

如何开发一款vscode插件

撸起袖子造轮子。

插件功能

vscode 赋予了插件 非常多的能力，让它们几乎可以在方方面面面对 vscode 进行武装升级，插件的主要能力包括：

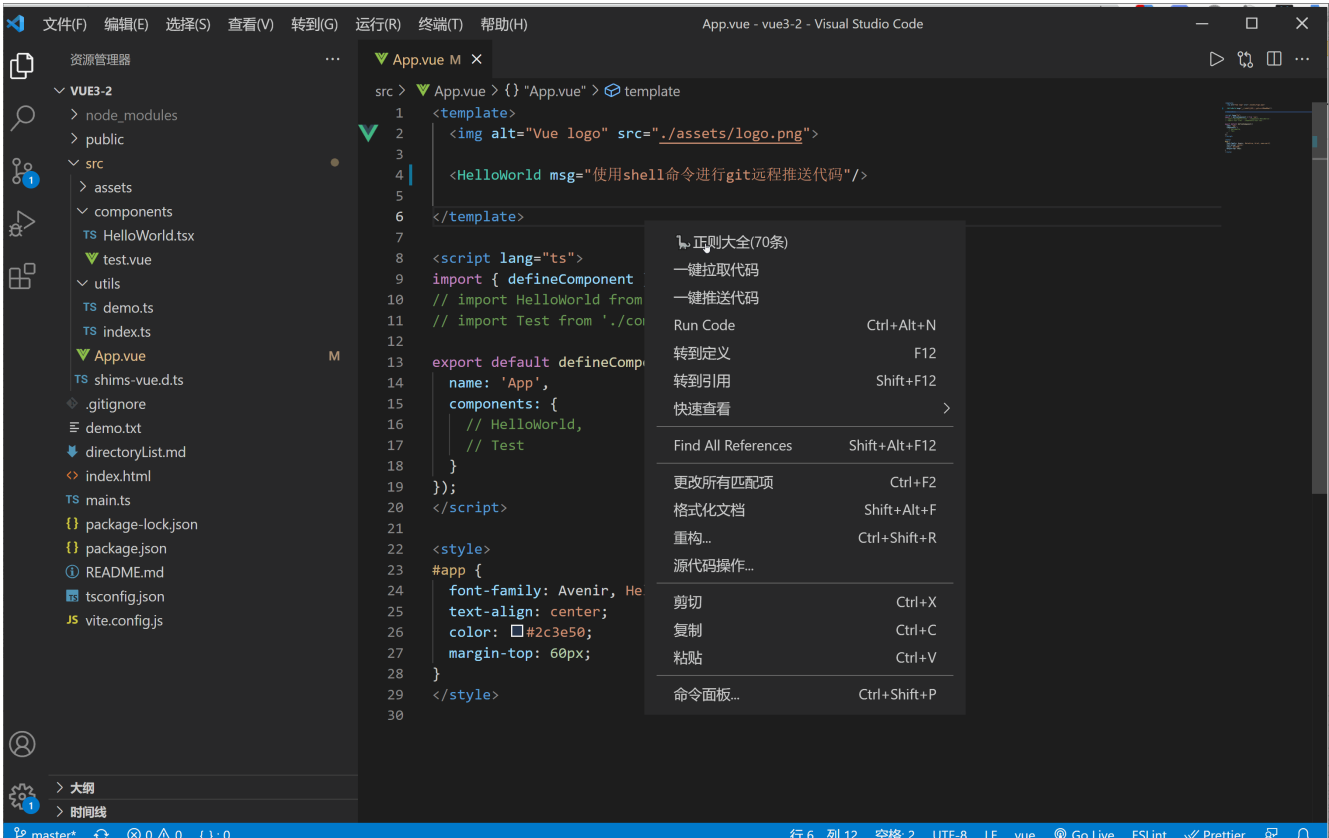
- 提供新主题（包括编辑器配色，文件图标等等）
- 支持新编程语言的语法（提供高亮，代码补全，错误检查等等功能）
- 提供程序调试功能
- 自定义命令，快捷键，菜单，输入框，选择框提示框
-

开发怎样的插件？

平时在vscode写完代码需要手动去git进行git add .、git commit...等等一系列操作，非常麻烦。

所以开发了一款vscode插件，快速进行推送代码和远程拉取代码，

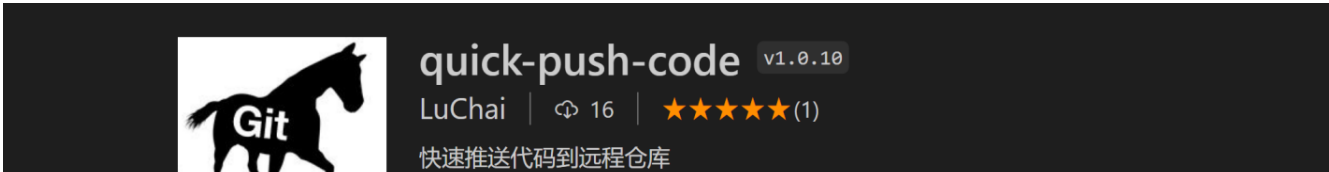
推送代码效果如下



插件功能：

- 推送代码：本地开发完代码进行一键推送代码到远程仓库，使用脚本进行执行shell进行推送代码，进行commit的时候要检查项目是否有git推送规范，如果commit不规范则推送失败，commit后会弹出代码是否进行打包后推送和直接推送进行选择。
- 拉取代码：执行拉取代码脚本命令会将本地代码进行暂存后进行拉取，拉取成功后释放暂存代码会提示用户代码是否有冲突

你可以在vscode插件市场搜索 quick-push-code进行体验：



环境准备

- nodejs，建议稳定版
- Git

首先安装 [yeoman](#) 脚手架工具，以及 vscode 官方提供的脚手架工具：

👉 `npm install -g yo generator-code`

接下来执行以下命令交互式创建插件项目：

👉 `yo code`

项目结构

插件项目已经创建完成后，项目目录结构如下：

Bash

```
1  .
2  |— node_modules
3  |— CHANGELOG.md
4  |— README.md
5  |— package.json
6  |— src
7  |   |— extension.ts
8  |   └─ test
9  |       |— runTest.ts
10 |       └─ suite
11 |           |— extension.test.ts
12 |           └─ index.ts
13 |— tsconfig.json
14 |— vsc-extension-quickstart.md
15 |— .vscode
16 |   |— extensions.json
17 |   |— launch.json
18 |   |— settings.json
19 |   └─ tasks.json
20 └─ yarn.lock
```

其中：

- `.vscode` 文件夹存放程序运行和调试相关命令
- `vsc-extension-quickstart.md` 文档介绍了插件的入门知识和运行方式，是项目创建后需要首先关注的点
- `src` 文件夹中存放插件源代码，需要重点关注这一块（`src/test` 存放插件 e2e 测试代码，暂时不用关注）

- `package.json` 除了包含常规 node 项目配置属性外，还包含插件配置属性，也需要重点关注
- `package.json` 主要部分展示，插件配置相关代码片段如下：
- `name`和`displayName`是插件的名字和显示名字（即在插件市场的名字）
 - `main` 插件程序入口
 - `activationEvents` 表示何时激活插件
 - `contributes` 表示如何定制插件的功能

```
1 {
2   "name": "quick-push-code",
3   "displayName": "quick-push-code",
4   "description": "快速推送代码到远程仓库",
5   "version": "1.0.10",
6   "repository": {
7     "private": true
8   },
9   "publisher": "LuChai",
10  "engines": {
11    "vscode": "^1.59.0"
12  },
13  "icon": "git.jpg",
14  "categories": [
15    "Other"
16  ],
17  "activationEvents": [
18    "onCommand:quick-push-code.helloWorld",
19    "onCommand:quick-push-code.test"
20  ],
21  "main": "./out/extension.js",
22  "contributes": {
23    "commands": [
24      {
25        "command": "quick-push-code.push",
26        "title": "一键推送代码"
27      },
28      {
29        "command": "quick-push-code.pull",
30        "title": "一键拉取代码"
31      }
32    ],
33    "menus": {
34      "editor/context": [
35        {
36          "when": "editorFocus",
37          "command": "quick-push-code.push",
38          "group": "navigation"
39        },
40        {
41          "when": "editorFocus",
42          "command": "quick-push-code.pull",
43          "group": "navigation"
44        }
45      ]
46    }
47  }
48 },
```

主入口文件

插件命令

快捷配置鼠标右键菜单

打开文件，右键可以看到配置的选项

```
<template>
  

  <HelloWorld msg="使用shell命令进行git远程推送代码"/>
</template>

<script lang="ts">
import { defineComponent } from 'vue';
// import HelloWorld from './components/HelloWorld';
// import Test from './components/test.vue';

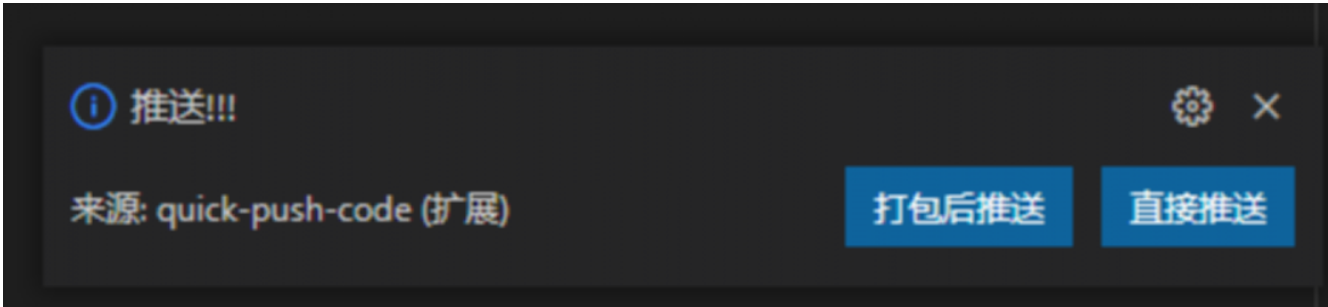
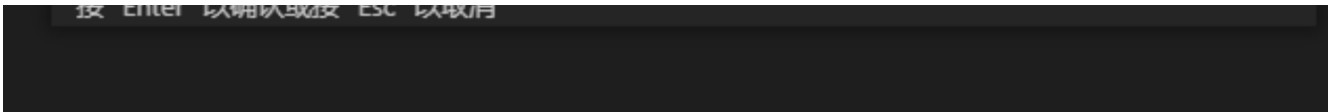
export default defineComponent({
  name: 'App',
  components: {
    // HelloWorld,
    // Test
  }
});
</script>

<style>
#app {
  font-family: Avenir, Helvetica, Arial, sans-serif;
  text-align: center;
  color: #2c3e50;
  margin-top: 60px;
}
</style>
```

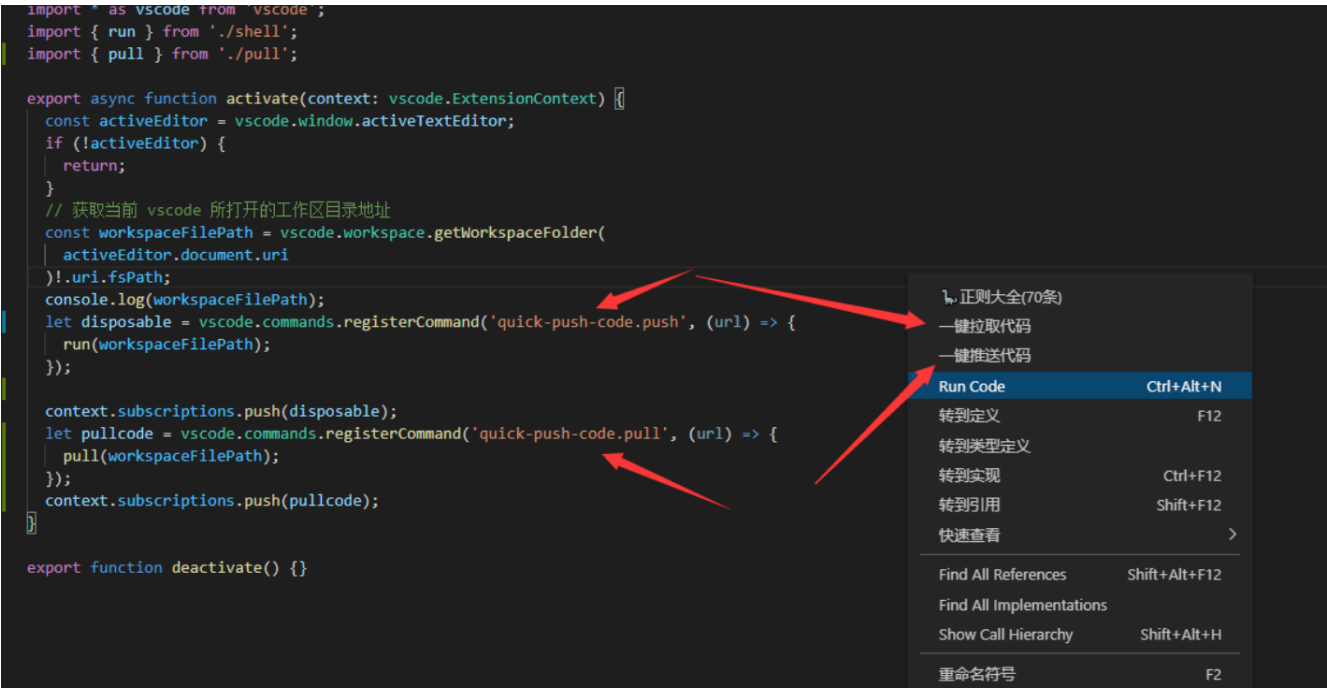
正则大全(70条)	
一键拉取代码	
一键推送代码	
Run Code	Ctrl+Alt+N
转到定义	F12
转到引用	Shift+F12
快速查看	>
Find All References	Shift+Alt+F12
更改所有匹配项	Ctrl+F2
格式化文档	Shift+Alt+F
重构...	Ctrl+Shift+R
源代码操作...	
剪切	Ctrl+X
复制	Ctrl+C
粘贴	Ctrl+V
命令面板...	Ctrl+Shift+P

执行结果：






src/extension.ts 中的内容：



extension.ts 暴露两个方法：

- `active`：在插件激活时运行，通常在其中注册自定义命令，主要使用这个
- `deactive`：在插件禁用时运行

插件注册

 `let disposable = vscode.commands.registerCommand('quick-push-code.push', (url) => {`
 // 执行你的脚本方法
});
`context.subscriptions.push(disposable);`

推送代码脚本代码展示

```
JavaScript

1
2
3 import * as vscode from 'vscode';
4 import * as colors from "colors";
5 import * as shell from "shelljs";
6 const nodePath = (shell.which('node').toString());
7 shell.config.execPath = nodePath;
8 shell.exec('git');
9
10
```

```
11 export const run = async (url:any) => {
12     // shell.config.execPath = url;
13     let commit: string | undefined = await
    vscode.window.showInputBox({placeholder: "请输入commit: "});
14
15     // 取消推送
16     if(typeof(commit) === "undefined") {return;}
17     vscode.window.showInformationMessage('推送!!!',...['打包后推送','直接推
    送']).then(res=> {
18         shell.cd(url);
19         let currentBranch = "";
20         let buildParmas = null;
21         if(!commit) {commit = "perf: update";}
22
23         // 去除空格
24         // commit = commit.replace(/\s*/g, "");
25         // 获取当前分支
26         const { stdout } = shell.exec('git symbolic-ref --short -q HEAD');
27         currentBranch = stdout;
28         console.log(colors.green(`当前分支为 ${currentBranch}`));
29         if(res === "打包后推送") {
30             console.log(colors.green('项目打包中，请稍等片刻~~~'));
31             buildParmas = new Promise((resolve,reject) => {
32                 const {code} = shell.exec(`npm run build`); // 打包失败不走catch
33                 // code = 0 成功
34                 if(code) {
35                     reject('打包失败');
36                 } else {
37                     resolve('打包成功');
38                 }
39             });
40         }
41
42
43         // 代码推送 code = 0 成功
44         const pushCode = () => {
45             shell.exec('git add .');
46
47
48             // shell中变量名是中间不能有空格的，有空格时，shell会把变量当成一个命令执行 解决
            办法添加双引号
49             const commitStatus = shell.exec(`git commit -m "${commit}"`);
50             console.log(commitStatus);
51             if(commitStatus.code) {return vscode.window.showWarningMessage('commit失
            败，请重新commit');}
52
53
54             const pullStatus = shell.exec(`git pull origin ${currentBranch}`);
55             if(pullStatus.code) {return vscode.window.showWarningMessage('拉取代码失
            败');}
56
57
58             try {
59                 console.log(colors.green(`尝试推送分支 ${currentBranch} 至远程仓库`));
60
61                 const { code } = shell.exec(`git push origin ${currentBranch}`);
62                 console.log(code);
63                 if(!code) {
64                     console.log(colors.green(`${currentBranch} 分支推送成功`));
65                     return vscode.window.showInformationMessage('推送成功!!!');
```

```
66         } else {
67             throw '';
68         }
69     } catch(error) {
70         console.log(colors.red(`推送分支失败: ${error}`));
71         return vscode.window.showWarningMessage('推送失败, 请检查代码是否有冲突后重新推送');
72     }
73 };
74 if (buildParmas) {
75     buildParmas.then(res => {
76         console.log(colors.green('打包成功'));
77         pushCode();
78     }).catch(err => {
79         return vscode.window.showInformationMessage('打包失败, 请重新打包失败');
80     });
81 } else {
82     pushCode();
83 }
84 });
85 };
86
87
```

发布插件

代码写完需要将插件进行本地打包测试

打包插件需要安装 `vsce` 库：

```
CoffeeScript

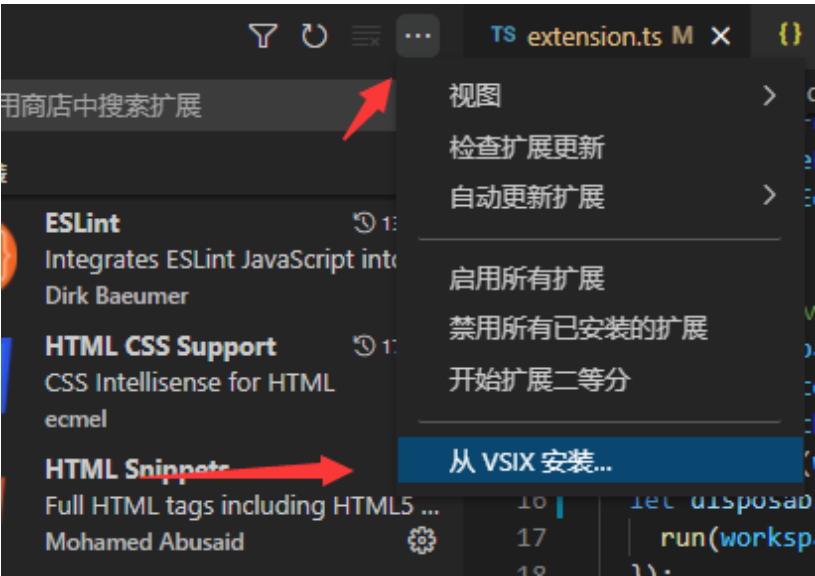
1  npm i -g vsce
```

然后执行打包命令：

```
Ada

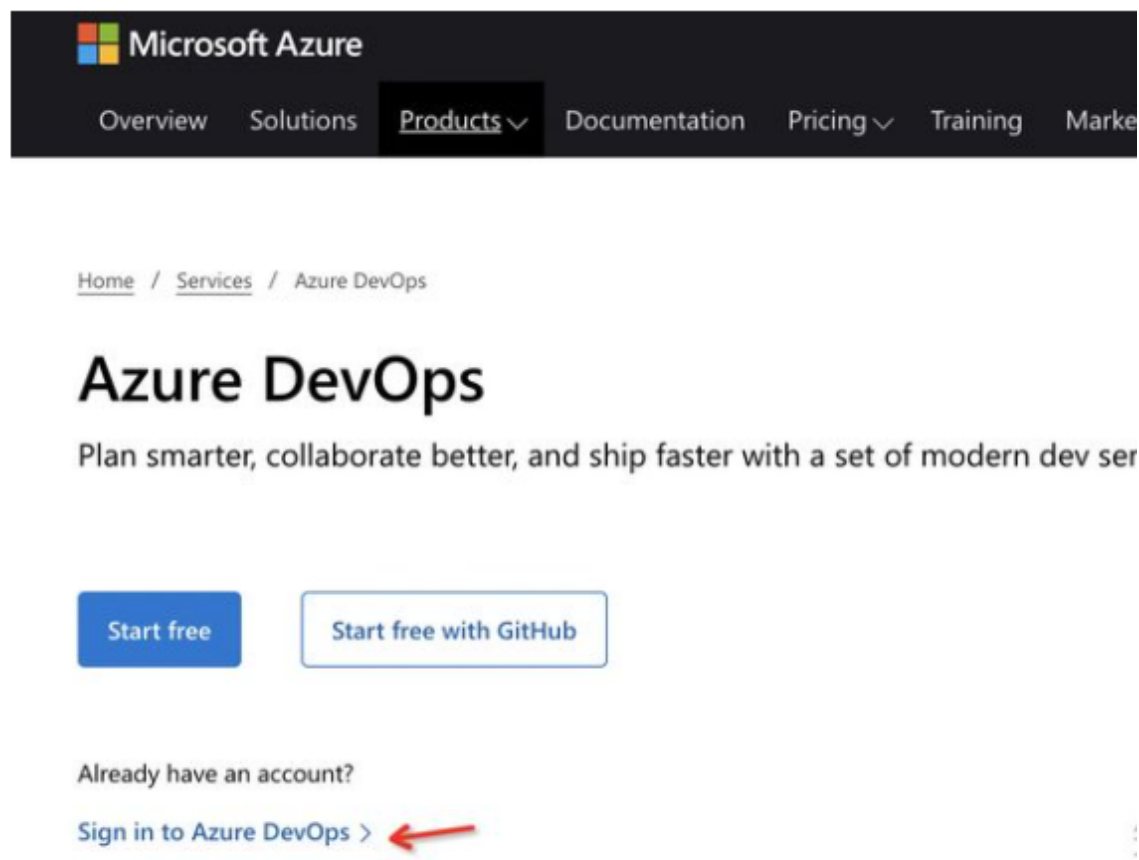
1  vsce package
```

打包完成后，会生成 `find-test-file-x.x.x.vsix` 文件。此时可以通过插件市场直接安装插件，验证效果：

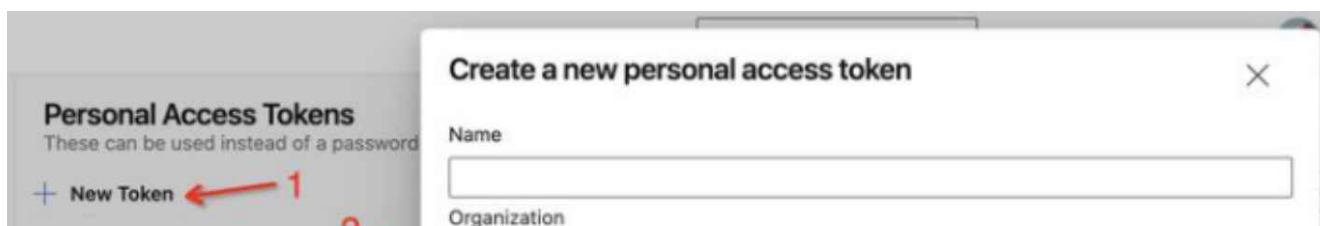
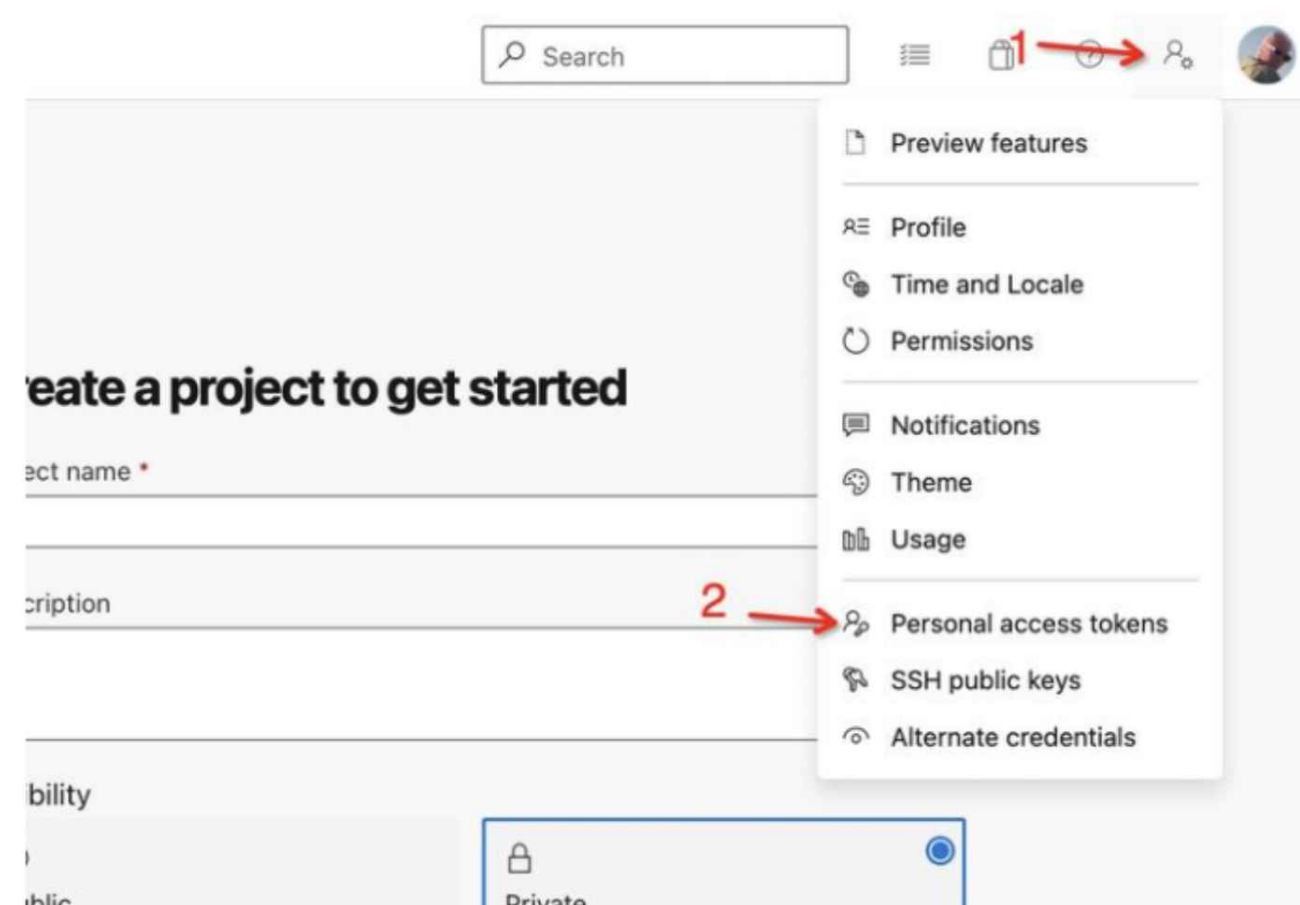


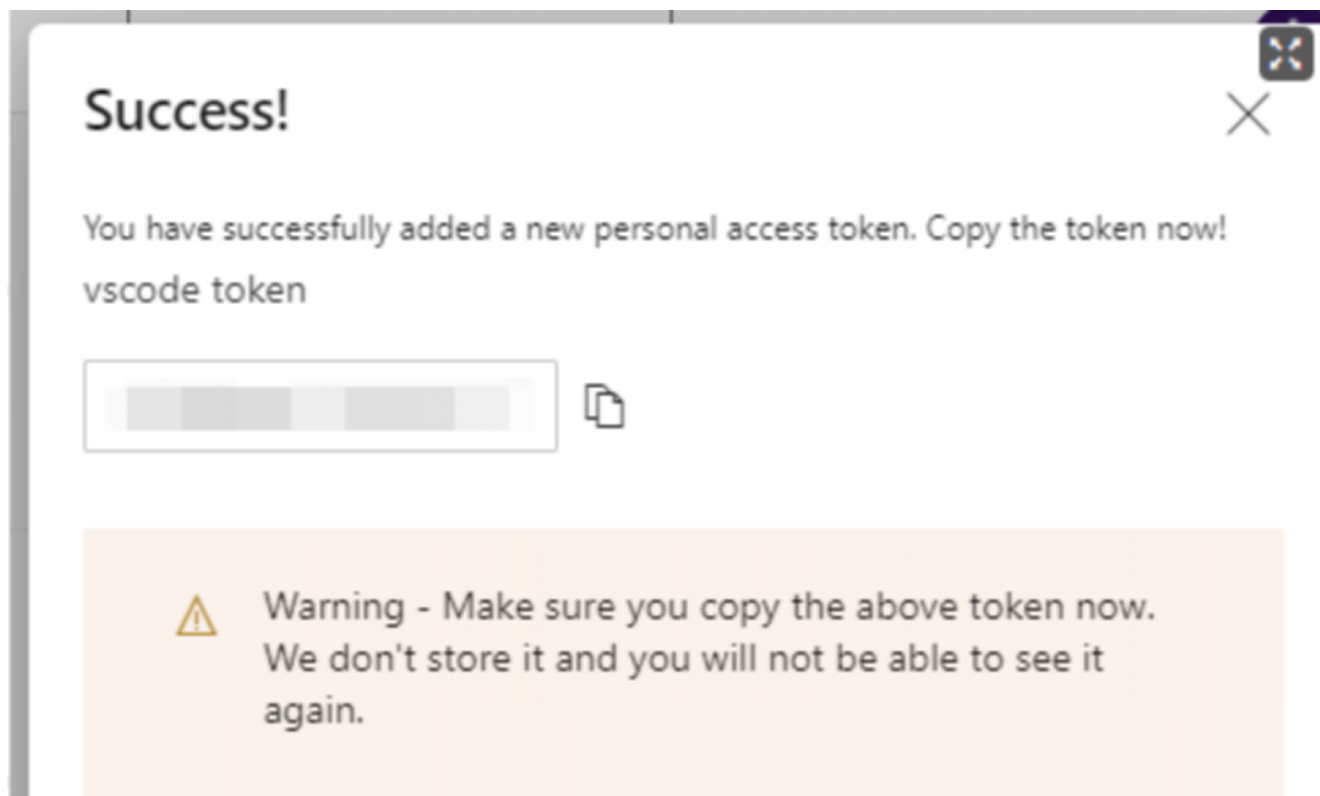
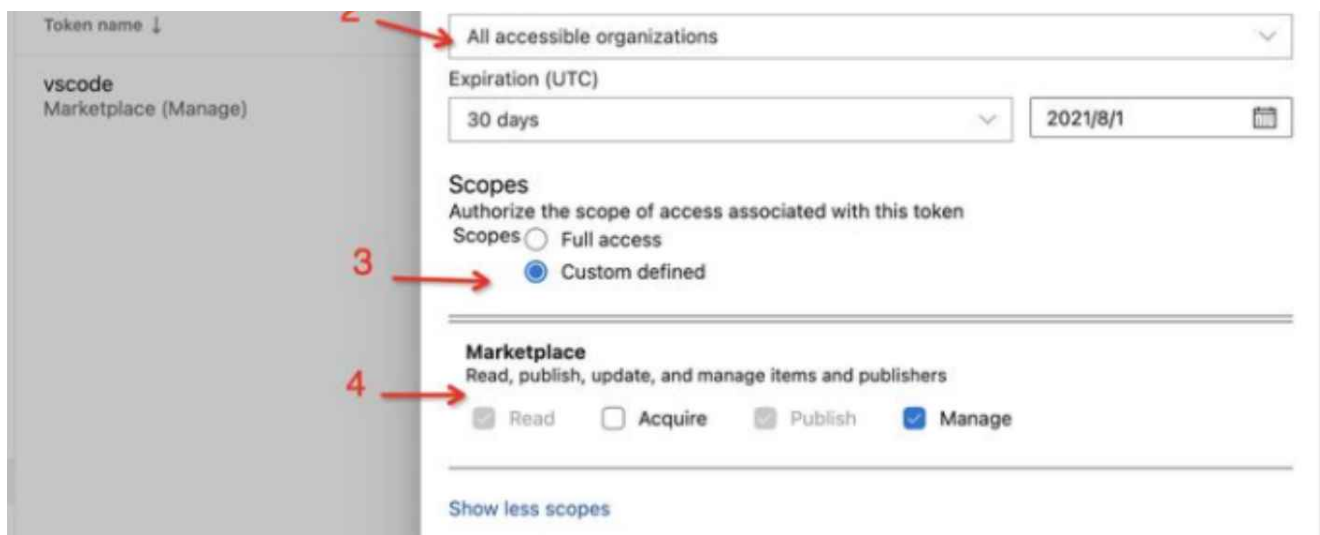
验证完功能的正确性后，就可以真正发布到插件市场：

vscode 的插件市场基于微软的 Azure DevOps，插件的身份验证、托管和管理都是在这里。所以发布前，首先得注册 [Azure DevOps](#) 账号，就像 npm 一样：

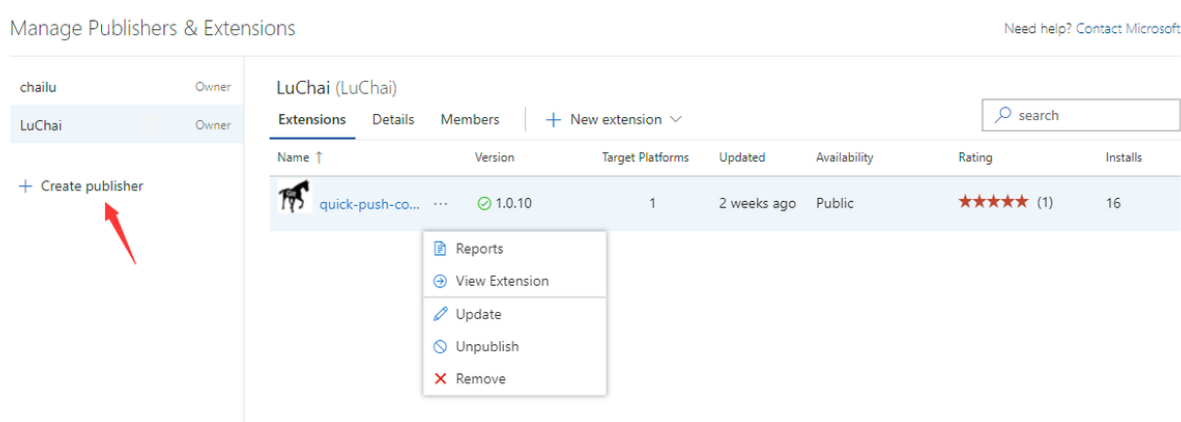


1. 注册完成后，创建 Personal Access Token：

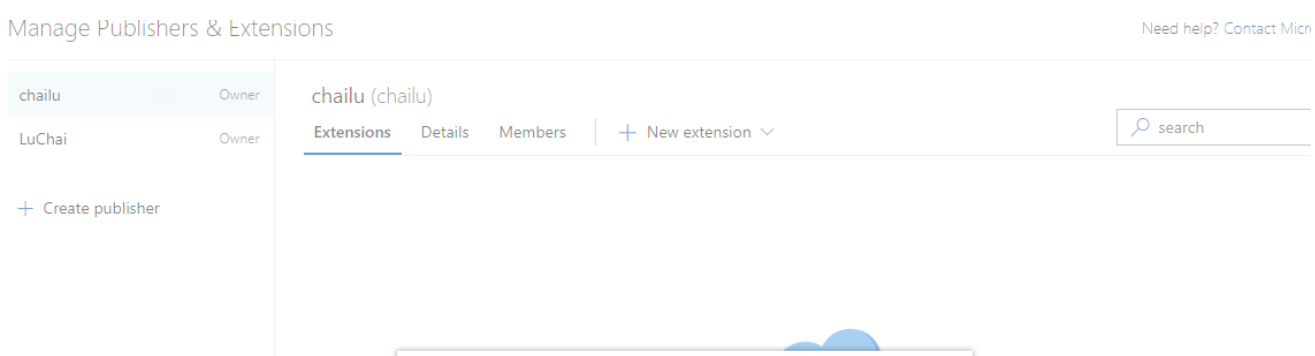


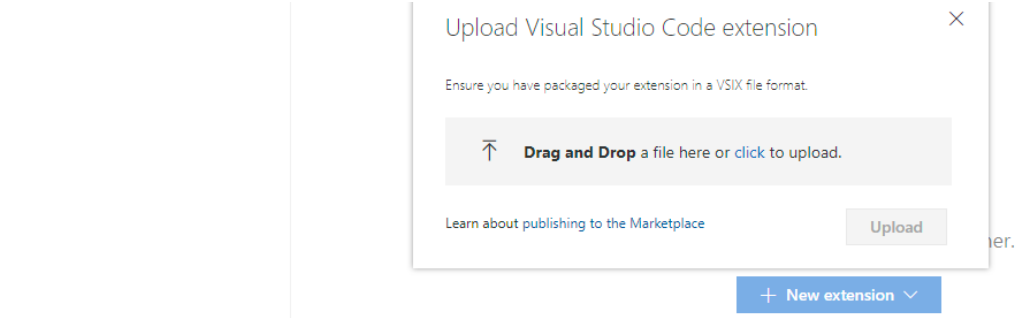


2. 接下来还需要在插件市场中 **新建一个 publisher** 用来发布插件：对你的插件进行命名，插件描述，插件logo配置等等



之后就可以对你本地测试打包好的代码进行上传





上传成功后过几分钟，会有邮件通知发布扩展成功，访问[quick-push-code - Visual Studio Marketplace](#)就可以看到刚刚发布的插件，进行下载安装

Visual Studio Code > Other > quick-push-code

New to Visual Studio Code? [Get it now.](#)

quick-push-code

LuChai | 16 installs | ★★★★★ (1) | Free

快速推送代码到远程仓库

Install

[Trouble Installing?](#)

Overview

Version History

Q & A

Rating & Review

quick-push-code README

quick-push-code

插件介绍

是一款一键推动代码插件，使用shelljs执行一系列git操作，将代码推送到远程仓库，提升效率

点击鼠标右键，一键推送代码点击后会弹出输入框输入commit信息(可不填)，默认 commit = "perf: update"

```
commit 规范

feat 增加新功能
fix 修复问题/BUG
style 代码风格相关无影响运行结果的
perf 优化/性能提升
refactor 重构
revert 撤销修改
test 测试相关
docs 文档/注释
chore 依赖更新/脚手架配置修改等
workflow 工作流改进
ci 持续集成
types 类型定义文件更改
wip 开发中

commit提交格式为:
git commit -m 'xxx: 你的修改信息'
```

Categories

Other

Works with

Universal

Resources

[Changelog](#)

[Download Extension](#)

More Info

Version

1.0.10

Released on

2021/8/27 下午2:11:51

Last updated

2021/9/30 下午12:29:17

Publisher

LuChai

Unique Identifier

LuChai.quick-push-code

Report

[Report Abuse](#)

也可以在vscode插件市场可搜到

quick-push

quick-push-code

快速推送代码到远程仓库

LuChai

quick-push-code v1.0.10

LuChai | 16 | ★★★★★ (1)

快速推送代码到远程仓库

禁用

卸载

此扩展已全局启用。

细节

功能贡献

更改日志

运行时状态

是一款一键推动代码插件，使用shelljs执行一系列git操作，将代码推送到远程仓库，提升效率

点击鼠标右键，一键推送代码点击后会弹出输入框输入commit信息(可不填)，默认 commit = "perf: update"

```
commit 规范

feat 增加新功能
fix 修复问题/BUG
style 代码风格相关无影响运行结果的
perf 优化/性能提升
refactor 重构
revert 撤销修改
test 测试相关
docs 文档/注释
```

总结

至此一款简单的vscode插件就完成了，我们还可以写一些一键新建文件模板（vue2,vue3,react等等），鼠标悬停提示，跳转到其它文件等等

参考文献

[vscode插件开发官方文档](#)

知乎讲vscode原理的

<https://zhuanlan.zhihu.com/p/99198980>

vscode入门的博文教程

<https://www.cnblogs.com/liuxianan/p/vscode-plugin-overview.html>