目　录

CC GUI - AI 编程助手插件

CC GUI - AI 编程助手插件

CC GUI —— IntelliJ IDEA 的 AI 编程助手插件

> *在 JetBrains IDE 内深度集成 Claude Code 与 Codex，支持与 CC Switch 无缝配合*

一、项目简介

CC GUI（原 Claude Code GUI）是一款 IntelliJ IDEA 插件，为 Claude Code 和 OpenAI Codex 提供可视化图形界面。它将 AI 编程能力直接嵌入 IDE，让开发者在不离开编码环境的情况下，享受 AI 辅助编程的便利。

- GitHub：github.com/zhukunpen...
- 原名：Claude Code GUI（为避免商标风险已更名）
- 协议：MIT 开源
- 支持 IDE：IntelliJ IDEA、WebStorm、PyCharm、GoLand、Rider 等 JetBrains 全系

二、核心功能

2.1 双 AI 引擎支持

引擎	提供商	说明
Claude Code	Anthropic	上下文感知编程助手，支持 Opus 系列模型
OpenAI Codex	OpenAI	强大的代码生成引擎

2.2 智能对话系统

- 上下文感知：自动识别当前项目代码上下文
- @file 引用：精准引用特定文件作为对话上下文
- 图片发送：支持上传图片描述视觉需求
- 对话回溯：灵活调整历史对话记录
- 增强提示词：自动优化用户输入，提升 AI 理解准确度

2.3 Agent 自动化

- 内置 Agent：自动化执行复杂多步骤任务

- Skills 命令：斜杠命令系统（ `/init` 、 `/review` 、 `/test` 等）
- MCP 扩展：支持 Model Context Protocol 服务器扩展 AI 能力

2.4 开发者体验优化

功能	说明
代码 DIFF 对比	直观展示 AI 修改前后的代码差异
文件导航	点击即可跳转到相关代码位置
代码跳转	支持跳转到定义、查找引用
主题切换	明暗双主题，适配 IDE 外观
字体同步	自动跟随 IDE 字体设置
国际化	中/英文自动切换

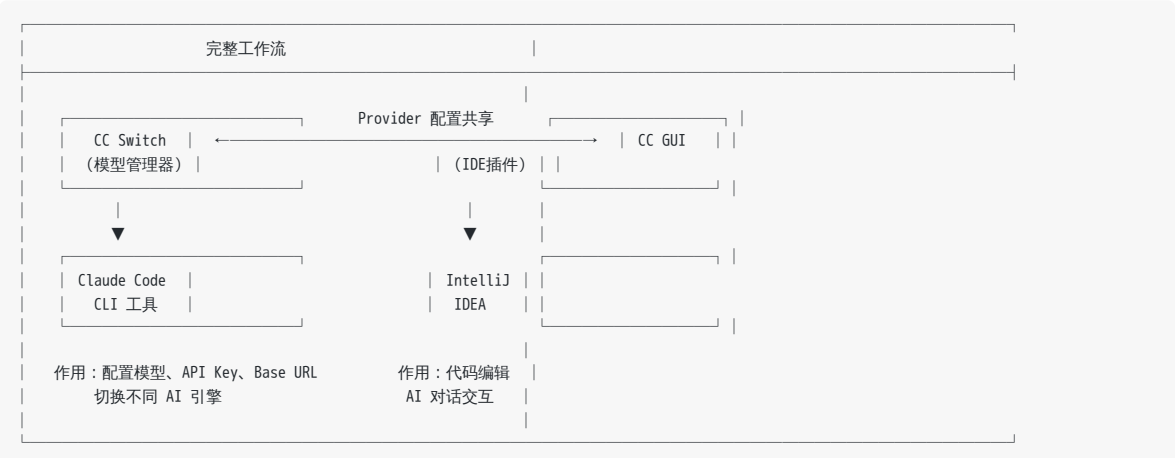
2.5 会话管理

- 历史会话记录与搜索
- 会话收藏（标记重要对话）
- 消息导出（保存为文件）
- 使用量统计分析

三、与 CC Switch 配合使用

3.1 为什么需要配合 CC Switch

CC GUI 是 IDE 内的操作界面，CC Switch 是模型配置管理器。两者配合：



3.2 配合流程

第一步：在 CC Switch 中配置模型

```
打开 CC Switch → 添加 Provider → 填写 API 信息 → 激活
```

支持的 Provider 类型：

- OpenAI Compatible（DeepSeek、智谱、硅基流动等）
- 官方 Claude（需特殊网络）
- 官方 Codex（需特殊网络）

第二步：在 CC GUI 中加载配置

CC GUI 会自动检测 CC Switch 的配置，或手动同步：

```
CC GUI 设置 → Provider → 选择「使用 CC Switch 配置」
```

第三步：在 IDE 中使用

```
打开项目 → 点击 CC GUI 面板 → 选择 AI 引擎 → 开始对话
```

3.3 配置共享说明

配置项	存储位置	共享方式
API Key	CC Switch 加密存储	CC GUI 读取调用
Base URL	CC Switch 配置文件	两者同步
模型选择	CC Switch 激活状态	CC GUI 自动跟随
会话历史	CC GUI 本地存储	独立管理

四、安装方法

4.1 方式一：插件市场安装（推荐）

```
IntelliJ IDEA → 设置 → 插件 → 市场 → 搜索 "CC GUI" → 安装 → 重启 IDE
```

4.2 方式二：离线安装

```
1. 从 GitHub Releases 下载插件包（.zip）
2. IDEA → 设置 → 插件 → 从磁盘安装 → 选择 .zip 文件
3. 重启 IDE
```

4.3 方式三：源码构建

```
git clone https://github.com/zhukunpenglinyutong/jetbrains-cc-gui.git
cd jetbrains-cc-gui
./gradlew clean buildPlugin
# 插件包位于 build/distributions/（约 40MB）
```

五、首次使用

5.1 打开 CC GUI 面板

```
IDEA 右侧工具栏 → 点击 "CCG" 图标 → 打开 AI 对话面板
```

5.2 连接 CC Switch

```
CC GUI 面板 → 设置齿轮 → Provider → 选择「CC Switch」→ 测试连接
```

5.3 开始对话

```
1. 在编辑器中选中代码
2. 右键 → "发送到 CC GUI" 或直接输入问题
3. AI 分析后给出建议
4. 点击「应用修改」或「复制代码」
```

六、典型使用场景

场景一：代码审查

```
选中代码 → 输入 "/review" → AI 自动分析代码问题
→ 查看 DIFF → 点击应用修改
```

场景二：生成测试

```
选中函数 → 输入 "给这个函数写单元测试" → AI 生成测试代码
→ 查看 DIFF → 应用到测试目录
```

场景三：重构建议

```
选中类文件 → 输入 "重构这个类，使用设计模式优化"
→ AI 给出重构方案 → 逐步应用
```

场景四：Bug 修复

```
选中报错代码 → 输入 "修复这个报错"
→ AI 分析错误原因 → 给出修复代码
```

七、安全审计

项目承诺定期安全审计：

- 每个小版本发布前进行 `/security-review` 审计
- 每 10 个小版本进行全面 `claude-code-security` 审计

八、常见问题

问题	解决
CC GUI 无法连接 CC Switch	确认 CC Switch 已启动并激活 Provider
模型响应慢	在 CC Switch 中切换到更快的 Provider
中文显示乱码	IDEA 设置 → 编辑器 → 文件编码 → UTF-8
代码应用失败	检查文件是否被其他程序锁定
插件崩溃	检查 IDEA 版本兼容性，更新到最新版

九、相关资源

资源	地址
CC GUI GitHub	github.com/zhukunpen...
CC Switch GitHub	github.com/farion123...
Claude Code 官网	code.claude.com
JetBrains 插件市场	IDEA 内搜索 "CC GUI"

十、与 CC Switch 的关系总结

维度	CC Switch	CC GUI
定位	模型配置管理器	IDE 内 AI 操作界面
运行环境	桌面系统	JetBrains IDE 内
核心功能	管理 API、切换模型	代码交互、AI 对话
是否必须	是（提供模型接入）	否（可选增强体验）

维度	CC Switch	CC GUI
配合效果	配置一次，多处使用	编码不离 IDE，效率最高

> *建议：先安装配置好 CC Switch，再安装 CC GUI，两者配合使用效果最佳。*