1.要安装依赖/安装脚手架

* 从0到1开发插件：安装脚手架

// 安装需要的包

npm install -g yo generator-code

// 运行

yo code

* clone别人的repository到本地运行：安装依赖

|  |
| --- |
| npm install |
| # 或 |
| yarn install |

2.检查是否包含开发插件的重要文件

package.json：必须包含 "activationEvents" 和 "contributes" 字段

src/ 或 extension.js：插件的主代码文件

* 从0到1开发插件：

当运行完以上的代码之后，会直接创建好一个仓库，里面一般来说都会有以上这些文件

* clone别人的repository到本地运行：

就需要自己在去检查有没有

3.对于package.json来说：

重要的是activationEvents、main以及contributes三部分内容

* main：指明了该插件的主入口在哪，只有找到主入口整个项目才能正常的运转
* activationEvents：指明该插件在何种情况下才会被激活，如：[onLanguage](https://link.segmentfault.com/?enc=8apLa6nJx8v4bVS6x%2FWWWQ%3D%3D.GCb1ha5OAj6dKsCW3Wy90KRV7mpk4opyElqQVI6s%2F4cDr9DYMCbkvJCQ1Hg5Xoj5z%2B9L6wp0edo9C5KGqPPspEEjG96hnlg3TPXbT0ZPv64%3D) 打开解析为特定语言文件时被激活，[onCommand](https://link.segmentfault.com/?enc=LPzbgYrTebDEhmmN79%2BdQQ%3D%3D.S1FNT7VRD4gMvjdb9UWyF%2Bv5WW4Ja2VKqo%2BCwCSbvUBZf104Frq2frIgb4Fx2wg7OfcqZrlmolizkfUEOvfxS9BtI92UUNIFDEa%2BX1BYZLo%3D" \t "_blank) 在调用命令时被激活
* contributes：通过扩展注册contributes用来扩展Visual Studio Code中的各项技能

4.对于extension.js来说：

它就是即package.json中main字段对应的文件，它有两种导出两个方法：activate和deactivate，两个方法的执行时机如下所示：

* activate：这是插件被激活时执行的函数
* deactivate：这是插件被销毁时调用的方法，比如释放内存等。

小结：以上是插件开发的前期准备，简单的插件开发直接在刚刚所提到的两个文件里面修改内容以后运行即可。

5. 复现显示文件状态的vscode插件

对于package.json：

日程表

AI 生成的内容可能不正确。

对于extension.js：

const vscode = require('vscode');

const fs = require('fs');

function activate(context) {

console.log('插件已经被激活');

// 注册命令

let commandOfGetFileState = vscode.commands.registerCommand('getFileState', uri => {

// 文件路径

const filePath = uri.path.substring(1);

fs.stat(filePath, (err, stats) => {

if (err) {

vscode.window.showErrorMessage(`获取文件时遇到错误了${err}!!!`)

}

if (stats.isDirectory()) {

vscode.window.showWarningMessage(`检测的是文件夹，不是文件，请重新选择！！！`);

}

if (stats.isFile()) {

const size = stats.size;

const createTime = stats.birthtime.toLocaleString();

const modifyTime = stats.mtime.toLocaleString();

vscode.window.showInformationMessage(`

文件大小为:${size}字节;

文件创建时间为:${createTime};

文件修改时间为:${modifyTime}

`, { modal: true });

}

});

const stats = fs.statSync(filePath);

console.log('stats', stats);

console.log('isFile', stats.isFile());

});

// 将命令放入其上下文对象中，使其生效

context.subscriptions.push(commandOfGetFileState);

}

function deactivate() {}

module.exports = {

activate,

deactivate

}

6. Run and Debug

按左侧图标（或按 F5）。

选择 "Extension" 配置（通常自动生成）。

VS Code 会启动一个 新的扩展开发窗口，并自动激活插件

7.打包

插件开发完毕之后可以分享出去供别人使用

* 直接把文件夹发给别人，让别人找到vscode的插件存放目录并放进去，然后重启vscode，一般不推荐；
* 打包成vsix插件，然后发送给别人安装，因为我的插件就
* 安装对应的模块vsce

npm i vsce -g

* 利用vsce进行打包，生成对应的vsix文件

vsce package

实现效果

图形用户界面, 应用程序

AI 生成的内容可能不正确。

图形用户界面, 文本, 应用程序

AI 生成的内容可能不正确。

图形用户界面, 应用程序

AI 生成的内容可能不正确。