

CodeArts IDE

常见问题

文档版本 01
发布日期 2025-09-17



版权所有 © 华为云计算技术有限公司 2025。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

商标声明



HUAWEI和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受华为云计算技术有限公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为云计算技术有限公司对本文档内容不做任何明示或暗示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

华为云计算技术有限公司

地址：贵州省贵安新区黔中大道交兴功路华为云数据中心 邮编：550029

网址：<https://www.huaweicloud.com/>

目录

1 通用常见问题	1
1.1 IAM 账号登录时如何开启编程访问权限	1
1.2 创建工程时勾选 git 仓库，添加远程仓库后推送代码时报错	2
2 CodeArts IDE for C/C++ 常见问题	4
2.1 如何处理 C/C++ 组件激活失败报错的问题	4
2.2 如何配置外部 terminal?	5
2.3 如何处理 CMake Build Tool 插件运行调试时中文乱码的问题?	7
3 CodeArts IDE for Java 常见问题	11
3.1 如何处理报错“等待语言服务初始化完成”?	11
3.2 如何从 IDE 直接打开文件实际存放路径?	12
3.3 如何处理修改文件树后颜色变化和使用中报错不高亮的问题?	12
3.4 如何处理终端输出中文乱码问题?	13
3.5 如何设置默认的 Maven 配置文件 settings.xml	15
3.6 如何设置自定义代码片段和代码模板?	17
3.7 如何设置 xml 文件中自动补全结束标签对?	21
3.8 如何设置自动展示工程中的 TODO 列表?	22
4 CodeArts IDE for RemoteShell 常见问题	24
4.1 如何处理登录 RemoteShell 客户端失败的问题?	24
4.2 如何处理 proxy 配置常见问题	25
4.3 如何处理主机连接失败的问题	26
4.3.1 如何处理新建主机连接失败的问题?	26
4.3.2 如何处理连接远程主机失败，出现报错的问题?	27
5 CodeArts IDE 鸿蒙版本常见问题	28

1 通用常见问题

1.1 IAM 账号登录时如何开启编程访问权限

- 若未创建IAM账号，请参考[创建IAM用户](#)注册或创建。创建IAM用户时，请确保“编程访问”已勾选，开启编程访问权限。

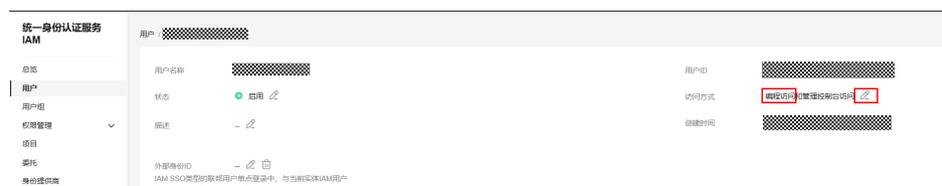
图 1-1 勾选编程访问方式



The screenshot shows the 'Create User' page in the IAM console. It is divided into three steps: 1. Configure user basic information, 2. Add user groups (optional), and 3. Complete. Under 'Configure user basic information', there are fields for 'Username', 'Email address', and 'Mobile number'. Below these is the 'Access Mode' section, which has two options: 'Program Access' (checked) and 'Management Console Access' (checked). The 'Program Access' option is highlighted with a red box. A tooltip icon is visible next to the 'Program Access' option.

- 若已创建IAM账号，需要确保该账号的“编程访问”已开启。若未开启，请联系管理员编辑如下访问方式，添加“编程访问”权限。可参考文档[管理IAM用户的基本信息](#)进行操作。

图 1-2 添加编程访问权限



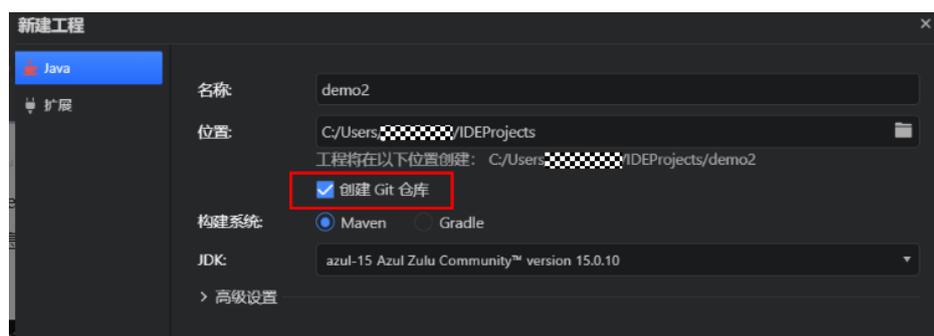
The screenshot shows the 'IAM User Management' page. On the left is a navigation menu with options like 'Overview', 'Users', 'User Groups', 'Roles', 'Projects', 'Delegation', and 'External Providers'. The main area shows a table of users. The 'Access Mode' column for a user is highlighted with a red box, showing 'Program Access' and 'Management Console Access' both checked.

1.2 创建工程时勾选 git 仓库，添加远程仓库后推送代码时报错

问题现象

以创建一个Java工程为例，选中“创建Git仓库”。工程创建并且打开后，添加远程存储库。如下图所示：

图 1-3 创建 Java 工程



说明

默认远程仓库的名称是origin。本地分支是master，推送到远程的master的分支。
将本地修改的文件提交之后，再推送到远程。

- 拉取远程仓库的代码，执行“git pull origin master”，报错信息如下图所示：

图 1-4 执行 git pull origin master 报错

```
PS C:\Users\...\IDEProjects\demo1> git pull origin master
From ssh://codehub-dg-g.huawei.com:2222/.../java-demo
 * branch          master       -> FETCH_HEAD
fatal: refusing to merge unrelated histories
```

- 将本地分支推送到远程，执行“git push -u origin master”，报错信息如下图所示：

图 1-5 执行 git push -u origin master 报错

```
2025-06-28 09:44:08.425 [info] > git push -u origin master [642ms]
2025-06-28 09:44:08.426 [info] To ssh://.../java-demo.git
|! [rejected]          master -> master (non-fast-forward)
error: failed to push some refs to 'ssh://.../java-demo.git'
hint: Updates were rejected because the tip of your current branch is behind
hint: its remote counterpart. If you want to integrate the remote changes,
hint: use 'git pull' before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
```

解决方法

当操作过程中出现“fatal: refusing to merge unrelated histories”的错误，原因是两个仓库不同而导致的。需要在代码后面加上“--allow-unrelated-histories”进行允许合并，问题即可解决。

即执行“`git pull origin master --allow-unrelated-histories`”后，同步远程分支代码，关联远程分支。

2 CodeArts IDE for C/C++常见问题

2.1 如何处理 C/C++组件激活失败报错的问题

CMake 项目配置失败

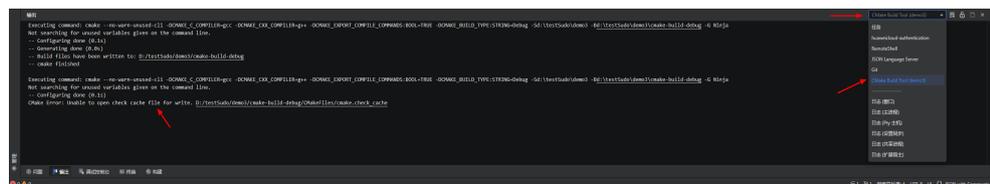
该错误表明CMake项目配置失败，并将错误的信息日志打印输出到输出窗口。

图 2-1 CMake 项目配置失败



该报错需要切换到输出窗口获取更多详细信息。此外，可以通过下拉菜单进行手动切换。

图 2-2 手动切换项目



Error: Unable to open check cache file for write

该错误表明CodeArts IDE在此工作区文件夹无写入权限，需用户自行确认此文件夹的读写权限。

Error: command ‘clangd.applyFix’ already exists

该错误表明命令clangd.applyFix被重复注册了，原因是插件冲突，目前已知的冲突插件包括clangd、5G-clangd、VSCode C/C++ Plugin、Nextcode Reference，出现此类错误时，请在插件列表中卸载以上插件。

Error: wecode-db port file has no content. Make sure there is space available for the folder “/tmp”

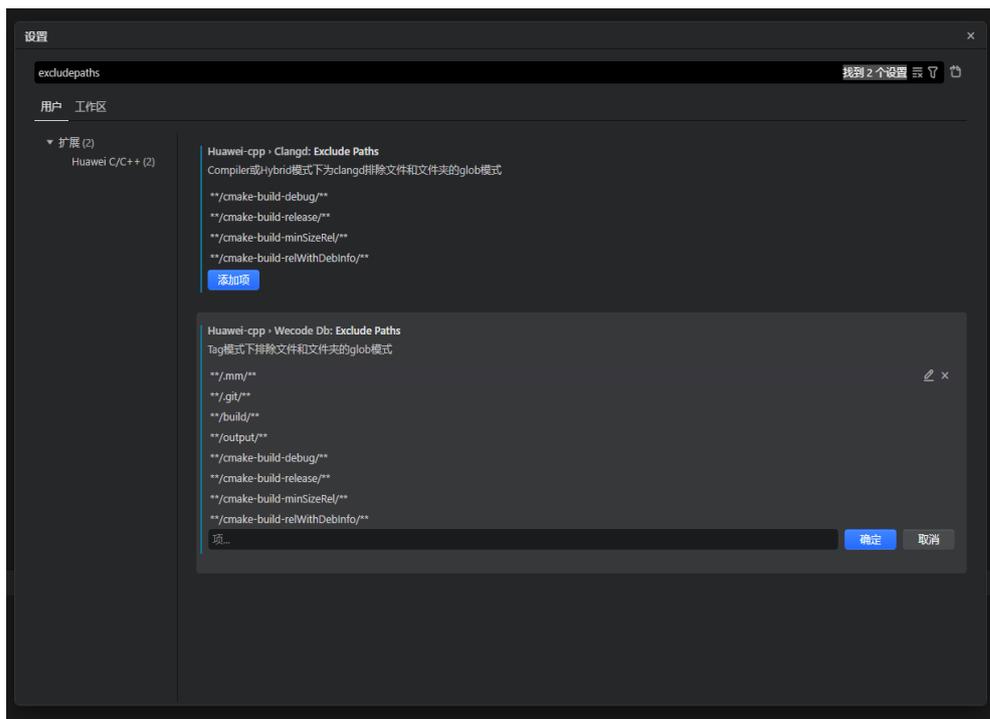
该错误表明WeCode-DB无法创建端口文件，可检查下当前的磁盘使用情况，确认磁盘空间占用是否已满，清理部分空间后重新尝试。

WeCode-DB is unable to watch for file changes in this large workspace

该警告表明WeCode-DB的文件监视程序已经耗尽文件句柄，文件句柄通常由操作系统决定，可通过cat /proc/sys/fs/inotify/max_user_watches查看，这种情况通常发生在工程非常大，包含的目录非常多的情况下，有以下两种方法解决：

- 在CodeArts IDE的设置中搜索excludepaths，尽可能的将较大的文件夹排除，如output,build目录，例如：

图 2-3 搜索 excludepaths

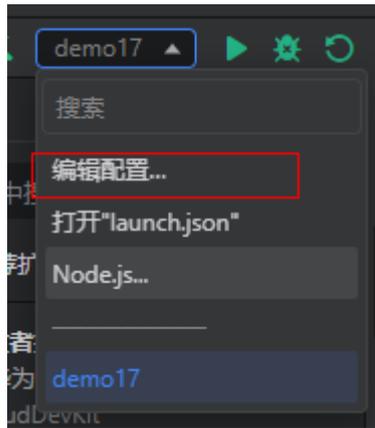


- 增加操作系统的可监控文件句柄数量，执行以下两条命令，将数量增加至524288，该操作最高将会多消耗约 540MB的内存：
sudo sh -c "echo fs.inotify.max_user_watches=524288 >> /etc/sysctl.conf"
sudo sysctl -p

2.2 如何配置外部 terminal?

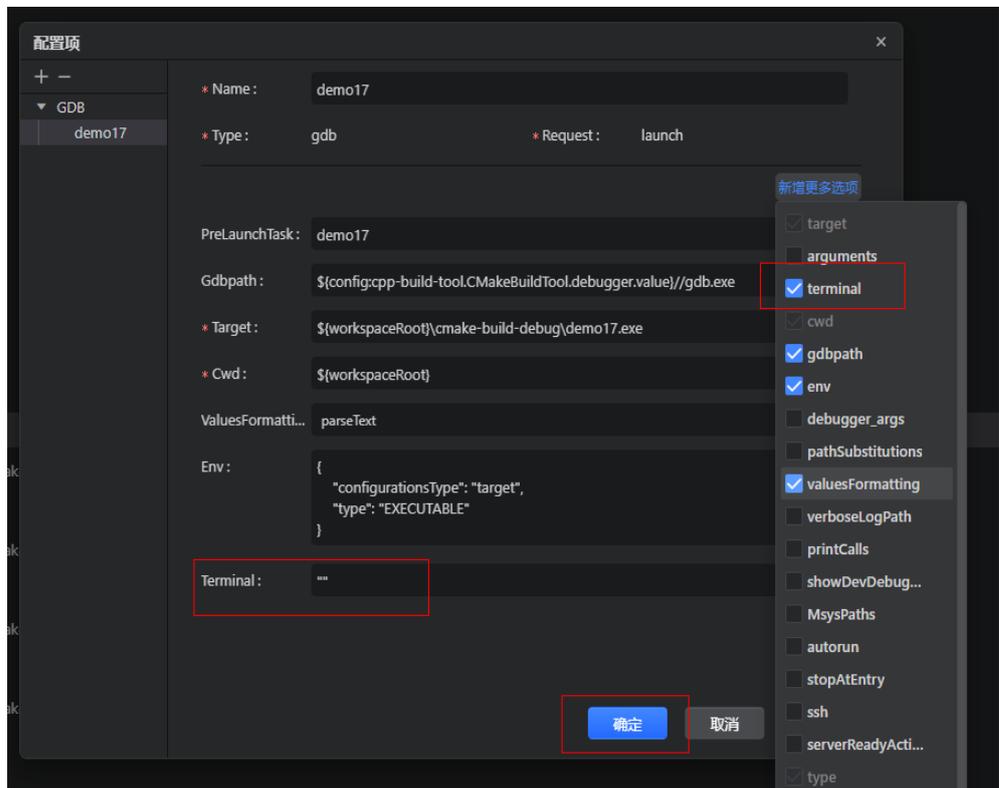
1. 编辑调试配置项，单击“编辑配置”。

图 2-4 编辑配置



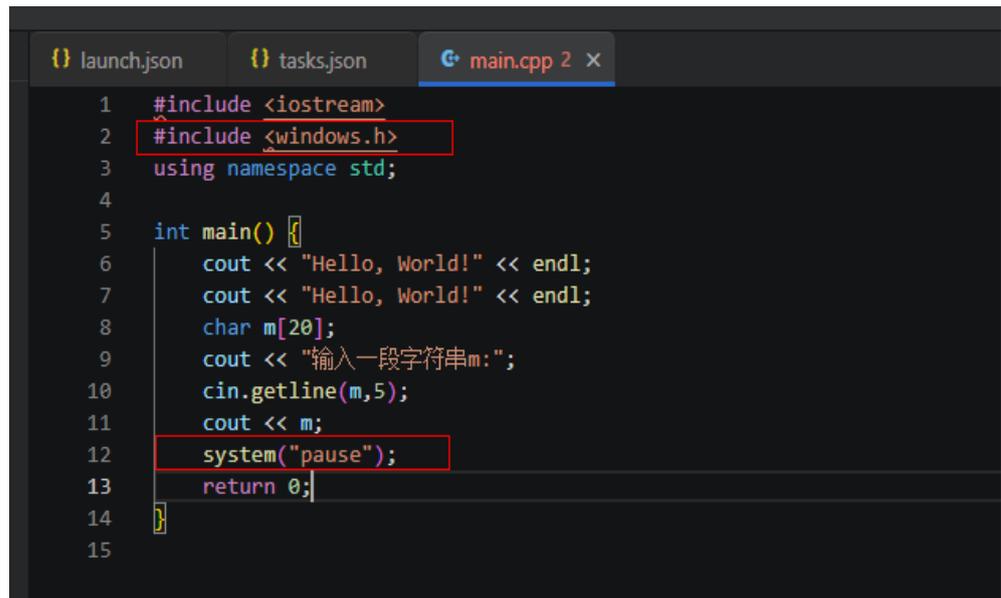
2. 修改terminal的配置项信息为“ ”，单击“确定”。

图 2-5 修改 terminal 的配置项信息



3. 配置完成后，运行时，在外部terminal输入字符串，回车后，terminal自动关闭。如果还需要在外部terminal里查看输出，在代码里面添加以下代码：

图 2-6 编辑代码信息



```
1 #include <iostream>
2 #include <windows.h>
3 using namespace std;
4
5 int main() {
6     cout << "Hello, World!" << endl;
7     cout << "Hello, World!" << endl;
8     char m[20];
9     cout << "输入一段字符串m:";
10    cin.getline(m,5);
11    cout << m;
12    system("pause");
13    return 0;
14 }
15
```

2.3 如何处理 CMake Build Tool 插件运行调试时中文乱码的问题?

问题现象

- 文件运行在内部终端的乱码：main.cpp文件包含中文文字，构建生成可执行文件，右键单击运行。
运行结果出现中文乱码：

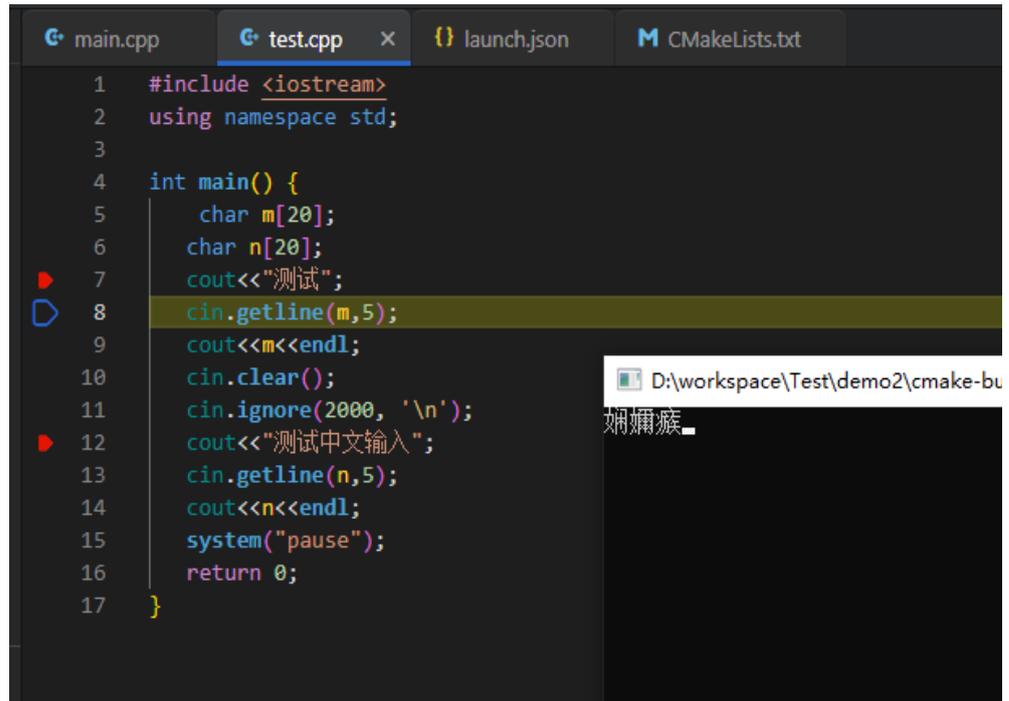
图 2-7 运行结果出现乱码



```
PS D:\workspace\Test\demo17> d:\workspace\Test\demo17\cmake-build-debug\demo17.exe
输入一段字符串m:
乱码
```

- 运行调试使用外部终端出现乱码：
文件里面有输入输出，配置外部terminal，在terminal中打印出来的中文乱码。

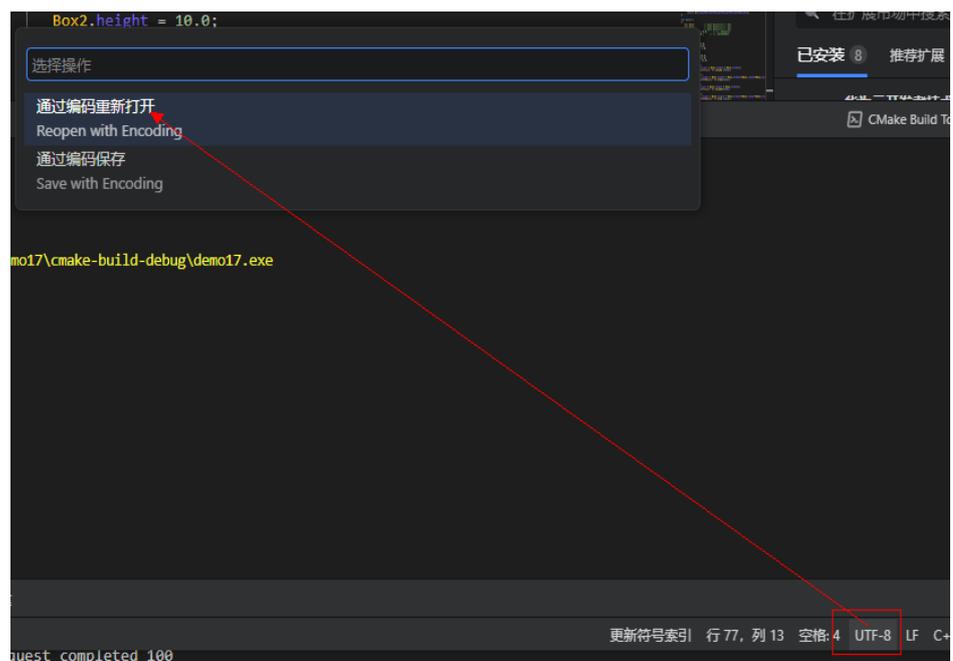
图 2-8 外部终端出现乱码



解决办法

- 方法一
 - a. 单击编码类型，选择Reopen with Encoding，选择GB 2312编码类型（或选择gbk、GB 18030类型编码）。

图 2-9 选择编码文件



- b. 选择之后，文件中出现中文乱码，将乱码文字改为简体中文。

图 2-10 修改中文文字

```
#include <iostream>
using namespace std;

int main() {
    char m[20];
    char n[20];
    cout<<"嫫嫫癡";
    cin.getline(m,5);
    cout<<m<<endl;
    cin.clear();
    cin.ignore(2000, '\n');
    cout<<"嫫嫫癡滑 緜囟緜緯";
    cin.getline(n,5);
    cout<<n<<endl;
    system("pause");
    return 0;
}
```

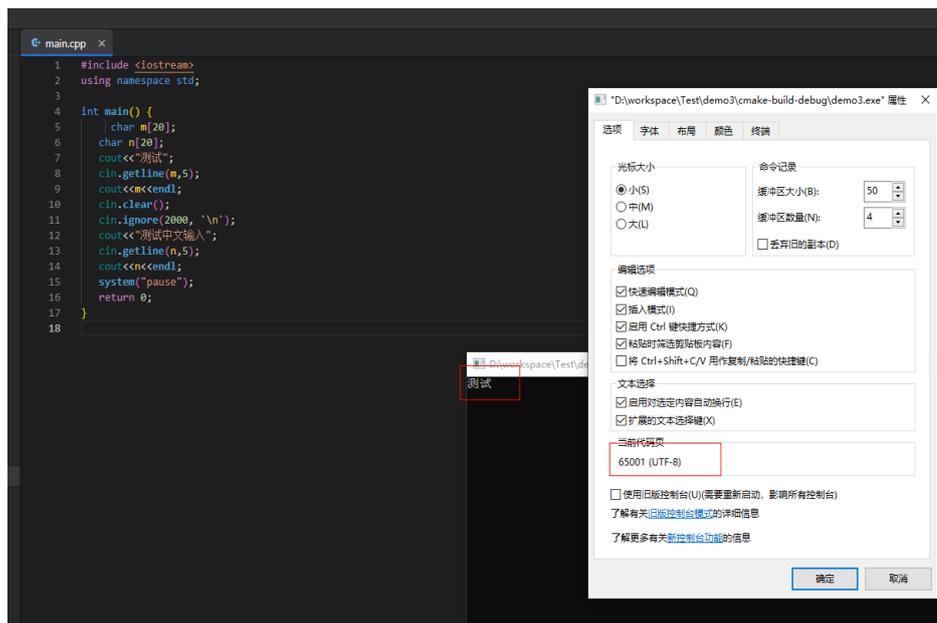
- c. 重新构建，生成可执行文件，运行后显示正常。

图 2-11 重新构建

```
PS D:\workspace\Test\demo17> d:\workspace\Test\demo17\cmake-build-debug\demo17.exe
测试
```

- 方法二
修改计算机系统的默认编码格式，将其改为utf-8，修改方法如下：
 - a. 在开始菜单中搜索“区域与语言设置 > 管理语言设置”
 - b. 在弹出的对话框中的管理选项卡中单击“更改系统区域设置”。
 - c. 如图勾选Unicode UTF_8，并单击“确定”和“应用”。
 - d. 重启系统，并验证，查看运行结果。

图 2-12 查看运行结果



3 CodeArts IDE for Java 常见问题

3.1 如何处理报错“等待语言服务初始化完成”？

问题现象

依赖项视图长时间出现“等待语言服务初始化完成”的提示信息，Maven/Gradle视图也一直显示“等待语言服务初始化完成”。

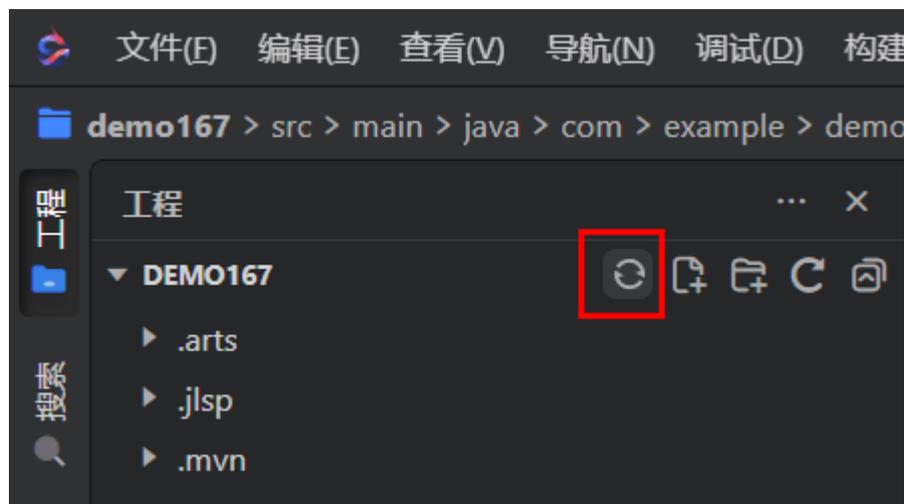
处理方法

请检查当前系统是否运行在虚拟机环境中。此类问题通常由虚拟机系统中缺少必要的证书引起。如确认为虚拟机环境，请按照以下步骤进行修复。

步骤1 单击CodeArts IDE for Java左下角的“管理->设置”菜单，在设置窗口中搜索代理关键字，在搜索出的结果中，找到Http: Proxy Strict SSL设置项，取消勾选该设置项。

步骤2 然后再单击工程视图的重新加载工程按钮，重新加载工程即可。

图 3-1 重新加载工程



----结束

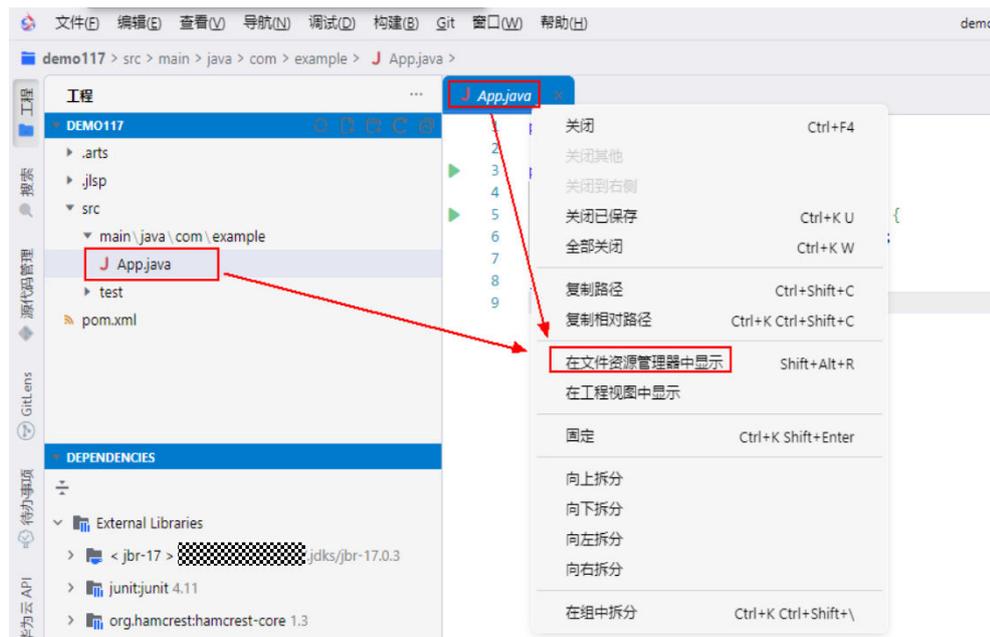
3.2 如何从 IDE 直接打开文件实际存放路径？

步骤1 登录到CodeArts IDE for Java客户端。

步骤2 在当前文件树或者已经打开的文件标题区域单击鼠标右键。

步骤3 选择“在文件资源管理器中显示”即可直接跳转到文件的实际存放路径。

图 3-2 在文件资源管理器中显示



----结束

3.3 如何处理修改文件树后颜色变化和使用中报错不高亮的问题？

原因分析

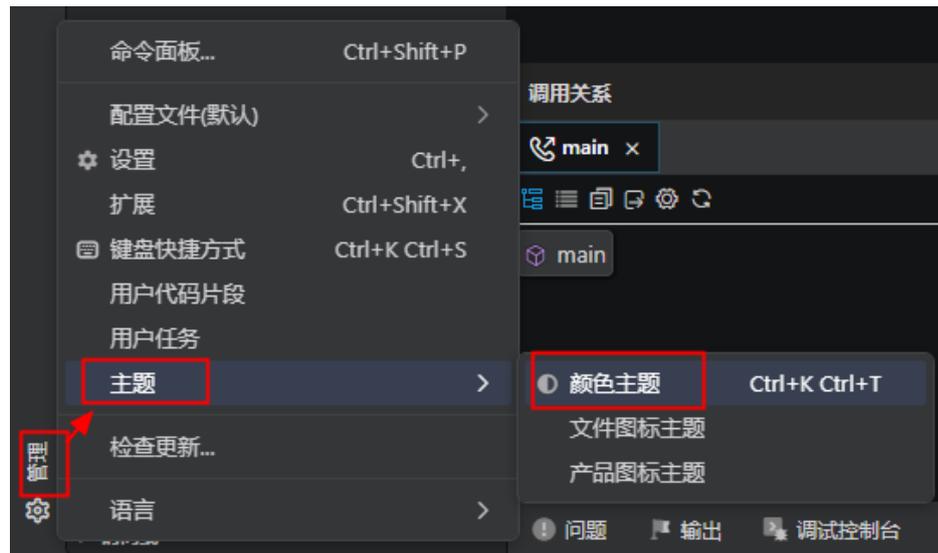
该问题产生的原因是因为选择了不合适的主题色。

处理方法

可以通过切换到其他主题色来处理，具体方法如下：

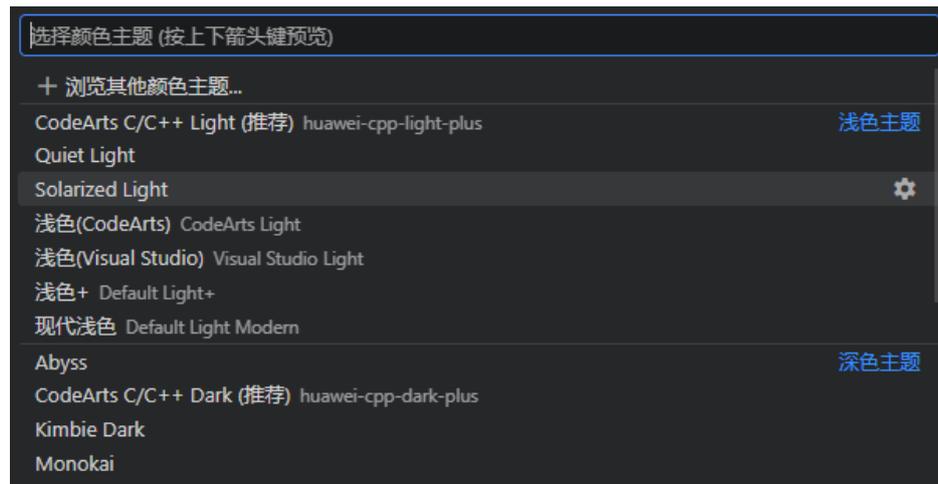
步骤1 单击右下角的“管理 > 主题 > 颜色主题”。

图 3-3 颜色主题



步骤2 切换到不同主题后问题可以解决。

图 3-4 切换主题颜色



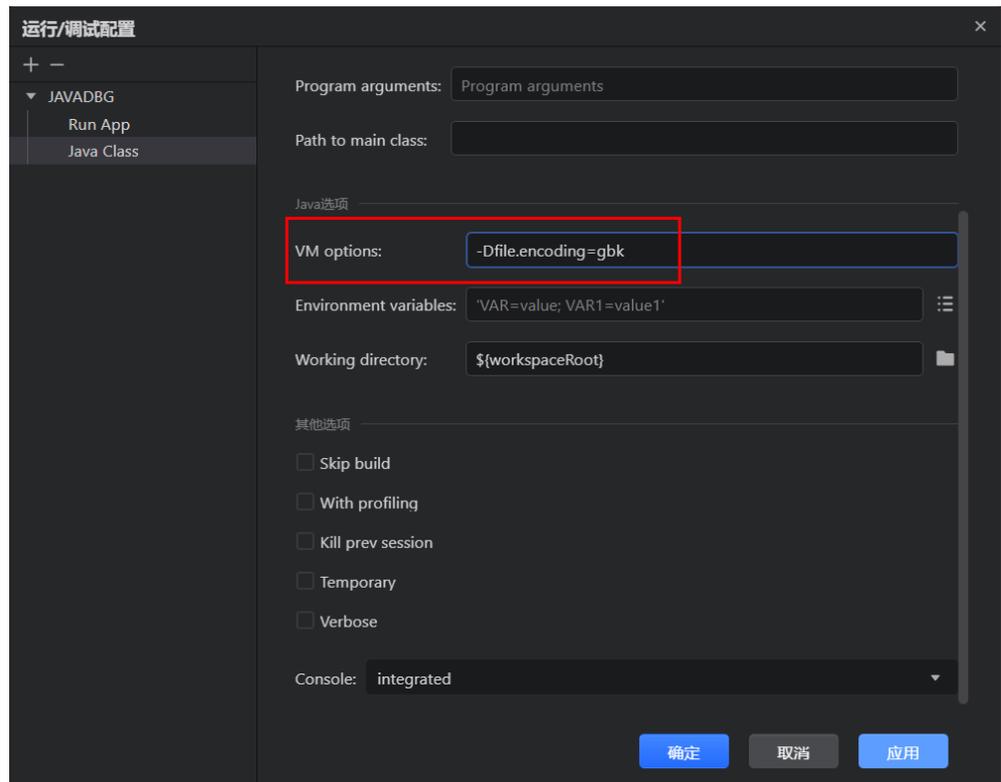
----结束

3.4 如何处理终端输出中文乱码问题?

针对终端输出中文乱码的问题，解决方式有如下两种：

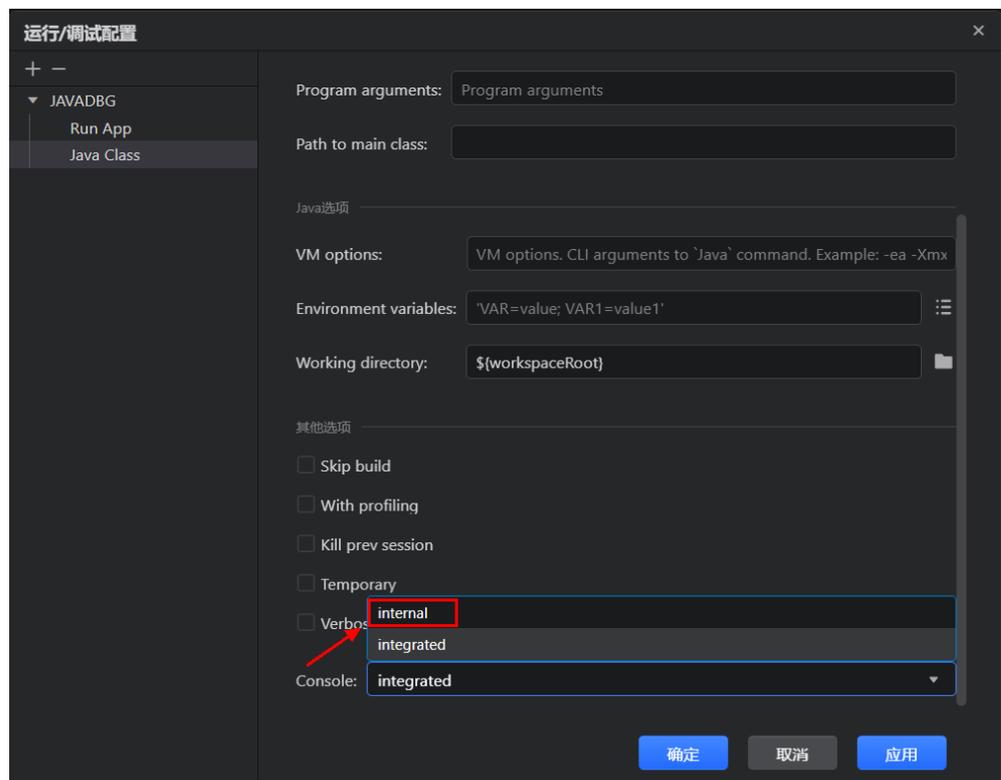
1. 在运行配置启动参数中加入VMOptions参数: `-Dfile.encoding=gbk`;
这种方法可以解决部分的乱码问题，考虑到实际项目中可能用到Scanner的交互式输入场景较少，可以先尝试此方法。如果无法解决，请跳转到2。

图 3-5 添加 VM options 参数



2. 将输出重定向到调试控制台里面：将运行配置中Console设置项中的integrated改成internal。

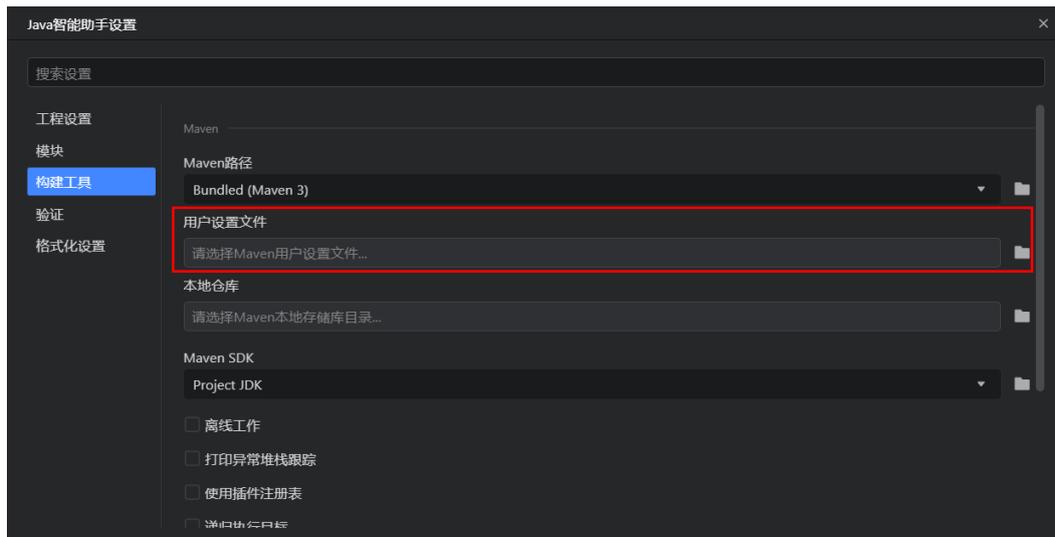
图 3-6 修改 Console 设置项



3.5 如何设置默认的 Maven 配置文件 settings.xml

自定义此settings.xml文件存放路径，然后在工程中指定具体的settings.xml文件的路径：

- 步骤1** 单击左侧任务栏下方“管理 > Java智能助手设置”。
- 步骤2** 在Java智能助手设置中，单击“构建工具”，找到Maven相关设置，替换用户设置文件即可。



----结束

请参考如下settings.xml文件配置，若要使用参考配置，可以在“用户路径/.m2”文件夹下，新建或替换settings.xml文件，Java语言服务会默认读取此路径下的maven配置文件。

settings.xml文件配置参考如下：

```
<?xml version="1.0" encoding="UTF-8"?>
<settings xmlns="http://maven.apache.org/SETTINGS/1.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/SETTINGS/1.0.0 http://maven.apache.org/xsd/
  settings-1.0.0.xsd">

  <!-- 默认的值是${user.home}/.m2/repository -->
  <!--<localRepository></localRepository>-->

  <!-- 如果Maven要试图与用户交互来得到输入就设置为true，否则就设置为false，默认为true。 -->
  <!--
  <interactiveMode>true</interactiveMode>
  -->

  <!-- 如果Maven使用${user.home}/.m2/plugin-registry.xml来管理plugin的版本，就设置为true，默认为false。
  -->
  <!--
  <usePluginRegistry>false</usePluginRegistry>
  -->

  <!-- 如果构建系统要在离线模式下工作，设置为true，默认为false。如果构建服务器因为网络故障或者安全问题
  不能与远程仓库相连，那么这个设置是非常有用的。 -->
  <!--
  <offline>false</offline>
  -->
```

```
<servers>
<!-- server
| Specifies the authentication information to use when connecting to a particular server, identified by
| a unique name within the system (referred to by the 'id' attribute below).
|
| NOTE: You should either specify username/password OR privateKey/passphrase, since these pairings are
| used together.
|-->
<!-- server标签的作用,如下 -->
<!-- 使用mvn install时, 会把项目打的包安装到本地maven仓库 -->
<!-- 使用mvn deploy时, 会把项目打的包部署到远程maven仓库, 这样有权限访问远程仓库的人都可以访问你的jar包 -->
<!-- 通过在pom.xml中使用distributionManagement标签, 来告知maven部署的远程仓库地址, -->

</servers>

<mirrors>
<mirror>
<id>huaweimaven</id>
<name>huaweiyun-maven</name>
<mirrorOf>central</mirrorOf>
<url>https://mirrors.huaweicloud.com/repository/maven/</url>
</mirror>

</mirrors>

<!-- settings.xml中的profile是pom.xml中的profile的简洁形式。它包含了激活(activation), 仓库
(repositories), 插件仓库(pluginRepositories)和属性(properties)元素。
profile元素仅包含这四个元素是因为他们涉及到整个的构建系统, 而不是个别的POM配置。如果settings中的
profile被激活, 那么它的值将重载POM或者profiles.xml中的任何相等ID的profiles。 -->
<!-- 如果setting中配置了repository, 则等于项目的pom中配置了 -->
<profiles>
<profile>
<!-- 指定该 profile的id -->
<id>dev</id>
<!-- 远程仓库-->
<repositories>
<repository>
<id>huaweicloud</id>
<name>huaweicloud maven Repository</name>
<url>https://mirrors.huaweicloud.com/repository/maven/</url>
<releases>
<enabled>true</enabled>
</releases>
<snapshots>
<enabled>>false</enabled>
</snapshots>
</repository>
<repository>
<id>spring-milestone</id>
<name>Spring Milestone Repository</name>
<url>https://repo.spring.io/milestone</url>
<releases>
<enabled>true</enabled>
</releases>
<snapshots>
<enabled>>false</enabled>
</snapshots>
<layout>default</layout>
</repository>
<repository>
<id>spring-snapshot</id>
<name>Spring Snapshot Repository</name>
<url>https://repo.spring.io/snapshot</url>
<releases>
<enabled>>false</enabled>
</releases>
</profiles>
```

```
<snapshots>
<enabled>true</enabled>
</snapshots>
<layout>default</layout>
</repository>
</repositories>
<pluginRepositories>
<!-- 插件仓库。插件从这些仓库下载 -->
<pluginRepository>
<id>huaweicloud</id>
<url>https://mirrors.huaweicloud.com/repository/maven/</url>
<releases>
<enabled>true</enabled>
</releases>
<snapshots>
<enabled>false</enabled>
</snapshots>
</pluginRepository>
</pluginRepositories>
</profile>
</profiles>
<!-- activations是profile的关键，就像POM中的profiles，profile的能力在于它在特定情况下可以修改一些值。
而这些情况是通过activation来指定的。 -->
<!-- <activeProfiles/> -->
<activeProfiles>
<activeProfile>dev</activeProfile>
</activeProfiles>
</settings>
```

3.6 如何设置自定义代码片段和代码模板？

设置自定义代码片段

CodeArts IDE支持自定义代码片段的能力，可以通过代码补全的方式插入自定义的代码片段。

效果如下：

图 3-7 代码补全前

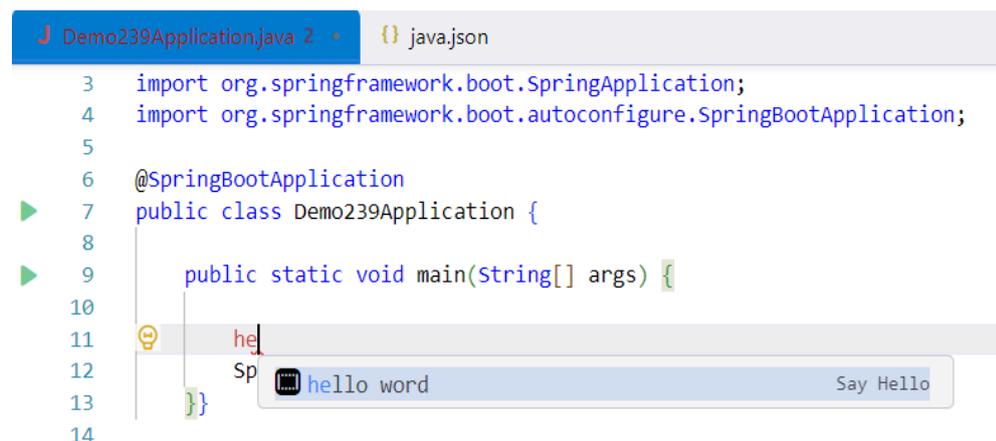
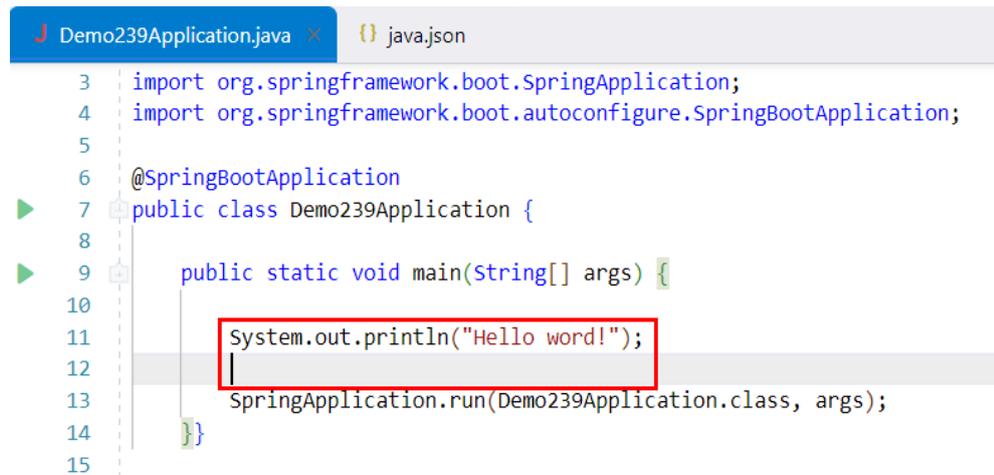


图 3-8 代码补全后

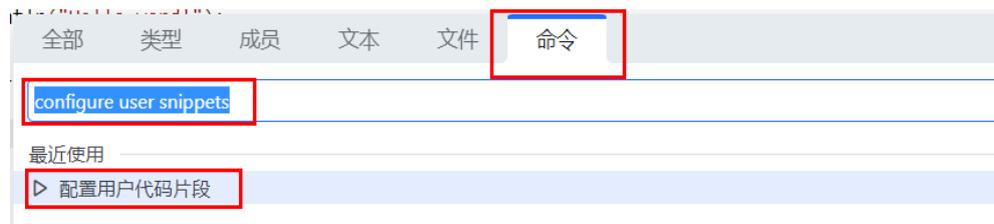


```
3 import org.springframework.boot.SpringApplication;
4 import org.springframework.boot.autoconfigure.SpringBootApplication;
5
6 @SpringBootApplication
7 public class Demo239Application {
8
9     public static void main(String[] args) {
10
11         System.out.println("Hello word!");
12
13         SpringApplication.run(Demo239Application.class, args);
14     }
15 }
```

具体相关设置过程如下：

步骤1 使用ctrl+shift+p打开命令面板，输入configure user snippets，单击“配置用户代码片段”。

图 3-9 配置用户代码片段



步骤2 选择新建全局代码片段文件或新建特定语言文件。如下图所示：

图 3-10 新建代码片段



步骤3 以Java配置文件为例，在文件中自定义代码片段。

图 3-11 自定义代码片段



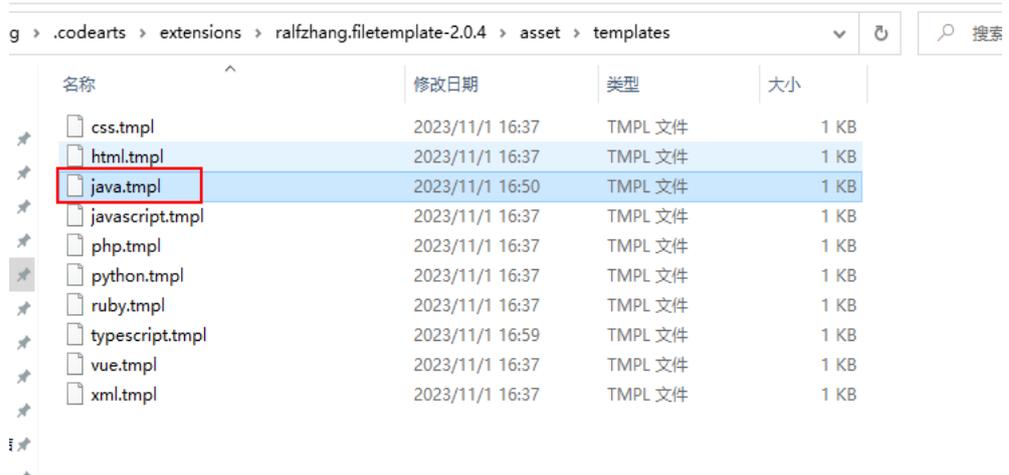
----结束

设置代码模板，支持动态设置日期

1. 从[插件市场](#)下载最新的File Template插件，然后安装到CodeArts IDE中，以 RalfZhang.filetemplate-2.0.4.rar为例。

- 在插件安装路径：“用户路径\codearts\extensions\ralfzhang.filetemplate-2.0.4\asset\templates” 新增一个java.tpl的文件（默认没有此文件）。

图 3-12 java.tpl 文件

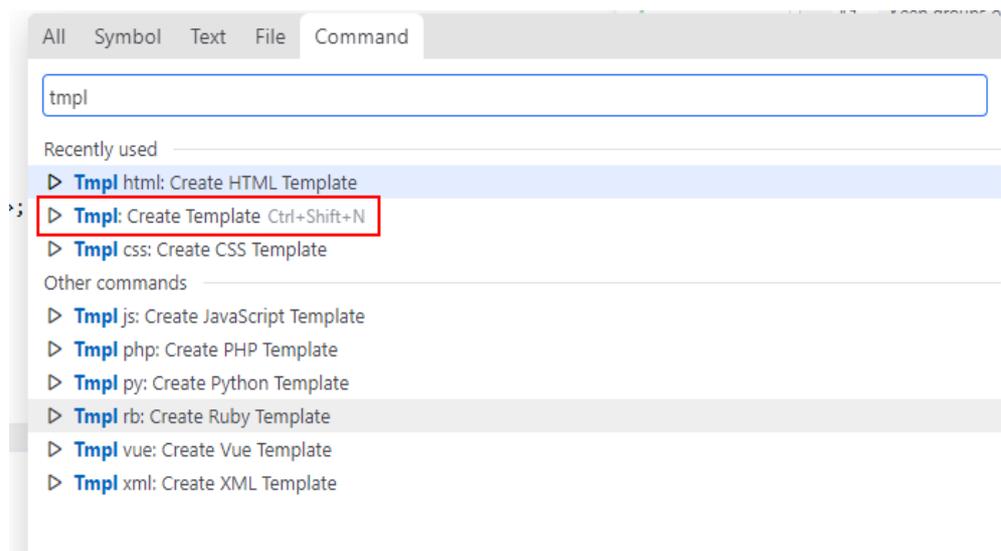


- 在文件中自定义Java语言相关的代码片段，如下：

```
/**
 * ${1:Description}
 * @authors ${2:Your Name} (${3:you@example.org})
 * @date ${date}
 * @version ${4:1.0.0}
 */
$0
```

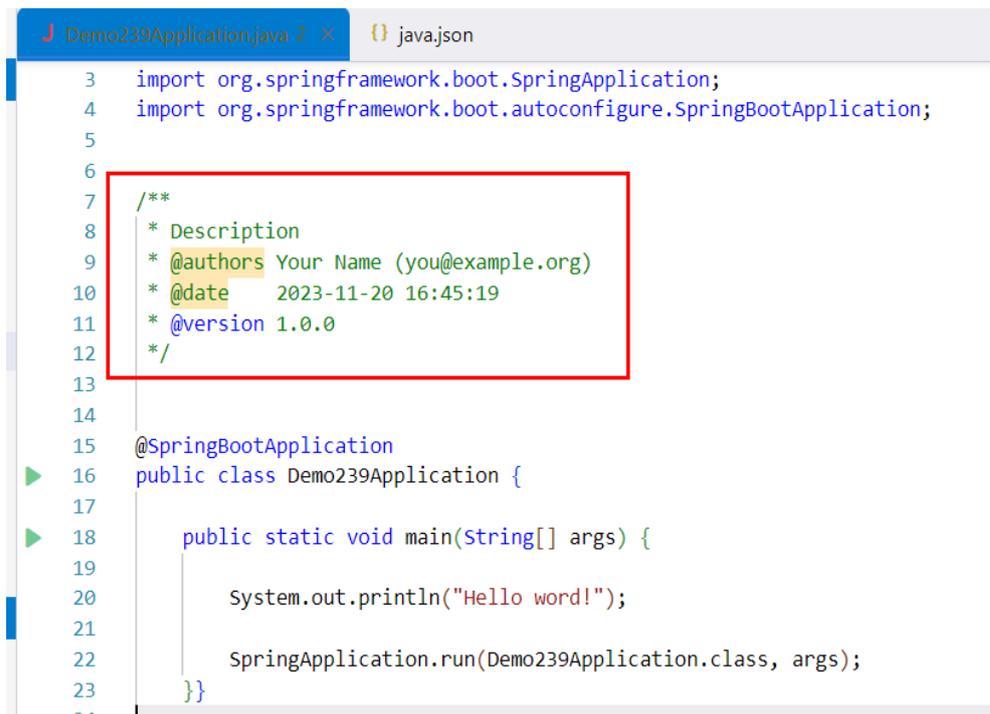
- 使用关键字“tpl”搜索相关模板命令，使用“Tmpl: Create Template”命令就能根据该文件所对应的语言生成对应的代码片段（注意：没有Java相关的tpl命令，只能在打开的Java文件中使用“Tmpl: Create Template”命令）：

图 3-13 生成代码片段



效果如下：

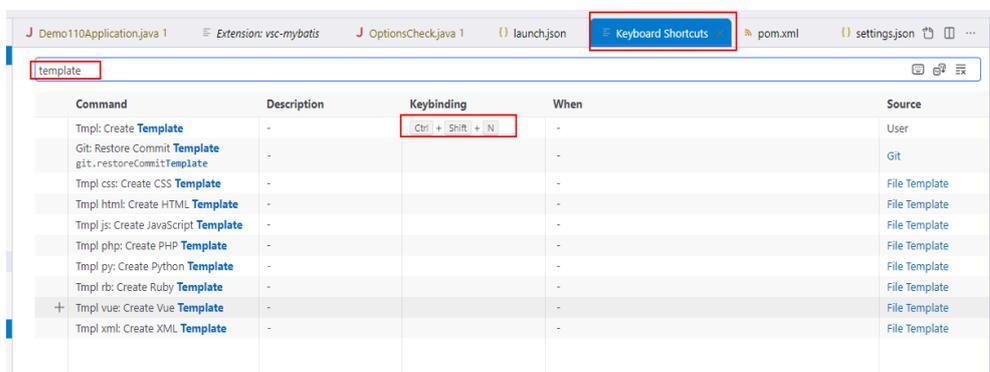
图 3-14 生成结果



```
3 import org.springframework.boot.SpringApplication;
4 import org.springframework.boot.autoconfigure.SpringBootApplication;
5
6
7 /**
8  * Description
9  * @authors Your Name (you@example.org)
10 * @date 2023-11-20 16:45:19
11 * @version 1.0.0
12 */
13
14
15 @SpringBootApplication
16 public class Demo239Application {
17
18     public static void main(String[] args) {
19
20         System.out.println("Hello word!");
21
22         SpringApplication.run(Demo239Application.class, args);
23     }}
24
```

5. 默认的Create Template命令是能够根据当前打开的文件所对应的语言（如Java），来插入对应语言的模板文件中定义的代码片段。
6. 同时也可以给如下命令添加快捷键。

图 3-15 添加快捷键



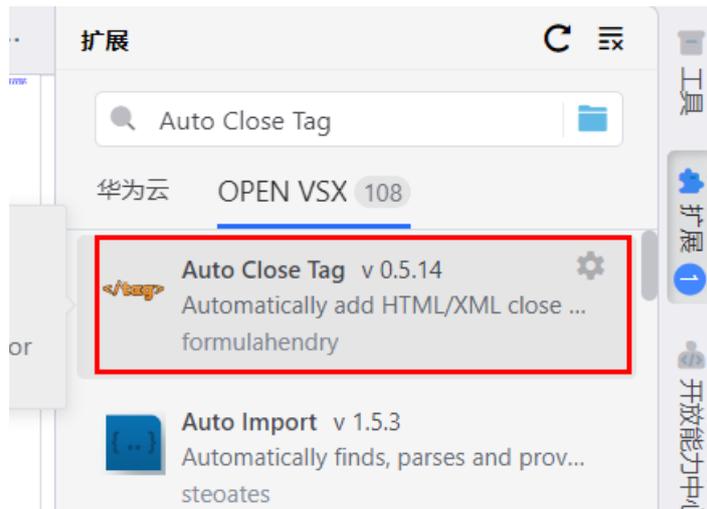
Command	Description	Keybinding	When	Source
Tmpl: Create Template	-	Ctrl + Shift + N	-	User
Git: Restore Commit Template	-		-	Git
git.restoreCommit Template	-		-	Git
Tmpl css: Create CSS Template	-		-	File Template
Tmpl html: Create HTML Template	-		-	File Template
Tmpl js: Create JavaScript Template	-		-	File Template
Tmpl php: Create PHP Template	-		-	File Template
Tmpl py: Create Python Template	-		-	File Template
Tmpl rb: Create Ruby Template	-		-	File Template
Tmpl vue: Create Vue Template	-		-	File Template
Tmpl xml: Create XML Template	-		-	File Template

3.7 如何设置 xml 文件中自动补全结束标签对?

针对该问题可以通过如下方法解决:

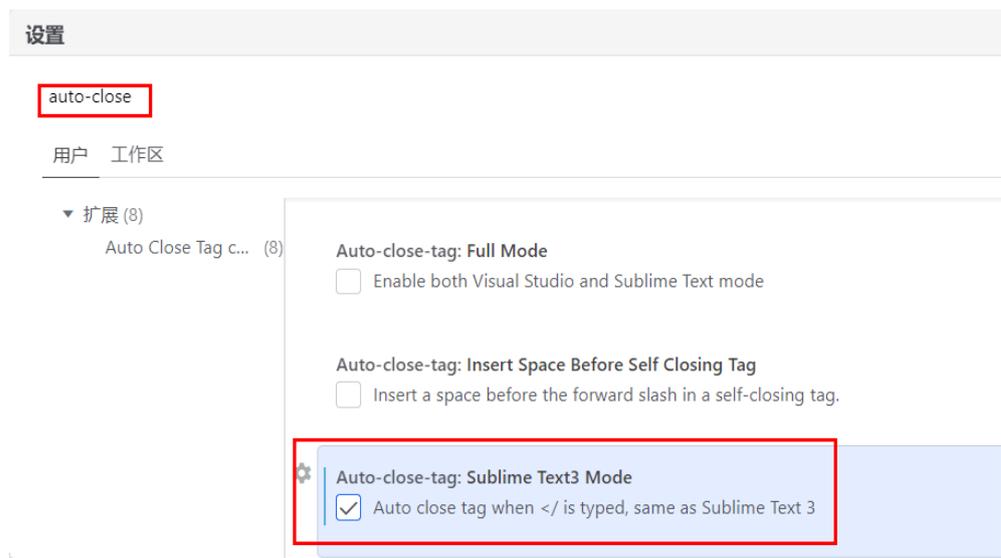
1. 在华为云扩展市场安装“Auto Close Tag”插件。

图 3-16 安装插件



2. 启用如下配置项，就可以在输入</ 的时候自动补全标签对。

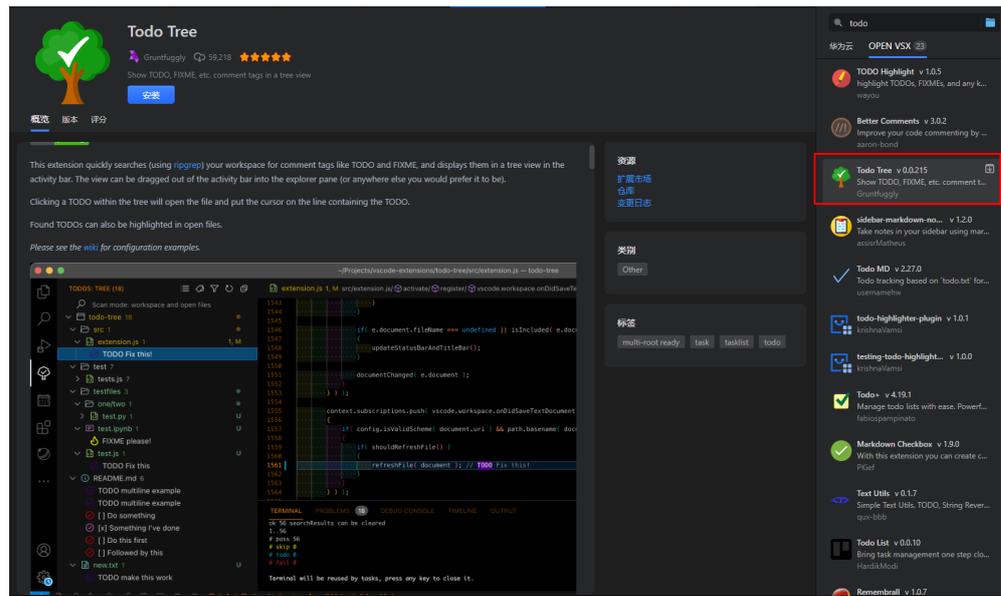
图 3-17 启用配置项



3.8 如何设置自动展示工程中的 TODO 列表?

1. 在华为云扩展市场下载并安装“Todo Tree”插件。
2. 插件安装后CodeArts IDE编辑器左侧会生成“待办事项”视图。
3. 如图所示，会自动识别工程中的TODO标签。

图 3-18 自动识别 TODO 标签



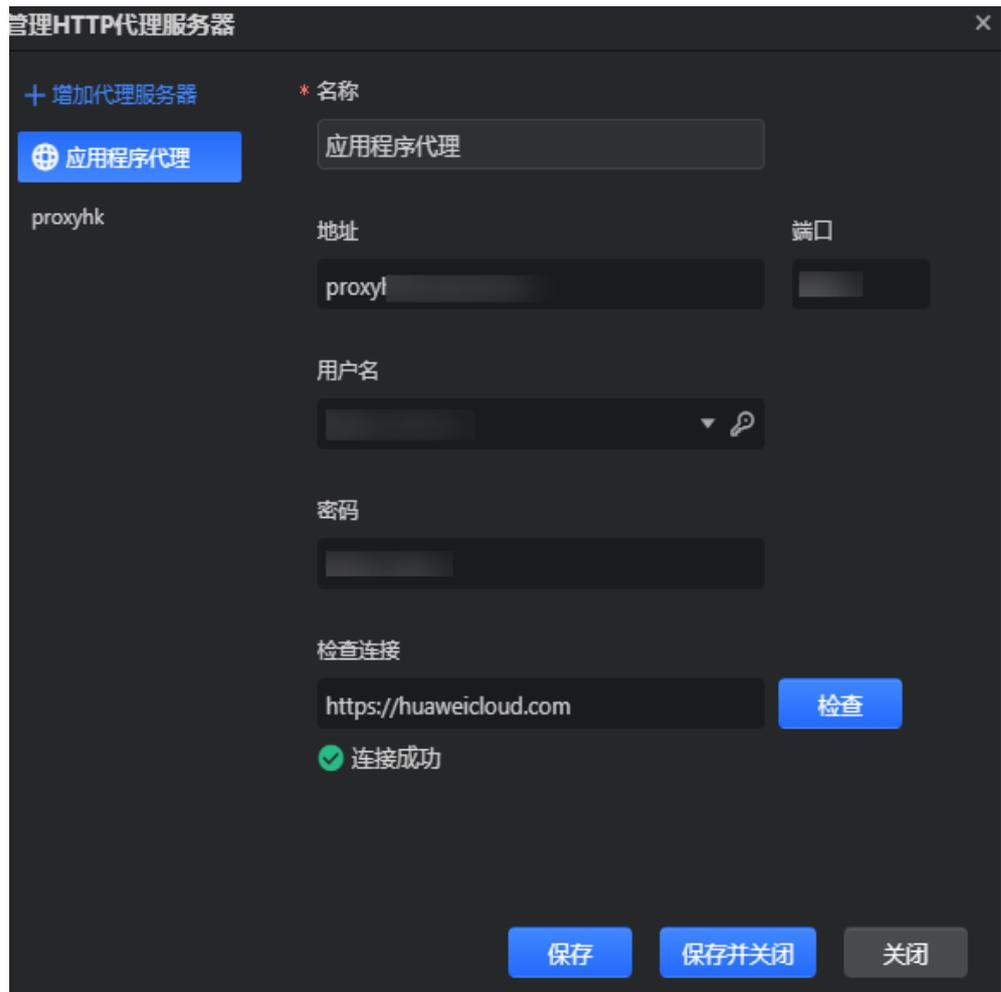
4 CodeArts IDE for RemoteShell 常见问题

4.1 如何处理登录 RemoteShell 客户端失败的问题？

解决办法：

1. 检查本机是否需要配置proxy登录，如已配置请检查proxy是否有效。
2. 单击“管理代理服务器”，配置全局代理。
3. 填写代理相关信息，单击“保存”。

图 4-1 配置全局代理



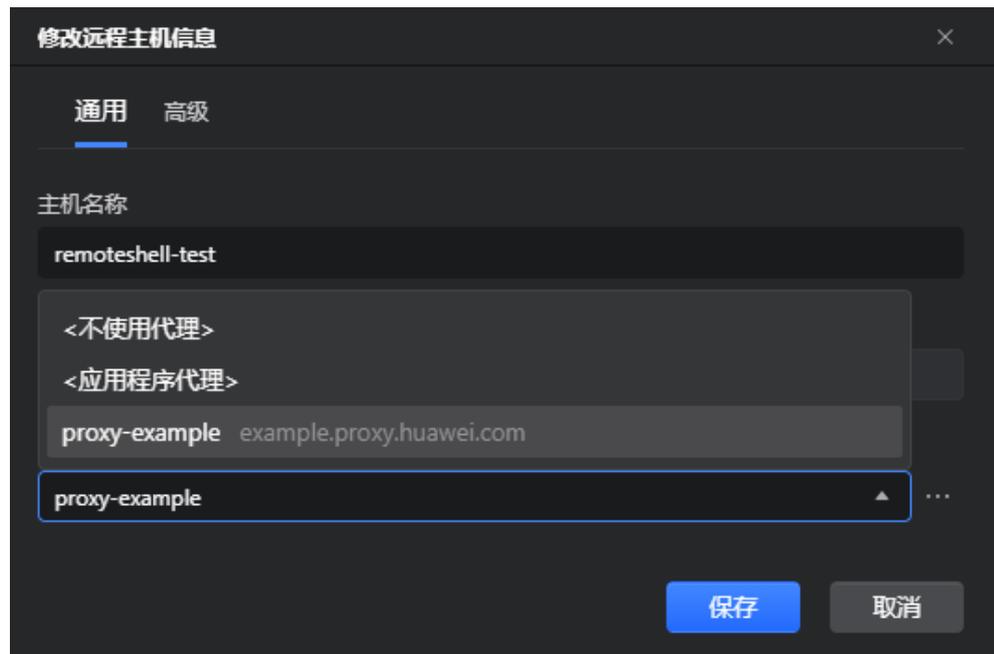
4.2 如何处理 proxy 配置常见问题

登录远程主机，proxy 该如何配置？

解决办法：

1. 在“管理代理”中新增、删除或修改代理配置。
2. 远程主机如需配置proxy或者修改当前使用的proxy，单击“修改远程主机信息”进行修改。

图 4-2 修改远程主机信息



proxy 配置修改后不生效，如何解决？

解决办法：可以尝试单击“关闭该主机的所有连接”，再重新操作连接。

proxy 配置检查连接失败，如何解决？

解决办法：检查proxy的地址和端口是否有效，用户名或密码是否输入错误，修改后重新尝试。

4.3 如何处理主机连接失败的问题

4.3.1 如何处理新建主机连接失败的问题？

可以尝试通过如下方法解决：

- 检查用户名、密码是否输入正确。
- 确保SSH服务已开启（默认端口为22），如果修改SSH默认的22端口，请确认要连接的远程端口已配置且生效，并添加对应的防火墙规则。
- 确认当前使用proxy代理配置出口IP的全部地址的远程端口（SSH默认端口为22）已经允许。
- 检查弹性云服务器的防火墙是否允许当前该主机使用的proxy代理配置的出口IP作为入口IP。
- 请确认网络ACL没有禁用当前该主机使用的proxy代理配置的出口IP。
- 请检查与远程主机之间的网络连通，如需要可绑定弹性公网IP（EIP）。

4.3.2 如何处理连接远程主机失败，出现报错的问题？

- 在连接远程主机时，遇到"407 Proxy Authentication Required"报错时，即为代理认证信息错误，该代理需要用户认证。
解决办法：需在“**管理代理**”弹窗中，找到所使用的代理，并为它输入用户名和密码。
- 在连接远程主机时，遇到"connect ETIMEDOUT"报错时，即为网络超时引起。
解决办法：确认proxy是否有效。
- 在连接远程主机时，遇到"All configured authentication methods failed"报错时，即为错误的密码或主机IP引起。
解决办法：修改为正确的密码后，尝试重新连接。或尝试移除该远程主机后，使用正确的主机地址进行新建远程连接。

5 CodeArts IDE 鸿蒙版本常见问题

配套版本

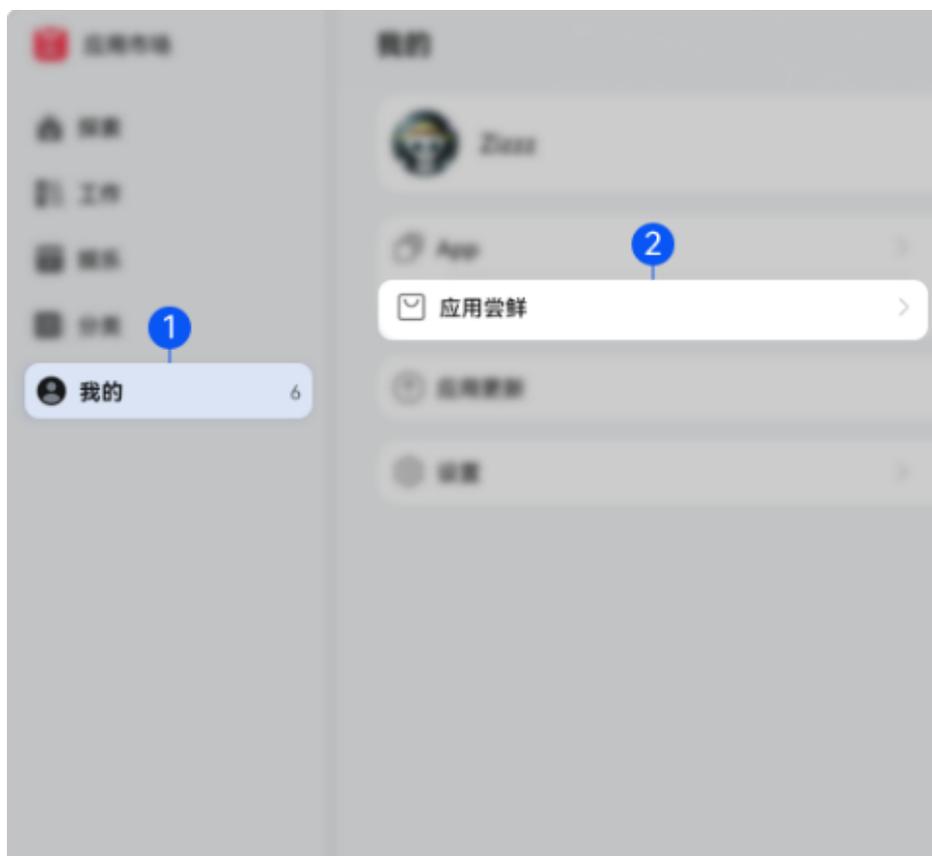
CodeArts IDE客户端配套的鸿蒙操作系统为：Harmony OS 5.0、Harmony OS 6.0。

支持的语言

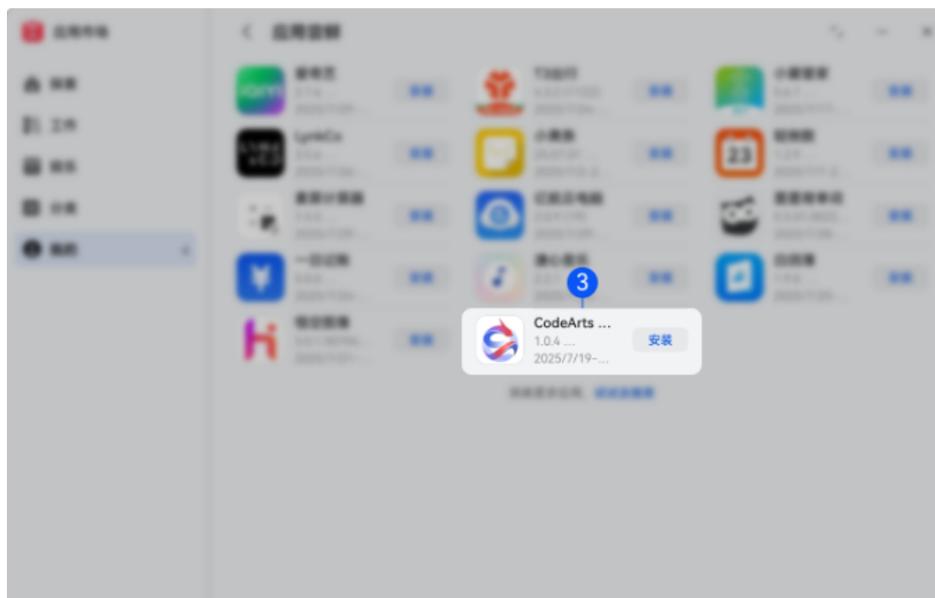
目前支持Java、Python、JS&TS语言。

下载客户端

步骤1 打开鸿蒙应用市场，单击“我的 > 应用尝鲜”。



步骤2 在下载列表中选择“CodeArts IDE”，单击“安装”。



---结束

核心特性

- 快捷的Java/Python工程创建向导：一站式创建工程。
- 内置Java/Python调试器：加快编辑、编译、运行和调试循环的速度。
- 强大的搜索引擎：快速查找和定位代码、文件、符号等。
- 内置Git：无需离开IDE即可进行源代码管理。
- 内置终端命令行：无需离开IDE即可进行命令行操作。
- 强大的代码编辑辅助能力：支持代码生成、重构、导航、高亮和验证等能力。