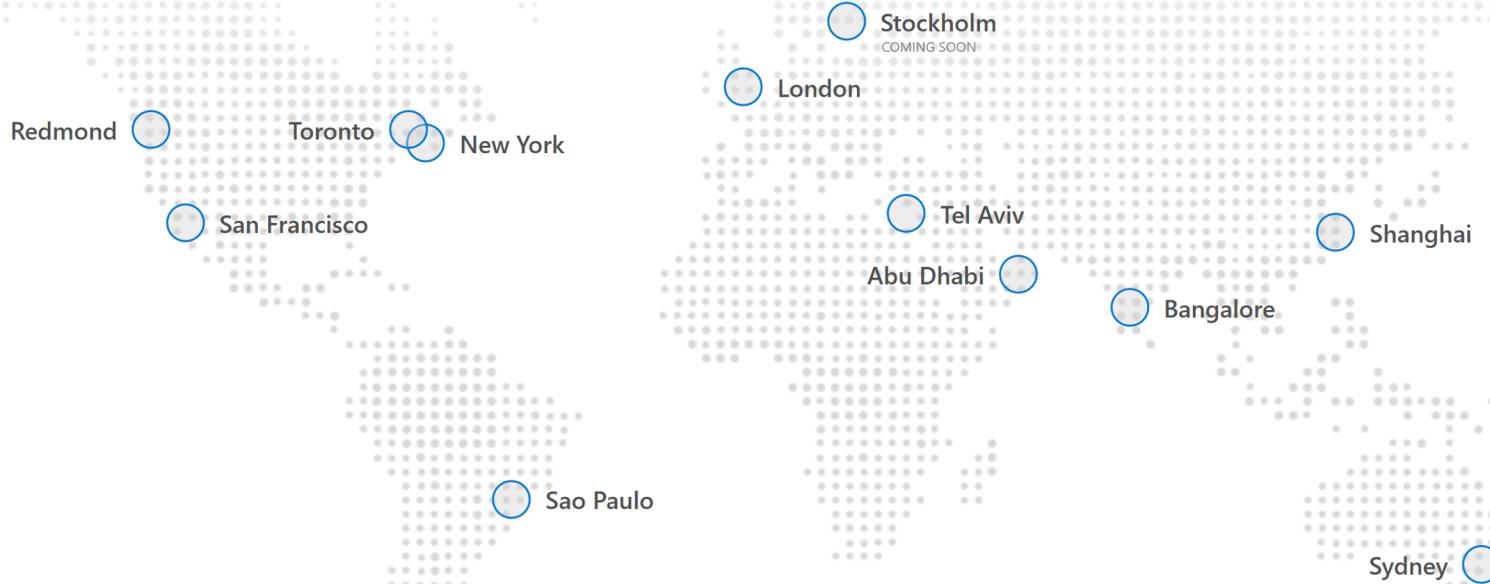




# Reactor

VS Code 下的聊天机器人开发

# Map



# 个人介绍



## Kinfey Lo – (卢建晖)

**Microsoft Cloud Advocate**

前微软MVP、Xamarin MVP和微软RD，拥有超过10年的云原生、人工智能和移动应用经验，为教育、金融和医疗提供应用解决方案。 Microsoft Ignite , TechEd 会议讲师，Microsoft AI 黑客马拉松教练，目前在微软，为技术人员和不同行业宣讲技术和相关应用场景。



爱编程(Python , C# , TypeScript , Swift , Rust , Go )

专注于人工智能，云原生，跨平台移动开发

**Github :** <https://github.com/kinfey>

**Email :** [kinfeylo@microsoft.com](mailto:kinfeylo@microsoft.com)    **Blog :** <https://blog.csdn.net/kinfey>

**Twitter :** @Ljh8304

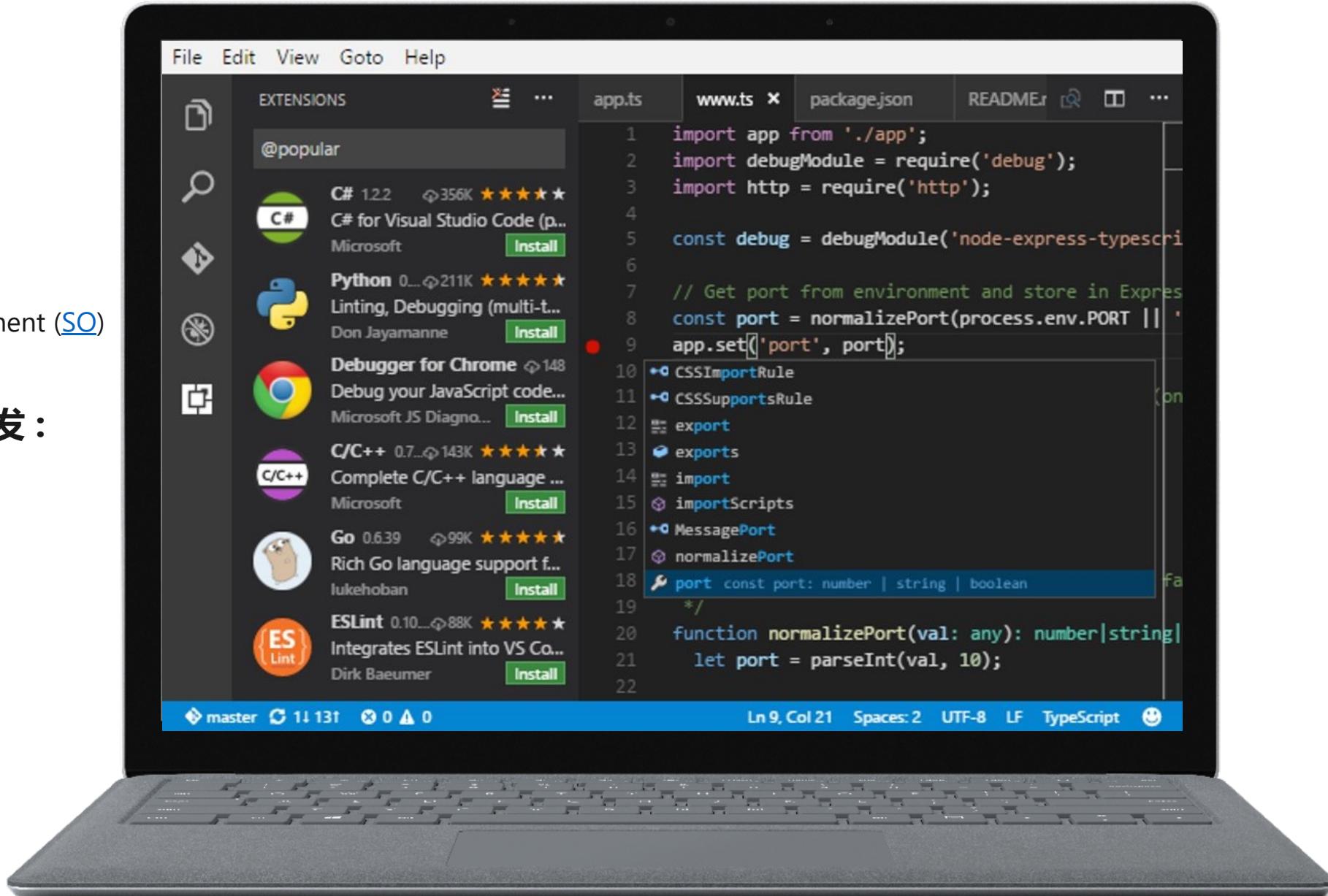
# 你好 Visual Studio Code

深受各社区开发者的喜爱：

- #1 JavaScript editor ([State of JS](#))
- #1 Go editor ([Go](#))
- #1 Rust editor ([Rust](#))
- #2 Python editor ([PSF/Jetbrains](#))
- Most popular development environment ([SO](#))

使用强大的 OSS 生态系统开发：

- 19K contributors on GitHub ([#1](#))
- Daily insider releases
- Public roadmap, issues, backlog



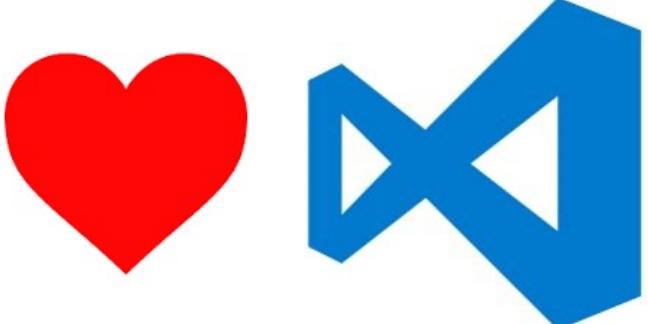
# Visual Studio Code 中的插件

<https://marketplace.visualstudio.com/VSCodium>

使 VS Code 如此受欢迎的原因是市场中可用的扩展数量。有超过 30,000 个扩展在流通，选择几乎是无限的——同时也是压倒性的。

根据面向程序员的时间跟踪工具 WakaTime，Visual Studio Code 在 2020 年的开发工作时长超过 1800 万小时。此外，Visual Studio Code 占开发人员使用的所有编辑器跟踪的总时间的 46%。

Developer



Extensions for the Visual Studio family of products

Search Visual Studio Code extensions  🔍

Featured

New Relic CodeStream New Relic CodeStream codestream.com <span>207K</span> <span>★★★★★</span> FREE	Chalice Icon Theme Artem Laman <span>34.1K</span> <span>★★★★★</span> FREE	Mintlify Doc Writer Mintlify mintlify.com <span>18.8K</span> <span>★★★★★</span> FREE	Trunk Trunk trunk.io <span>14.3K</span> <span>★★★★★</span> FREE	Playwright Test for VS Code Microsoft microsoft.com <span>12.5K</span> <span>★★★★★</span> FREE	Truffle for VS Code Consensys Software trufflesuite.com <span>1.3K</span> <span>★★★★★</span> FREE
--	---	--	---	--	---

Trending this week

CodeSandbox Theme CodeSandbox <span>245</span> <span>★★★★★</span> FREE	Mindustry Logic Vortetty <span>216</span> <span>★★★★★</span> FREE	LeetCode with labuladong labuladong <span>3.1K</span> <span>★★★★★</span> FREE	SQL Bindings Microsoft microsoft.com <span>137K</span> <span>★★★★★</span> FREE	GruvBox Ransh <span>161</span> <span>★★★★★</span> FREE	Polaris for VS Code Shopify shopify.com <span>454</span> <span>★★★★★</span> FREE
--	---	---	--	--	--

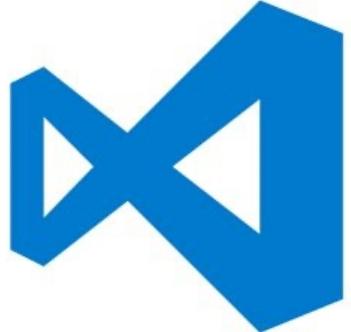
Most Popular

Python Microsoft microsoft.com <span>55M</span> <span>★★★★★</span> FREE	Jupyter Microsoft microsoft.com <span>35.5M</span> <span>★★★★★</span> FREE	C/C++ Microsoft microsoft.com <span>30.5M</span> <span>★★★★★</span> FREE	Pylance Microsoft microsoft.com <span>28.3M</span> <span>★★★★★</span> FREE	Live Server Ritwick Dey <span>20.9M</span> <span>★★★★★</span> FREE	Prettier - Code formatter Prettier <span>20.5M</span> <span>★★★★★</span> FREE
---	--	--	--	--	---

[See more](#) >

# Visual Studio Code 中的插件种类

<https://marketplace.visualstudio.com/VSCodium>



VSCode 涵盖了开发的所有场景，开发者众生伴侣

Azure

Data Science

Debuggers

Education

Extension Packs

Formatters

Keymaps

Language Packs

Linters

Machine Learning

Notebooks

Programming Languages

SCM Providers

Snippets

Testing

Themes

Visualization

Other

# 开发一个 Visual Studio Code 插件有多难？

```
npm install -g yo generator-code
```

```
yo code
```

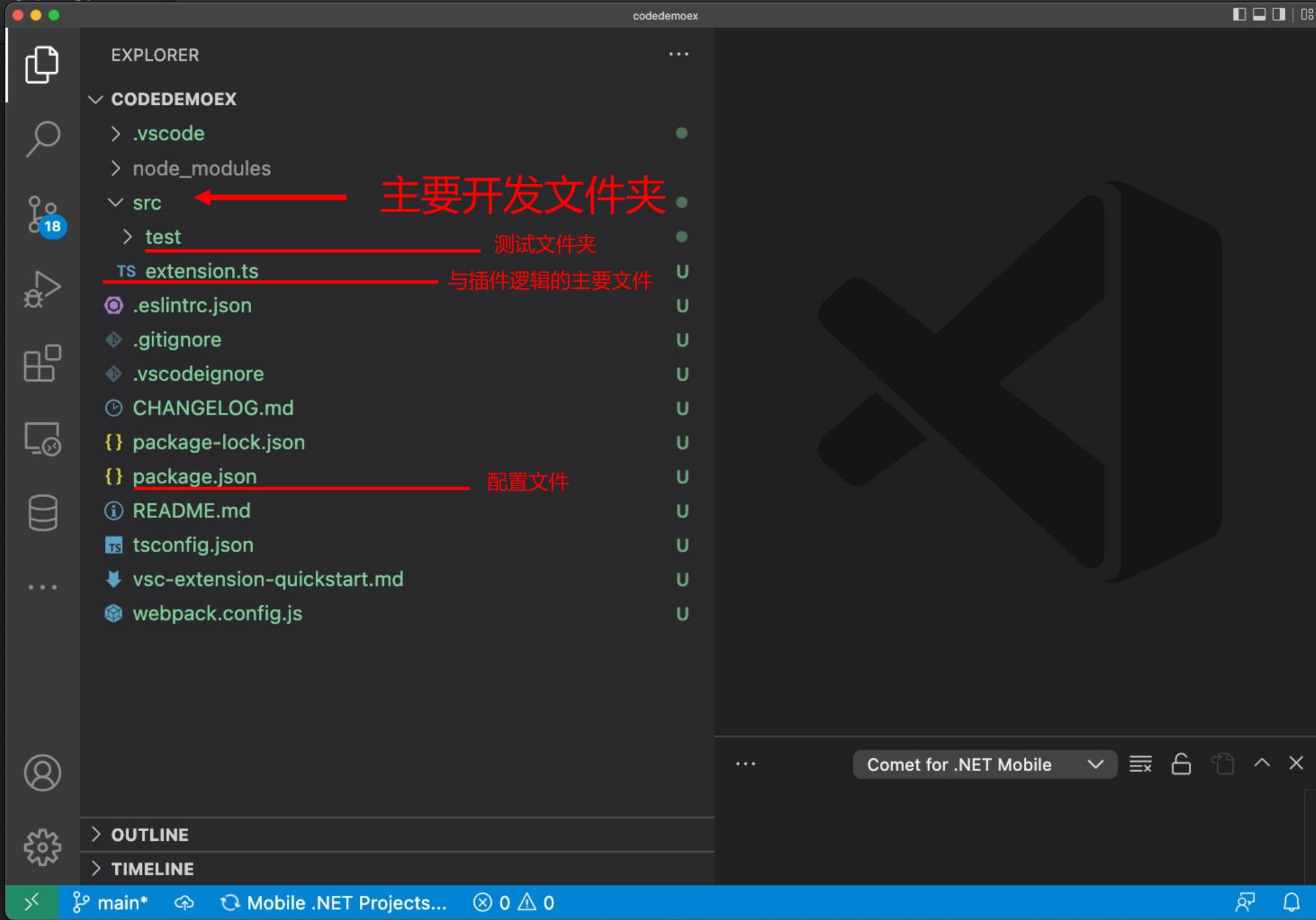
```
(base) → code yo code

  _---(o)---_
  |           |
  ( _ 'U' _ )
 /   A   \   /
  |   ~   |
  '---.---' '
          |°| Y |

Welcome to the Visual
Studio Code Extension
generator!

? What type of extension do you want to create? New Extension (TypeScript)
? What's the name of your extension? qaextensiondemo
? What's the identifier of your extension? qaextensiondemo
? What's the description of your extension?
? Initialize a git repository? Yes
? Bundle the source code with webpack? Yes
? Which package manager to use? npm
```





# package.json

```
"name": "codedemoex",
"displayName": "codedemoex", 插件名
"description": "",
"version": "0.0.1",
"engines": {
    "vscode": "≥1.66.0" 插件支持的 vscode 版本
},
"categories": [
    "Other"
],
"activationEvents": [
    "onCommand:codedemoex.helloWorld"
],
"main": "./dist/extension.js",
"contributes": {
    "commands": [
        {
            "command": "codedemoex.helloWorld", 绑定的响应事件
            "title": "Hello World"
        }
    ]
}
}, 对应 CMD+ Shift + P 对应的命令
```

# extension.ts

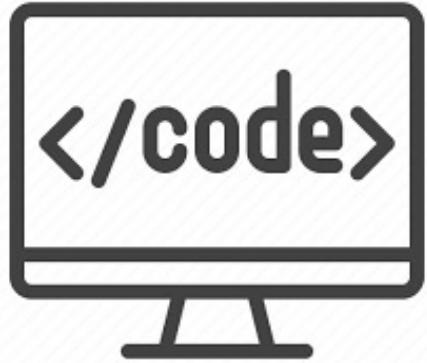
```
1 // The module 'vscode' contains the VS Code extensibility API
2 // Import the module and reference it with the alias vscode in your code below
3 import * as vscode from 'vscode';
4
5 // this method is called when your extension is activated
6 // your extension is activated the very first time the command is executed
7 export function activate(context: vscode.ExtensionContext) {
8
9     // Use the console to output diagnostic information (console.log) and errors (console.error)
10    // This line of code will only be executed once when your extension is activated
11    console.log('Congratulations, your extension "codedemoex" is now active!');
12
13    // The command has been defined in the package.json file
14    // Now provide the implementation of the command with registerCommand
15    // The commandId parameter must match the command field in package.json
16    let disposable = vscode.commands.registerCommand('codedemoex.helloWorld', () => {
17        // The code you place here will be executed every time your command is executed
18        // Display a message box to the user
19        vscode.window.showInformationMessage('Hello World from codedemoex!'); 响应后的结果
20    });
21
22    context.subscriptions.push(disposable);
23 }
24
25 // This method is called when your extension is deactivated
26 export function deactivate() {}
```

# Debug 呢？

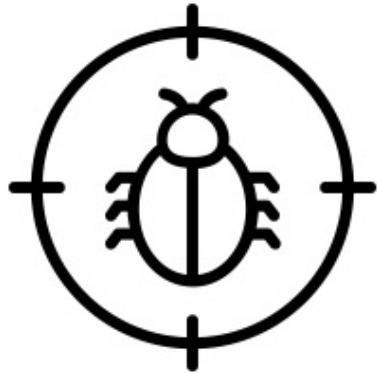
The screenshot shows the Visual Studio Code interface with the "RUN AND DEBUG" tab selected. The "extension.ts" file is open in the editor, displaying code for a VS Code extension. The code includes logic for activation, logging a message to the console, and displaying an information message box. A yellow arrow points to the line of code that shows how to display a message box using `vscode.window.showInformationMessage`. The "CALL STACK" panel shows the execution path, pausing at a breakpoint. The "BREAKPOINTS" panel shows that the current file, "extension.ts", has a breakpoint set. The "DEBUG CONSOLE" tab is active, showing the output of the debug session, which includes the message "Congratulations, your extension "codedemoex" is now active!". The bottom status bar shows the file name "extension.ts" and the line number "Ln 10, Col 85".

```
src > TS extension.ts U X
src > TS extension.ts > activate
1 // The module 'vscode' contains the VS Code extensibility API
2 // Import the module and reference it with the alias vscode in your code below
3 import * as vscode from 'vscode';
4
5 // this method is called when your extension is activated
6 // your extension is activated the very first time the command is executed
7 export function activate(context: vscode.ExtensionContext) {
8
9     // Use the console to output diagnostic information (console.log) and errors (console.error)
10    // This line of code will only be executed once when your extension is activated
11    console.log('Congratulations, your extension "codedemoex" is now active!');
12
13    // The command has been defined in the package.json file
14    // Now provide the implementation of the command with registerCommand
15    // The commandId parameter must match the command field in package.json
16    let disposable = vscode.commands.registerCommand('codedemoex.helloWorld', () => {
17        // The code you place here will be executed every time your command is executed
18        // Display a message box to the user
19        vscode.window.showInformationMessage('Hello World from codedemoex!');
20    });
21
22    context.subscriptions.push(disposable);
23
24
25 // this method is called when your extension is deactivated
26 export function deactivate() {}
```

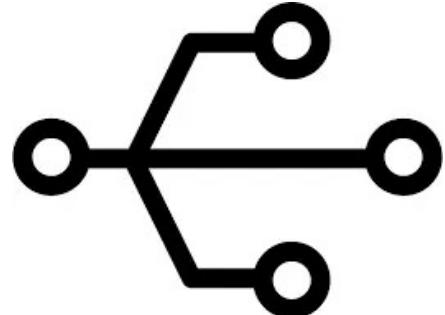
# 我们现在可以做什么样的插件



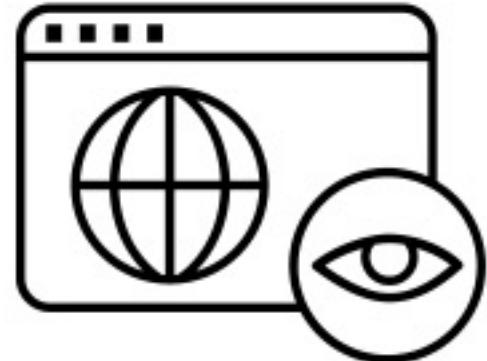
**Language**



**Debug**



**Source Control**



**WebView**

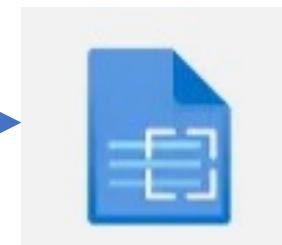
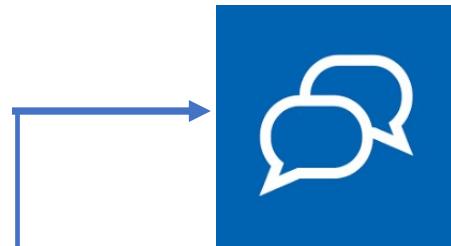
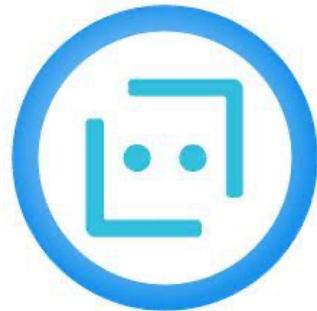
<https://code.visualstudio.com/api/language-extensions/overview>

<https://code.visualstudio.com/api/extension-guides/overview>

我是一名布道师  
也是一名开发者



# 架构



什么是 Dapr?

Dapr 如何离线安装?

QnA Rust 的基本要点

又要996啦好伤心

终于把Bug调整好了

帮我生成一个JSON格式

在哪里学习 PHP?

学习云原生的基本条件

知识图谱

# 上一代的 Cognitive Language Services

## Text Analytics



预建功能，包括命名实体提取、情感分析、PII 提取、语言检测、摘要等。

## Language Understanding (LUIS)



为会话应用程序构建自定义 NLU 模型以对意图进行分类并提取实体。

[www.luis.ai](http://www.luis.ai)

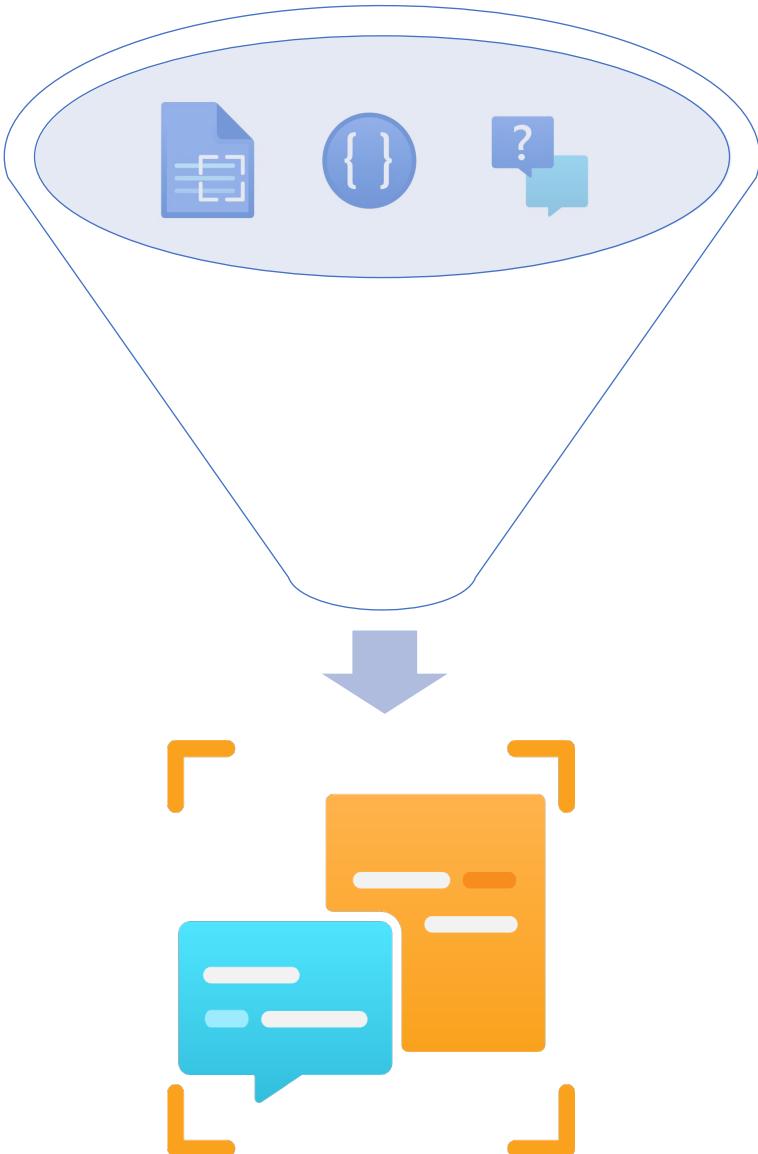
## QnA Maker



构建自定义知识库并为来自非结构化文本的问题提供答案。

[www.qnamaker.ai](http://www.qnamaker.ai)

# 新一代 Cognitive Service for Language



- 以前的单独服务合并为一个功能组下
- **统一服务**
  - 一个 Azure 资源
  - 一种用户体验
  - 一个 SDK
  - 一组 API
- **一个有凝聚力、更简单、功能强大的产品**
  - 最先进的
  - 多种语言
  - 新的和改进的

# The Language Studio

<https://language.cognitive.azure.com>

Cognitive Services | Language Studio

Language Studio >

What would you like to do?

Extract information | Classify text | Understand conversational language | Extract health information | Answer questions

[Extract information](#)

Use Natural Language Understanding (NLU) to extract information from unstructured text. Use these tools in order to do things like identify key phrases or Personally Identifiable Information (PII), summarize text, recognize and categorize named entities, or customize an entity extraction model on top of your domain set. [Learn more about text extraction](#)

[Back to top](#)



**Extract PII** (GA)

Identify sensitive entities in text that are associated with an individual

[Try it out](#)



**Extract key phrases** (GA)

Identify the most important points in a piece of text

[Try it out](#)



**Find linked entities** (GA)

Identify and disambiguate the identity of an entity found in text

[Try it out](#)



**Extract named entities** (GA)

Identify different entities in text and categorize them into pre-defined types

[Try it out](#)



**Custom named entity recognition** (Preview)

Train an extraction model to identify your domain categories using your own data.

[Open custom named entity recognition](#)

[Back to top](#)

**Classify text**

Use Natural Language Understanding (NLU) to detect the language or classify the sentiment of any piece of text you have. You can also classify your text documents by customizing a classification model over your dataset. [Learn more about custom text classification](#)



**Analyze sentiment and mine opinions** (GA)

Detect positive, negative and neutral sentiment in text. Get more insights by mining opinions.

[Try it out](#)



**Detect language** (GA)

Evaluate text and detect a wide range of languages, and variant, dialects.

[Try it out](#)



**Custom text classification** (Preview)

Train a classification model to classify text using your own data.

[Open custom text classification](#)

# The Language Studio

<https://language.cognitive.azure.com>

帮助我们改进 Language Studio. 参与我们的调查!

语言工作室 > 自定义问题解答 > daprqa - 编辑知识库

## 编辑 daprqa 中的问题

1非结构化源。 [查看源](#)

[添加问题对](#) [保存更改](#) [编辑](#) [测试](#) [删除](#) [显示列](#)

[Question answer pairs](#) [Synonyms](#)

Dapr 是一个开源、可移植的、事件驱动的运行时，可以帮助开发人员构建在云和边缘上运行的弹性的、微服务的、无状态和有状态应用程序，并且关注于业务逻辑而不用考虑分布式相关的问题，相关的问题可以交给Dapr 来解决。

2022年4月8日正式发布了1.7 版本，这是这是自Dapr v1.0 发布以来的第七次小版本更新，Dapr 1.7.0 版本的贡献者数量非常多。我们要感谢所有 83 位新的和现有的贡献者，他们帮助我们完成了这个版本。

如何离线安装

在这个版本中，现在共有 97 个组件，包括 41 个绑定、1 个配置、8 个 HTTP 中间件、3 个名称解析、13 个 pub/sub、8 个秘密存储和 23 个状态存储。

键入邮件并按 Enter 

测试

答复选项

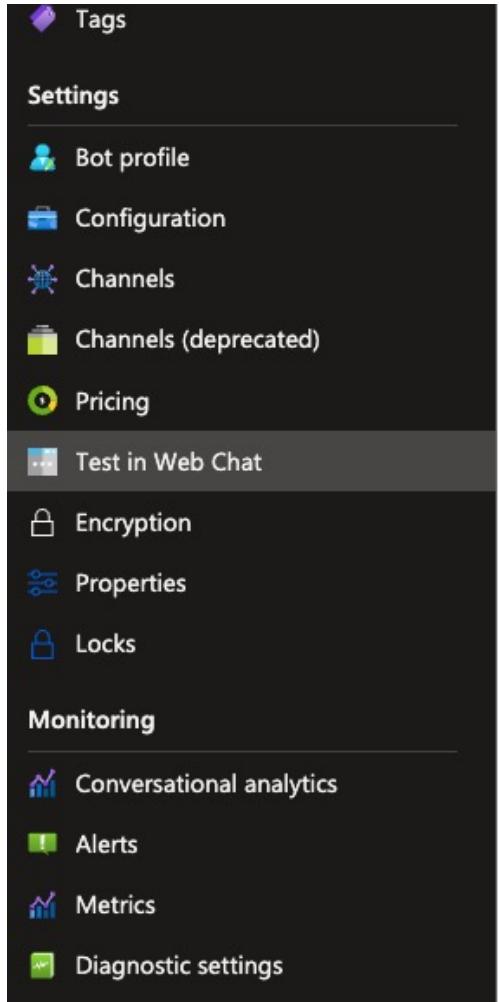
包括简短答案响应 ①

使用已部署的知识库 ①

显示高级选项

清除 

Dapr 



开源、可移植的、事件驱动的运行时，可以帮助开发人员构建在云和边缘上运行的弹性的、微服务的、无状态和有状态应用程序，并且关注于业务逻辑而不用考虑分布式相关的问题，相关的问题可以交给Dapr来解决

Dapr 是一个开源、可移植的、事件驱动的运行时，可以帮助开发人员构建在云和边缘上运行的弹性的、微服务的、无状态和有状态应用程序，并且关注于业务逻辑而不用考虑分布式相关的问题，相关的问题可以交给 Dapr 来解决。

2022年4月8日正式发布了1.7 版本，这是自Dapr v1.0 发布以来的第七次小版本更新，Dapr 1.7.0 版本的贡献者数量非常多。我们要感谢所有 83 位新的和现有的贡献者，他们帮助我们完成了这个版本。

Just now

如何离线安装

Just now

下载二进制文件并从网络中提取Image 来设置开发环境

Just now

# chatbot

```
activate(context: vscode.ExtensionContext) {  
    console to output diagnostic information (console.log) and errors (C  
of code will only be executed once when your extension is activated.  
Congratulations, your extension "qaextensiondemo" is now active!  
  
    // has been defined in the package.json file  
    // the implementation of the command with registerCommand  
    // id parameter must match the command field in package.json  
    const commandId = 'qaextensiondemo.helloWorld';  
    const disposable = vscode.commands.registerCommand('qaextensiondemo.helloWorld', () => {  
        // Code that will be executed every time you run this command  
        // You can also add user input to this command  
        // For example, let message = await vscode.window.showInformationMessage('Hello World!');  
        // message then needs to be disposed, otherwise it will remain open until the extension is deactivated  
        // .window.showInformationMessage('Hello World!').then(message => message.dispose());  
  
        const webviewPanel = vscode.window.createWebviewPanel("chat", "Chat", vscode.ViewColumn.Two, {  
            enableScripts: true,  
            context: {  
                onContextWhenHidden: true,  
  
                onDidFocus: () => {  
                    const webview = webviewPanel.webview;  
                    webview.html = `<body style="margin:0px;padding:0px;overflow:scroll"><div>${  
                        'me src="https://webchat.botframework.com/embed/daprkg-bot?&botId=' +  
                        context.botId}</div></body>`;  
                },  
            },  
            onDidDispose: () => {  
                context.subscriptions.push(disposable);  
            }  
        });  
        webviewPanel.onDidFocus(() => {  
            const webview = webviewPanel.webview;  
            webview.html = `<body style="margin:0px;padding:0px;overflow:scroll"><div>${  
                'me src="https://webchat.botframework.com/embed/daprkg-bot?&botId=' +  
                context.botId}</div></body>`;  
        });  
        webviewPanel.onDidDispose(() => {  
            context.subscriptions.push(disposable);  
        });  
        webviewPanel.onDidChangeViewState((change) => {  
            if (change === 'closed') {  
                context.subscriptions.push(disposable);  
            }  
        });  
        webviewPanel.onDidReceiveMessage((message) => {  
            if (message.type === 'openChatRoom') {  
                context.chatClient.sendEvent("openChatRoom");  
            }  
        });  
        webviewPanel.onDidDispose(() => {  
            context.subscriptions.push(disposable);  
        });  
    });  
}  
};
```

# 如何绑定到 VS Code 插件上



# I Studio Code evolved

Repository...

~/Desktop/code

emo ~/Desktop/code

text-analytics ~/Desktop

esktop

-Code-Using-Python-and...

## Walkthroughs

### Get Started with VS Code

Discover the best  
customizations to make VS  
Code yours.

### Learn the Fundamentals

Jump right into VS Code and  
get an overview of the must-  
have features.

### Boost your Productivity

### Getting Started Updated

### Get started with Updated Python development

More...

Hello and welcome!

Just now

开源、可移植的、事件驱动的运行时，可以帮助开发人员构建在云和边缘上运行的弹性的、微服务的、无状态和有状态应用程序，并且关注于业务逻辑而不用考虑分布式相关的问题，相关的问题可以交给Dapr来解决

Dapr 是一个开源、可移植的、事件驱动的运行时，可以帮助开发人员构建在云和边缘上运行的弹性的、微服务的、无状态和有状态应用程序，并且关注于业务逻辑而不用考虑分布式相关的问题，相关的问题可以交给 Dapr 来解决。

2022年4月8日正式发布了1.7 版本，这是这是自Dapr v1.0 发布以来的第七次小版本更新，Dapr 1.7.0 版本的贡献者数量非常多。我们要感谢所有 83 位新的和现有的贡献者，他们帮助我们完成了这个版本。

Just now

Type your message

# VS Code Chatbot

DEBUG CONSOLE TERMINAL

Comet for .NET M



# Reactor

## Thank You!