HW4 Tool Prototype

Intelligent @types Installer

Ram Chandra Ramaraju- G01384424
Dubba Srikanth Reddy- G01353043
Sai Rohith Pasham- G01348426
Karthik Doguparthi - G01383990

Intelligent @types Installer - Enhancing TypeScript Development in Visual Studio Code

Describes the overall challenge the tool is designed to address

By automatically detecting and installing missing type dependencies and presenting visual feedback within the VS Code editor, the Intelligent @types for TypeScript Code in VS Code aims to solve the difficulty of enhancing the development experience for TypeScript projects. The manual procedure that developers must go through to manage TypeScript type dependencies and make sure the required types are installed in their projects for improved code quality and tooling support is the source of the challenge.

Offers a clear and well-reasoned explanation of how the implemented tool features address the challenge

Features of the Implemented Tool:

1. Installing Missing Type Dependencies Automatically:
    - The tool looks for dependencies without matching type definitions in the “package.json” file.
    - Use the ‘@types’ convention to install the missing type dependencies automatically.

2. Visual Feedback for Missing Types:
    - Indicates issues in the editor by highlighting missing type dependencies in the “package.json” file.
    - Highlights the names of the missing packages and offers helpful tooltips.

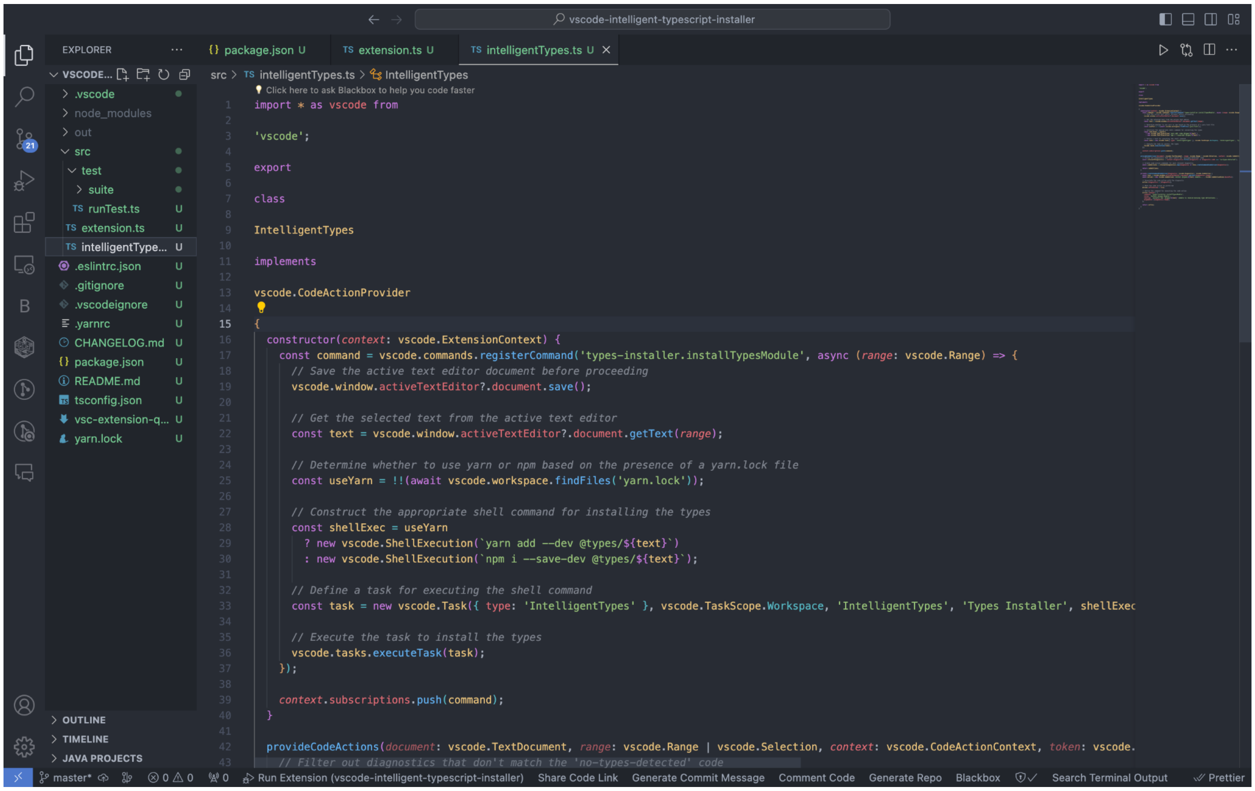
3. Code Action Provider:
    - Incorporates a Code Action Provider to recommend and implement solutions for type dependencies that are lacking.
    - Provides easy ways to install required types right from the editor.

Illustrates the use of the tool through one or more detailed scenarios illustrated with screenshots or a video

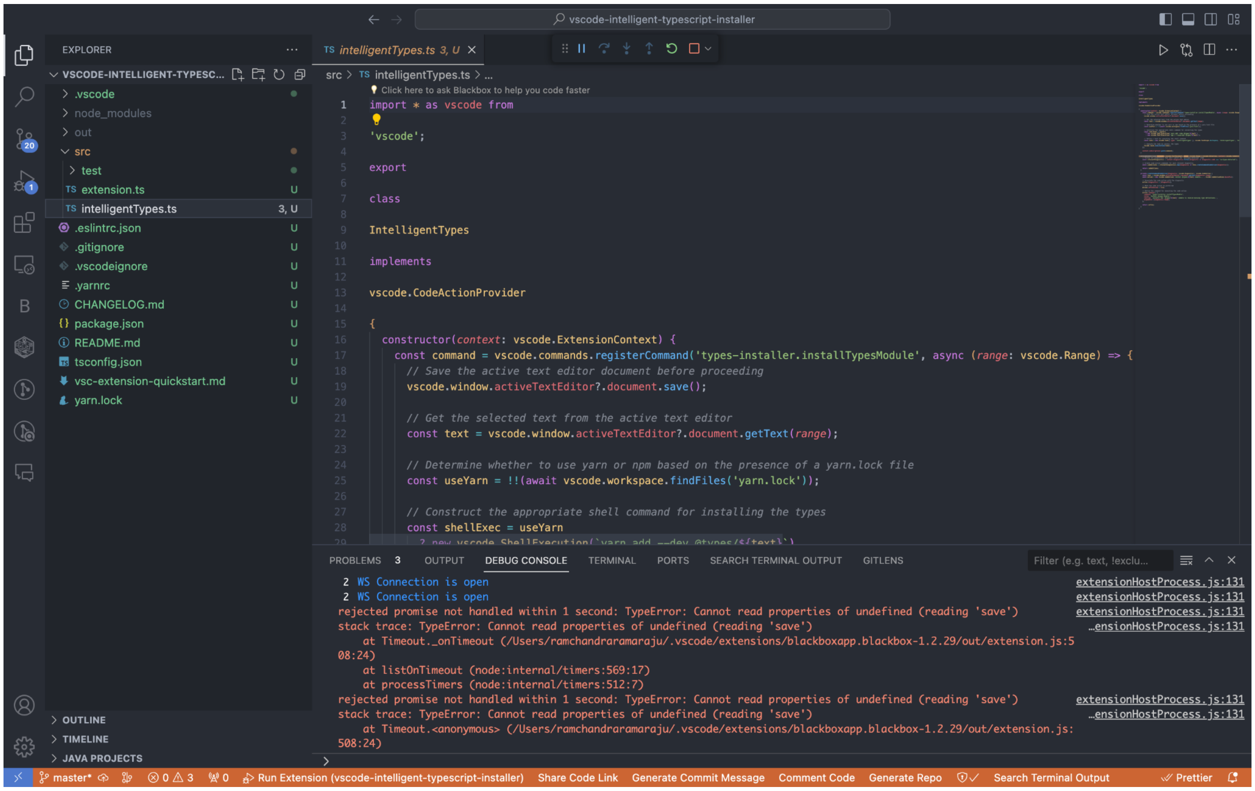
Scenario: Finding and Installing Type Dependencies That Are Missing

1. In VS Code, the developer creates a TypeScript project.
2. Dependencies without matching type definitions can be found in the "package.json" file.
3. The tool finds missing types and shows them in the editor as issues.
4. When the developer moves their cursor over the underlined package name, a tooltip containing details about the absent type appears.
5. The developer triggers the code action provider to install the missing type automatically.
6. The utility installs the necessary type dependency and modifies the "package.json" file.
7. The installation was successful, as indicated by the visual feedback, and the developer can now take advantage of enhanced TypeScript tools support.

Screenshots:

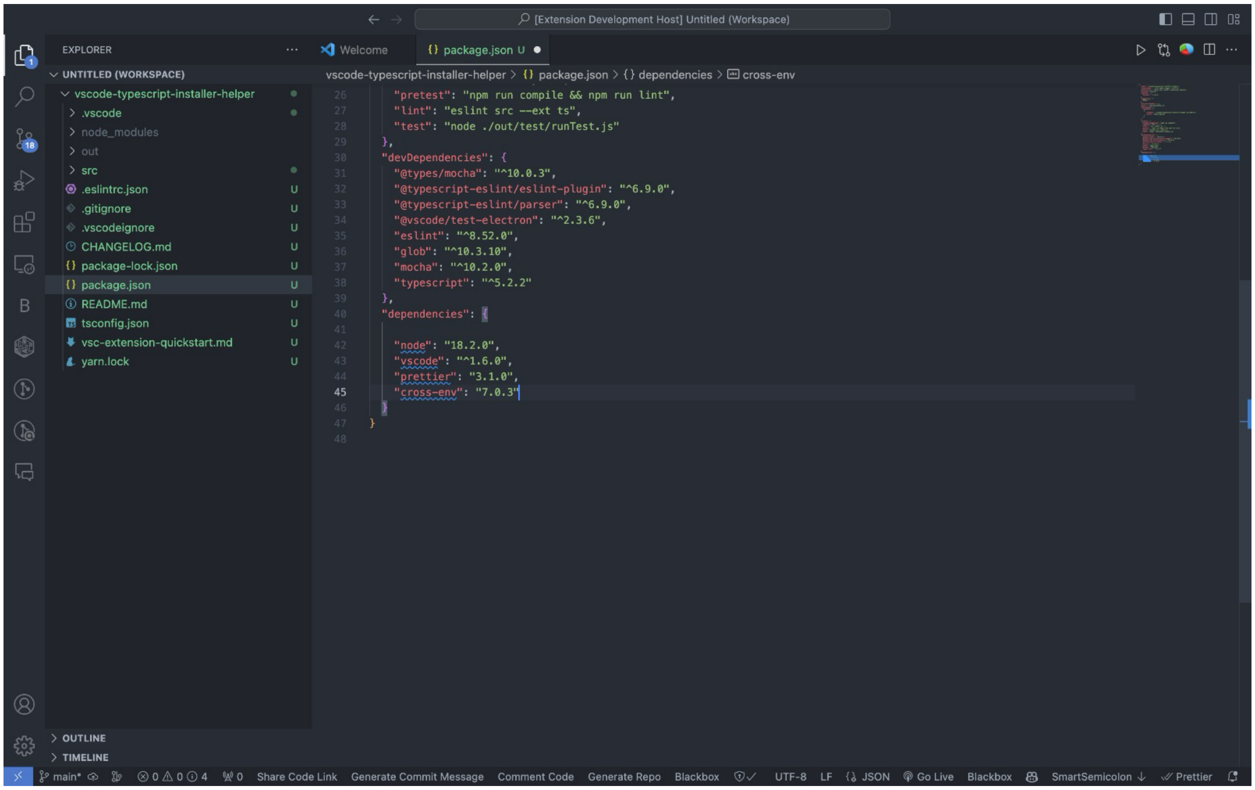


```
import * as vscode from

'vscode';

export

class

IntelligentTypes

implements

vscode.CodeActionProvider

{
  constructor(context: vscode.ExtensionContext) {
    const command = vscode.commands.registerCommand('types-installer.installTypesModule', async (range: vscode.Range) => {
      // Save the active text editor document before proceeding
      vscode.window.activeTextEditor?.document.save();

      // Get the selected text from the active text editor
      const text = vscode.window.activeTextEditor?.document.getText(range);

      // Determine whether to use yarn or npm based on the presence of a yarn.lock file
      const useYarn = !!(await vscode.workspace.findFiles('yarn.lock'));

      // Construct the appropriate shell command for installing the types
      const shellExec = useYarn
        ? new vscode.ShellExecution(`yarn add --dev @types/${text}`)
        : new vscode.ShellExecution(`npm i --save-dev @types/${text}`);

      // Define a task for executing the shell command
      const task = new vscode.Task({ type: 'IntelligentTypes' }, vscode.TaskScope.Workspace, 'IntelligentTypes', 'Types Installer', shellExec

      // Execute the task to install the types
      vscode.tasks.executeTask(task);
    });

    context.subscriptions.push(command);
  }

  provideCodeActions(document: vscode.TextDocument, range: vscode.Range | vscode.Selection, context: vscode.CodeActionContext, token: vscode.
    // Filter out diagnostics that don't match the 'no-types-detected' code
```

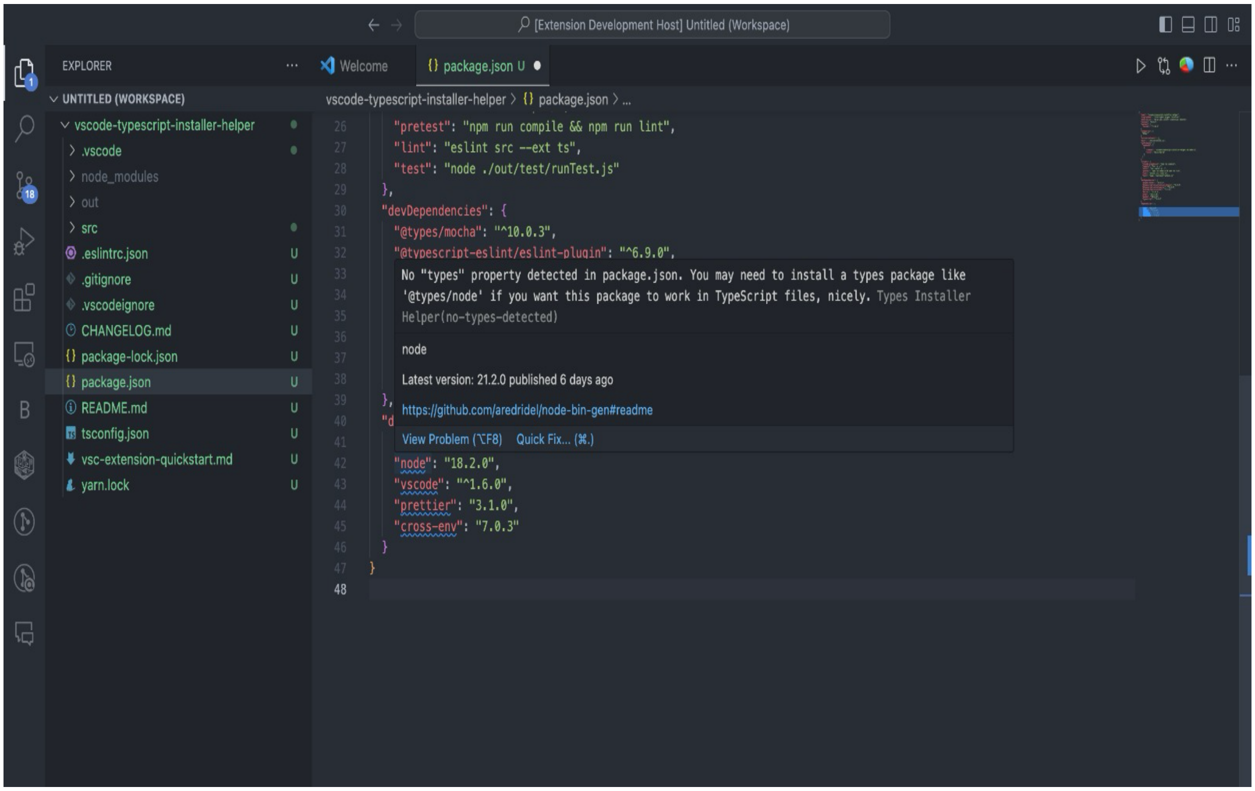


```
import * as vscode from

'vscode';

export

class

IntelligentTypes

implements

vscode.CodeActionProvider

{
  constructor(context: vscode.ExtensionContext) {
    const command = vscode.commands.registerCommand('types-installer.installTypesModule', async (range: vscode.Range) => {
      // Save the active text editor document before proceeding
      vscode.window.activeTextEditor?.document.save();

      // Get the selected text from the active text editor
      const text = vscode.window.activeTextEditor?.document.getText(range);

      // Determine whether to use yarn or npm based on the presence of a yarn.lock file
      const useYarn = !!(await vscode.workspace.findFiles('yarn.lock'));

      // Construct the appropriate shell command for installing the types
      const shellExec = useYarn
        ? new vscode.ShellExecution(`yarn add --dev @types/${text}`)
```

```
 2 WS Connection is open                                                                     extensionHostProcess.js:131
 2 WS Connection is open                                                                     extensionHostProcess.js:131
rejected promise not handled within 1 second: TypeError: Cannot read properties of undefined (reading 'save')   extensionHostProcess.js:131
stack trace: TypeError: Cannot read properties of undefined (reading 'save')                 …ensionHostProcess.js:131
    at Timeout._onTimeout (/Users/ramchandraramaraju/.vscode/extensions/blackboxapp.blackbox-1.2.29/out/extension.js:5
08:24)
    at listOnTimeout (node:internal/timers:569:17)
    at processTimers (node:internal/timers:512:7)
rejected promise not handled within 1 second: TypeError: Cannot read properties of undefined (reading 'save')   extensionHostProcess.js:131
stack trace: TypeError: Cannot read properties of undefined (reading 'save')                 …ensionHostProcess.js:131
    at Timeout.<anonymous> (/Users/ramchandraramaraju/.vscode/extensions/blackboxapp.blackbox-1.2.29/out/extension.js:
508:24)
```


[Extension Development Host] Untitled (Workspace)
EXPLORER
UNTITLED (WORKSPACE)
vscode-typescript-installer-helper
.vscode
node_modules
out
src
.eslintrc.json U
.gitignore U
.vscodeignore U
CHANGELOG.md U
package-lock.json U
package.json U
README.md U
tsconfig.json U
vsc-extension-quickstart.md U
yarn.lock U
Welcome
package.json U
vscode-typescript-installer-helper > package.json > dependencies > cross-env
OUTLINE
TIMELINE
main* ⊗ 0 ⚠ 0 ⓘ 4 0 Share Code Link Generate Commit Message Comment Code Generate Repo Blackbox UTF-8 LF JSON Go Live Blackbox SmartSemicolon Prettier

```json
    "pretest": "npm run compile && npm run lint",
    "lint": "eslint src --ext ts",
    "test": "node ./out/test/runTest.js"
  },
  "devDependencies": {
    "@types/mocha": "^10.0.3",
    "@typescript-eslint/eslint-plugin": "^6.9.0",
    "@typescript-eslint/parser": "^6.9.0",
    "@vscode/test-electron": "^2.3.6",
    "eslint": "^8.52.0",
    "glob": "^10.3.10",
    "mocha": "^10.2.0",
    "typescript": "^5.2.2"
  },
  "dependencies": {

    "node": "18.2.0",
    "vscode": "^1.6.0",
    "prettier": "3.1.0",
    "cross-env": "7.0.3"
  }
}
```



```json
    "pretest": "npm run compile && npm run lint",
    "lint": "eslint src --ext ts",
    "test": "node ./out/test/runTest.js"
  },
  "devDependencies": {
    "@types/mocha": "^10.0.3",
    "@typescript-eslint/eslint-plugin": "^6.9.0",
```

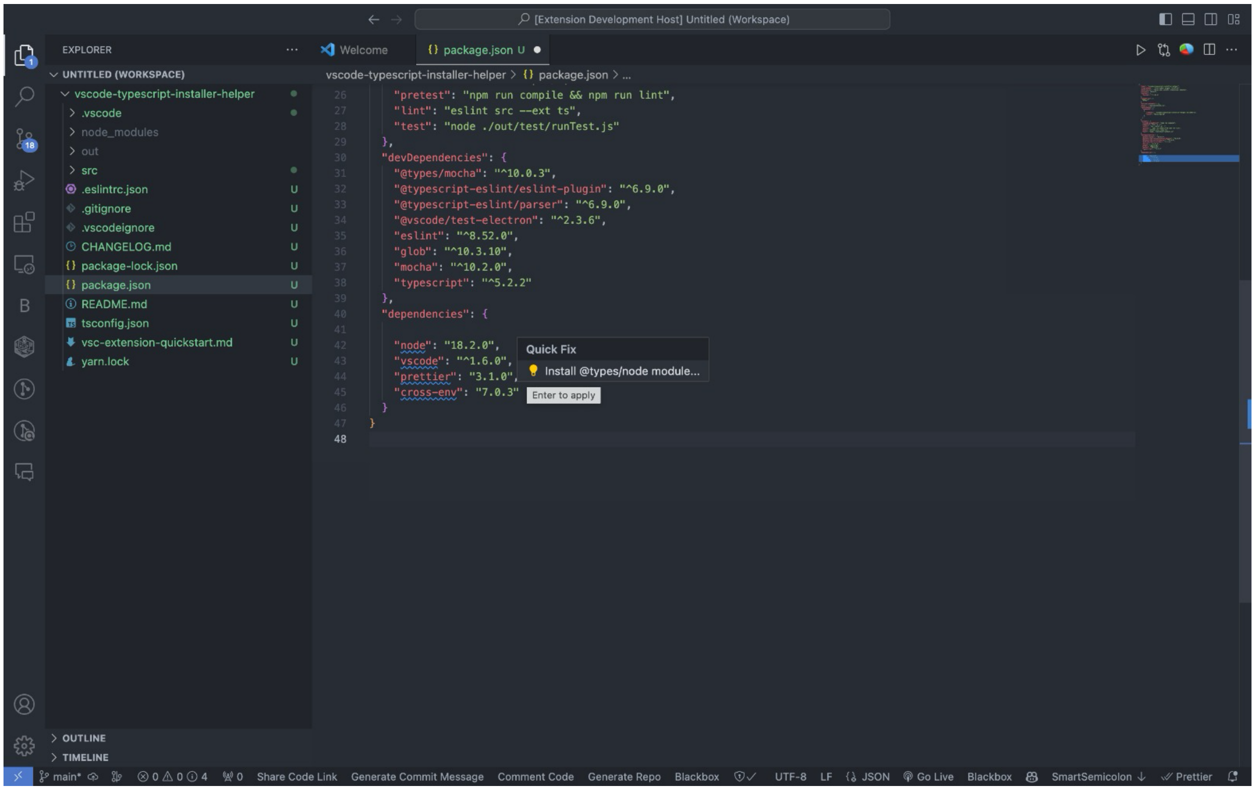
```
No "types" property detected in package.json. You may need to install a types package like
'@types/node' if you want this package to work in TypeScript files, nicely. Types Installer
Helper(no-types-detected)
```

node

Latest version: 21.2.0 published 6 days ago

https://github.com/aredridel/node-bin-gen#readme

```json
  },
  "d
    "node": "18.2.0",
    "vscode": "^1.6.0",
    "prettier": "3.1.0",
    "cross-env": "7.0.3"
  }
}
```

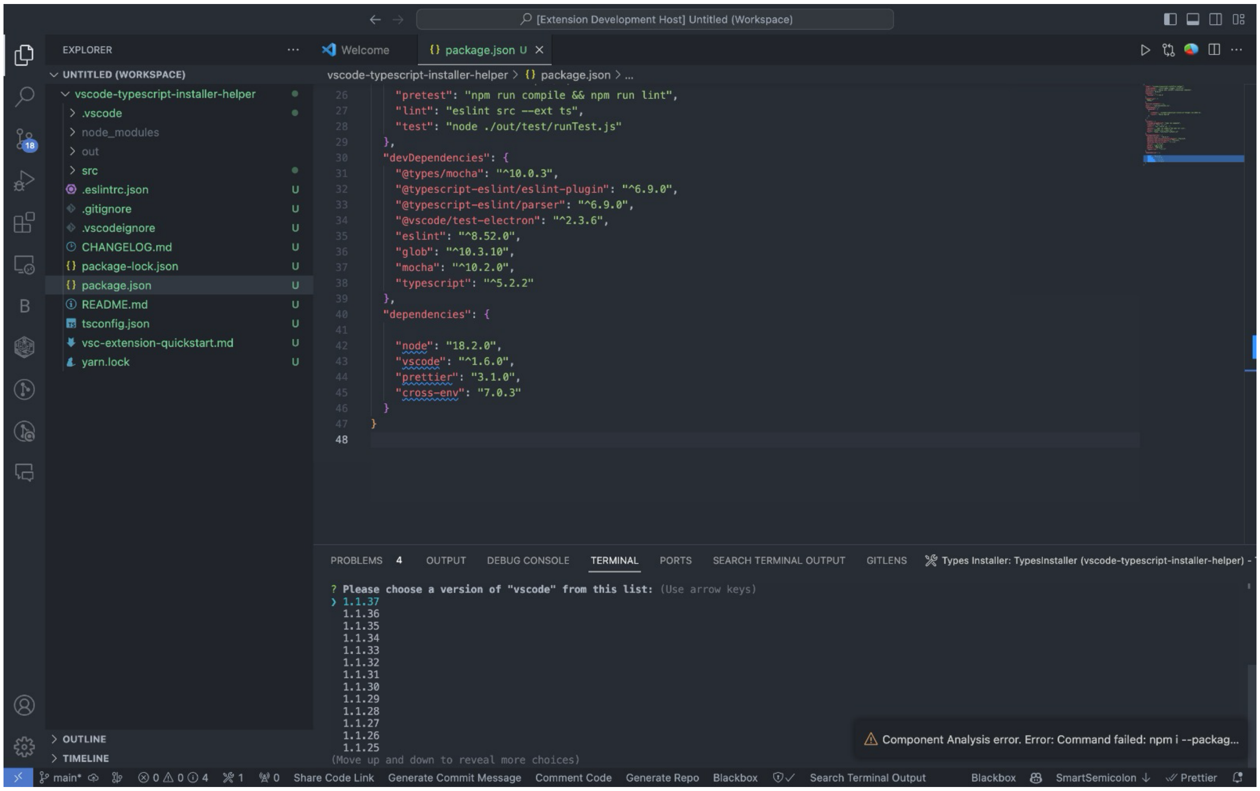

[Extension Development Host] Untitled (Workspace)
EXPLORER
UNTITLED (WORKSPACE)
vscode-typescript-installer-helper
.vscode
node_modules
out
src
.eslintrc.json U
.gitignore U
.vscodeignore U
CHANGELOG.md U
package-lock.json U
package.json U
README.md U
tsconfig.json U
vsc-extension-quickstart.md U
yarn.lock U
OUTLINE
TIMELINE
Welcome
package.json U
vscode-typescript-installer-helper > {} package.json > ...
main* ⊗ 0 ⚠ 0 ⓘ 4 0 Share Code Link Generate Commit Message Comment Code Generate Repo Blackbox UTF-8 LF {} JSON Go Live Blackbox SmartSemicolon Prettier

```
    "pretest": "npm run compile && npm run lint",
    "lint": "eslint src --ext ts",
    "test": "node ./out/test/runTest.js"
  },
  "devDependencies": {
    "@types/mocha": "^10.0.3",
    "@typescript-eslint/eslint-plugin": "^6.9.0",
    "@typescript-eslint/parser": "^6.9.0",
    "@vscode/test-electron": "^2.3.6",
    "eslint": "^8.52.0",
    "glob": "^10.3.10",
    "mocha": "^10.2.0",
    "typescript": "^5.2.2"
  },
  "dependencies": {

    "node": "18.2.0",
    "vscode": "^1.6.0",
    "prettier": "3.1.0",
    "cross-env": "7.0.3"
  }
}
```

Quick Fix

Install @types/node module...

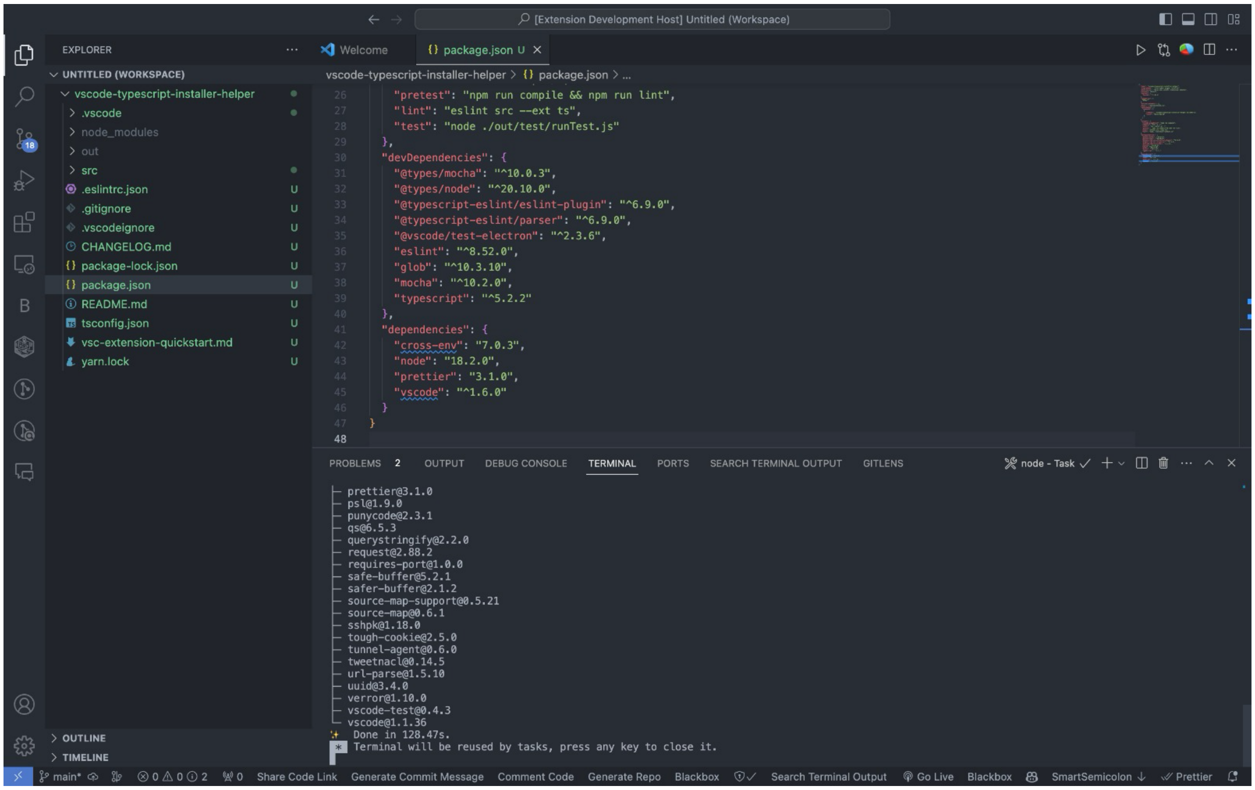
Enter to apply


[Extension Development Host] Untitled (Workspace)
EXPLORER
UNTITLED (WORKSPACE)
vscode-typescript-installer-helper
.vscode
node_modules
out
src
.eslintrc.json U
.gitignore U
.vscodeignore U
CHANGELOG.md U
package-lock.json U
package.json U
README.md U
tsconfig.json U
vsc-extension-quickstart.md U
yarn.lock U
OUTLINE
TIMELINE
Welcome
package.json U
vscode-typescript-installer-helper > {} package.json > ...
PROBLEMS 4
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
SEARCH TERMINAL OUTPUT
GITLENS
Types Installer: TypesInstaller (vscode-typescript-installer-helper) -
main* ⊗ 0 ⚠ 0 ⓘ 4 1 0 Share Code Link Generate Commit Message Comment Code Generate Repo Blackbox Search Terminal Output Blackbox SmartSemicolon Prettier

```
    "pretest": "npm run compile && npm run lint",
    "lint": "eslint src --ext ts",
    "test": "node ./out/test/runTest.js"
  },
  "devDependencies": {
    "@types/mocha": "^10.0.3",
    "@typescript-eslint/eslint-plugin": "^6.9.0",
    "@typescript-eslint/parser": "^6.9.0",
    "@vscode/test-electron": "^2.3.6",
    "eslint": "^8.52.0",
    "glob": "^10.3.10",
    "mocha": "^10.2.0",
    "typescript": "^5.2.2"
  },
  "dependencies": {

    "node": "18.2.0",
    "vscode": "^1.6.0",
    "prettier": "3.1.0",
    "cross-env": "7.0.3"
  }
}
```

```
? Please choose a version of "vscode" from this list: (Use arrow keys)
❯ 1.1.37
  1.1.36
  1.1.35
  1.1.34
  1.1.33
  1.1.32
  1.1.31
  1.1.30
  1.1.29
  1.1.28
  1.1.27
  1.1.26
  1.1.25
(Move up and down to reveal more choices)
```

⚠ Component Analysis error. Error: Command failed: npm i --packag...



```
    "pretest": "npm run compile && npm run lint",
    "lint": "eslint src --ext ts",
    "test": "node ./out/test/runTest.js"
  },
  "devDependencies": {
    "@types/mocha": "^10.0.3",
    "@types/node": "^20.10.0",
    "@typescript-eslint/eslint-plugin": "^6.9.0",
    "@typescript-eslint/parser": "^6.9.0",
    "@vscode/test-electron": "^2.3.6",
    "eslint": "^8.52.0",
    "glob": "^10.3.10",
    "mocha": "^10.2.0",
    "typescript": "^5.2.2"
  },
  "dependencies": {
    "cross-env": "7.0.3",
    "node": "18.2.0",
    "prettier": "3.1.0",
    "vscode": "^1.6.0"
  }
}
```

```
├─ prettier@3.1.0
├─ psl@1.9.0
├─ punycode@2.3.1
├─ qs@6.5.3
├─ querystringify@2.2.0
├─ request@2.88.2
├─ requires-port@1.0.0
├─ safe-buffer@5.2.1
├─ safer-buffer@2.1.2
├─ source-map-support@0.5.21
├─ source-map@0.6.1
├─ sshpk@1.18.0
├─ tough-cookie@2.5.0
├─ tunnel-agent@0.6.0
├─ tweetnacl@0.14.5
├─ url-parse@1.5.10
├─ uuid@3.4.0
├─ verror@1.10.0
├─ vscode-test@0.4.3
└─ vscode@1.1.36
✨  Done in 128.47s.
 *  Terminal will be reused by tasks, press any key to close it.
```

Presents results from the think-aloud usability study illustrating how developers made use of the tool and both positive and negative aspects of how it supported their development activity

DEMO of the Proposed prototype Tool

Open this link for demo : Intelligent @types Installer

Usability Study by Thinking Aloud

1. In a TypeScript project, my friends who are already working with Typescript are tasked with purposefully adding a dependency without the matching type definition.
2. To locate and install the missing type, participants utilize the Intelligent @types tool.
3. We recorded their ideas, activities, and opinions.
4. The simplicity of automatic installation, ease of use, and speedy identification of missing types are possible positive factors.
5. Unexpected behavior, difficulties encountered during the procedure, and uncertainty in the user interface are examples of negative features.

6. The usability study's findings offer information about the tool's effectiveness and accessibility.

Results and Feedback:

Positive Feedback:

- The automatic detection of missing types is appreciated by developers.
- Automated installation and quick fixes simplify the development process.
- Visual feedback facilitates problem understanding and resolution.

Negative Feedback:

- Possible problems with incompatible versions or strange behavior that arises during automatic installations.

Visual Studio Code's Intelligent @types for TypeScript Code is a helpful tool for managing type dependencies in TypeScript applications. Through scenarios and user studies, it demonstrates how it enhances developer productivity and provides a seamless experience for managing TypeScript-type dependencies within the VS Code editor. The continuous development of the instrument to increase its effectiveness and usability can be influenced by user feedback.