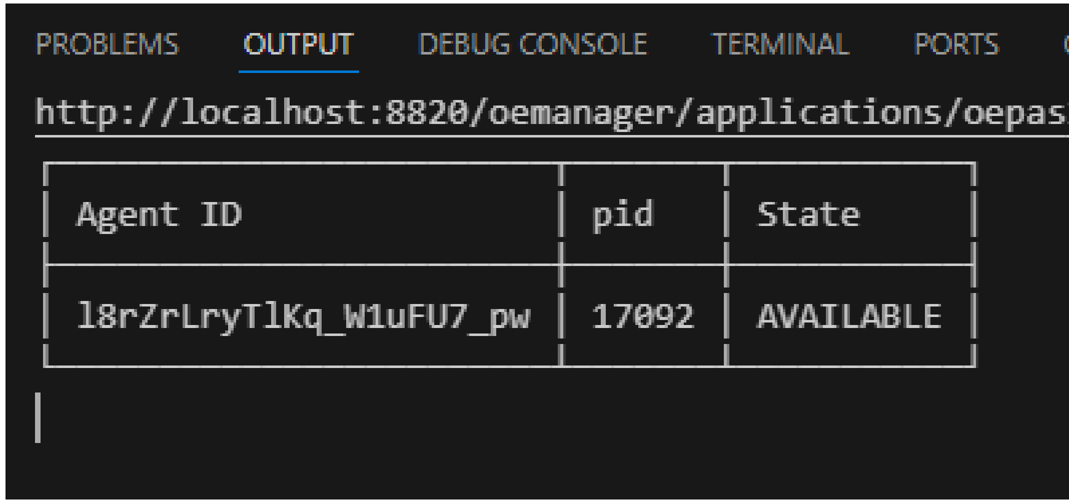
- Retrieve and display all active PASOE agents.
- Provides quick visibility into Agent ID, process ID (pid), and current state.
- Output is shown in the VS Code Output Channel

Technical Flow:

- Load extension
- Validate that application name is set in settings
- Construct GET request URL for /agents/ endpoint
- Fetch agent data safely with errorSafeFetch,
- Parse JSON response and extract agent list.



```
>PASOE
PASOE: List of PASOE Agents                                   recently
PASOE: List of Agent Sessions
PASOE: Ping request
PASOE: Trim Agents
PASOE: Edit Connections
PASOE: Trim most recent Appserver                             other co
Perforce: Edit and Save - Open the current file for edit, and then save it
```

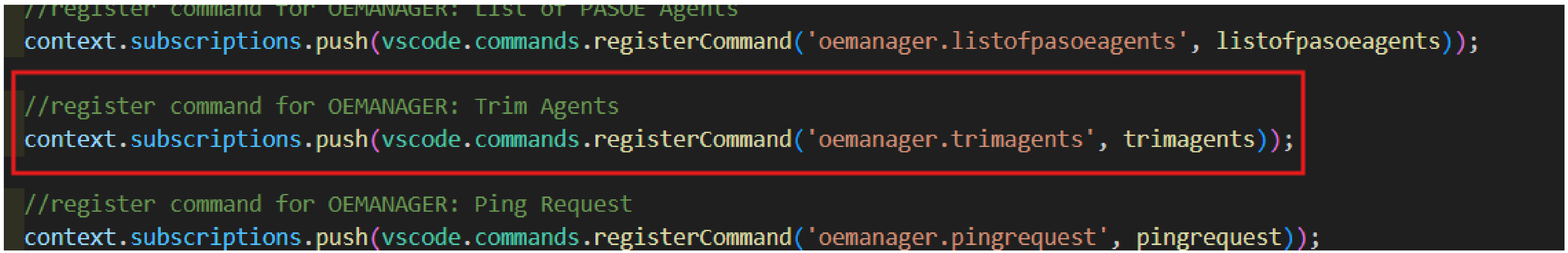


```
http://localhost:8820/oemanager/applications/oepas
┌──────────────────────┬───────┬───────────┐
│ Agent ID             │ pid   │ State     │
├──────────────────────┼───────┼───────────┤
│ l8rZrLryTlKq_W1uFU7_pw │ 17092 │ AVAILABLE │
└──────────────────────┴───────┴───────────┘
```

OE Manager Web App of PASOE – PASOE – trim agents

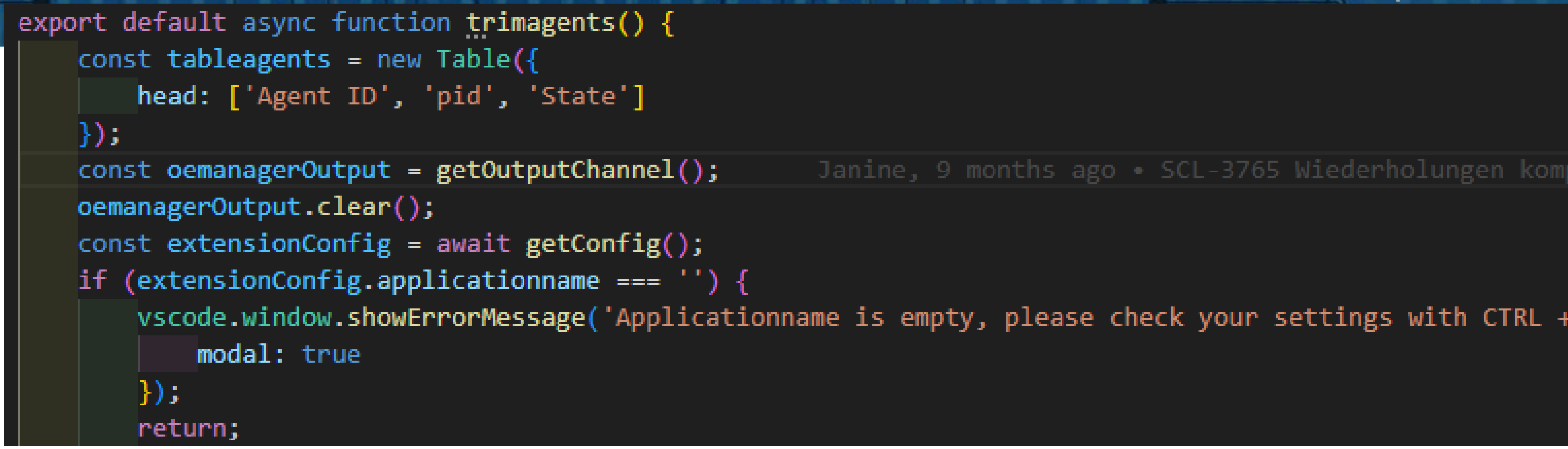
context.subscriptions.push(vscode.commands.registerCommand('oemanager.trimagents', trimagents));

- Registers the trimagents function as a VS Code command.
- Makes the command available in the Command Palette (Ctrl+Shift+P) under OEMANAGER.
- Connected to the PASOE Manager Extension functionality for trimming agents.



```
//register command for OEMANAGER: List of PASOE Agents
context.subscriptions.push(vscode.commands.registerCommand('oemanager.listofpasoeagents', listofpasoeagents));

//register command for OEMANAGER: Trim Agents
context.subscriptions.push(vscode.commands.registerCommand('oemanager.trimagents', trimagents));

//register command for OEMANAGER: Ping Request
context.subscriptions.push(vscode.commands.registerCommand('oemanager.pingrequest', pingrequest));
```

OE Manager Web App of PASOE



```
export default async function trimagents() {
    const tableagents = new Table({
        head: ['Agent ID', 'pid', 'State']
    });
    const oemanagerOutput = getOutputChannel();       Janine, 9 months ago • SCL-3765 Wiederholungen kom
    oemanagerOutput.clear();
    const extensionConfig = await getConfig();
    if (extensionConfig.applicationname === '') {
        vscode.window.showErrorMessage('Applicationname is empty, please check your settings with CTRL +
            modal: true
        });
        return;
```

- The function is asynchronous because HTTP requests (fetch) are used.
- Creates a table with the columns: Agent ID, Process ID, State.
- Retrieves the VS Code output channel ‘oemanager’.
- Deletes previous entries.
- Reads the extension configuration.
- If applicationname is empty → display warning in VS Code (modal pop-up).
- Abort because no agent can be reached without an application name.

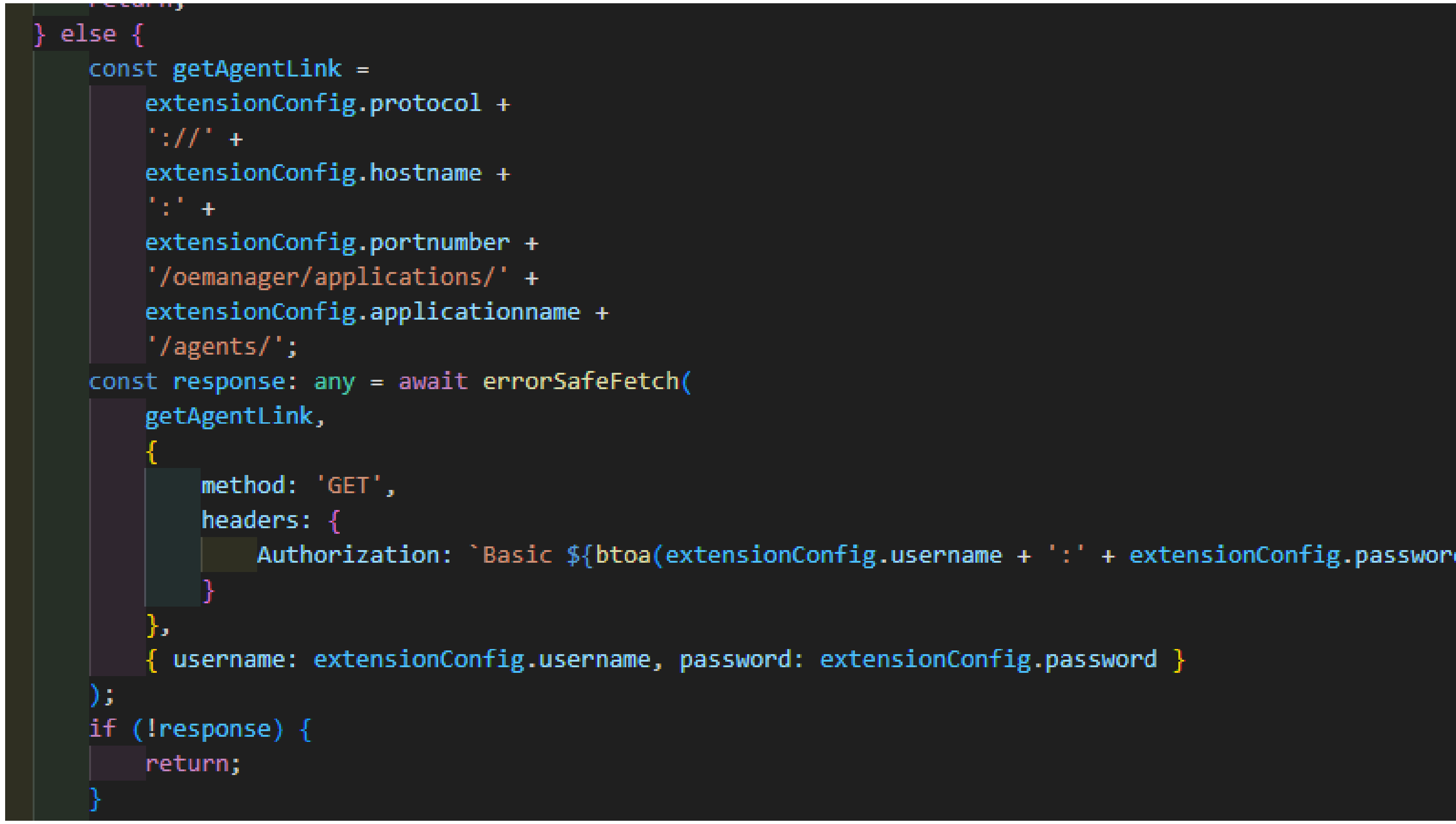
OE Manager Web App of PASOE

Prepare API endpoint for agents

- Put the URL together,
- e.g.:http://localhost:8820/oemanager/applications/oepas2/agents/

Retrieve agents from the server

- Retrieve all agents via GET.
- Authentication with Basic Auth (username:password).
- If no response → return.

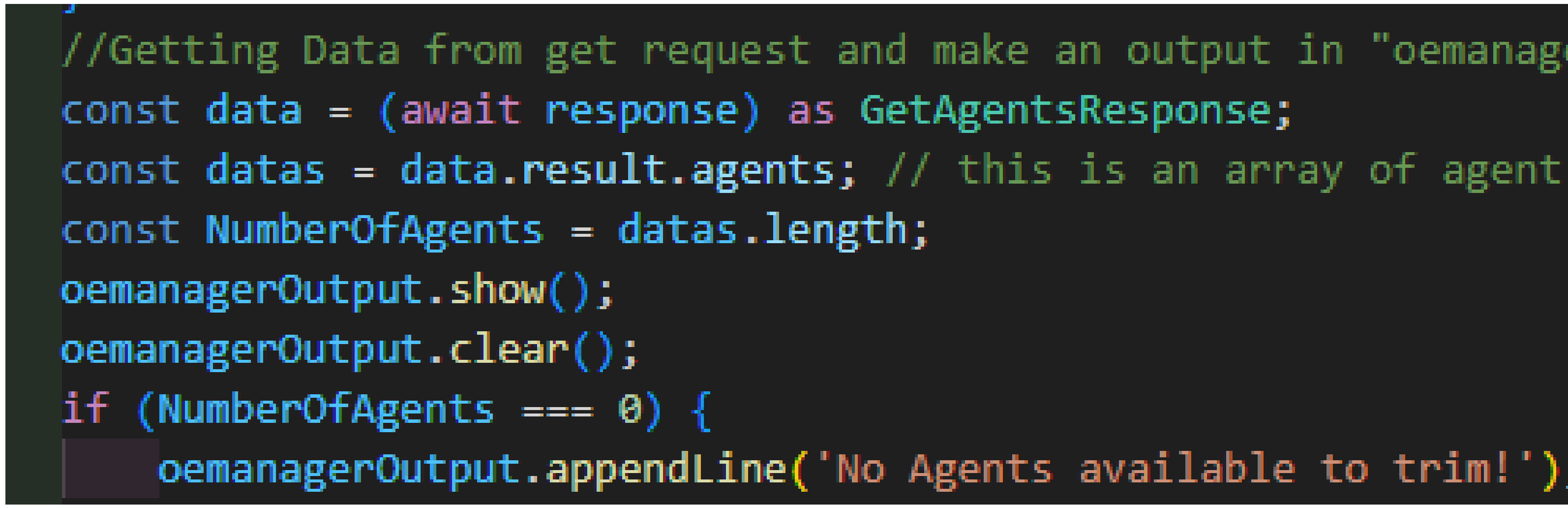


```
} else {
    const getAgentLink =
        extensionConfig.protocol +
        '://' +
        extensionConfig.hostname +
        ':' +
        extensionConfig.portnumber +
        '/oemanager/applications/' +
        extensionConfig.applicationname +
        '/agents/';
    const response: any = await errorSafeFetch(
        getAgentLink,
        {
            method: 'GET',
            headers: {
                Authorization: `Basic ${btoa(extensionConfig.username + ':' + extensionConfig.passwor
            }
        },
        { username: extensionConfig.username, password: extensionConfig.password }
    );
    if (!response) {
        return;
    }
```

OE Manager Web App of PASOE

Process reply

- Cast response to GetAgentsResponse format.
- Extract agents array.
- Calculate number of agents.
- Show output channel and clea it.
- If no agents are available → short message in the output.

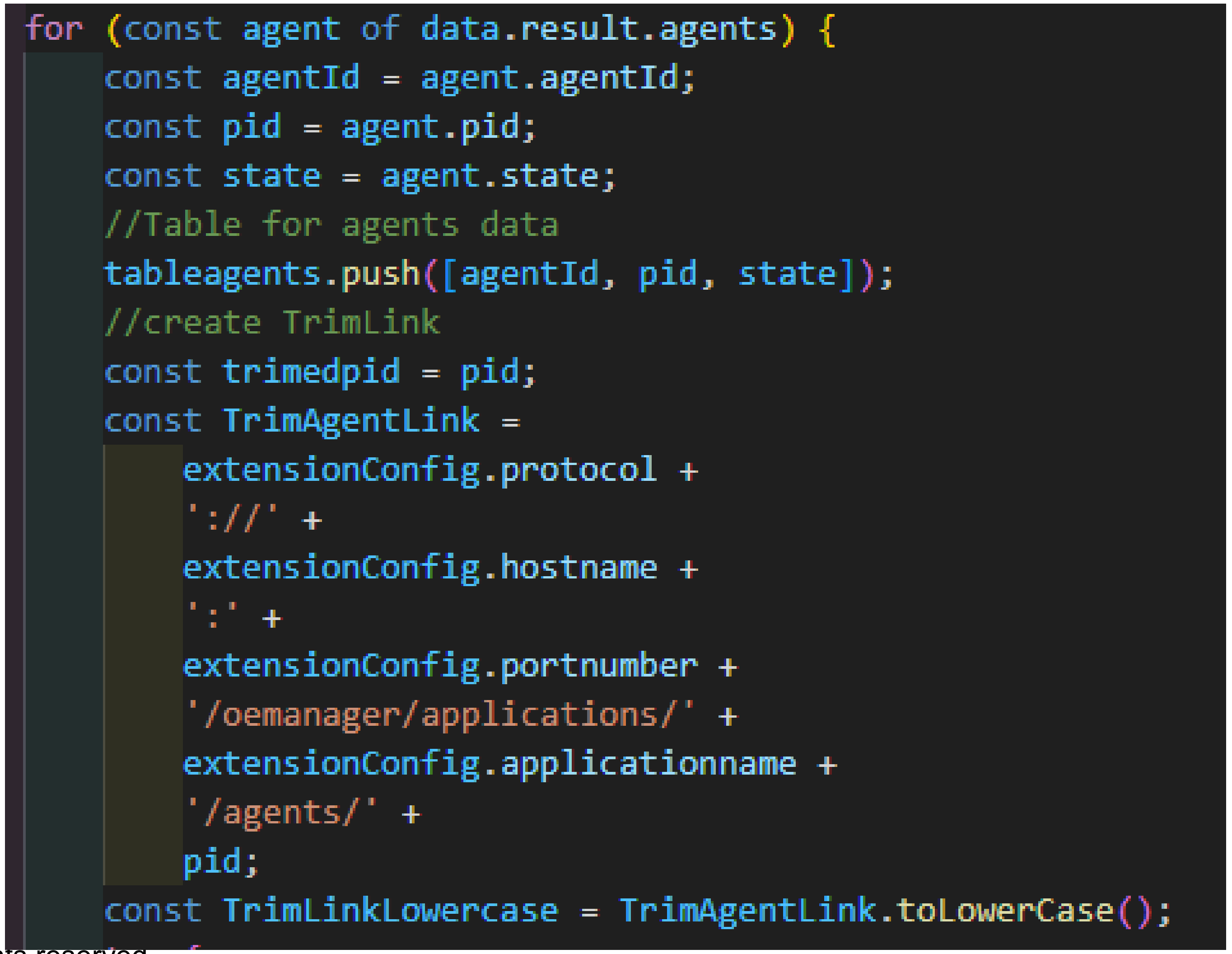


```
//Getting Data from get request and make an output in "oemanag
const data = (await response) as GetAgentsResponse;
const datas = data.result.agents; // this is an array of agent
const NumberOfAgents = datas.length;
oemanagerOutput.show();
oemanagerOutput.clear();
if (NumberOfAgents === 0) {
    oemanagerOutput.appendLine('No Agents available to trim!')
```

OE Manager Web App of PASOE

Agents pass through and trim

- Iterate through all agents.
- Extract agent information (ID, PID, state).
- Insert the values into the table.
- Build the DELETE URL for the trim command.



```
for (const agent of data.result.agents) {
    const agentId = agent.agentId;
    const pid = agent.pid;
    const state = agent.state;
    //Table for agents data
    tableagents.push([agentId, pid, state]);
    //create TrimLink
    const trimedpid = pid;
    const TrimAgentLink =
        extensionConfig.protocol +
        '://' +
        extensionConfig.hostname +
        ':' +
        extensionConfig.portnumber +
        '/oemanager/applications/' +
        extensionConfig.applicationname +
        '/agents/' +
        pid;
    const TrimLinkLowercase = TrimAgentLink.toLowerCase();
```

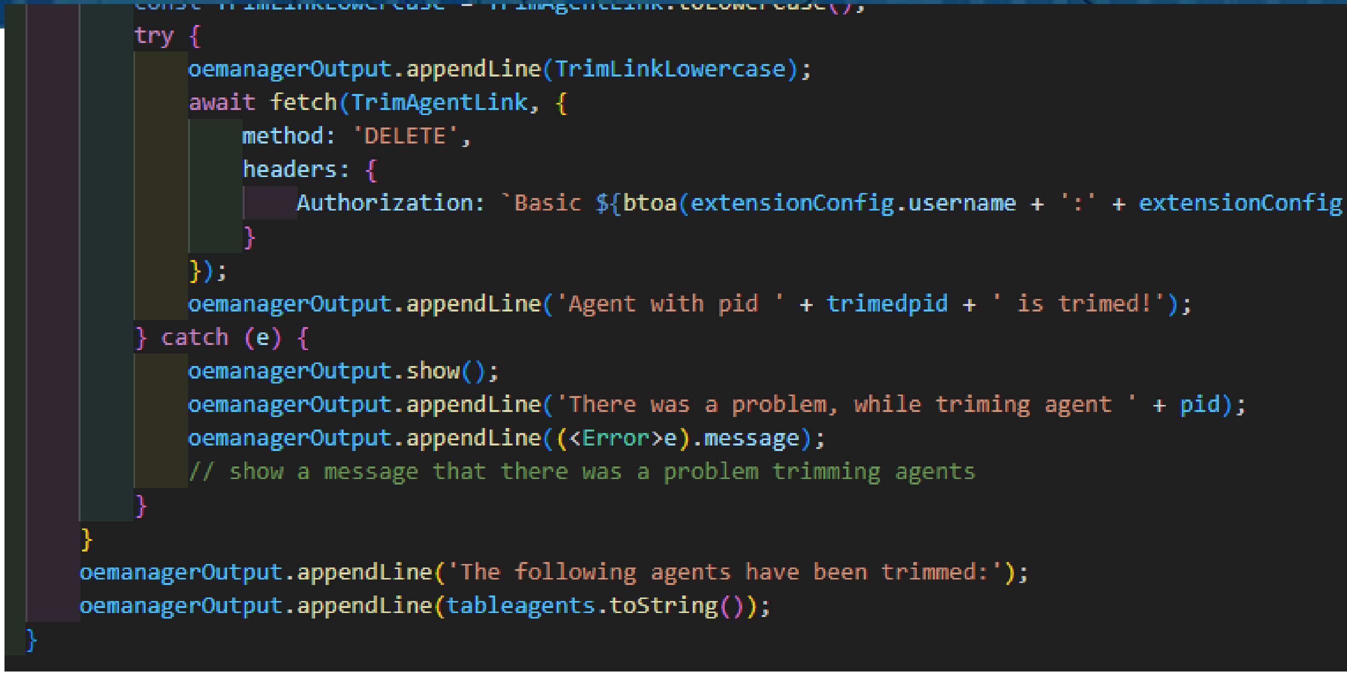
OE Manager Web App of PASOE

Execute trim request

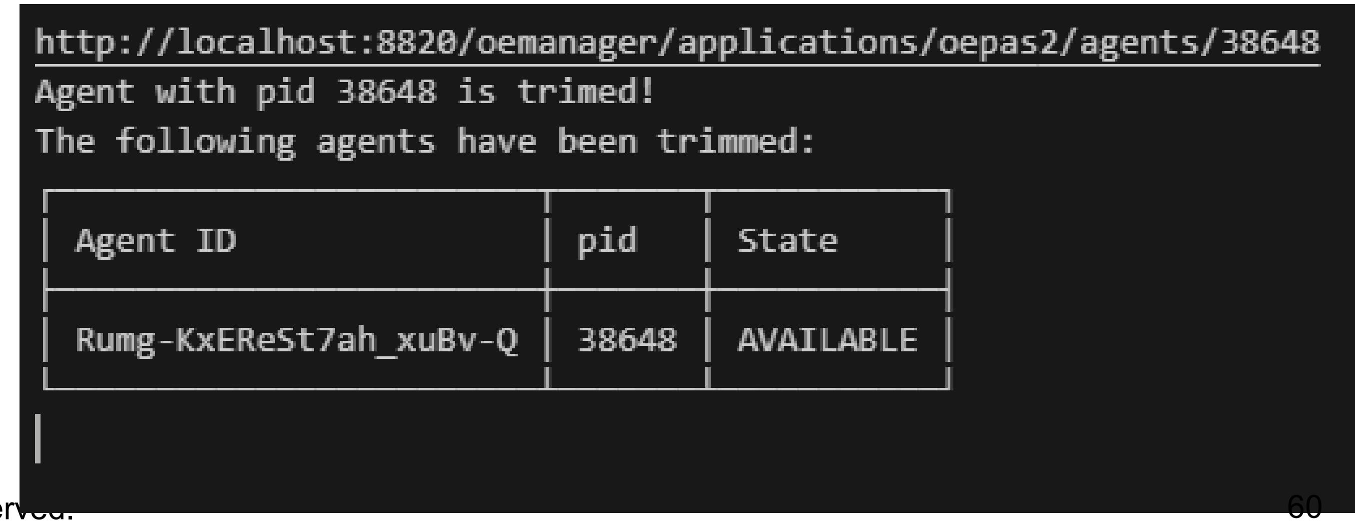
- DELETE request is sent to the agent (terminates the process).
- Success: Log message ‘Agent with pid ... is trimmed’.
- Error: Error message in the output channel.

Final output

- Summary in tabular form in the output channel.



```
        const TrimLinkLowercase = TrimAgentLink.toLowerCase();
        try {
            oemanagerOutput.appendLine(TrimLinkLowercase);
            await fetch(TrimAgentLink, {
                method: 'DELETE',
                headers: {
                    Authorization: `Basic ${btoa(extensionConfig.username + ':' + extensionConfig
                }
            });
            oemanagerOutput.appendLine('Agent with pid ' + trimedpid + ' is trimed!');
        } catch (e) {
            oemanagerOutput.show();
            oemanagerOutput.appendLine('There was a problem, while triming agent ' + pid);
            oemanagerOutput.appendLine((<Error>e).message);
            // show a message that there was a problem trimming agents
        }
    }
    oemanagerOutput.appendLine('The following agents have been trimmed:');
    oemanagerOutput.appendLine(tableagents.toString());
}
```

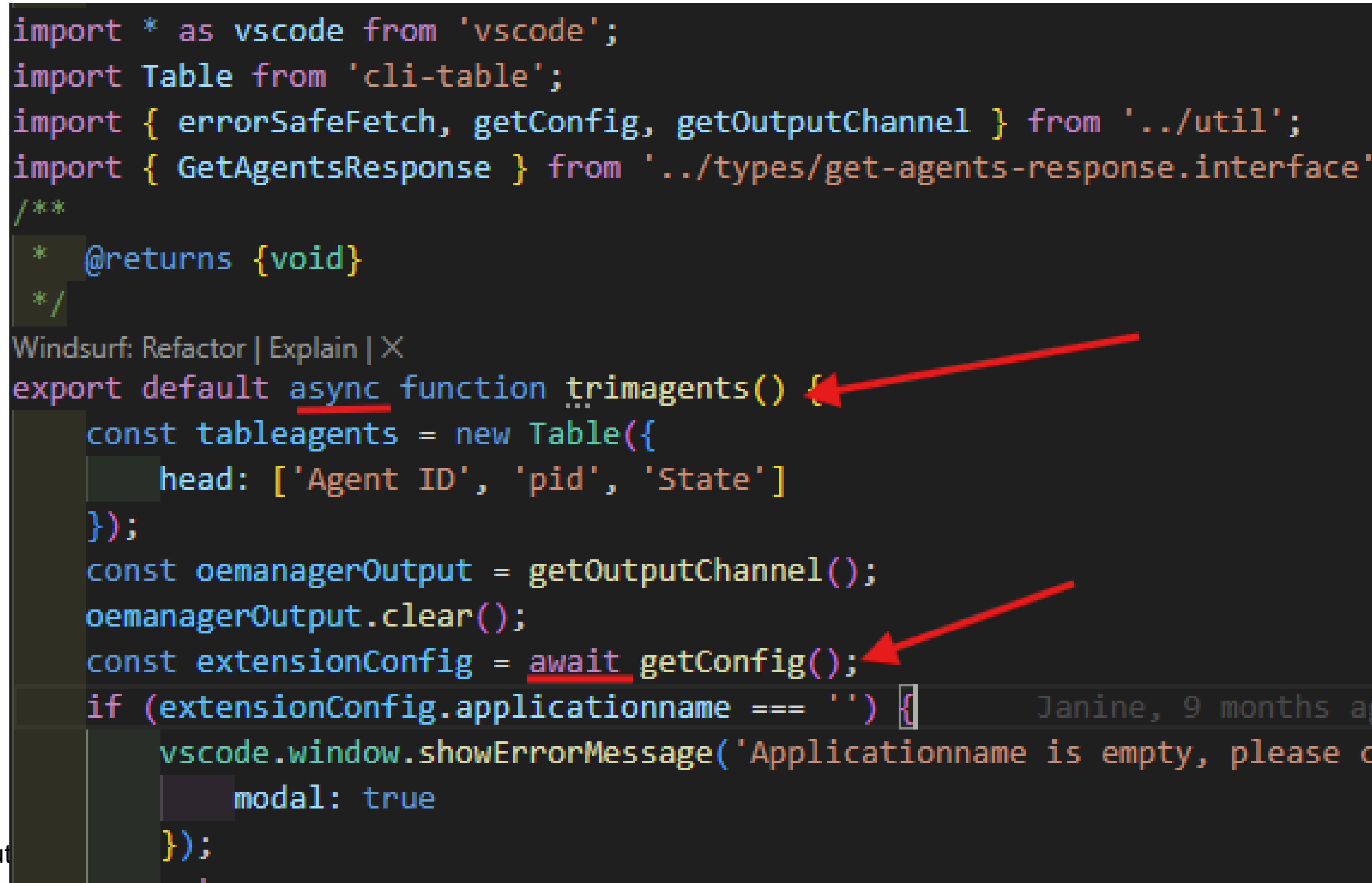


```
http://localhost:8820/oemanager/applications/oepas2/agents/38648
Agent with pid 38648 is trimed!
The following agents have been trimmed:

┌──────────────────────────┬───────┬───────────┐
│ Agent ID                 │ pid   │ State     │
├──────────────────────────┼───────┼───────────┤
│ Rumg-KxEReSt7ah_xuBv-Q   │ 38648 │ AVAILABLE │
└──────────────────────────┴───────┴───────────┘
```

OE Manager Web App of PASOE

Asynchronicity

- Handles time-consuming tasks (e.g. network, file I/O)
- Keeps VS Code responsive via async programming
- Uses async / await for smooth execution
- Example: trimagents() calls getConfig(),

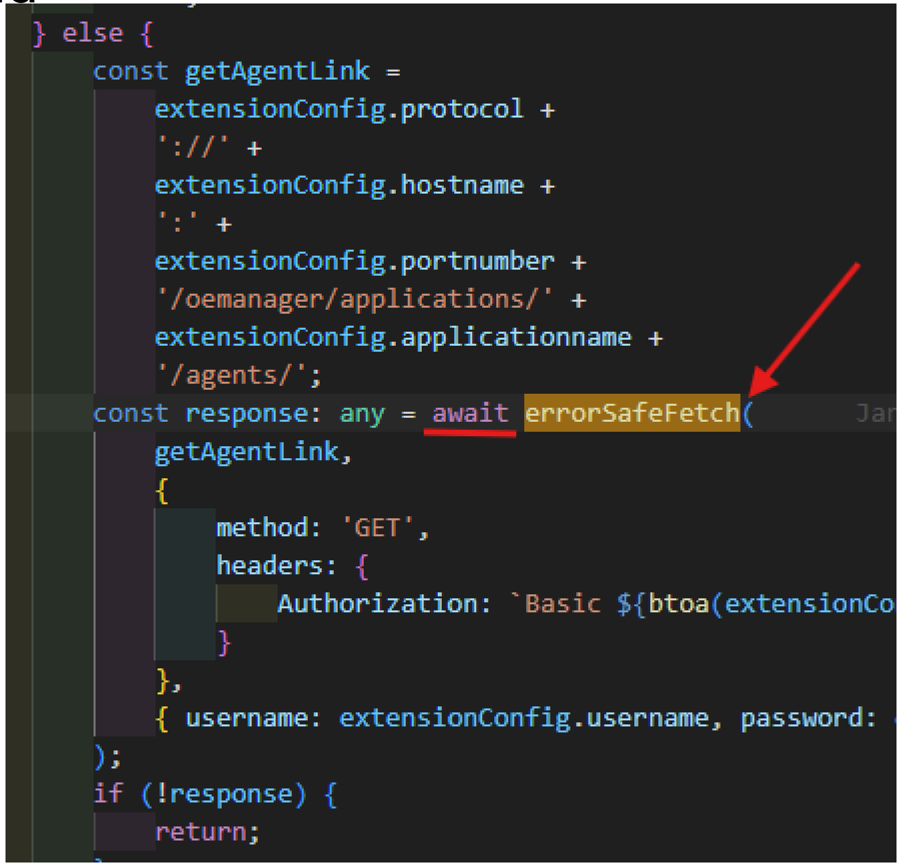


```
import * as vscode from 'vscode';
import Table from 'cli-table';
import { errorSafeFetch, getConfig, getOutputChannel } from '../util';
import { GetAgentsResponse } from '../types/get-agents-response.interface'
/**
 *  @returns {void}
 */
Windsurf: Refactor | Explain | ✕
export default async function trimagents() {
    const tableagents = new Table({
        head: ['Agent ID', 'pid', 'State']
    });
    const oemanagerOutput = getOutputChannel();
    oemanagerOutput.clear();
    const extensionConfig = await getConfig();
    if (extensionConfig.applicationname === '') {        Janine, 9 months a
        vscode.window.showErrorMessage('Applicationname is empty, please c
            modal: true
        });
```

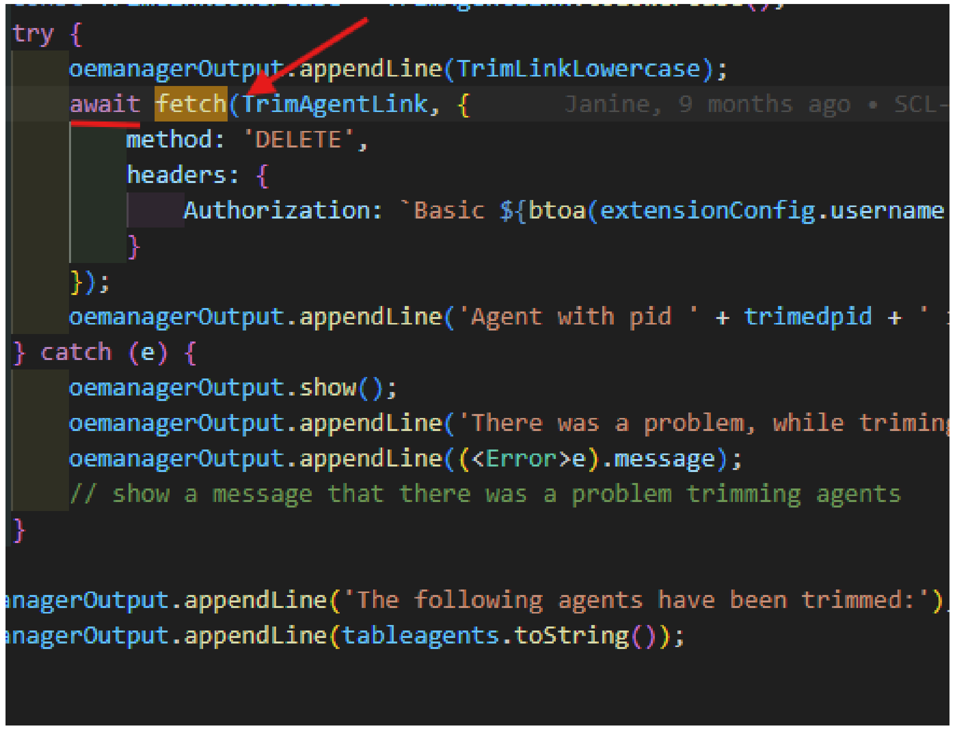
OE Manager Web App of PASOE

errorSafeFetch(), and fetch() asynchronously

- **Benefits:** non-blocking, better performance, cleaner code



```
} else {
    const getAgentLink =
        extensionConfig.protocol +
        '://' +
        extensionConfig.hostname +
        ':' +
        extensionConfig.portnumber +
        '/oemanager/applications/' +
        extensionConfig.applicationname +
        '/agents/';
    const response: any = await errorSafeFetch(          Jar
        getAgentLink,
        {
            method: 'GET',
            headers: {
                Authorization: `Basic ${btoa(extensionCo
            }
        },
        { username: extensionConfig.username, password:
    );
    if (!response) {
        return;
```



```
try {
    oemanagerOutput.appendLine(TrimLinkLowercase);
    await fetch(TrimAgentLink, {        Janine, 9 months ago • SCL-
        method: 'DELETE',
        headers: {
            Authorization: `Basic ${btoa(extensionConfig.username
        }
    });
    oemanagerOutput.appendLine('Agent with pid ' + trimedpid + '
} catch (e) {
    oemanagerOutput.show();
    oemanagerOutput.appendLine('There was a problem, while triming
    oemanagerOutput.appendLine((<Error>e).message);
    // show a message that there was a problem trimming agents
}

nagerOutput.appendLine('The following agents have been trimmed:')
nagerOutput.appendLine(tableagents.toString());
```

Agenda

- Sample PASOE Manager Extension
- VS Code Marketplace and alternatives
- VS Code as an IDE
- Microsoft SDK for VS Code Extensions
- Building an Extension with Visual Studio Code
- Programming languages and file types used in a VS Code extension
- Event Callbacks
- Implementing the PASOE Manager Extension
- Package and publish your extension
- Important Common Capabilities
- Questions

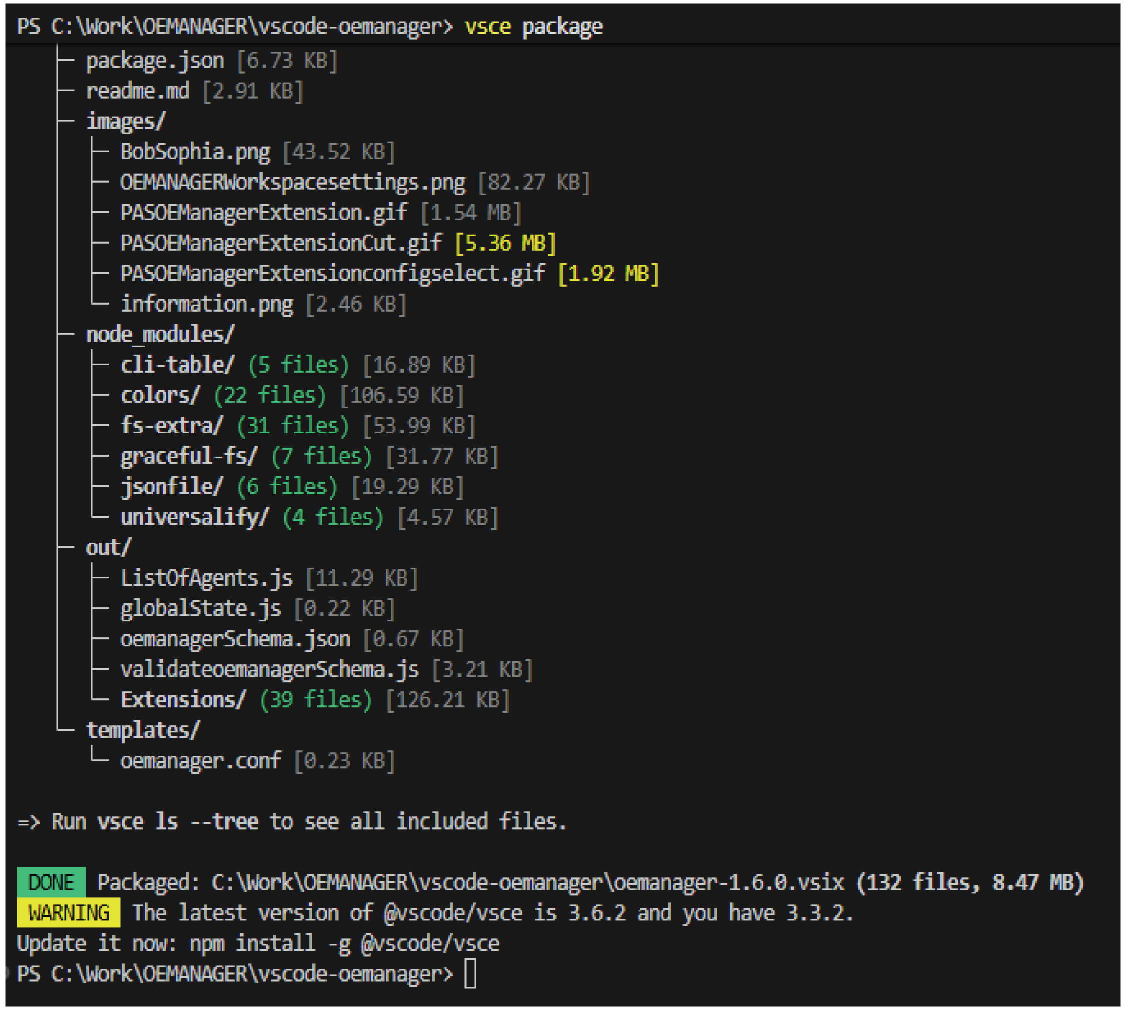
Package and publish your extension

To install an extension manually, it must be packaged as a .vsix file. This can be done directly from the VS Code Terminal using the following command:

```
PS C:\Work\OEMANAGER\vscode-oemanager> vsce package
```

Install the package using the command:

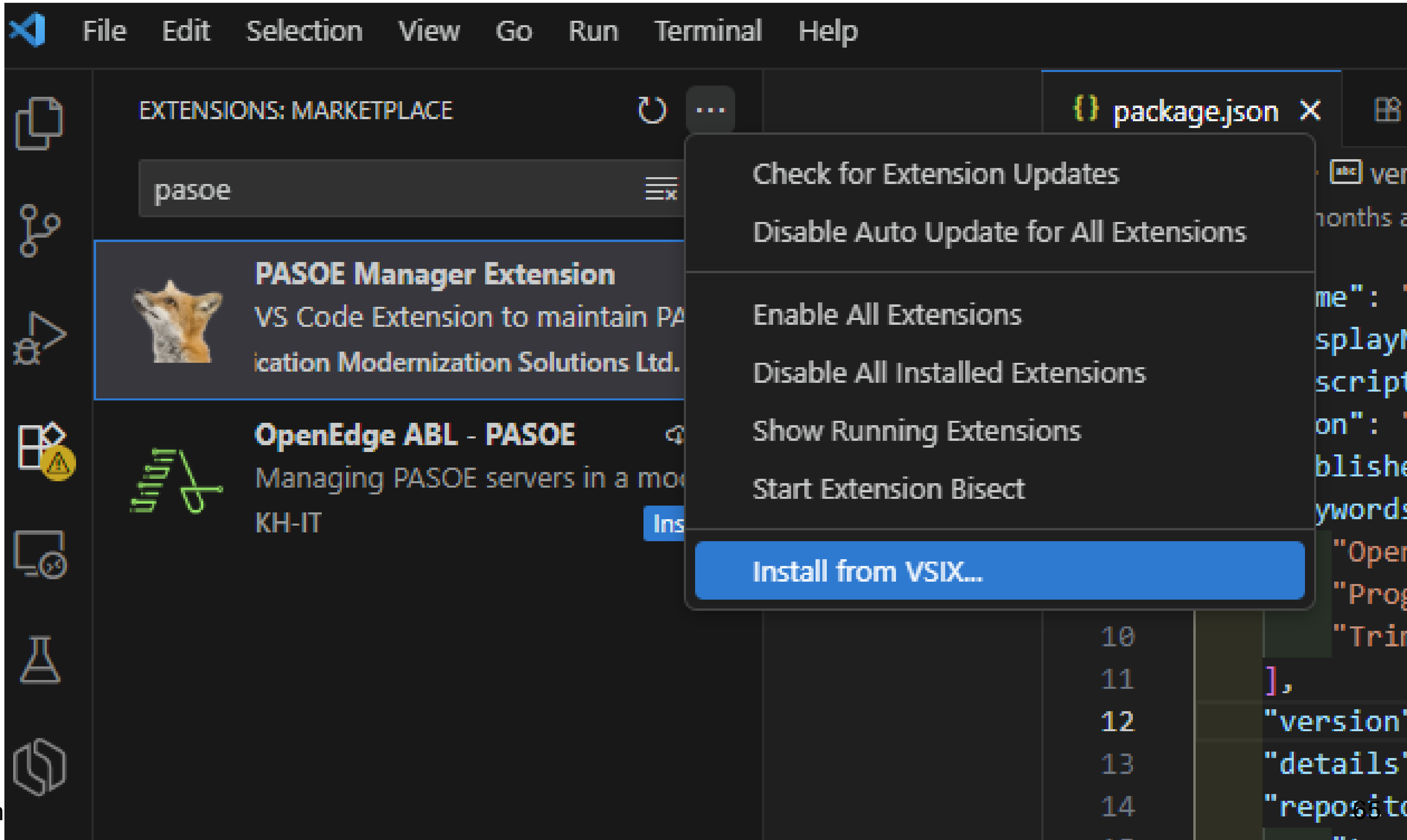
```
code --install-extension oemanager-1.6.0.vsix
Installing extensions...
Extension ‘oemanager-1.6.0.vsix' was successfully installed.
```



```
PS C:\Work\OEMANAGER\vscode-oemanager> vsce package
  ├─ package.json [6.73 KB]
  ├─ readme.md [2.91 KB]
  ├─ images/
  │  ├─ BobSophia.png [43.52 KB]
  │  ├─ OEMANAGERWorkspacesettings.png [82.27 KB]
  │  ├─ PASOEManagerExtension.gif [1.54 MB]
  │  ├─ PASOEManagerExtensionCut.gif [5.36 MB]
  │  ├─ PASOEManagerExtensionconfigselect.gif [1.92 MB]
  │  └─ information.png [2.46 KB]
  ├─ node_modules/
  │  ├─ cli-table/ (5 files) [16.89 KB]
  │  ├─ colors/ (22 files) [106.59 KB]
  │  ├─ fs-extra/ (31 files) [53.99 KB]
  │  ├─ graceful-fs/ (7 files) [31.77 KB]
  │  ├─ jsonfile/ (6 files) [19.29 KB]
  │  └─ universalify/ (4 files) [4.57 KB]
  ├─ out/
  │  ├─ ListOfAgents.js [11.29 KB]
  │  ├─ globalState.js [0.22 KB]
  │  ├─ oemanagerSchema.json [0.67 KB]
  │  ├─ validateoemanagerSchema.js [3.21 KB]
  │  └─ Extensions/ (39 files) [126.21 KB]
  └─ templates/
     └─ oemanager.conf [0.23 KB]

=> Run vsce ls --tree to see all included files.

 DONE  Packaged: C:\Work\OEMANAGER\vscode-oemanager\oemanager-1.6.0.vsix (132 files, 8.47 MB)
 WARNING  The latest version of @vscode/vsce is 3.6.2 and you have 3.3.2.
Update it now: npm install -g @vscode/vsce
PS C:\Work\OEMANAGER\vscode-oemanager> 
```

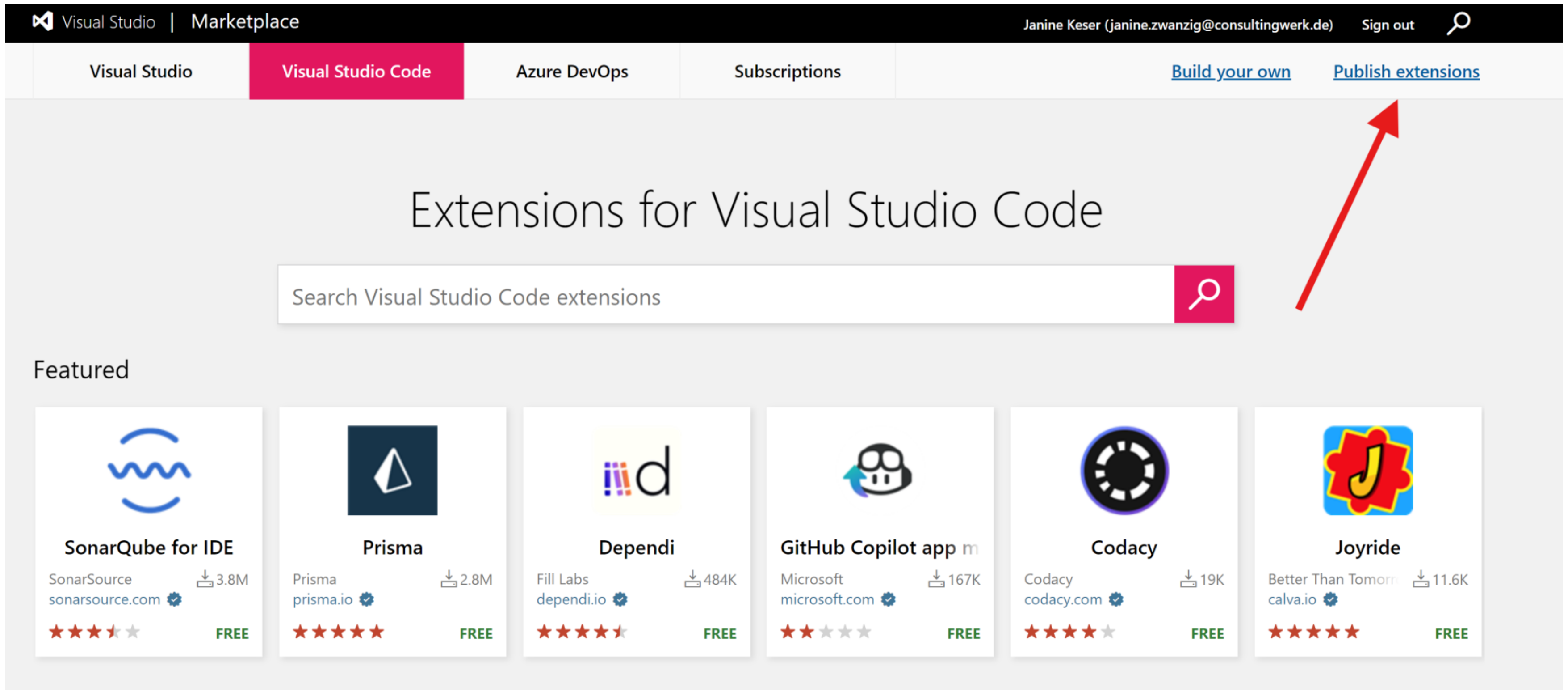
Package and publish your extension

- Alternative installation method: use the “Install from VSIX…” button in the Extensions view
- Select the .vsix file from a local folder
- Installs the extension directly into VS Code without using the terminal



Package and publish your extension

- VS Code Marketplace home page features the “Publish extensions” button
- Manage your own extensions: publish, update, and monitor usage



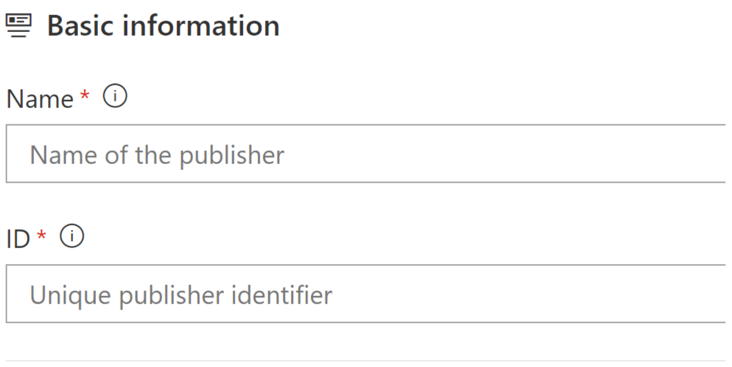
Package and publish your extension

Steps before publishing an extension:

Create a publisher

Publisher name

Unique ID

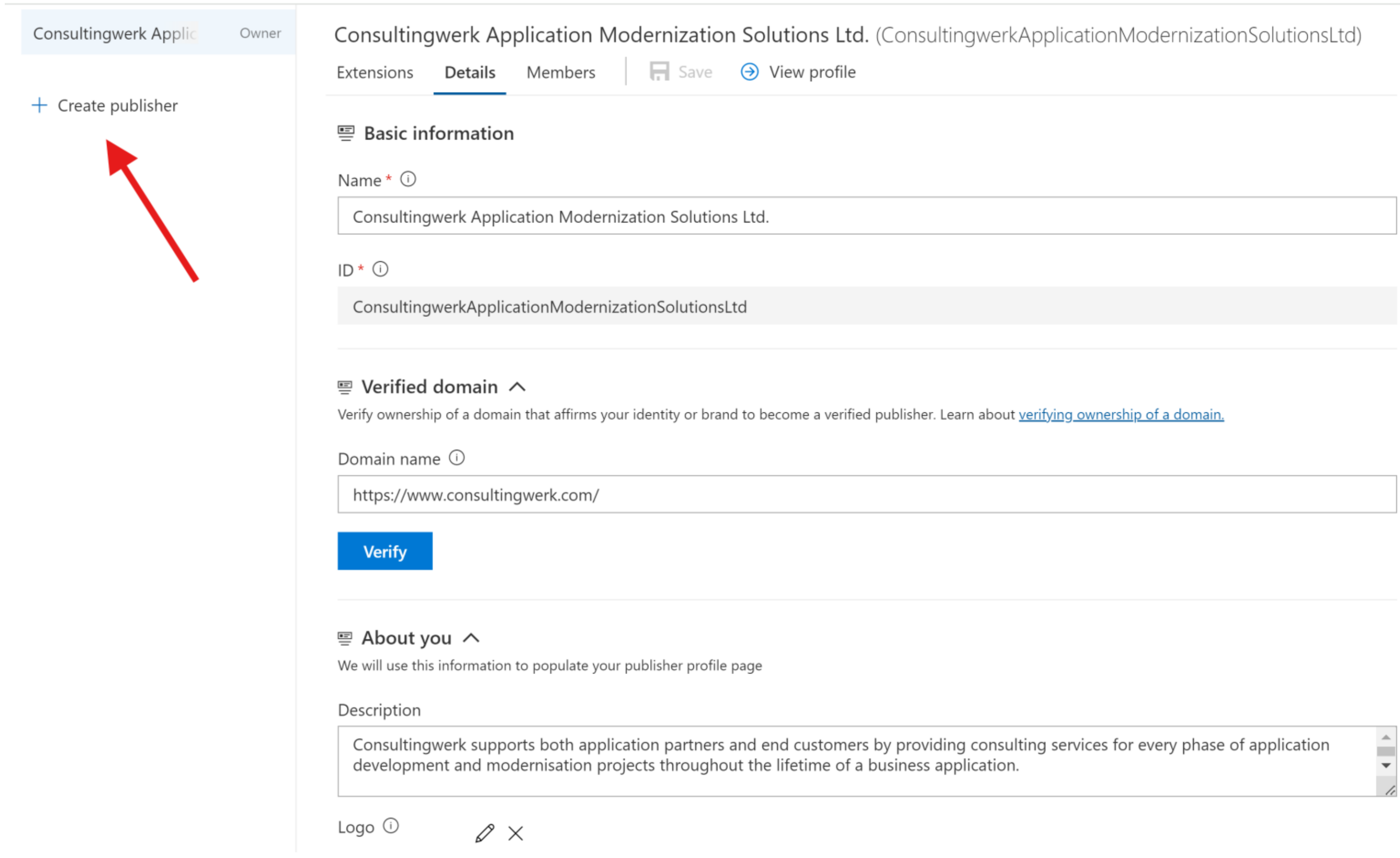

Basic information

Name *

Name of the publisher

ID *

Unique publisher identifier



Consultingwerk Application Modernization Solutions Ltd. (ConsultingwerkApplicationModernizationSolutionsLtd)

Basic information

Name *

Consultingwerk Application Modernization Solutions Ltd.

ID *

ConsultingwerkApplicationModernizationSolutionsLtd

Verified domain

Verify ownership of a domain that affirms your identity or brand to become a verified publisher. Learn about verifying ownership of a domain.

Domain name

https://www.consultingwerk.com/

About you

We will use this information to populate your publisher profile page

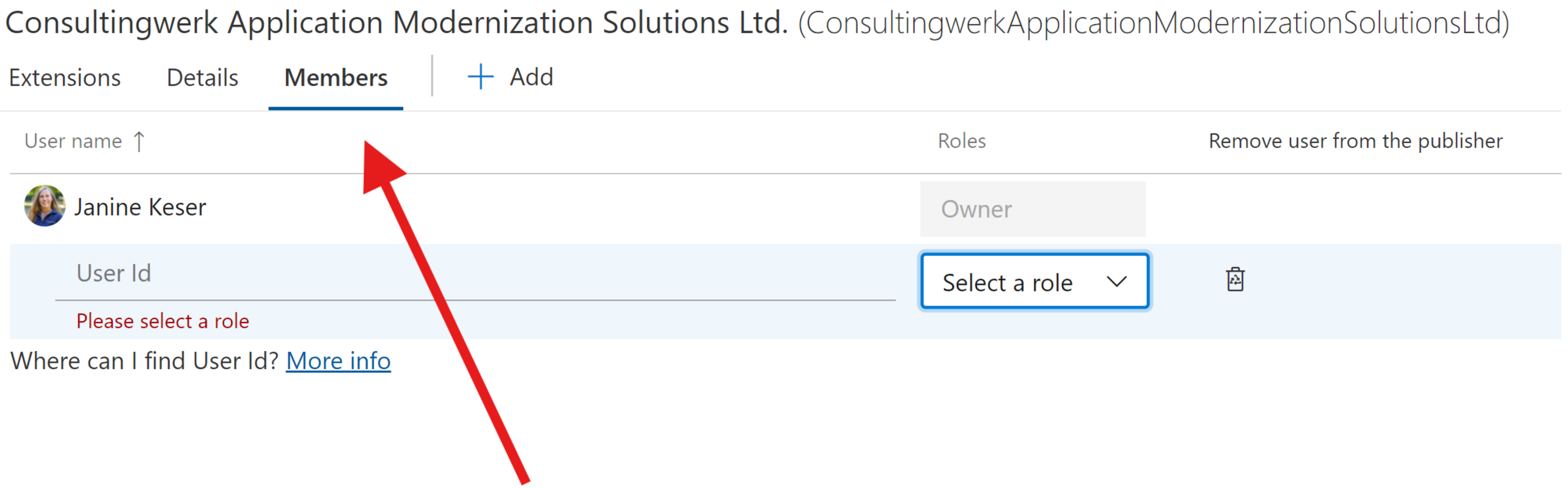
Description

Consultingwerk supports both application partners and end customers by providing consulting services for every phase of application development and modernisation projects throughout the lifetime of a business application.

Logo

Package and publish your extension

If you have created the publisher, you need to create members and you can select different roles for them.



Consultingwerk Application Modernization Solutions Ltd. (ConsultingwerkApplicationModernizationSolutionsLtd)

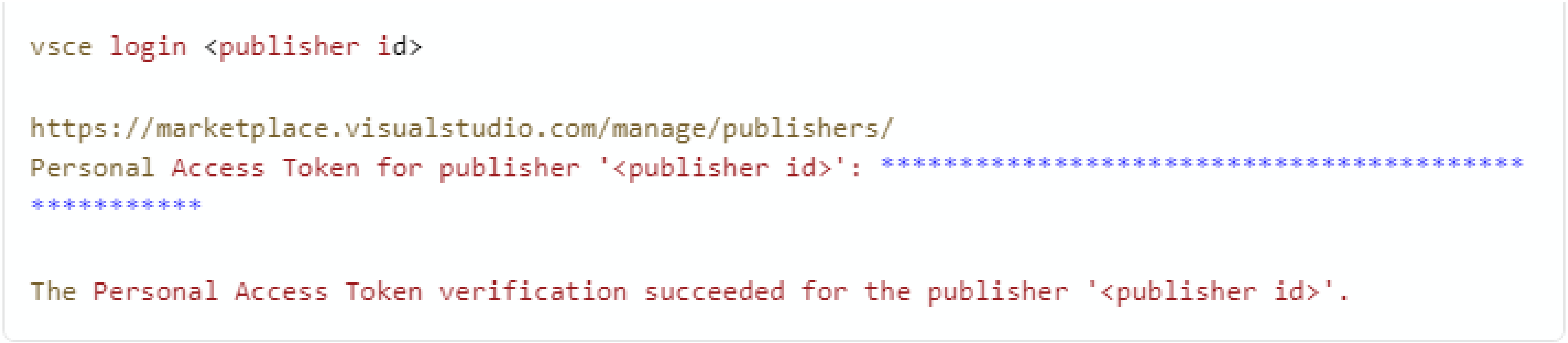
Extensions Details Members + Add

User name ↑	Roles	Remove user from the publisher
Janine Keser	Owner	
User Id Please select a role	Select a role	

Where can I find User Id? More info

Package and publish your extension

Verify the newly created publisher using vsce. In your terminal, run the following command, and when prompted, type the Personal Access Token created in the previous step:



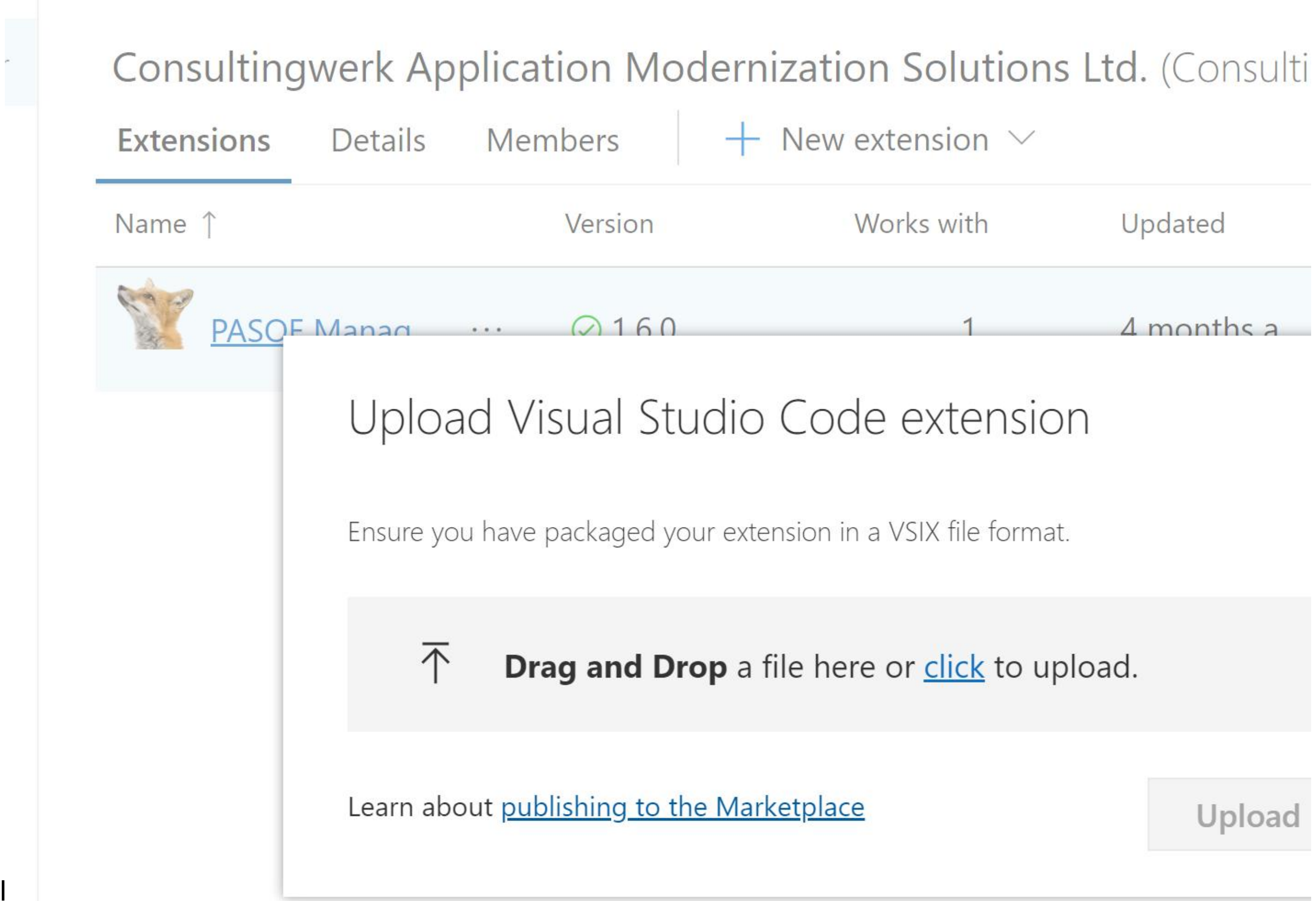
```
vsce login <publisher id>

https://marketplace.visualstudio.com/manage/publishers/
Personal Access Token for publisher '<publisher id>': ******************************************
***********

The Personal Access Token verification succeeded for the publisher '<publisher id>'.
```

Package and publish your own extension

You can then publish your extension.

There are three different ways to do this:

- Drag and drop into this window.
- Click to upload from a local folder.



Consultingwerk Application Modernization Solutions Ltd. (Consulti

Upload Visual Studio Code extension

Ensure you have packaged your extension in a VSIX file format.

Drag and Drop a file here or click to upload.

Learn about publishing to the Marketplace

Package and publish your own extension

With the following command from VS Code Terminal

```
vsce publish
```

Automatically increment its version number

update an extension from 1.0.0 to 1.1.0,

you would specify:

```
vsce publish 1.1.0
```

Changes automatically the version attribute in the package.json before publishing with the new Version number

Agenda

- Sample PASOE Manager Extension
- VS Code Marketplace and alternatives
- VS Code as an IDE
- Microsoft SDK for VS Code Extensions
- Building an Extension with Visual Studio Code
- Programming languages and file types used in a VS Code extension
- Event Callbacks
- Implementing the PASOE Manager Extension
- Package and publish your extension
- Important Common Capabilities
- Questions

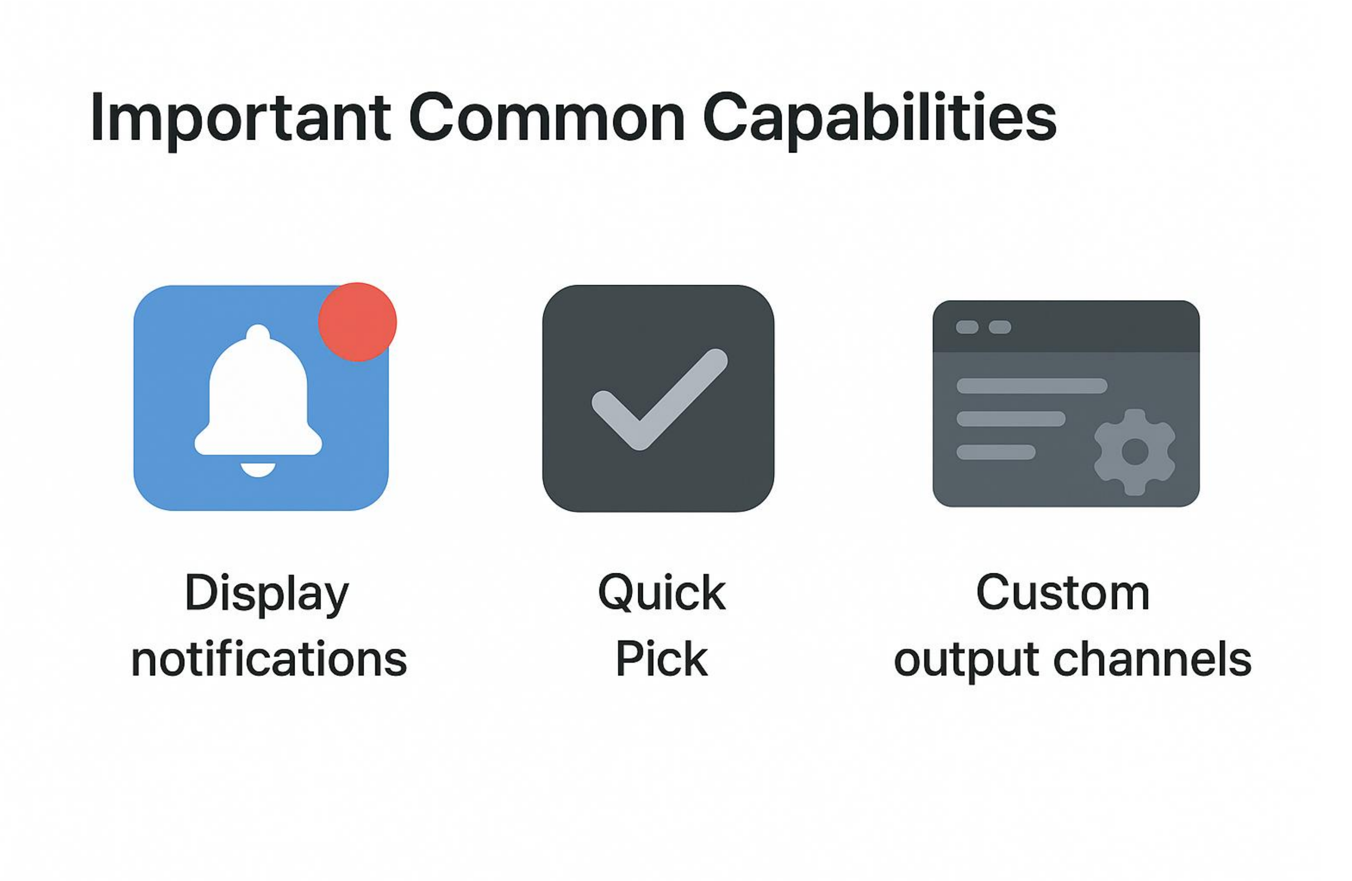
Important Common Capabilities

Visual Studio Code extensions offer a variety of common features, help developers

- design user interactions,
- display data
- improve workflows.

Examples of frequently used capabilities

- Display notifications
- Quick Pick
- Custom output channels



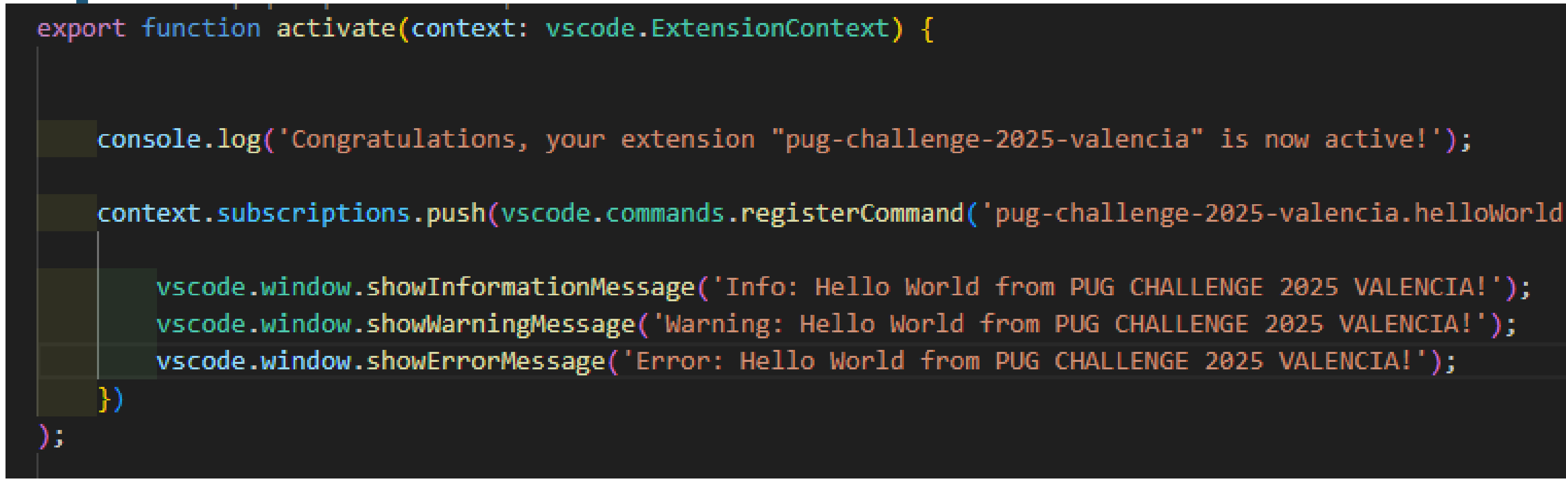
Important Common Capabilities

Display notifications:

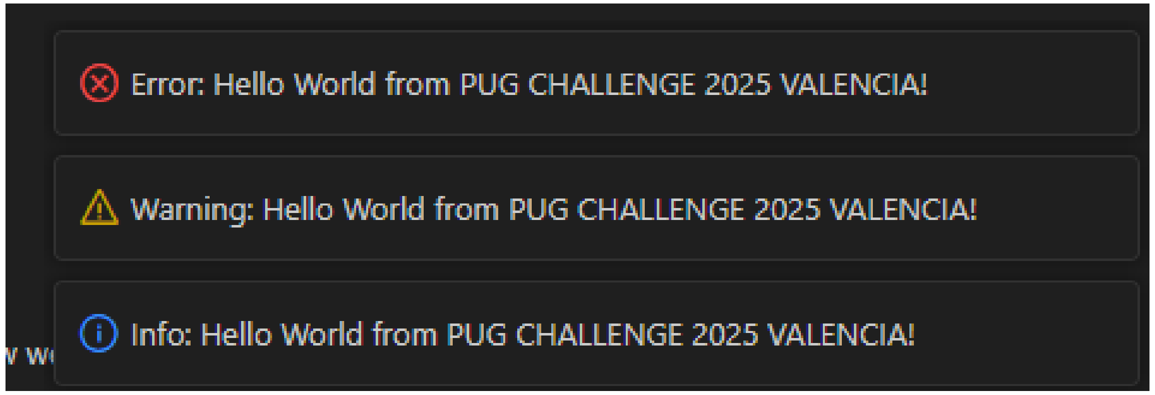
There are three types of notifications that make it easy to display a message to the user:

- Info
- Warning
- Error

The picture shows the difference between the three notifications.



```
export function activate(context: vscode.ExtensionContext) {


    console.log('Congratulations, your extension "pug-challenge-2025-valencia" is now active!');

    context.subscriptions.push(vscode.commands.registerCommand('pug-challenge-2025-valencia.helloWorld

        vscode.window.showInformationMessage('Info: Hello World from PUG CHALLENGE 2025 VALENCIA!');
        vscode.window.showWarningMessage('Warning: Hello World from PUG CHALLENGE 2025 VALENCIA!');
        vscode.window.showErrorMessage('Error: Hello World from PUG CHALLENGE 2025 VALENCIA!');
    })
);
```



Error: Hello World from PUG CHALLENGE 2025 VALENCIA!

Warning: Hello World from PUG CHALLENGE 2025 VALENCIA!

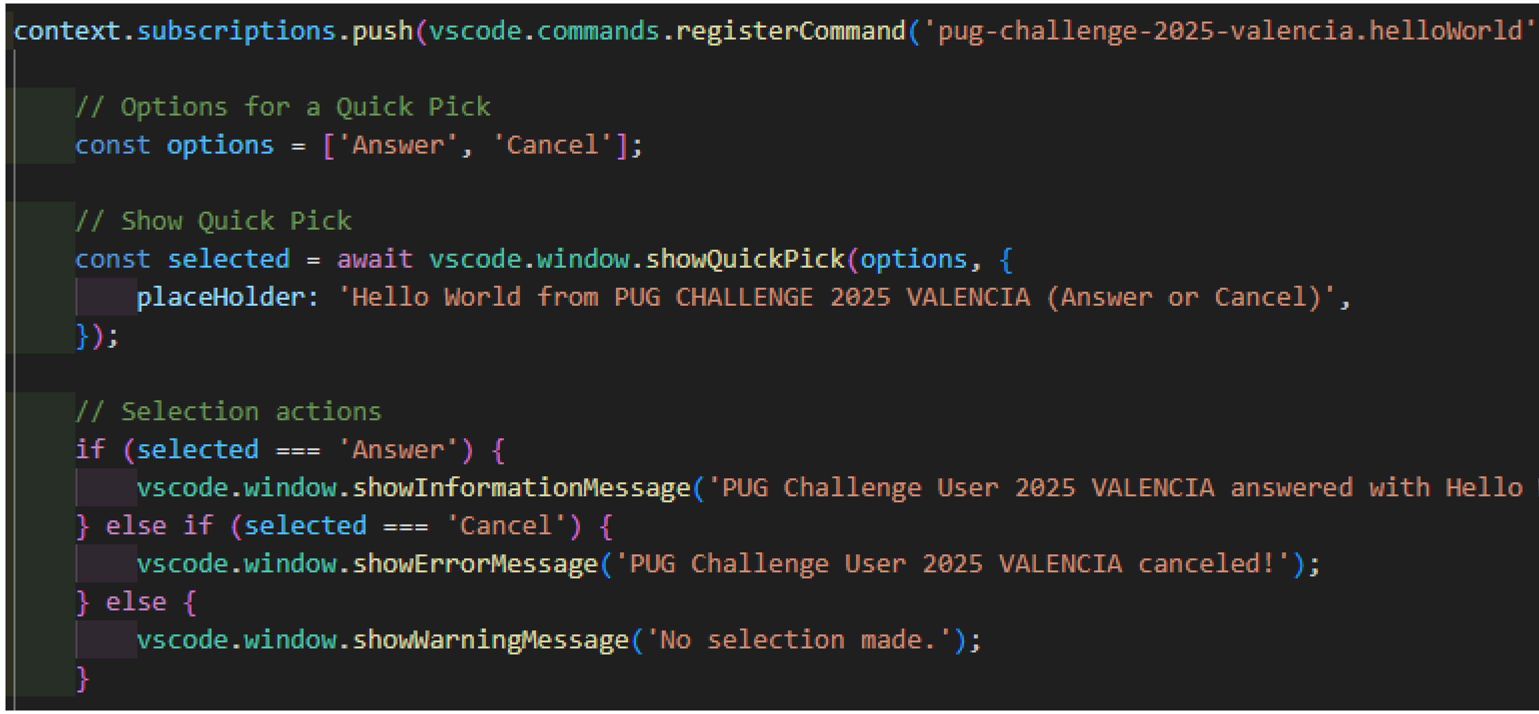
Info: Hello World from PUG CHALLENGE 2025 VALENCIA!

Important Common Capabilities

Quick Picks

Our Hello World example is also a good way to demonstrate a Quick Pick.

- We output our Hello World message in the command palette.
- The user has the option to select Answer or Cancel.
- Depending on the selection, the corresponding messages are displayed.

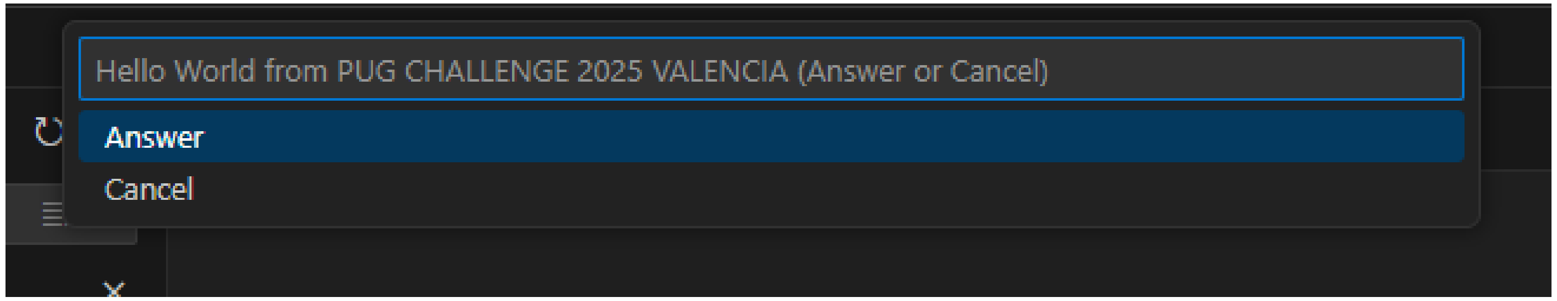


```
context.subscriptions.push(vscode.commands.registerCommand('pug-challenge-2025-valencia.helloWorld'

    // Options for a Quick Pick
    const options = ['Answer', 'Cancel'];

    // Show Quick Pick
    const selected = await vscode.window.showQuickPick(options, {
        placeHolder: 'Hello World from PUG CHALLENGE 2025 VALENCIA (Answer or Cancel)',
    });

    // Selection actions
    if (selected === 'Answer') {
        vscode.window.showInformationMessage('PUG Challenge User 2025 VALENCIA answered with Hello
    } else if (selected === 'Cancel') {
        vscode.window.showErrorMessage('PUG Challenge User 2025 VALENCIA canceled!');
    } else {
        vscode.window.showWarningMessage('No selection made.');
    }
```

Important Common Capabilities

Quick Picks

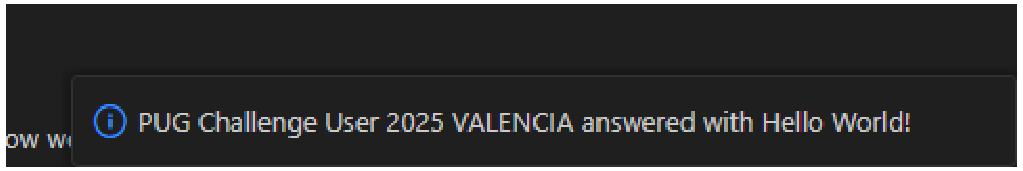
Let's take a look
at this in practice:



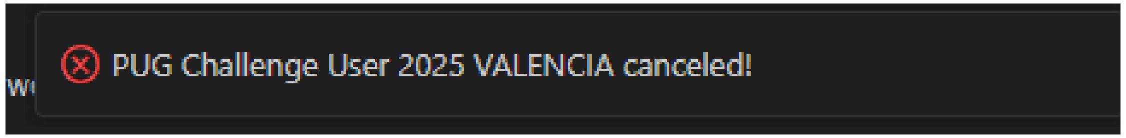
Hello World from PUG CHALLENGE 2025 VALENCIA (Answer or Cancel)

Answer

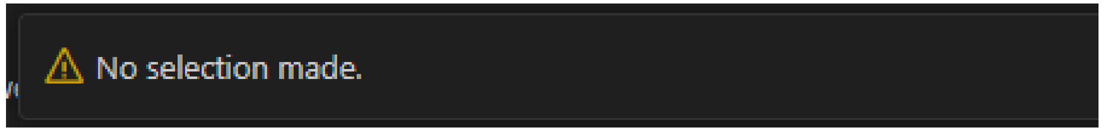
Cancel



PUG Challenge User 2025 VALENCIA answered with Hello World!



PUG Challenge User 2025 VALENCIA canceled!

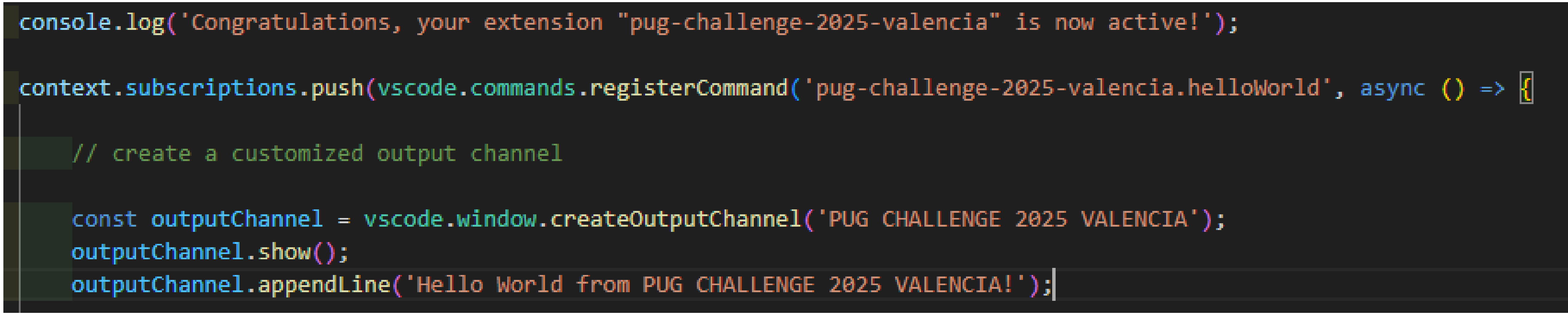


No selection made.

Important Common Capabilities

Customized output channels

vscode.window.createOutputChannel

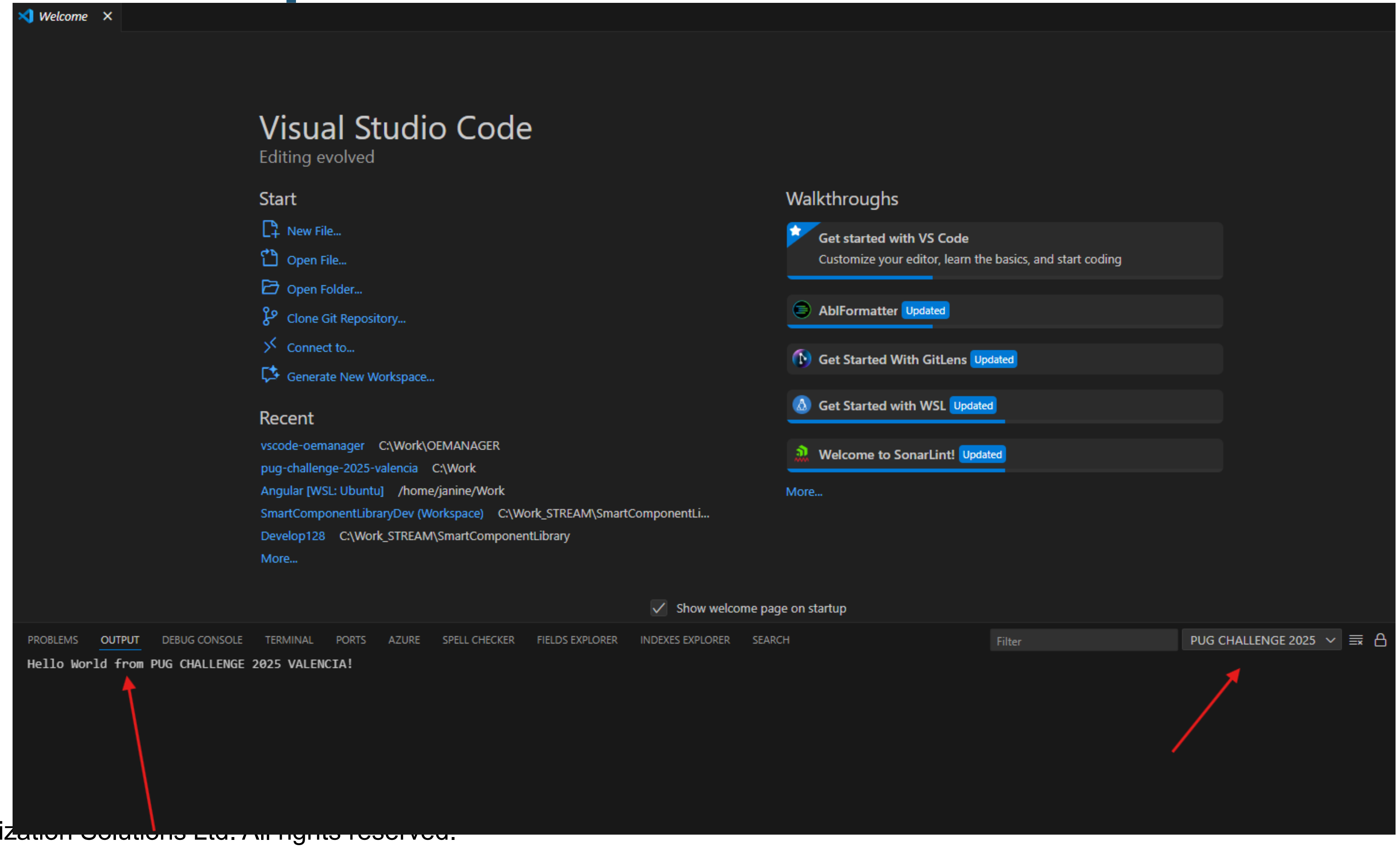


```
console.log('Congratulations, your extension "pug-challenge-2025-valencia" is now active!');

context.subscriptions.push(vscode.commands.registerCommand('pug-challenge-2025-valencia.helloWorld', async () => {

    // create a customized output channel

    const outputChannel = vscode.window.createOutputChannel('PUG CHALLENGE 2025 VALENCIA');
    outputChannel.show();
    outputChannel.appendLine('Hello World from PUG CHALLENGE 2025 VALENCIA!');
```

Important Common Capabilities

Customized output channels

In Practice:



Welcome

Visual Studio Code
Editing evolved

Start
- New File...
- Open File...
- Open Folder...
- Clone Git Repository...
- Connect to...
- Generate New Workspace...

Recent

vscode-oemanager	C:\Work\OEMANAGER
pug-challenge-2025-valencia	C:\Work
Angular [WSL: Ubuntu]	/home/janine/Work
SmartComponentLibraryDev (Workspace)	C:\Work_STREAM\SmartComponentLi...
Develop128	C:\Work_STREAM\SmartComponentLibrary

More...

Walkthroughs
- Get started with VS Code — Customize your editor, learn the basics, and start coding
- AblFormatter Updated
- Get Started With GitLens Updated
- Get Started with WSL Updated
- Welcome to SonarLint! Updated

More...

Show welcome page on startup

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS AZURE SPELL CHECKER FIELDS EXPLORER INDEXES EXPLORER SEARCH — Filter — PUG CHALLENGE 2025

```
Hello World from PUG CHALLENGE 2025 VALENCIA!
```

Consultingwerk
software architecture and development

Agenda

- Sample PASOE Manager Extension
- VS Code Marketplace and alternatives
- VS Code as an IDE
- Microsoft SDK for VS Code Extensions
- Building an Extension with Visual Studio Code
- Programming languages and file types used in a VS Code extension
- Event Callbacks
- Implementing the PASOE Manager Extension
- Package and publish your extension
- Important Common Capabilities
- Questions

Questions

Consultingwerk

software architecture and development