

移动云代码大模型



用户手册





移动云代码大模型,融合海量代码 库、技术文档、研发知识,为研发 人员提供智能、高效的编程支持。

移动云代码大模型能够**准确理解开 发者的意图**,将产品需求、原型图 转化为高效、精确的代码,从而大 幅提升编码效率,优化代码质量, 助力企业实现研发管理的数字化转型。



语言能力

兼容 Visual Studio Code、JetBrains IDEs 等主流 IDE;

支持 Java、Python、Go、C/C++/C#、JavaScript、TypeScript、PHP、 Ruby、Rust、Scala 等主流编程语言。

/ 产品优势

【多语言支持】能够理解和使用100种编程语言, 满足多样化的开发需求; 【高效精确】通过智能技术极大提升编码速度和准确性。 【多模态支持】能够解析图片内容,智能生成相应代码。 【持续学习】不断更新和学习新的编程知识和技术,保持行业领先。 日录

一、 插件安装与登录

1.1 VS Code中的湛卢插件安装

1.2 JetBrains中的湛卢插件安装

二、使用前配置

2.1 VS Code中的插件配置

2.2 JetBrains中的插件配置

三、操作指南

3.1 代码补全模式

- 3.1.1 行补全
- 3.1.2 块补全
- 3.1.3 注释生成补全

3.2 Chat模式

- 3.2.1 生成代码(支持多伦对话)
- 3.2.2 解释代码
- 3.2.3 生成代码注释
- 3.2.4 生成单元测试
- 3.2.5 指定代码库智能问答
- 3.2.6 代码调优纠错
- 3.2.7 代码翻译
- 3.2.8 图生代码
- 3.2.9 针对报错提示一键生成修复建议
- 3.2.10 互联网检索问答
- 3.2.11 代码评审
- 3.2.12 Vue升级

3.3 AI程序员

3.3.1 项目级单元测试

3.4 常见问题解答

一、插件安装与登录

1.1 VS Code中的湛卢插件安装

1.1.1 运行环境

适配 VS Code 集成开发环境(**1.86及以上版本**),支持**Windows、macOS**等 主流操作系统。

1.1.2 安装步骤

▶ 步骤一:

打开VS Code点击左侧扩展(Extensions)按钮,进入插件市场,搜索 "zhanlu",点击install进行安装。





安装成功后,打开湛卢页面,湛卢提供两种登录方式,可通过**手机验证码登** 录或通过填写AccessKey,SecretKey,License信息登录。未注册过的用户 可通过手机验证码自动完成注册并登录。

若用户未登录, IDE会弹出提示用户登录的弹窗。



点击前往登录,自动跳转至插件的登录页。

湛卢		≣ 3 \$ 0		
	ZHAN/U 湛卢代码大模型		ZHA 法卢代码	N/U 大模型
			Access Key	
手机号			· · · · · · · · · · · · · · · · · · ·	
请输入手机	1号		Secret Key	
验证码			请输入	
请输入验证	E码 获取验	证码	License	
			请输入 ————————————————————————————————————	





登录成功后,进入湛卢首页。

S Ei	le <u>E</u> dit <u>S</u> election <u>V</u> iew <u>Go R</u> un <u>T</u> erminal <u>H</u> elp	$\leftarrow \rightarrow$		𝒫 Search [Administrator]	
C.	港卢	≣ 3 ∯ 0	🗉 Untitled-1 🍨 🔒	¹ Extension: zhanlu:AI Coding Assistant-Intelligent partner in development ×	⊟ …
Q % 2 ~	SWIPS BEHGA BEHGA ELLAR BEHGA ELLAR BEHGA ELLAR BEHGA ELLAR CALLAR ELLAR	我▼		zhanlu:Al Coding Assistant-Intelligent partne 湿声 ゆ 10,333 ★★★★★(1) zhanlu:Al Coding Assistant-your intelligent partner in software development with auto Disable Uninstall マ Switch to Pre-Release Version ②	r in development v1.;
₿	双击 Alt ,快速唤起智能问答			DETAILS FEATURES CHANGELOG	
Ø			Ì	甚卢代码大模型	Categories Programming Languages
			ží Đ	甚卢代码大模型,融合海量代码库、技术文档、研发知识,为研发人员提供智能、高效的编程支 寺。可准确理解开发者的意图,将产品需求、原型图转换为高效的代码,从而大幅度提升编码效 氧、优化代码质量。	Machine Learning Education
				 東容 Visual Studio Code、JetBrains IDEs 等主流 IDE; 支持 Java、Python、Go、C/C++/C#、JavaScript、TypeScript、PHP、Ruby、Rust、Scala 等主流编程语言。 	Resources Marketplace 湛卢
			J	功能使用	More Info
			f g X	代码补全 忽可以在编辑器内通过自动或手动方式触发代码补全,一经触发,编辑器状态栏右下的湛卢图标将指示现在的请求 状态。	Published 2024-06-04, 10:58:27 Last released 2024-12-04, 17:25:18 Last updated 2024-12-05, 18:56:31 Identifier ecloud.zhanlu
			4		
			3	if (i < right) {	
	Hi,我是湛卢代码大模型,您的智能编码助手			<pre>quickSort(arr, i, right);</pre>	
	您可以证我与您一起编与代码,我支持单元测试、代码解释、代码纠错、API又相主成、 功能,您还可以向我询问技术问题。	vue2转vue3、等			
	🏂 更多功能,请查看快速上手			多行代码状作全 Multi-line Completions	
	● 窓可以试试这么问我:	♡ 换—换	3	TORKOLEIFIDET大,装置初亚的元星包有大家(AIT T 新田的方法、包歇、Fillin、幅中44号),当为日间版及保 式时,按下Enter键或按下Ctrl+K可推荐一个完整的代码块。(您可自行配置手动触发或者自动触发)	
	Spring MVC的流程是什么?			@~	
8	反下 Ctrl+K 键,手动触发代码补全			private TimeWatch(String tagName) {	
563	/指令 @知识库 粤开启互联网检索	1		this.tagName = tagName;	
× 8					the Contract of the Contract o

3

1.2 JetBrains中的湛卢插件安装

1.2.1 运行环境

适配 JetBrains 集成开发环境(支持Ultimate Edition 2020.3-2024.3版本、 Community Edition 2020.3-2024.3版本),支持Windows、macOS等主流操作 系统。

1.2.2 安装步骤

▶ 步骤一:

打开JetBrains,点击顶部"File"-->"Setting"按钮。

۲	<u>Eile [</u> dit <u>V</u> iew <u>N</u> avigate	e <u>C</u> ode <u>R</u> efactor	<u>B</u> uild R <u>u</u> n	<u>T</u> ools VC <u>S</u>	<u>W</u> indow	<u>H</u> elp			2+	¢ j	٥	×
	New	>										Ç
80	<u>Recent Projects</u>	>										
•••	Remote Development	£										
	Settings	Ctrl+Alt+S										
	Ca Project Structure	Ctri+Alt+Shift+S										
	File Properties				Sear	ch Everywhere Double Shi	ift					
	Local <u>H</u> istory				Proie	ect View Alt+1						
	🖹 <u>S</u> ave All	Ctrl+S										
	🕑 Reload All from Disk				Go to	o File Ctrl+Shift+N						
	Repair IDE				Rece	nt Files Ctrl+E						
	Invalidate Caches				Navio	nation Bar Alt+Home						
	Manage IDE Settings				Havi	gation bar Parthonio						
	New Projects Setup	>			Drop	files here to open them						





点击"Plugins",在"Marketplace"中搜索"zhanlu",点击"Install"。



▶ 步骤三:

安装成功后,打开湛卢页面,湛卢提供两种登录方式,可通过**手机验证码登** 录或通过填写AccessKey,SecretKey,License信息登录。未注册过的用户 可通过手机验证码登录并自动完成注册。

若用户未登录, IDE会弹出提示用户登录的弹窗。

\blacksquare \equiv T test \checkmark Version control \checkmark	🔲 DrawCircle 🗸 ▷ 🗘 🗄	옥Q 🕸 - @ 🛛
<mark>┣</mark> Projŧ ⊕ ≎ X i —	港户	\$? : — L



点击前往登录,自动跳转至插件的登录页。



▶ 步骤四: 登录成功后,进入湛卢首页。





2.1 VS Code中的插件配置

▶ 步骤一:

点击插件上端的配置按钮,进入配置页面,可根据实际需求更改下图的默认 配置。

- 🔁	文件(F) 编辑(E) 选择(S) 查看(V) 转到(G) 运行(R) ···		,				- 0 X
G) 日 扩展: zhanlu:Al Coding As	ssistant-Intelligent partner in development	ಘ 设置 X	🕏 score_prompts_4o.py	test_score_prompts_4o.py	• 🖞 🔲 …
Q	智能问答《图生代码	▼ ©ext:ecloud.zhanlu				找到	14 个设置 🔜 🍸
ر م		用户 远程 [SSN: 36138.22	2.228] 工作区				备份和同步设置
5	👌 ZHAN/U	〜 扩展 (14)	归嫣心址, 刂 毗直 松月化放伞 归 螨 地址				
æ	■	湛卢 (14)	https://api-wuxi-1.cmecloud.cn:8443				
₿	双击 Alt ,快速唤起智能问答	4	湖方: Debounce_time				
G			代码补至熙友可她毫秒(ms)	~			
π							
A A			湛卢: Completion_number 代码补全生成候选个数				
RI R			3	~			
			湛卢: Inline Completion_granularity 补全粒度的偏好				
	Hi,我是湛卢代码大模型,您的智能编码助手		均衡	~			
	您可以让我与您一起编写代码,我支持代码注释、单元测试、 代码解释、代码纠错、API文档生成、等功能,您还可以向我 询问技术问题。		湛卢: Python_unit_testing Python生成单元测试选用的框架				
	🧩 更多功能, 请查看快速上手		pytest	~			
	🤞 您可以试试这么问我: 🗘 换一换	问题 12 输出 调试控制	台 终端 端口 100	Ĵ	^{斋选器} (例如 text、!exclude、\esc	ape)	ר א <u>א</u> א
	Spring MVC的流程是什么?						





【**补全配置**】在设置页面中,可以配置编辑器内代码补全的以下四个方面:

(1) 触发延时:配置从停止输入到触发获取补全建议请求的延时,也可配置为 手动,此时仅当按下热键ctrl + k时获取补全建议,无触发时延。

(2)补全偏好:

行级补全:优先保证响应速度,返回单行建议;

平衡:兼顾响应速度和补全建议长度;

最大长度:依据配置的代码补全的max_tokens,在最大允许范围内自行确定返回内容。

- (3) 补全建议返回的最大token数
- (4) 候选建议数量:设置返回候选条目的数量。
- 【其他配置】 还支持配置单元测试框架类型,以及问答的最大token数。
- **可根据需求自行配置**,配置完成后,返回 Visual Studio Code 即可体验港 卢的各项功能。

▶ 步骤三:

- 【自定义prompt配置】可根据实际需求,自定快捷操作。
- (1) 点击"/"后,进入配置页面,点击修改自定义提示按钮 📓 。

		EIT(i)		nangsinjun (born borrboilelleto)			J 00 U	
G	湛卢	≣ 3 ∰ 0	昭 扩展: zhanlu:AI Coding Assistant-Intel	ligent partner in development 🛛 孝 设	2置 👌 自定义快捷操作	E 🗙 🌻 score_prompts_4o.py	🗬 test_score_p	• 🛙 ••
Q	智能问答 图生代码	我▼	自定义快捷操作列表					
20	ZHAN/L	Ĵ					+ 新3	建
å	▶ 湛卢代码大模型	旦	名称		描述		操作	
₿	双击 Alt ,快速唤起智能问	塔	代码注释		/生成代码注释		٢	>
G			单元测试		/生成单元测试		C	>
			代码解释		/解释代码		Ø	>
			代码纠错		/代码调优纠错		0	>
ß			API文档生成		生成遵循Swagger2.0规范的API描述	JSON格式文件,以代码块形式返回。	ℤ₫◎	>
	Hi,我是湛卢代码大模型,您的智能编码助到	F						
		Ľ						



(2) 点击"新建",进行快捷操作配置,配置完成后保存。



(3)保存后,此自定义prompt即可出现在快捷触发位置。

>	文件(F) 编辑(E) 选择(S) 查看(V) 转到(G) 运行	$f(R) \cdots \leftarrow \rightarrow$					08 — 🗇
G	湛卢	≣ 🕲 🛱 0 E	出 扩展: zhanlu:AI Coding Assistant-Intelligent partner in development	幸 设置 🛛 🖞	自定义快捷操作 🗙 🏾 🍨 sc	ore_prompts_4o.py	🕏 test_score_p 🔲 ·
Q	智能问答 图生代码	我 ▼	自定义快捷操作列表				
jo	2HAN/11						+ 新建
å	送卢代码大模型		名称	描述			操作
₿	双击 Alt ,快速唤起智能问答		代码注释	/生成代码注释			©
G			单元测试 	/生成单元测试			©
π			代码解释	/解释代码			۲
A			代码纠错	/代码调优纠错			۲
ß			API文档生成	生成遵循Swagger2.	.0规范的API描述JSON格式文件	,以代码块形式返回。	∠ ⊡ ⊚



(4) 使用时,先选中目标代码,再在问答输入框输入 "/",选择对应提示语即可。

2.2 JetBrains中的插件配置

▶ 步骤一:

中文显示问题配置。

(1) IDE**顶部菜单栏**中,选择**Help-Edit Custum VM Options**。

Eile Edit View Navigate Code Refactor Build Run Tools Git Window	Help	\square Application \lor \triangleright \textcircled{O}	2+ Q 🕸 - 🗆	×
	Eind Action Ctrl+Shift+A	湛卢		Ļ
<u>ب</u>	? <u>H</u> elp	智能问答	① 已登录,占击退出 日	• m
<u> </u>	😔 Learn IDE Features		000 01	
8	What's <u>N</u> ew in IntelliJ IDEA			
	Configure the New UI			
•••	<u>G</u> etting Started			
	IntelliJ IDEA on YouTube			
	Keyboard Shortcuts PDF			
	Tip of the Day	2 [7]		
	My <u>P</u> roductivity		代码十档刑	
	Contact <u>S</u> upport	· · · · · · · · · · · · · · · · · · ·	,快捷键。和15	
	Submit a Bug Report			
	Submit <u>F</u> eedback			
	Show Log in Explorer			
	Collect Logs and Diagnostic Data			
	Delete Leftover IDE Directories			
	Diagnostic Tools >			
	Change Memory Settings			
	Edit Custom Properties	Hi,我是湛卢代码大模型,您	的智能编码助手	
	Edit Custom VM Options	您可以让我与您一起编写代码,我 代码纠错等功能,您还可以向我询	支持代码注释、单元测试、代码解释、 问技术问题。	
(9	<u>C</u> heck for Updates			
	<u>R</u> egister Plugins	🔰 🎽 更多功能,请查看 ⁴	快速上手 🛛 🔺	
Ð	About	≫ 你可以试试这么问我。	€ , #_#	
8	C Statistic			
		Spring MVC的流程是怎样的		
0			4	
		/指令 @知识库 ●开启到	夏联网检索	
٣				
ilot-server-data-processor > 🗆 acepilot-server-data-consumer > src > main > jav	a > com > chinamobile > cmss > acepilot	> server > data > repository >	/lessageRepository > 🎯 save	\$

(2) 增加-Dfile.encoding=UTF-8 后重启IDE。



▶步骤二:

通用配置。

(1) 点击插件顶端配置按钮 💿 ,进入配置页面。

Ľ	≡ 🚺 acepilot-server ∽ 🍄 dev_AcepilotServer_V1.0.6_Myh ∽	Applicat	tion ~	⊳⇒	Ĉŧ :	ĉ+	Q	ତ୍ତେ –		×
	\equiv idea64.exe.vmoptions \times		湛卢					@?		¢
8	1 -Xmx1 U Settings				×	⊕	3	已登录,点击退	出 C+	т

3	-jvm	Q-	Tools → 湛卢 ←		
		Tools	Connection Settings		
		Qodana			
		Actions on Save	■ 普通问答最大token数: 2048 Haloe问答最大t	oken	
÷.		Web Browsers and Preview	读取超时时间(毫秒): 50000 连接超时时间(毫	秒):	
		External Tools			
		Terminal	■ Java单元测试框架: Junit5		
		> Android	Go单元测试框架: testify	_{框架} 甚卢代码大模型	
		Black		E问答,快捷键 Alt S	
		Code With Me	服务端地址: https://api-wuxi-1.cmecloud.cn:8443		
2		> Diff & Merge			
		Features Suggester			
		Features Trainer			
		Python External Documentation			
		Python Integrated Tools			
		Server Certificates			
		Shared Indexes		』,您的智能编码助手	
		Startup Tasks		肖,我支持代码注释、单元测试、代码解释、 向我询问技术问题。	
10		Statistic			
		> Tasks		查看快速上手	
⊘		XPath Viewer			
~		> 湛卢		问我: ひ 換一換	
Sec. 1		Settings Sync		样的 🛁	
2		Advanced Settings			
		?	OK Cancel Apply		
(!)					
የዎ				◎ 开启互联网检索	
□ C: > Us	ers > 6	1689 > AppData > Roaming > JetBrain	> IdealC2023.2 > ≡ idea64.exe.vmoptions	3:15 CRLF UTF-8 💋 4 spaces	්

(2) 可根据实际需求配置,如单次接受的最大token数,代码补全粒度、补全 候选答案个数、补全抖动时间、单元测试框架选择等。



▶ 步骤三:

自定义提示词配置。

	≡ 🚺 acepilot-server ∽ 🍄 dev_AcepilotServer_V1.).6_Myh ∨	Applica	tion ~ 🗅 🗘 :	ද <u>ි</u> Q	6 – 5	×
	\equiv idea64.exe.vmoptions \times			湛卢			Ļ
3	1 -Xmx: 2 -Dfi Settings			×	⊕ ©	已登录,点击退出 E	• m
_	3 -jvm	Tools 〉 湛卢 〉 白定义提示词					

80		IOOIS 〉 准户 〉 日定义提示词	$\leftarrow \rightarrow$		
	Tools	+ - a			
	Qodana 📼	Prompt Name (2)	Prompt Value ②		
	Actions on Save 🔳	API文档生成	生成遵循Swagger2.0规范的API描述JSON格式文件		
	Web Browsers and Preview	Solution and the second			
	External Tools				
	Terminal 🖉			[ZHAN/U]	
	> Android			甚卢代码大模型	
	Black			老问答,快捷键 Alt S	
	Code With Me				
. ²	> Diff & Merge				
	Features Suggester				
	Features Trainer				
	Python External Documentation				
	Python Integrated Tools 🗉				
	Server Certificates				
	Shared Indexes			」。您的智能编码助手	
	Startup Tasks			丙,我支持代码注释、单元测试、代码解释、	
	Statistic			司我间问技术问题。	
C	> Tasks 🖉 🗉			ᄷᆂᅝᇾᄔᆍ	
	XPath Viewer				
	~ 湛卢			问我: ↔ 换一换	
	代码补全			样的	
·	自定义提示词			174.3	
·					
<u> </u>	(?)		OK Cancel Apply		
			/指令 @知识库	◎ 开启互联网检索	
역					
□ C: > Use	rs > 61689 > AppData > Roaming > JetBrains >	> IdealC2023.2 > ≡ idea64.exe.vmoptions		3:15 CRLF UTF-8 💋 4 space	es വി

下面详细描述下补全和自定义提示词的配置:

【代码补全】

(1) 触发延时:配置从停止输入到触发获取补全建议请求的延时,也可配置为

手动,此时仅当按下热键ctrl + alt+ k时获取补全建议,无触发时延。

(2) 补全偏好:

行级补全:优先保证响应速度,返回单行建议;

平衡:兼顾响应速度和补全建议长度;

最大长度:依据配置的代码补全的max_tokens,在最大允许范围内自行确定返回内容。

(3) 补全建议返回的最大token数

(4) 候选建议数量:设置返回候选条目的数量。

【自定义提示词】

(1) 如何进行自定义提示?

提示模板中可以包含 {{code}} 来标识该提示需要抓取用户框选的代码,一并提交湛卢处理。提示模板中可以包含用户填空,可以使用 {{input}} 来标识 需要填写对应位置的信息后再通过主动点击发送图标 提问。为了更好地提示 用户,您可以为当前填空编写占位符,在内容为空时显示 提示。如 {{input:language}} 将会显示可编辑文本框,并以 language 作 为提示信息占位显示。以下是几个示例:

编写一个 Dockerfile 模板

将{{input:十进制数}}转换为十六进制

提取以下代码中的所有函数签名

{{code}}

将以下代码翻译为{{input:目标语言}}语言代码

{{code}}

(2) 如何使用?

用户可根据实际需求,自定义提示模板,以更好地适用于您的使用场景。在输入框点击"**/快捷指令**",插件将在上方生成对应的操作按钮。

▶ 步骤四:

配置完成后重启IDE,即可体验湛卢智能编码助手的各项功能。

三、操作指南

> 快捷键介绍:

首先介绍下的快捷键,使用Tab键采纳推荐,使用Esc键取消推荐,当触发模式 为自动触发时,使用Enter键或Ctrl+K触发多行补全;当为手动触发模式, 仅 Ctrl+K触发多行补全。候选数量大于 1,可以使用 Alt+[以及 Alt+]进行 翻页浏览,确定要接受的建议项后,使用 Tab 键插入编辑器。您也可以部分接 受结果,使用Ctrl+↓ 接受单行建议。VS Code编辑器在触发补全事件后, 如遇 用 户 的 其 他 选择、移动、输入 行为,则会取 消补全操作。VS Code和 Jetbrains 具体使用方式将在下面介绍。

(1) VS Code

快捷功能	快捷键 (Windows)	快捷键 (Mac)
召出问答框	Alt+Alt	cmd+cmd
接纳补全建议	Tab	Tab
取消补全建议	Esc	Esc
多个补全候选答案 生成时,前后选择	Alt +[(Alt+])	cmd+[(cmd+]
补全单行接受	Ctrl+↓	Ctrl+↓
手动触发	Ctrl+K	Ctrl+K

(2) Jetbrains

快捷功能	快捷键(Windows)	快捷键 (Mac)
召出问答框	Alt+S	cmd+S
接纳补全建议	Tab	Tab
取消补全建议	Esc	Esc
多个补全候选答案生 成时,前后选择	Alt +[(Alt+])	cmd+[(cmd+]
补全单行接受	Ctrl+Alt+↓	Ctrl+cmd+↓
手动触发	Ctrl+Alt+K	Ctrl+cmd+K

3.1 代码补全模式

湛卢代码大模型为多种语言和各种框架提供了建议,但特别适用于

Python · JavaScript · Java · Go · C++ ·

以下以Java语言为例,湛卢代码大模型将自动显示灰色文本建议(即自动代码补 全或代码续写)。

3.1.1 行补全

用户在IDE编写某行代码时,根据改行前序内容,编码过程中稍做停顿(默认1s,您可自行配置),自动补全该行内容,补全内容灰色显示,一经触发,编辑器状态栏右下的湛卢()》)图标将指示现在的请求状态。按下**Tab**键接受补全自动将代码补充,按下**Esc**键或鼠标焦点移开则取消此次补全。



3.1.2 块补全

用户在IDE编写代码时,当待触发推荐的上下文,具备明显的完整逻辑关系(如一个新的方法、函数、判断、循环体等),当为自动触发模式时,按下Enter键或按下Ctrl+K可推荐一个完整的代码块(您可自行配置手动触发或者自动触发)。 一经触发,编辑器状态栏右下的 湛卢() 图标将指示现在的请求状态。 根据用户编写的上下文代码情况,如果能触发后端代码补全,则出现补全内容,

如果不能触发代码补全,则不会显示补全内容。按下**Tab**键接受补全自动将代码补充,按下**Esc**键或鼠标焦点移开则取消此次补全。

Ш	Ξ [A acepilot-server 🗸 🍄 dev_AcepilotServer_V1.0.6_Myh 🗸		Application	rv ⊳ ¢t i	육 Q @ -	0	×
	er.java	© PromptRepository.java © DateTimeUtils.java ©	SnowflakeldGenerator.java ×	CodegenRepository.java	luserRepository.java	CodeMessageRep	~ :	Ċ
3		}				9 1 ▲11 🛫	1 ^ ~	т
		} else {						
80		sequence = 0L;						
00		}						
• • • •								
		lastTimestamp = <u>timestamp;</u>						
							<u>_</u>	
		<pre>return ((timestamp - twepoch) << timestampLeft</pre>	tShift)					
		(datacenterId << datacenterIdShift)						
		(workerId << workerIdShift)						
		sequence;					_	
							_	
		2 usages 🔺 mayuhang						
		<pre>protected long tilNextMillis(long lastTimestamp) {</pre>	{					
	56	long timestamp = timeGen();						
		while (timestamp <= lastTimestamp) {						
		<pre>timestamp = timeGen();</pre>						
		}						
		return timestamp;						
		块级补全,按下Tab键接受						
		2 usages 🔹 mayuhang						
C		protected long timeGen() {						
~		peturn System currentTimeMillis():						
⊘								
\diamond		¥					1	
*								
P ₂							_	
(!)								
약								
lot-se	rver-dat	a-consumer > src > main > java > com > chinamobile > cmss :	> acepilot > server > data >	utils 🔸 🌀 SnowflakeldGenerator	r > 🎯 tilNextMillis 56:	9 CRLF UTF-8 💋 4	spaces	ර

3.1.3 注释生成补全

用户在IDE编写代码时,在注释中例如写"//交换两个元素",按下Enter键 或 Ctrl+K,触发代码补全,在IDE中显示补全内容。按下Tab键接受补全自动将 代码补充,按下Esc键或鼠标焦点移开则取消此次补全。

3.2 Chat模式

湛卢通过连续对话的方式为您解决问题。输入问题描述,湛卢为您生成代码。

(1) 支持多轮对话

(2) 支持指定开源知识的问答,当前已具备halo2组件库、haloe3组件库、网络安全规范、研发管理规范、代码规范、经分系统操作规范、BCOP操作手册、MOP对接规范、全栈专属云ECSO规范、OpenAPI、资源编排知识库。

(4)快速召唤代码助手

方式1:随时通过连按两次 Alt键(JetBrains为Alt+S), 召唤侧边栏代码助手。 方式2:在编辑器内单击右键选择湛卢-->功能按键,转到侧边栏代码助手提问或 使用预设提示提问。 您可以在**提问框中输入问题**,或通过**键入 / 选择预设提示**(部分操作可能需要用 户补全提示,完成补全后可以点击发送图标提交内容)开始对话。如在编辑器 内有活动的代码文件内容被选中,则会随问题一起发送;

对话结果为**流式输出**模式,即回答将在产生过程中逐步显示,以便于您提早审阅结果,在这种模式下,您可以随时点击**取消生成**按钮来终止此次回答。

对于结果中可识别的代码内容,插件将以代码框形式显示,并提供**生成到** 新文件、复制到剪贴板和插入到当前文件位置的按钮,便于您的查看和操作。

麊 关于上下文

在提问过程中,湛卢会按照算法 token 长度要求,携带合适数量的上下文信息 一起发送。

或者您可以点击湛卢侧边栏顶部标题栏上的 + 按钮来新建对话,可以清空当前上 下文内容。

✗ 历史记录

您可以通过 ↑/↓ 方向键调取当前对话窗的历史提问,通过 Enter 键直接定 位到历史对话处。

为了提供更加连续的开发体验,对于侧边栏代码助手页面,湛卢会将历史会话信

息本地缓存,这些信息会随着您的问答,操作(删除、清空)等同步更新内容。

点击对话框内"**互联网问答**"按钮,并输入相关关键词,可以发起互联网检索,湛 卢会展示检索结果,使您不离开 IDE 就能浏览需要的信息。再次点击"互联网问 答"按钮,可关闭互联网搜索,基于大模型问答。

击对话框内**@知识库**按钮,可展示所有知识库,点击选择与您问题相关的 知识库,输入关键信息,点击发送即可;知识库中**@湛卢**,支持在问答框询问 湛卢相关使用问题。

在回答结束或回答被终止后,答案底部将显示评价按钮,如果您接受了湛卢的 隐私协议,当您点击对应按钮时,我们将收集你的评价反馈,这对我们改进算 法和产品设计很有帮助,十分感谢您的支持。

下面对各个功能进行详细介绍:

3.2.1 生成代码(支持多轮对话)

(1) 直接提问

在chat框中,输入"写一段xx代码",chat框输出代码片段.

1	文件(F) 编辑(E) 选择(S) 查看(V) 转到(G)	运行(R) ・・・ ← -		⁾ zhangshiyun [SSH: 36.138.22.228]			08 — 0	
ß	湛卢	≣ 3 ∰ 0	昭 扩展: zhanlu:Al Coding Assistant-Int	telligent partner in development 🛛 🕏	2 设置 🕴 自定义快捷	操作 × 🍦 score_prompts_4o.py	🗬 test_score_p 🛛]
ρ	· 智能问答 图生代码	我 ▼	自定义快捷操作列表					
ço	我 写一段快速排序代码 (java)	16:55					+ 新建	
₽	· · · · · · · · · · · · · · · · · · ·		名称		描述		操作	
R			代码注释		/生成代码注释		۲	
_	道海		 单元测试		/生成单元测试		۲	
∍Ø	以下是一个简单的Java快速排序代码:				/解释代码			
<u>д</u>	java public class QuickSort { static int partition(int arr	□ Ξ Ξ			/代码调优纠错			-2)
\$	<pre>int pivot = arr[high]; int i = (low - 1);</pre>	[]) inc ion) i	API文档生成		生成遵循Swagger2.0规范的API	描述JSON格式文件,以代码块形式返回。	2 ⊡ ⊚	
	<pre>for (int j = low; j < hi</pre>	igh; j++) { {			将code十进制转换为十六进制		∠ ⊡ ⊚	
	<pre>int temp = arr[i arr[i] = arr[j] arr[i] = terr[;</pre>	i]; ;						

(2) 结合代码提问

用户可以在编辑器中选中代码,再输入款加上问题描述,点击发送。

用户对生成的答案可进行插入、复制、生成到新文件、点赞和踩,意见反馈。

23

(3) 历史记录查看

点击上方的历史记录按钮 🕥 ,可查看历史问答记录

点击对应的记录,可直接点定位到相应问答位置。

×	文件(F) 编辑(E) 选择(S) 查看(V) 转到(G) 运行(R) ···	。 1993年1月1日日日日日日日日日日日日日日日日日日日日日日日日日日日日日日日日日日日		
ď	· 湛卢 · □ ⑦ ⑫ · □	问答历史搜索	label.py 1 💿 🏾 🗬 filter_data.py	🔹 filter_rules.py 1 🔹 ▷ 🗠 🛄 😶
\$	○ 智能问答 图生代码 我▼	这段代码是什么意思		
۰ و	م ع 16:57	写一段快速排序代码(Java) /生成单元测试	<pre>{}".format(len(instances),</pre>	* Bigger and *

3.2.2 解释代码

支持对用户代码进行解释,触发方式支持选中代码**右击选择按钮触发**及选中代码后在**对话框中输入命令触发**。

用户在IDE代码编辑页面选中代码进行右击,选择湛卢下的"代码解释"。

文 🕅	件(F) 编辑(E) 选择(S) 查看(V) 转到(G) 运行(R) ・	$\cdots \leftarrow \rightarrow$		₽ zhangshiyun [[SSH: 36.138.22.228]					ð X
ر <mark>ل</mark> م	海 ラ 、	S 🛱 O 1	{} chat_2024-11-15.	json 🗢 get_service	_label.py 1 🔹 🏾 🚔 f	filter_data.py 🏾 🍦 filter	r_rules.py 1 ×	🕏initpy	🕏 log_process_new.py	⊳ ∽ ⊞ …
P	智能问答 图生代码	我 ▼	data_process > prompt_fil 15 class FilterF 27 def filte	ter〉utils〉♥ filter_rules Rules: er_get_plugin_versio	s.py〉聲FilterRules n_dict(self, ins	tances, min_version)	:			Harris and Anna Anna Anna Anna Anna Anna Anna
ف] 13 مح 14 مح	 我 写一段快速排序代码(JAVA) 删除对话 》混声 以下是一个简单的Java快速排序代码实现: 	14:06	29 fo 30 31 32 33 34 35 36 36 37 37 38	湛卢 转到定义 转到声明 转到类型定义 转到引用 快速查看 Find All References	F12 F12 Shift+F12 Shift+Alt+F12	代码注释 2 前 单元测试 代码解释 代码编辑 代码编辑	标右键 in_version) ing smaller	• than {}, has	{}".format(len(inst	
A Ø	<pre>java</pre>		$ \begin{array}{c ccccccccccccccccccccccccccccccccccc$	Show Call Hierarchy Show Type Hierarchy 重命名符号 更改所有匹配项 格式化文档	Shift+Alt+H F2 Ctrl+F2 Shift+Alt+F	tances): None) and (instance ce)	e["completion"	'] != "") :		
	<pre>int temp = arr[i]; arr[i] = arr[j]; arr[j] = temp; } int temp = arr[i + 1]; arr[i + 1] = arr[high]; arr[high] = temp:</pre>		46 47 pr 48 re 49 €]∨ \$ 50 def fi	使用格式化文档 格式化选定内容 重构 源代码操作	Ctrl+Shift+R	, after filtering ha	as {}".format((len(instances)	, len(instances_cle	WHEELEN
Q	<pre>static void quickSort(int arr[], int] static void quickSort(int arr[], int] if (low < high) { int pi = partition(arr, low, H quickSort(arr, low, pi - 1); guickSort(arr, ni + 1, high); </pre>	low, i nigh);	问题 3 输出 14	通义灵码 剪切 复制 粘贴	Ctrl+X Ctrl+C Ctrl+V	筛选器((例如 text、 !exclude,	、\escape)		ρ <u>⇒</u> ∧ ×

智能助手在对话框中对该代码进行解释,可对回答进行可以进行插入及复制、生 成到新文件、差异对比、点赞和踩,意见反馈,操作同上。

触发方式2:选中后通过命令触发

用户在代码编辑页面选中代码,在对话框中输入命令"解释代码"或直接点击 "/代码解释"触发快捷操作选择解释代码,对话框中自动将选中代码带入对话中。

智能助手在对话框中对该代码进行解释,可对回答进行可以进行插入及复制、生成到新文件、差异对比、点赞和踩,意见反馈,操作同上。

26

3.2.3 生成代码注释

用户选中一个函数,则进行函数注释和函数中行间代码注释,如仅选择一 行代码,则进行该行代码注释。触发方式支持选中代码**右击选择工具触发**及选 中在对**话框中输入命令触发**:

触发方式1:右击触发

选中代码后,右击选择相应按钮触发:

文 🚺	.(件(F) 编辑(E) 选择(S) 查看(V) 转到(G) 运行(R) ··· (\rightarrow (C	${\cal P}$ zhangshiyun [SSH:	36.138.22.228]			o x
D1	^{⊯⊨}		{) chat_2024-11-15.json	🗬 get_service_labe	el.py 1 🌒 🍦 filter_	_data.py 🕏 filter_rules.py 1 🗙 🍖 _initpy	🗬 log_process_new.py	\triangleright ~ \square …
ρ	智能问答 图生代码 我	▼ da	ata_process > prompt_filter > 15 class FilterRule 27 def filter g	utils〉♥ filter_rules.py s: et plugin version d:	〉给 FilterRules ict(seff, instand	ces, min version):		A
ဠ၀	我 14:06 写—段曲读排这代码(IAVA)		29 for i 30 t		F40	代码注释		
å	删除对话		31 32 33	转到定义 转到声明	F12	单元测试 /ersion)		
₿			34 e 35	转到类型定义 转到引用	Shift+F12	代码解释 		
<u>⊡</u>	以下是一个简单的Java快速排序代码实现:		36 print 37 38选中代码	快速查看 Find All References	> Shift+Alt+F12	smaller than {}, has 代码翻译 >	<pre>{}".format(len(inst</pre>	
A	java		∄∨∣‰∨ 39 def flite 40 insta	Show Call Filerarchy Show Type Hierarchy	SIIIIt+Ait+⊓	ces):		
đ	<pre>int pivot = arr[high]; int i = (low - 1); for (int j = low; j < high; j++) { if (arr[j] < pivot) { i++;</pre>		41 for i 42 t 43 44	重命名符号 更改所有匹配项 格式化文档	F2 Ctrl+F2 Shift+Alt+F	<pre>ne) and (instance["completion"] != ""):</pre>	1	
	<pre>int temp = arr[i]; arr[i] = arr[j]; arr[j] = temp; </pre>	1	46 47 print 48 retur	使用格式化文档 格式化选定内容	Ctel : Chift : D	<pre>fter filtering has {}".format(len(instances</pre>), len(instances_cle	na anna an anna an anna an an an an an a
	代码注释		d∨∣∳ 50 def filte	源代码操作		time. end time):		
	单元测试	问	·题 3 输出 调试控	通义灵码	>	筛选器(例如 text、!exclude、\escape)	۶	⊃ <u>≡</u> ~ ×
	代码解释			剪切	Ctrl+X			
	代码纠错			复制	Ctrl+C			
	API文档生成			粘贴	Ctrl+V			

智能助手在对话框中对该代码进行解释,可对回答进行可以进行插入及复制、生成到新文件、差异对比、点赞和踩,意见反馈,操作同上。

触发方式2:选中后通过命令触发

用户在代码编辑页面选中代码,在对话框中**输入命令"生成代码注释"**或点击 "/代码注释"触发快捷操作选择"生成代码注释",对话框中自动将选中代码带 入对话中。

C.	應 ⑦ 缴 ⑦	1 {} chat_2024-11-15.json	•
ρ	智能问答 图生代码 我▼	data_process > prompt_filter > utils > ♥ filter_rules.py > ♥ FilterRules 15 class FilterRules: 27 def filter get plugin version dict(self, instances, min version):	
an 20 20 20 11 11 11 11 11 11 11 11 11 11 11 11 11	我 14:06 写一段快速排序代码 (JAVA) 删除对话	29 for instance in instances: 30 try: 31 rel_res = compare_version(instance["plugin_version"], min_version) 32 if rel_res >= 0: 33 instances_clean.append(instance) 34 except: 35 continue	
G	▲ 湛卢 以下是一个简单的Java快速排序代码实现:	36 print("Version Select: total input instance is {}, after filtering smaller than {}, has {}".format(len(inst 37	
⊥ ¢	<pre>java □ 王 任 public class QuickSort { static int partition(int arr[], int low, in int pivot = arr[high]; int i = (low - 1); for (int j = low; j < high; j++) { if (arr[j] < pivot) { i++; int temp = arr[i]; arr[i] = arr[j]; arr[j] = temp;</pre>	39 def fliter_delete_output_empty_dict(self, instances): 40 instances_clean = [] 41 for instance in instances: 42 try: 43 if (instance["completion"] is not None) and (instance["completion"] != ""): 44 instances_clean.append(instance) 45 except: 46 continue 47 print("Empty remove: total instance is {}, after filtering has {}".format(len(instances), len(instances_clean)) 48 return instances_clean 49 \$\overlish \sigma \sigm	
	单元测试 代码解释 代码纠错	问题 3 输出 调试控制台 终端 端口 100 筛选器(例如 text、lexclude、\escape)	^ ×
() () () () () () () () () () () () () (API文档生成 / ②点击"/指令" / 週知识库 ●开启互联网检索	>	

3.2.4 生成单元测试

支持用户单元测试代码生成,触发方式支持选中代码右击选择工具触发及选中在对话框中输入命令触发。

▲ 触发方式1:右击触发

用户在IDE代码编辑页面选中代码进行右击,选择湛卢下的"**单元测试**"

智能助手在对话框中对该代码进行解释,可对回答进行可以进行插入及复制、生

成到新文件、差异对比、点赞和踩,意见反馈,操作同上。

触发方式2:选中后通过命令触发

用户在代码编辑页面选中代码,在对话框中输入命令"生成单元测试"或点击 "/单元测试"触发快捷操作,对话框中自动将选中代码带入对话中。智能助手 在对话框中对该代码生成单元测试,代码片段可以进行插入及复制、生成到新文 件、差异对比、点赞和踩,意见反馈,操作同上。

3.2.5 指定代码库智能问答

支持针对指定代码仓库进行检索,或针对制定知识库问答,实现更精准的代码 生成。

代码仓库问答

点击问答输入框下方"@知识库",选择目标代码仓库,并对问题加以描述,点击发送。

Image:	× 🚺	て件(F) 编辑(E) 选择(S) 查看(V)	转到(G) 运行(R) ··· ← →	, P zhangshiyun [SSH: 36.138.22.228]	
data_process > prompt.Hme > Units > 4 filter.get > 14 filter.dets is class filters.get > 14 dos is continue	C.	湛卢	≣ 3 ∰ 0 1	🕻 chat_2024-11-15.json 🔮 get_service_label.py 1 💿 🔮 filter_data.py 🔮 filter_rules.py 1 🗙 🍨initpy	🔮 log_process_new.py 🛛 🏷 🗸
<pre>S=Retwaining(G) (wave) S=Retwaining(G) (</pre>	م م	智能问答 图生代码 我	我▼ 14:06	<pre>data_process > prompt_filter > utils ></pre>	Vicition Vicition Magnetic Mag
Image: State in the state	₽ ₽	写一段快速排序代码(JAVA)		<pre>31 31 32 33 33 34 except: 35 36 37 37 37 37 38 39 39 39 30 30 30 30 30 30 30 30 30 30 30 30 30</pre>	
<pre></pre>	G	▲港卢 以下是一个简单的Java快速排序代码	^{要现:} ②选择知识库	<pre>35 continue 36 print("Version Select: total input instance is {}, after filtering smaller than {}, has 37 return instances_clean 38 38</pre>	[]".format(len(inst
43 if (instance["completion"] is not None) and (instance["completion"] I= ""); instances_clean.append(instance) 44 except: 45 except: 46 47 print("Empty remove: total instance is (), after filtering has ()".format(len(instances), len(instances_clean) 49 () < !\$> 60 lef filter get date dict(self. instances. start time. end time): 60 0 61 0 62 100 63 0 64 0 65 0 66 0 67 0 68 0 69 0 69 0 60 0 60 0 61 0 62 0 63 0 64 0 65 0 66 0 67 0 68 0 69 0 69 0 69 0 60 0 60 0 60	⊥ Ø	iav-		<pre>39 def fliter_delete_output_empty_dict(self, instances): 40 instances_clean = [] 41 for instance in instances: 42 try:</pre>	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
◎ ③ 输出 测试控制台 终端 端□ 100 筛选器(例如 text, lexclude, \escape) 15				<pre>43 43 43 if (instance["completion"] is not None) and (instance["completion"] l= ""): 44 45 45 45 45 46 47 47 47 47 47 47 47 47 50 48</pre>	, len(instances_clε
		tarraan haloe2		问题 3 输出 调试控制台 终端 端口 100 筛选器(例如 text、lexclude、\escape)	,
	8 #3		·····································		

知识库问答

点击问答输入框下方"@知识库",选择目标知识库,并对问题加以描述,点击发送。

× 🕅	5件(F) 编辑(E) 选择(S) 查看(V) 转到(G)	运行(R) ··· ← →			♀ zhangshiyun [SSH: 36.138.22.22]					\circ
ر <mark>ل</mark> م	湛卢	≣ 5 錄 0 1	0	chat_2024-11-15.json	🕏 get_service_label.py 1 💿	🕏 filter_data.py	🔹 filter_rules.py 1 🗙	🕏initpy	🕏 log_process_new.py	⊳∽ш·
ے م	智能问答 图生代码 	#▼ ♪ ₽ ©	data_pro 15 27 29	cess > prompt_filter > utils class FilterRules: def filter_get_ for instance	<pre>s > @ filter_rules.py > & FilterRu plugin_version_dict(self, e in instances:</pre>	ules〉♡ fliter_delete_ instances, min_v	output_empty_dict version):			EXAMPLE AND
2 2 2 2 2 3	我 @移动云网络安全管理规范 多动云网络规范 副除对话:目标"@知识库"	<u>页描述</u> 15:04	30 31 32 33 34 35 36 37	try: rel if r except: con print("Vers: return inst	_res = compare_version(ins rel_res >= 0: instances_clean.append(in tinue ion Select: total input in ances clean	<pre>stance["plugin_ve nstance) nstance is {}, af</pre>	ersion"], min_versio Fter filtering small	on) .er than {}, has	<pre>{}".format(len(inst</pre>	
_⊚ ⊉	 · 湛卢 移动云网络规范强调了以下几个关键点: 1.后台服务器安全:后台服务器(如数据库) IP地址,只使用局域网地址,确保其与外部 经授权的访问。 2.运维通道控制:如果已经建立了内部运维通 	应避免配置互联网 3网络隔离,防止未 6道,后台管理入	38 39 40 41 42 43 44	①~1歩~ def fliter_delet instances_ci for instance try: if	<pre>te_output_empty_dict(self, lean = [] e in instances: (instance["completion"] is instances_clean.append(in)</pre>	, instances): s not None) and (nstance)	(instance["completic	on"] != ""):		
	口、运维和远程控制端口应严格限制,禁止 减少潜在安全风险。 3.最小化访问原则:产品设计时应遵循最小机 略应根据租户的实际需求进行最小化配置, 访问权限。云上产品会根据租户是否订购等 络策略。	L向互联网开放,以 Q限原则,即网络策 只提供必要的网络 转定服务动态调整网	46 47 48 49 50	cont print("Empty return insta の人体へ def filter get d	tinue y remove: total instance : ances_clean date dict(self, instances	is {}, after filt . start time. end	tering has {}".forma	ut(len(instances), len(instances_cle	Kintigen-
	这些规范源于《研发安全通用要求》,确保网 租户数据的私密性,符合云服务的安全标准。 计、部署和运维时,必须严格遵守这些规定。	路资源的安全性和 在进行云服务设	问题 3	输出 调试控制台	终端 端口 100		筛选器(例如 text、 !exclu	de、\escape)		₽ ☴ ^ゝ
8	生新生成 按下 Ctrl+K 键,手动触发代码补全									
£63	/指令 @知识库 ●开启互联网检索	1	>							

点击"@知识库"中的"@湛卢",支持在问答框询问湛卢相关使用问题。

3.2.6 代码调优纠错

触发方式支持选中代码右击选择工具触发及选中在对话框中输入命令触发:

触发方式1:右击触发

选中代码后,右击选择按钮"代码纠错"触发:

د 😽	ζ件(F) 编辑(E) 选择(S) 查看(V) 转到(G) 运行	ī(R) ···	\leftrightarrow \rightarrow (♀ zhangshiyun [SSH: 36.138.22.2	228]				0 X
ر <mark>ل</mark> ا	湛卢	≣ © ∰ (୭ <mark>1</mark>	{} chat_2024-11-15.json	🔮 get_service_label.py 1 🏾	🕏 filter_data.py	🕏 filter_rules.py 1 🗙	🍖initpy	🔹 log_process_new.py	⊳ ~ Ш …
ρ	智能问答 图生代码	Į	tt ▼ data 15 27	_process > prompt_filter > utils class FilterRules: def filter get r	s〉∉ filter_rules.py>锋Filter n]ugin version dict(self	Rules . instances. min ver	sion):			
ço	重新生成	८ए©	29 30	for instance try:	e in i <u>湛</u> 卢	>	代码注释			
å	我	15:04	31 32 33	rel_ if n	_res = ^{牧到定义} rel_re 转到声明 insta	FIZ	单元测试	の鼠;	标右键	
₿	@移动云网络安全管理规范移动云网络规范 删除对话		34 35 36	except: cont print("Vers	特到会型定义 tinue 转到引用 ion Se 中读香着	Shift+F12	代码纠错	has {	}".format(len(inst	
_ø -	▲ 【 湛卢		37 38	return_inst 122	ances Find Ail References	Shift+Alt+F12	代码翻译			
4	移动云网络规范强调了以下几个关键点:		39 40	def fliter_delet instances_cl	te_out Show Call Hierarchy 1ean = Show Type Hierarch	y Shift+Alt+H Iy				
Ð	 1. 后台服务器安全:后台服务器(如数据库)应遗 IP地址,只使用局域网地址,确保其与外部网络 经授权的访问。 	避免配置互联网 3隔离,防止未	41 42 43	for instance try:	e in i 重命名符号 (insta 更改所有匹配项	F2 Ctrl+F2	tance["completion	n"]!= ""):		
	 运维通道控制:如果已经建立了内部运维通道, 口、运维和远程控制端口应严格限制,禁止向互 减少潜在安全风险。 	后台管理入 1联网开放,以	44 45 46	except: cont	Insta 格式化文档 tinue 使用格式化文档	Shift+Alt+F	ng has ()" farmed	t(]an(instance)	lon/instances als	Kalenaria
	 3.最小化访问原则:产品设计时应遵循最小权限。 略应根据租户的实际需求进行最小化配置,只想访问权限。云上产品会根据租户是否订购特定 	?则,即网络策 是供必要的网络 服务动态调整网	47 48 49	return insta	y Tellic 格式化选定内容 ances重构 渡代码操作	Ctrl+Shift+R		t(Ten(Instances),	Ten(Instances_cie	
	络策略。 这些规范源于《研发安全通用要求》,确保网络资	资源的安全性和	50 问题	def filter get d 3 输出 调试控制台	date c 终端 通义灵码	>	me): 訪选器(例如 text、lexclud	le、\escape)		ρ≣ ∧ ×
	租户数据的私密性,符合云服务的安全标准。在进 计、部署和运维时,必须严格遵守这些规定。	抗云服务设				Ctrl+X Ctrl+C				
0	重新生成	₽₽©			粘贴	Ctrl+V				

智能助手在对话框中生成对该段代码问题的描述、优化后的代码等,优化后的代码片段可以进行插入及复制、生成到新文件、差异对比、点赞和踩,意见反馈, 操作同上。

触发方式2:选中后通过命令触发:

用户在代码编辑页面选中代码,在对话框中输入命令"代码调优"或点击"/代码 纠错"触发快捷操作,对话框中自动将选中代码带入对话中,后续操作同上。

>	文件(F) 编辑(E) 选择(S) 查看(V) 转到(G) 运行(R) ····		,⊅ zh	angshiyun [SSH: 36.138.22.228					o >
G	är ⇒ 3 🕸 (୭ <mark>1</mark>	🚯 chat_2024-11-15.json 🛭 🔹 g	get_service_label.py 1 🔹	🕏 filter_data.py	🔹 filter_rules.py 1 🗙	🕏initpy	🕏 log_process_new.py	\triangleright \checkmark \square \cdots
ر م	智能问答图生代码 我	t≹ ▼ 1 2 2	a_process > prompt_filter > utils > 5 class FilterRules: 7 def filter_get_plugi 9 for instance in	<pre>filter_rules.py > ♣ FilterRul n_version_dict(self, instances:</pre>	^{les} instances, min_v	ersion):			ETTER Barrow Martin """"""""""""""""""""""""""""""""""""
° æ	ZHAN/U 湛卢代码大模型		0 try: 1 rel_res 2 if rel_r 3 inst 4 except:	= compare_version(ins es >= 0: ances_clean.append(in	stance["plugin_ve nstance)	rsion"], min_versio	n)		
F	双击 Alt ,快速唤起智能问答		5 continue 6 print("Version S 7 return instances 8	elect: total input in _clean ①选中代码	istance is {}, af	ter filtering small	er than {}, has	{}".format(len <mark>(</mark> inst	All
Ð			9 def fliter_delete_ou 1 instances_clean 1 for instance in 2 try: 3 if (inst 4 inst 5 except: 6 continue	<pre>tput_empty_dict(self, = [] instances: ance["completion"] is ances_clean.append(in</pre>	instances): not None) and (nstance)	instance["completio	n"]·!=·""):		
	Hi,我是湛卢代码大模型,您的智能编码助手		7 print("Empty rem 8 return instances 9 ① ① 〇 〇 〇 〇 〇 〇 〇 〇 〇 〇 〇 〇 〇 〇 〇 〇 〇 〇	ove: total instance i _clean dict(self. instances.	s {}, after filt	ering has {}".forma	t(len(instances)), len(instances_cl€	
	单元测试 代码解释 代码纠错 400文形体 ct		题 3 输出 <u>调试控制台</u> 终端	端口 100		筛选器(例如 text、 !exclud	ie、\escape)		0 ☴ ^ >
833 833 833	API又有主成 / ②点击"/指令" /指令 @知识库 ●开启互联网检索] _,							

3.2.7 代码翻译

触发方式支持选中代码右击选择工具触发及选中在对话框中输入命令触发:

触发方式:右击触发

选中代码后,右击选择按钮"代码翻译"触发:

8	文件(F) 编辑(E) 选择(S)	查看(V) 转到(G) 运行(R) ···	÷	→		${\cal P}$ zhangshiyun [SSH: 36	5.138.22.228]				· o ×
ر ل	湛卢	5 ≡	0 # O	1 () cha	at_2024-11-15.json	🔮 get_service_label.p	oy 1 🌒 🏾 🍨 filter_data	a.py 🏾 🇬 filter_rules.py 1	× 🔹 _initpy	🍨 log_process_new.py	⊳ ~ □ …
P	智能问答 图生代码		我 ▼	data_proces 15 cl 27	s>prompt_filter> ass FilterRules def filter_ge	utils > 🌻 filter_rules.py > ' : t_plugin_version_dic	铪 FilterRules t(self, instances,	, min_version):			Description of the second seco
°°° ℃	4	ZHAN/U 湛卢代码大模型		29 30 31 32 33	for inst try:	湛卢 转到定义 转到声明	> F12	代码注释 单元测试	sion)		
	双击	Alt ,快速唤起智能问答		34 35 36	exce print("\	转到类型定义 转到引用 快速查看	Shift+F12		allon than () has	다".format(len(inst	
_⊚ ⊿					def fliter_c	Find All References	Shift+Alt+F12):	Java Python		
Ø				40	for inst try:	snow Type Hierarchy 重命名符号 更改所有匹配项	F2 Ctrl+F2	and (instance["compl	etion"] != ""):		
	Hi,我是湛卢代码大档	塑,您的智能编码助手		4 45 47 43 43	exce print("E return i	格式化文档 使用格式化文档 格式化选定内容 重构	Shift+Alt+F Ctrl+Shift+R	r filtering has {}".f	ormat(len(instances), len(instances_cl	
	代码注释				لع الله من الله الله الله الله الله الله الله الل	源代码操作 	Š	e. end time):	avaluate (accord)		0 = x
	单元测试 代码解释						Ctrl+X		exclude, (escape)		
	代码纠错						Ctrl+C Ctrl+V				
8	API文档生成					在交互窗口中运行	>				
503	/指令 @知识库	●开启互联网检索		\$		运行 Python	>)				

智能助手在对话框中生成对该段代码问题的翻译后的代码等,翻译后的代码片段 可以进行插入及复制、生成到新文件、差异对比、点赞和踩,意见反馈,操作同

_____ 。

3.2.8 图生代码

在左侧对话框标签页中选中"图生代码"标签,点击上传设计原型图,即可自动生 成相应代码。

注意:图片base64编码后大小不超过10M,支持JPG、JPEG、PNG、BMP格 式,图片的长和宽要求最大边不超1920px。

操作方式:

点击"图生代码"tab页,点击上传产品设计图:

自动生成代码如下:

>	文件(F) 编辑(E) 选择(S) 查看(V) 转到(G) 运行(R) ···· 〈			ho zhangshiy	un [SSH: 36.138.22.228]				- 0 ×
G	☞	e get	t_service_label.py 1 🏾	🍦 filter_data.py	🔹 filter_rules.py 1 🗙	🚔initpy	log_process_new.py	🗬 openai_utils_classification.py	⊳ ~ ⊡
<u>م</u>	智能问答图生代码 我,	data 1!	ta_process > prompt_filt L5 class FilterR 27 def filte	er > utils > 🌳 filter_ru ules: r_get_plugin_versi	<pre>les.py > & FilterRules ion_dict(self, insta</pre>	nces, min_versi	ion):		172 II Maria analas ang katala 172 II Maria ang katalan 172 II Maria ang katalan 172 II Maria ang katalan 172 II Maria ang katalan 172 II Maria ang katalan
ço	送港卢	29 30	29 for i 30 t	nstance in instand	ces:				The second secon
å	xulo Image: A state of the state	3: 3: 3:	31 32 33	<pre>rel_res = compa if rel_res >= @ instances_c</pre>	are_version(instance): clean.append(instanc	e["plugin_version)	on"], min_version)		
₿	<pre><c-badge :count="1" class="item"></c-badge></pre>	34 3!	34 e 35	xcept: continue			(:)+;		The second secon
<u>©</u>	<pre></pre>	30	36 print 37 retur 38	n instances_clean	total input instanc	æ <u>is </u> {}, atter	Tiltering smaller than	<pre>T {}, nas {} .Tormat(len(lnsta)</pre>	Martin Carlos and Annual Annua
Д	<pre></pre>	39	引くし続く 39 def flite 10 insta	r_delete_output_em nces clean = []	npty_dict(self, inst	ances):			

湛卢 P	layground			Vue Version 3.5.13 HaloE Version 3.5.1-alpha.1						ŝ
App.vue	+	tsconfig.json]	Emport Map	PREVIEW	JS CSS	SSR				
1 <	template≻				120					
2	<pre><div class="demo-inner-desc"></div></pre>					四小五日八十五日				
3	<pre><e-badge :count="1" class="item"></e-badge></pre>				找 必	段代码是什么意思				
4	<e-button type="primary">湛卢</e-button>									
5				0	0					
6	<pre><e-badge :count="1" class="item"></e-badge></pre>			湛卢	我这	段代码是什么意思				
7	<e-button type="secondary">我</e-button>									
8										i¥
9	<pre><e-badge class="item" type="error"></e-badge></pre>			删除对话						
10	<e-button type="error">这段代码是什么意思</e-button>									
11				湛卢						
12										
13	<pre><div class="demo-inner-desc"></div></pre>			•						
14	<pre><e-badge :count="1" class="item"></e-badge></pre>			+						
15	<e-button type="primary">湛卢</e-button>									
16				•						
17	<pre><e-badge :count="1" class="item"></e-badge></pre>			+						
18	<e-button type="secondary">我</e-button>			23						
19				•						
20	<pre><e-badge class="item" type="error"></e-badge></pre>			+						
21	<e-button type="error">这段代码是什么意思</e-button>									
22				•						
23				+						
24	<pre><div class="demo-inner-desc"></div></pre>			1						
25	<div class="button"></div>									
26	<e-tooltip content="删除对话" placement="top"></e-tooltip>			+						
27	<e-button type="text">删除对话</e-button>									
28										
29										
30	<pre><div class="button"></div></pre>									
31	<e-tooltip content="湛卢" placement="top"></e-tooltip>									
32	<e-button type="text">湛卢</e-button>									
33										

同时支持进行插入及复制、生成到新文件、差异对比、点赞和踩,意见反馈,操 作同上。

3.2.9 针对报错提示一键生成修复建议

当编译区出现编译错误或运行出现异常报错时,一键启动报错排查的智能答疑,可结合运行代码、异常堆栈等报错上下文,快速给出排查思路或修复建议代码。

(1)针对编辑区编辑错误,用户可选择红色波浪线(编译错误处),选择湛卢一键修

复功能

点击后, 在问答框中生成修复建议

(2)针对运行报的堆栈错误,用户可点击报错堆栈处湛卢图标,一键生成修复意见

3.2.10 互联网检索问答 点击对话框内"互联网问答"按钮,并输入相关关键词,可以发起互联网检索,湛卢 会展示检索结果,使您不离开 IDE 就能浏览需要的信息,实现代码知识实时检索。

3.2.11 代码评审

触发方式支持选中代码右击选择工具触发及选中在对话框中输入命令触发:

在要提交的代码文件中,右击选择按钮"代码评审"触发:

× .	文件(F) 编辑(E) 选择(S) 查看(V) 转到(G) ····	\leftrightarrow \rightarrow (₽ sujinling [SS	H: 36.138.22.228]	& ~			- 0 X
ζŋ	湛卢 录 ♡ 錄	0 🛢	check.sql 🏾 🍦 df_utils.py 🗙 🗉 membe	er_1220.csv III member_1128.csv	member_11	14.csv 🛢 insert.sql	🕏 constant.py	⊳ ~ ⊞ …
	法点 ③ ③ ④ 留部问答 图生代码 第二章 ● ○ ④ 留部问答 图生代码 正 こ ○ ○ 正 こ ○ ○ 正 こ ○ ○ 正 こ ○ ○ 正 こ ○ ○ 正 こ ○	⑦ ■ sjl > a 2 2 2 1 <	check.sql ● df_utils.py × I member acepilot-data > utils > ● df_utils.py > 1 import pandas as pd 2 import numpy as np 3 from scipy.stats import spearmann 4 import sqlite3 5 from loguru import logger 6 7 from utils.constant import db_pat 8 ● 9 class Report: 10 ● 0 11 definit(self, process_date = p	er_1220.csv 正 th 描 描 描 描 描 描 描 描 描 描 描 描 描	F12 Ctrl+F12 Shift+Alt+F12 Shift+Alt+F12 Shift+Alt+F12 Shift+Alt+F12	14.csv ■ insert.sql 代码注释 単元测试 代码解释 代码明 不 の 代码部译 へ 、 、 、 、 、 、 、 、 、 、 、 、 、	Constant.py	 V Martinette V Mar
	数据转换 vue2转vue3 代码评审 清空对话框	22 22 24 29 20	22 If "率" in col or "百 23 If df[col].max() 24 logger.warnin 25 df[col] = df[col]	比 in 更改所有匹配项 > 1: 重构 g(f"{c 源代码操作 .apply = · · · · · · · · · · · · · · · · · · ·	Ctrl+F2 Ctrl+Shift+R Ctrl+X	[{df.columns}]") (x) else x)		
	清除上下文	Material and a second se		复制 粘贴	Ctrl+C Ctrl+V	exclude, (escape)		,≫ =x ∧ ∧
8				在交互窗口中运行	>	打开要升级的vue文件		\$\$ ∨ X
503	/指令 @知识库 ● 🦪 🚀	,		运行 Python	>	hanlu:Al Coding Assistant-Inte	lligent partner in developm	ent

智能助手在对话框中对当前代码文件的增量代码生成评审意见和优化建议。

触发方式2:选中后通过命令触发:

用户在代码编辑页面,对话框中输入命令"代码评审"或点击"/代码评审"触发快捷操作,对话框中自动将自动对当前代码文件进行代码评审,生成评审建议。

3.2.12 Vue升级

触发方式支持在对话框中输入命令触发:

用户在要升级的Vue代码文件页面,对话框中输入命令 "vue2转vue3" 或点击 "/vue 转vue3" 触发快捷操作,对话框中自动将自动对当前代码文件进行代码进行vue升级。

β	湛卢	≣ ♡ ‡ 0 <	echeck.sql	🗬 df_utils.py 🗙	member_1220.csv	member_1128.csv	member_1114.csv	🛢 insert.sql	🍦 constant.py	\triangleright \checkmark \square \cdots
Q % 20	智能回答 图生代码 さいまたの さいまたの との での 大概	我▼	jl > acepilot-data 1 import 2 import 3 from s 4 import 5 from 1 6 7 from u	> utils > 🗬 df_utils.py pandas as pd numpy as np cipy.stats import sqlite3 oguru import logg	/≻… spearmanr ger					
₽ C© A	双击 Alt ,快速唤起智能问答		9 class 10 11 de 12 13	<pre>report: finit(self, self.conn = sql solf nprecess da</pre>	process_date): ite3.connect(db_path)					
ß	代码注释 单元测试		13 14 15 de 16	f _df_format(self	, df): columns:					The second secon
U	代码解释 代码纠错 API文档生成		17 18 19 20 21 22	if col in [df[col] else: df[col] if '牽' in	"邮箱", "姓名", "部门 = df[col].fillna("") = df[col].fillna(0) col or '占比' in col:	/公司" ," 班组" ," 角色"];			
	数据转换 vue2转vue3 1、時件単 造内マッドチャ		23 24 25 26	if df[c log df[col]	<pre>col].max() > 1: ger.warning(f"{col} + = df[col].apply(lambu </pre>	df[col].max()} exceed	d 100%. index [{df.col ') if pd.notna(x) else	umns}]") x)		Particular and and any and any
Ø	肩 立对 适框 清除上下文 help		问题输出调	试控制台 终端 端[筛选器(例如 text、 lexclude、	\escape) 的vue文件		ρ≣ ~ × ⊕ ~ ×

3.3 AI程序员

3.3.1 项目级单元测试

在对话框中输入命令触发,需指定项目文件或路径,如在对话框中输入"对 D:\Codetest\es_process批量生成单元测试",AI程序员会自动对任务进行分析和拆解, 点击"Run Command"表示接受并执行下一步,"Reject"表示不接受此次分析建议, 点击"Reject",AI程序员会重新分析,并进行下一步。

代码写好后,AI程序员会提示你保存与执行;如运行不通过,也能够自动deBug。

× >	文件(F) 编辑(E) 选择(S) 查看(V) 转到(G)					€ Codetest		& ~	0: 🗖 🖬 (1 –	D	
	湛卢	≣ ∰ 0	ру	🛢 create.sql 🏾	🛛 🛢 pre_user.sql	👻 constant.py	🔹 test.py	e test_add_partition.py: Original ↔ Zhanlu's Cl	hanges (Editable) 🏾	r ↓ 1	๗ 🛛	
	 智能问答 图生代码 APEPS 容。接下来,我们将为这个文件编写单元测 了编写有效的单元测试,我们需要了解每个 功能和预期行为。我们将从 parser_args 始,因为它处理命令行参数,是程序的入口 一。 我们将创建一个新的测试文件 test_add_partition.py,并在其中编写 试。首先,我们将编写 parser_args 函数 用例。 [1] 湛卢想要创建文件: es_process/test_add_partition.py ① 请求中 好的,我们已经成功创建了 test_add_partition.py 文件,并为 parser_args 函数编写了单元测试。接下 们将继续为 add_partition.py 中的其他 写单元测试。 我们将从 get_date_list 函数开始。这个成一个日期列表,我们需要确保它在各种编 下都能正确工作 	我▼ 就。数数开 〕 〕 〕 〕 〕 〕 〕 〕 〕 〕 〕 〕 〕	es_prod 1 2 3 4- 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26	<pre>cess > @ test_ac import unitto import argpan from unittes: from es_proce class TestAdd @patch(' def test args self self self self self self self sel</pre>	dd_partition.py > est rse t.mock import pat ess.add_partition dPartition(unitte sys.argv', ['add_ _parser_args_crea = parser_args() .assertTrue(args. .assertFalse(args .assertFalse(args .assertFalse(args .assertIsNone(arg .assertIsNone(arg .assertIsNone(arg .assertIsNone(arg .assertIsNone(arg .assertIsNone(arg .assertFalse(args .assertFalse(args .assertFalse(args .assertFalse(args .assertFalse(args .assertFalse(args .assertFalse(args .assertFalse(args .assertFalse(args .assertFalse(args	ch import parser_arg st.TestCase): partition.py', ' te_table(self): create_table) delete_table) delete_table) add_partition) transfer) alter) s.sql) args.start_date) s.end_date) partition.py', ' te_table(self): create_table) delete_table) add_partition) transfer) alter)	a,])	<pre>1 import unittest 2 import argparse 3 from unittest.mock import pat 4+ from es_process.add_partition 5 6 class TestAddPartition(unitte 7 @patch('sys.argv', ['add_ 8 def test_parser_args_crea 9 args = parser_args() 10 self.assertFalse(args 11 self.assertFalse(args 12 self.assertFalse(args 13 self.assertFalse(args 14 self.assertFalse(args 15 self.assertIsNone(arg 16 self.assertIsNone(arg 17 self.assertFalse(args 18 19 @patch('sys.argv', ['add_ 20 def test_parser_args_dele 21 args = parser_args() 22 self.assertFalse(args 23 self.assertFalse(args 24 self.assertFalse(args 25 self.assertFalse(args 26 self.assertFalse(args 27 self.assertFalse(args 28 self.assertFalse(args 29 self.assertFalse(args 20 self.assertFalse(args 21 self.assertFalse(args 22 self.assertFalse(args 23 self.assertFalse(args 24 self.assertFalse(args 25 self.assertFalse(args 26 self.assertFalse(args 27 self.assertFalse(args 28 self.assertFalse(args 29 self.assertFalse(args 20 self.assertFalse(args 21 self.assertFalse(args 22 self.assertFalse(args 23 self.assertFalse(args 24 self.assertFalse(args 25 self.assertFalse(args 26 self.assertFalse(args 26 self.assertFalse(args 27 self.assertFalse(args 28 self.assertFalse(args 29 self.assertFalse(args 20 self.assertFalse(args 21 self.assertFalse(args 22 self.assertFalse(args 23 self.assertFalse(args 24 self.assertFalse(args 25 self.assertFalse(args 26 self.assertFalse(args 27 self.assertFalse(args 28 self.assertFalse(args 29 self.assertFalse(args 20 self.assertFalse(args 20 self.assertFalse(args 21 self.assertFalse(args 22 self.assertFalse(args 23 self.assertFalse(args 24 self.assertFalse(args 25 self.assertFalse(args 26 self.assertFalse(args 27 self.assertFalse(args 28 self.assertFalse(args 29 self.assertFalse(args 20 self.assertFalse(args 20 self.assertFalse(args 20 self.assertFalse(args 21 self.assertFalse(args 22 self.assertFalse(args 23 self.assertFalse(args 24 self.assertFalse(args 25 self.assertFalse(args 26 self.assertFalse(args 27 self.assertFalse(args 28 self.assertFalse(args 29 self.assertFal</pre>	<pre>ch import parser_arg est.TestCase): partition.py', '-o te_table(self): create_table) delete_table) delete_table) add_partition) transfer) alter) s.sql) args.start_date) gs.end_date) partition.py', '-o ete_table(self): create_table) delete_table) add_partition) transfer) alter)</pre>	35, get_0 :']) 1'])	- late_lis	ť
0	Fapiliz正確工1F。 Ch 湛卢想要编辑文件: es_process/test_add_partition.py Save Reje	~ ect	问题	输出调试控制	<u> 台</u> 终端 端口			筛选器(例如 text、lexclude、\escape)		ې () = ^	×
503	[M	>									

刘 文	件(F) 编辑(E) 选择(S) 查看(V) 转到(G) ···		\leftrightarrow \rightarrow		₽ Code	test	8∼	08 🔳 🖬] [] -	- Ō	×
4	湛卢	≣ ∰ 0	pre_user.sql	🔹 constant.py	🇬 test.py	test_add_partition.py	🕏 test_data_store.py	📌 _initpy: New File (Editable) 🗙		¶ @ []	
Q	的 智能问答 图生代码 AI程序员	我▼	es_process > 👻 _ 1	_initpy			1				
	的test_add_partition.py能否运行										
fo											
à	ば上										
n0	 ✓ 请求完成 ▶ 湛卢想要执行指令: 										
Ш	python es_process/test_add_partition.py										
Ē	> Command Output										
Д	● \#+℃-₽										
 	看起来 test_add_partition.py 文件中导)	∖模									
LD	块的方式存在问题。Python 在当前日录下场个es_process 模块。我们需要确保 Python 能	▶到 够正									
•	确识别 es_process 目录为一个包。可以通过 es_process 目录下添加一个initpy	拉在 文件									
	来解决这个问题。										
	C+ 湛卢想要创建文件:										
	es_process/initpy	~									
	⑦ 请求中 □ 洪古相西地与地会。										
			问题输出源	試控制台 终端 う う	端口					+~ … ^	×
	python es_process/test_add_partition.py		PS D:\Codetest Traceback (mos	python es_proce t recent call lag	ess/test_add_parti st):	tion.py				区 湛卢 es_	<u>Д</u> pro
		t second	File "D:\Coo from es pr	letest\es_process ocess.add partiti	\test_add_partitio	n.py", line 4, in <module args, get date list, crea</module 	e> nte partition table, add	partition, delete partition		と powersh	ell
8	Run Command Reject		ModuleNotFound • PS D:\Codetest	Error: No module	named 'es_process						
£63	请输入您的任务	1							a		

除项目级单元测试外,AI程序员还支持项目级代码生成,以及项目级Vue2升级Vue3,同上只需在对话框中输入命令,如"用python写一个贪吃蛇html小游戏"、"对./XX文件中的所有Vue文件从Vue2升级到Vue3"即可。

3.4 常见问题解答

浴 湛卢支持哪些语言?Which languages does ZhanLu support?

湛卢支持C/C++、Java、Python、Go、JavaScript、TypeScript、Rust、 PHP、Kotlin、ObjectC等100+主流编程语言。

⋧ 是否会上传代码?上传哪些代码?

正常使用湛卢过程中需要利用正在编写的文件中的上文以及其他相关文件,来 推理出接下来的下文。因此确实会需要通过网络将编写的代码文件内容传输给 模型推理。我们承诺使用加密传输并保证不使用这部分上传的代码做任何 积累、分析、加工等后续动作。

✗ 生成的内容不准是怎么回事?

大模型类产品的本身输出内容存在一定的不准确性,同时我们也建议您进行更 具体的提问,并结合代码片段,提供足够的上下文信息,优化代码结构,添加 详细的注释,确保使用最新版本的湛卢插件,保持网络稳定,从而提升问答效果。

增加-Dfile.encoding=UTF-8 后重启IDE。

期待大家在使用过程中,动动你们的小手,积极"点赞"或"拉踩",帮助我们更 好地优化模型!

