
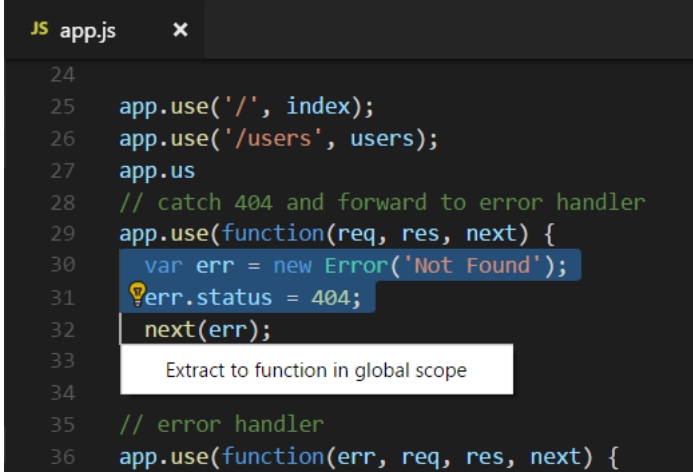


TOPICS Refactoring ▼

重构

 (<https://github.com/Microsoft/vscode-docs/blob/master/docs/editor/refactoring.md>)

(https://en.wikipedia.org/wiki/Code_refactoring)通过重构代码而不修改运行时行为，源代码重构 (https://en.wikipedia.org/wiki/Code_refactoring)可以提高项目的质量和可维护性。Visual Studio Code支持诸如“提取方法” (<https://refactoring.com/catalog/extractMethod.html>)和“提取变量” (<https://refactoring.com/catalog/extractVariable.html>)之类的重构操作（重构），以从编辑器内部改善代码库。



```
24
25 app.use('/', index);
26 app.use('/users', users);
27 app.use
28 // catch 404 and forward to error handler
29 app.use(function(req, res, next) {
30   var err = new Error('Not Found');
31   err.status = 404;
32   next(err);
33
34 // error handler
35 app.use(function(err, req, res, next) {
```

例如，用于避免代码重复的常见重构（麻烦的维护工作）是Extract Method (<https://refactoring.com/catalog/extractMethod.html>)重构，您可以在其中选择要在其他地方重用的源代码，并将其提取到自己的共享方法中。

重构由语言服务提供，并且VS Code通过TypeScript (<https://www.typescriptlang.org/>)语言服务对TypeScript和JavaScript重构具有内置支持。通过提供语言服务的VS Code扩展 (/docs/editor/extension-gallery)提供了对其他编程语言的重构支持。跨语言的UI和重构命令相同，在本主题中，我们将演示TypeScript语言服务对重构的支持。

代码操作=快速修复和重构

在VS Code中，Code Actions可以为检测到的问题提供重构和快速修复（以绿色花体突出显示）。当光标位于花形或选定的文本区域上时，源代码附近的灯泡将宣布可用的代码操作。单击“代码操作”灯泡或使用**快速修复**命令 Ctrl + 。 将显示快速修复和重构。

如果您只想查看没有快速修复的**重构**，则可以使用“**重构**”命令（Ctrl + Shift + R）。

注意：如果您不想在编辑器中看到Code Action灯泡，则可以使用 `editor.lightbulb.enable` 设置 (/docs/getstarted/settings)禁用灯泡。您仍然可以通过“**快速修复**”命令和 Ctrl + 打开“快速修复”。 键盘快捷键。

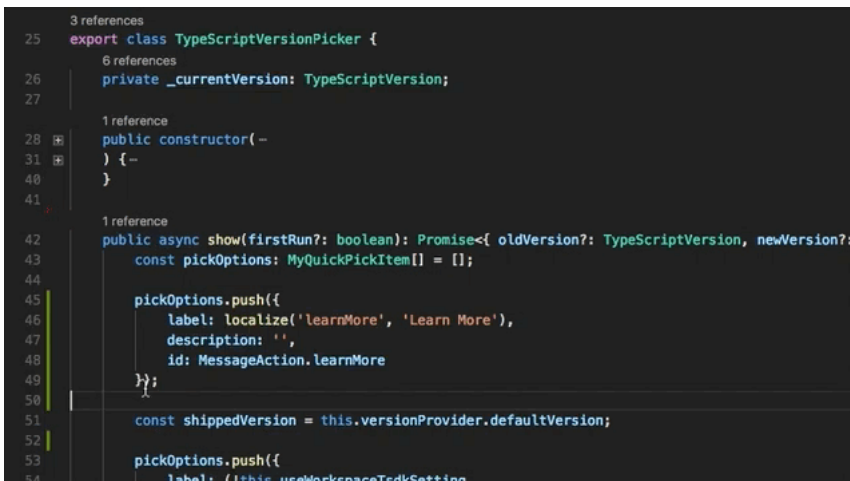
重构动作

提取方法

选择要提取的源代码，然后单击装订线中的灯泡或按（Ctrl + 。）以查看可用的重构。可以将源代码片段提取到新方法中，也可以提取到各种不同范围的新函数中。在提取重构期间，系统将提示您提供一个有意义的名称。

提取变量

TypeScript语言服务提供了Extract到const 重构，以为当前选择的表达式创建一个新的本地变量：



```
3 references
25 export class TypeScriptVersionPicker {
26   6 references
27   private _currentVersion: TypeScriptVersion;
28   1 reference
29   public constructor(-
30   ) {-
31   }
32   }
33   1 reference
34   public async show(firstRun?: boolean): Promise<{ oldVersion?: TypeScriptVersion, newVersion?:
35   const pickOptions: MyQuickPickItem[] = [];
36   44
37   pickOptions.push({
38     label: localize('learnMore', 'Learn More'),
39     description: '',
40     id: MessageAction.learnMore
41   });
42   50
43   const shippedVersion = this.versionProvider.defaultVersion;
44   52
45   pickOptions.push({
46     label: (!this.useWorkspaceTsdkSetting
```

使用类时，还可以将值提取到新属性。

重命名符号

重命名是与重构源代码相关的常见操作，而VS Code具有单独的“**重命名符号**”命令（F2）。某些语言支持跨文件重命名符号。按 F2，然后输入新的所需名称，然后按 Enter。符号的所有用法将在文件中重命名。



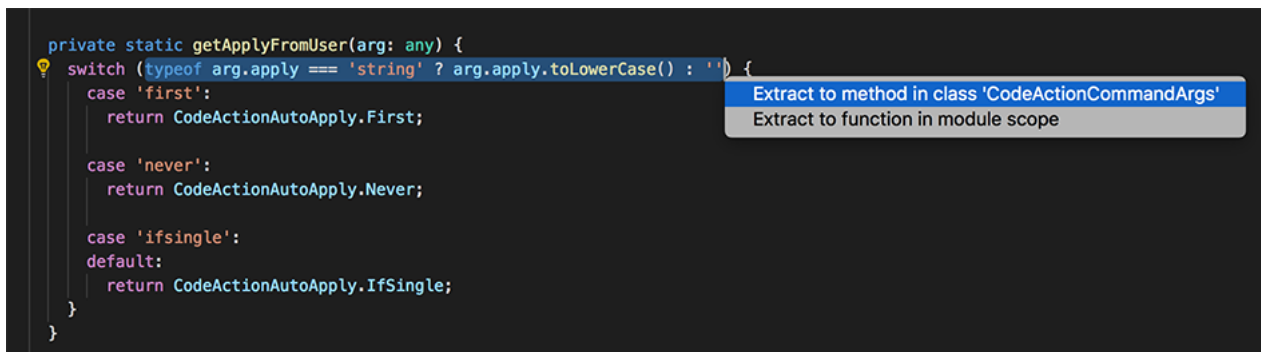
代码操作的键绑定

该 `editor.action.codeAction` 命令使您可以为特定的代码操作配置键绑定。例如，此键绑定触发 **Extract函数** 重构代码操作：

```
{
  "key": "ctrl+shift+r ctrl+e",
  "command": "editor.action.codeAction",
  "args": {
    "kind": "refactor.extract.function"
  }
}
```

扩展使用增强的 `CodeActionProvider` API 指定了代码操作类型。种类是分层的，因此 `"kind": "refactor"` 将显示所有重构的代码操作，而 `"kind": "refactor.extract.function"` 仅显示**提取函数**重构。

使用上面的键绑定，如果只有一个 `"refactor.extract.function"` Code Action可用，它将被自动应用。如果有多个“**提取**”功能的“代码操作”可用，我们将打开一个上下文菜单以选择它们：



您还可以使用 `apply` 参数来控制如何/何时自动执行代码操作：

```
{
  "key": "ctrl+shift+r ctrl+e",
  "command": "editor.action.codeAction",
  "args": {
    "kind": "refactor.extract.function",
    "apply": "first"
  }
}
```

有效值 `"apply"`：

- `"first"` -始终自动应用第一个可用的代码操作。
- `"ifSingle"` -默认。如果只有一个可用，则自动应用代码操作。否则，显示上下文菜单。
- `"never"` -即使只有一个“代码操作”可用，也始终显示“代码操作”上下文菜单。

当使用来配置代码操作键绑定时 `"preferred": true`，仅显示首选的快速修复和重构。首选的快速修复解决了潜在的错误，而首选的重构是最常见的重构选择。例如，虽然 `refactor.extract.constant` 可能存在多个重构，每个重构都提取到文件中的不同范围，但首选的 `refactor.extract.constant` 重构是提取到局部变量的重构。

此键绑定用于 `"preferred": true` 创建一个重构，该重构始终尝试将所选源代码提取为本地范围内的常量：

```
{
  "key": "shift+ctrl+e",
  "command": "editor.action.codeAction",
  "args": {
    "kind": "refactor.extract.constant",
    "preferred": true,
    "apply": "ifsingle"
  }
}
```

重构扩展

通过在VS Code Marketplace中 (<https://marketplace.visualstudio.com/vscode>) 查找，可以找到支持重构的扩展。您可以转到“扩展程序”视图（Ctrl + Shift + X），然后在搜索框中键入“refactor”。然后，您可以按安装数量或等级进行排序，以查看流行的扩展。



蟒蛇
ms-python
21.8百万
整理，调试（多线程，远程），集成...



语言支持...
红帽
620万
Java Linting, Intellisense, 格式化，重构...



PHP智能感知
费利克斯贝克尔
400万
先进的自动完成和重构支持



SCSS智能感知
麦克林克
388.9千
先进的自动完成和重构支持...

(<https://marketplace.visualstudio.com/items?itemName=ms-python.python>) (<https://marketplace.visualstudio.com/items?itemName=redhat.java>) (<https://marketplace.visualstudio.com/items?itemName=felixbecker.php-intellisense>) (<https://marketplace.visualstudio.com/items?itemName=mrmlnc.vscode-scss>)

提示：上面显示的扩展名是动态查询的。点击上方的扩展程序磁贴，以阅读说明和评论，以决定最适合您的扩展程序。

下一步

- 简介视频-代码编辑 (/docs/introvideos/codeediting) -观看有关代码编辑功能的入门视频。
- 代码导航 (/docs/editor/editingevolved) -VS Code使您可以快速浏览源代码。
- 调试 (/docs/editor/debugging) -了解有关使用VS Code进行调试的信息。

常见问题

如果我的代码有错误，为什么我看不到任何灯泡？
仅当光标悬停在显示错误的文本上时，才会显示灯泡（代码操作）。将鼠标悬停在文本上将显示错误说明，但是您需要移动光标或选择文本以查看用于快速修复和重构的灯泡。

该文档对您有帮助吗？

☒ 是

☐ 没有