# 从零开始开发一款 属于你的 VS Code 插件





"VS Code 中文社区"公众号

## 韩骏

- 微软研发工程师@ 开发平台事业部
- 《Visual Studio Code 权威指南》作者
- 20 多款 VS Code 插件作者
- "玩转VS Code"知乎专栏 & 微信公众号
- VS Code 中文社区创始人
- <a href="https://github.com/formulahendry/955.WLB">https://github.com/formulahendry/955.WLB</a>
- 内推 70 人拿到微软 offer
- Speaker @ PyCon、JSConf、.NET Conf、 Microsoft Tech Summit、Google Developer Group、COSCon ...



公众号"HJ说"

# 2019年11月30日

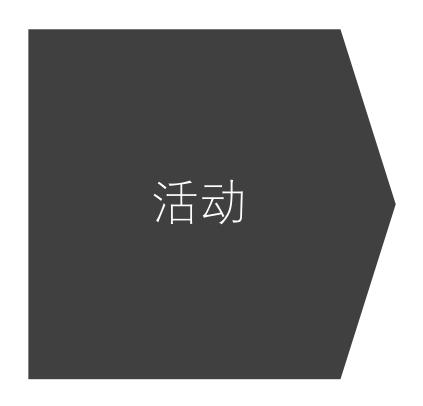




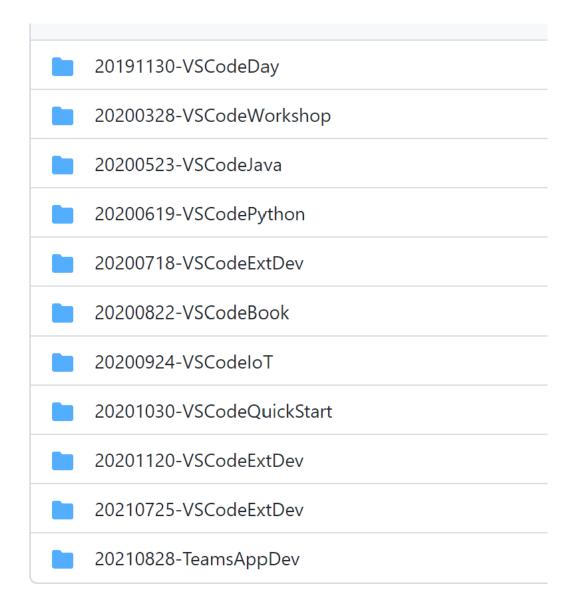
https://github.com/vscodecc/vscodecc/tree/master/meetups/20191130-VSCodeDay



https://github.com/vscodecc/vscodecc/tree/master/meetups/20191130-VSCodeDay



https://github.com/vscodecc/vscodecc



### **Announcing New Logo**



### 日程

- 14:00 14:10 主持人开场
- 14:10 14:50 VS Code TreeView API 概览与进阶使用技巧
- 14:50 15:30 DevContainer 更快构建跨平台应用开发
- 15:30 15:40 茶歇,交流
- 15:40 16:20 为VS Code增加新的语言支持——从插件到协议
- 16:20 16:30 Q&A,抽奖,合影

# Visual Studio Code TreeView API 概览与进阶使 用技巧

陈晟

```
Arror_mod = modifier_ob
  mirror object to mirror_mod.mirror_object
  peration == "MIRROR_X":
 1rror_mod.use_x = True
Irror_mod.use_x = True
Irror_mod.use_y = False
Irror_mod.use_z = False
Operation == "MIRROR_Y"
  irror_mod.use_x = False
  lrror_mod.use_y = True
  rror mod.use_z = False
   operation == "MIRROR_Z"
   rror_mod.use_x = False
   rror_mod.use_y = False
   rror_mod.use_z = True
    election at the end -add
    er ob.select=1
    text.scene.objects.action
     Selected" + str(modification
     bpy.context.selected_obj
    ta.objects[one.name].sel
   Int("please select exactly
   X mirror to the selected ject.mirror_mirror_x"
      pes.Operator):
    ontext):
xt.active_object is not
```

## 关于我

- VS Code Java 团队成员
- Eclipse JDT.LS Committer
- VS Code LeetCode 插件作者
- 《玩转 VS Code》知乎专栏作者
- GitHub: jdneo



陈晟 Software Engineer @ Microsoft

#### TreeDataProvider

```
/**
 * A data provider that provides tree data
 */
export interface TreeDataProvider<T> {
    getChildren(element?: T): T[];
    getTreeItem(element: T): TreeItem;
```

#### Treeltem

```
export class TreeItem {
    label?: string;
                                    id?: string;
                                    description?: string;
    iconPath?: ThemeIcon;
    resourceUri?: Uri;
                                    tooltip?: string;
                                    contextValue?: string;
    command?: Command;
    collapsibleState?: TreeItemCollapsibleState;
    accessibilityInformation?: AccessibilityInformation;
    constructor(label: string, collapsibleState?: TreeItemCollapsibleState);
    constructor(resourceUri: Uri, collapsibleState?: TreeItemCollapsibleState);
```

### VS Code API 的两种模式

- Pull 模式 VS Code 向插件拉取数据
  - 通常接口被命名为 \*Provider
  - 拉取的时机明确
- Push 模式 插件向 VS Code 推送数据
  - 数据发生变化的时机不明确
  - 例如 DiagnosticCollection

## 在 package.json 中注册

```
"contributes": {
 "views": {
  "explorer": [
     "id": "<Your Tree View ID>",
     "name": "<Your Tree View Name> "
```

### 注册 TreeDataProvider

```
vscode.window.registerTreeDataProvider(
    '<Your Tree View ID>',
    new CustomizedTreeDataProvider()
);
```



https://github.com/jdneo/tree-view-demo

### 插件开发的学习资料

- VS Code 官网
  - https://code.visualstudio.com/api
- 插件样例
  - https://github.com/microsoft/vscode-extension-samples