

CodeArts IDE Onilne

用户指南

文档版本 01
发布日期 2023-08-13



版权所有 © 华为技术有限公司 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。

安全声明

漏洞处理流程

华为公司对产品漏洞管理的规定以“漏洞处理流程”为准，该流程的详细内容请参见如下网址：

<https://www.huawei.com/cn/psirt/vul-response-process>

如企业客户须获取漏洞信息，请参见如下网址：

<https://securitybulletin.huawei.com/enterprise/cn/security-advisory>

目录

1 使用前必读	1
2 管理权限	2
2.1 授权 IAM 用户.....	2
2.2 自定义权限策略.....	3
3 开通服务	6
4 登录服务首页	9
5 新建实例	11
6 新建/导入项目	16
7 基础编辑操作	19
8 运行/调试代码	24
9 提交代码	26
10 通用操作	27
10.1 终端基本操作.....	27
10.2 上传/下载文件.....	29
10.3 自定义命令.....	30
11 Java 技术栈相关操作	31
12 C/C++技术栈相关操作	36
12.1 代码编辑.....	36
12.2 代码调试.....	40
12.3 常见问题.....	43
13 Python 技术栈相关操作	45
14 NodeJs 技术栈相关操作	49
15 Go 技术栈相关操作	51
16 CloudShell 相关操作	53
16.1 约束限制.....	53
16.2 文件树操作.....	53

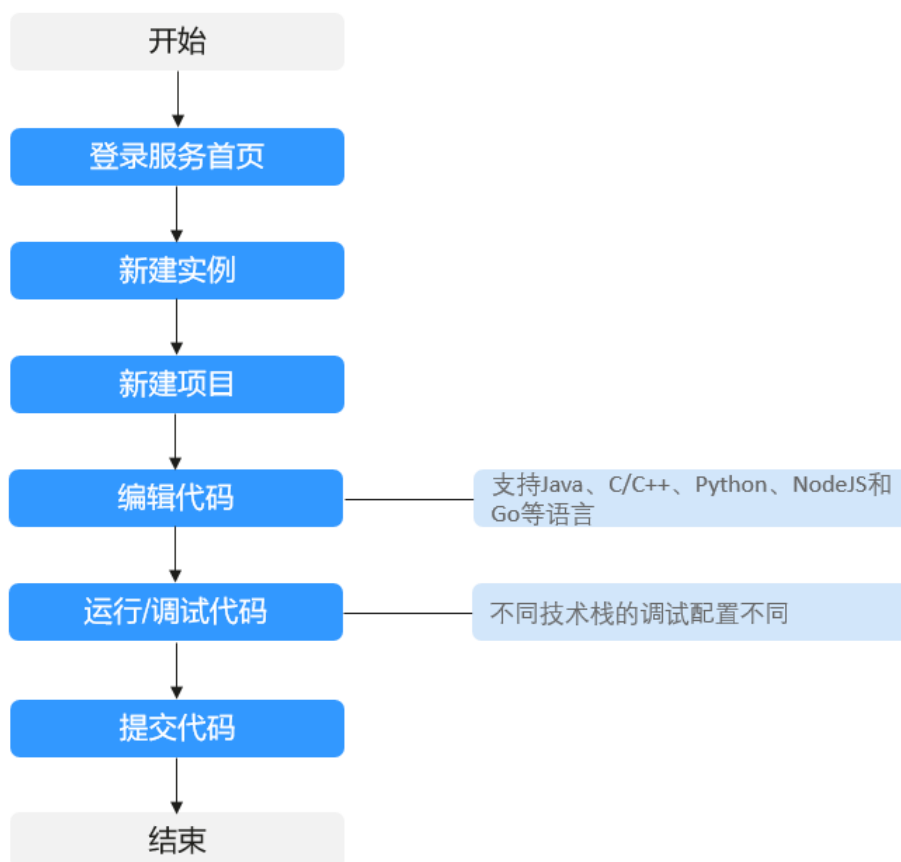
16.3 命令行终端操作..... 55

1 使用前必读

CodeArts IDE Online是云端开发环境服务，向开发者提供按需配置、快速获取的工作空间（包含编辑器和运行环境），支持完成环境配置、代码阅读、编写代码、构建、运行、调试、预览等操作，并支持对接多种代码仓库。

在使用CodeArts IDE Online服务之前，您需要先了解本服务的相关[权限说明](#)以及如何进行[权限管理](#)。

使用流程如下：



2 管理权限

2.1 授权 IAM 用户

概述

如果用户需要对所拥有的CodeArts IDE Online服务进行精细的权限管理，可以使用[统一身份认证服务](#)（Identity and Access Management，简称IAM），通过IAM，可以：

- 根据企业的业务组织，在华为云账号中，给不同职能部门的员工创建IAM用户，让员工拥有唯一安全凭证使用CodeArts IDE Online资源。
- 根据企业用户的职能不同，设置不同的访问权限，实现用户之间的权限隔离。
- 将CodeArts IDE Online资源委托给更专业、高效的其他华为云账号或者云服务，这些账号或者云服务可以根据权限进行代运维。

说明

如果华为云账号可以满足用户对权限管理的需求，则不需要创建独立的IAM用户，可以跳过本章节，不影响使用CodeArts IDE Online服务的其它功能。

本章节介绍对用户授权的方法，操作流程如[示例流程](#)所示。

前提条件

已了解用户组可以添加的CodeArts IDE Online系统策略，并结合实际需求进行选择。

CodeArts IDE Online支持的系统权限，请参见：[权限说明](#)。若您需要对除CodeArts IDE Online之外的其它服务授权，IAM支持服务的所有策略请参见[系统权限](#)。

示例流程

图1 为用户授权CodeArts IDE Online权限操作流程



1. 创建用户组并授权

在IAM控制台创建用户组，并授予CodeArts IDE Online只读权限“CloudIDE ReadOnlyAccess”。

2. 创建用户并加入用户组

在IAM控制台创建用户，并将其加入步骤1中创建的用户组。

3. 用户登录并验证权限

新创建的用户登录控制台，切换至授权区域，通过以下两种方式验证权限是否生效：

- 在“服务列表”中选择CodeArts IDE Online，进入CodeArts IDE Online主界面，单击“新建实例”，创建实例，如果无法创建实例（假设当前权限仅包含CloudIDE ReadOnlyAccess），表示“CloudIDE ReadOnlyAccess”已生效。
- 在“服务列表”中选择除CodeArts IDE Online外（假设当前权限仅包含CloudIDE ReadOnlyAccess）的任意服务，若提示权限不足，表示“CloudIDE ReadOnlyAccess”已生效。

2.2 自定义权限策略

如果系统预置的权限策略，不满足用户授权需求，CodeArts IDE Online支持自定义权限策略。

- 自定义权限策略中可以添加的授权项请参考[策略及授权项说明](#)。
- 自定义权限策略具体创建步骤请参见[创建自定义策略](#)。

本章为您介绍CodeArts IDE Online常用的自定义权限策略代码样例。

自定义策略样例

- 授权用户创建、启动、停止、访问和查看权限。

```
{  
  "Version": "1.1",
```



```
"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "cloudIDE:instance:create",
      "cloudIDE:instance:start",
      "cloudIDE:instance:stop",
      "cloudIDE:instance:access",
      "cloudIDE:instance:get"
    ]
  }
]
```

- 授权用户使用CodeArts IDE Online所有权限。

```
{
  "Version": "1.1",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cloudIDE:*:*"
      ]
    }
  ]
}
```

- 禁止用户删除CodeArts IDE Online实例。

用户被授予的策略中，一个授权项的作用如果同时存在Allow和Deny，则遵循Deny优先原则。因此禁止策略需要同时配合其他策略使用，否则没有实际作用。

例如：如果授予用户CloudIDE FullAccess的系统策略，但不希望用户拥有CloudIDE FullAccess中定义的删除CodeArts IDE Online实例权限，可以创建一条禁止删除CodeArts IDE Online实例的自定义策略，同时将CloudIDE FullAccess和禁止策略授予用户，根据Deny优先原则，则用户可以对CodeArts IDE Online执行除了CodeArts IDE Online实例外的所有操作。禁止策略示例如下：

```
{
  "Version": "1.1",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
        "cloudIDE:instance:delete"
      ]
    }
  ]
}
```

- 仅允许用户启动、停止自己拥有的实例或被分发给自己的实例。

可以在自定义策略中为启动和停止实例的授权项添加条件，使该授权项仅对用户自己的实例生效。CodeArts IDE Online通过IAM预置了两组条件键，其与授权项的关联关系详见[CodeArts IDE Online权限表4](#)授权项与对应条件键，您可以使用条件键检查请求者的id，然后再允许执行操作。为了保证数据安全，CodeArts IDE Online只允许实例的访问者访问实例，如果用户不是实例的访问者，即使配置权限也无法访问该实例。条件策略示例如下：

```
{
  "Version": "1.1",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cloudIDE:instance:start",
        "cloudIDE:instance:stop"
      ],
      "Condition": {
```

```
        "StringEquals": {  
            "cloudIDE:instanceUser": ["${g:UserId}"]  
        }  
    }  
}, {  
    "Effect": "Allow",  
    "Action": [  
        "cloudIDE:instance:start",  
        "cloudIDE:instance:stop"  
    ],  
    "Condition": {  
        "StringEquals": {  
            "cloudIDE:instanceVisitor": ["${g:UserId}"]  
        }  
    }  
}  
]  
}
```

3 开通服务

开通场景

在[CodeArts支持的区域](#)内，各区域独立开通购买、独立计费。

2020年10月16日前，未在CodeArts某个区域下产生过费用的用户，按照新版计费规则，需在该区域[开通/购买CodeArts](#)后使用CodeArts IDEOnline服务。

在该区域内产生过费用的用户，延续旧版计费规则，可参考[开通服务](#)在该区域开通CodeArts IDEOnline服务按需计费。

例如：

用户于2020年3月在“华北-北京四”购买了半年期的包月套餐。由于疫情影响业务，套餐到期后关闭了CodeArts服务；2020年10月20日将重新启用CodeArts。

- 若用户仍使用“华北-北京四”，可以开通/购买旧版套餐，或单独开通CodeArts IDEOnline服务使用。
- 若用户使用其它区域，则需开通/购买CodeArts新版计费。

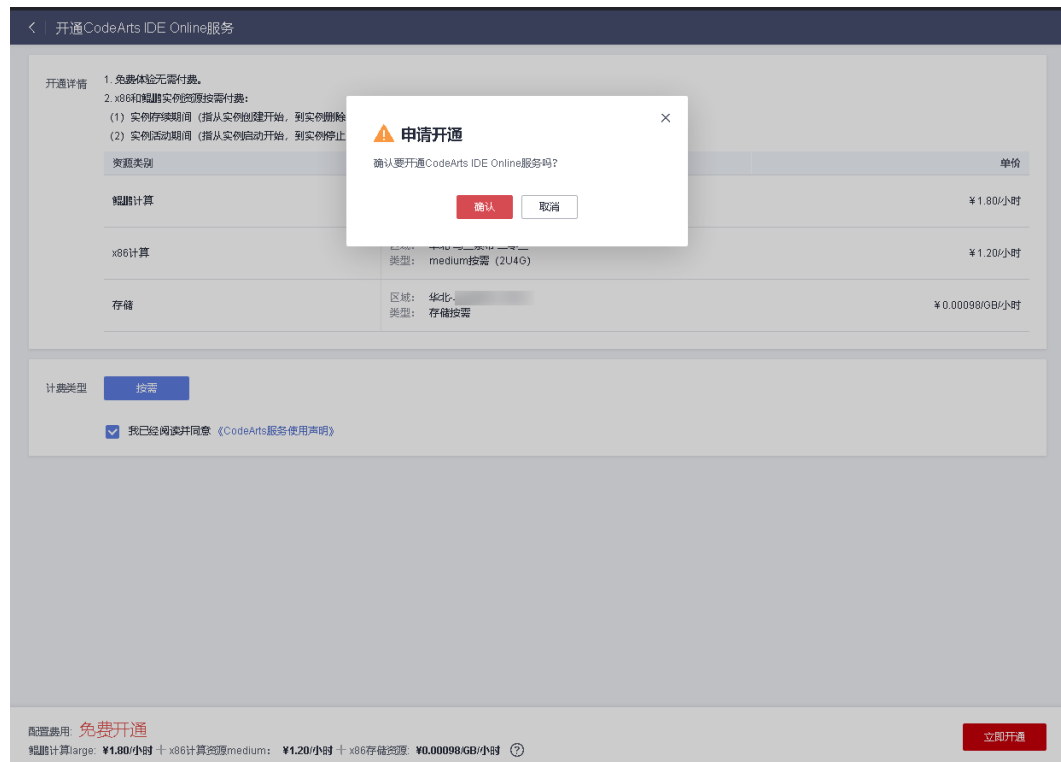
前提条件

已[注册华为云用户](#)（若已注册华为云用户，请忽略）。

开通服务

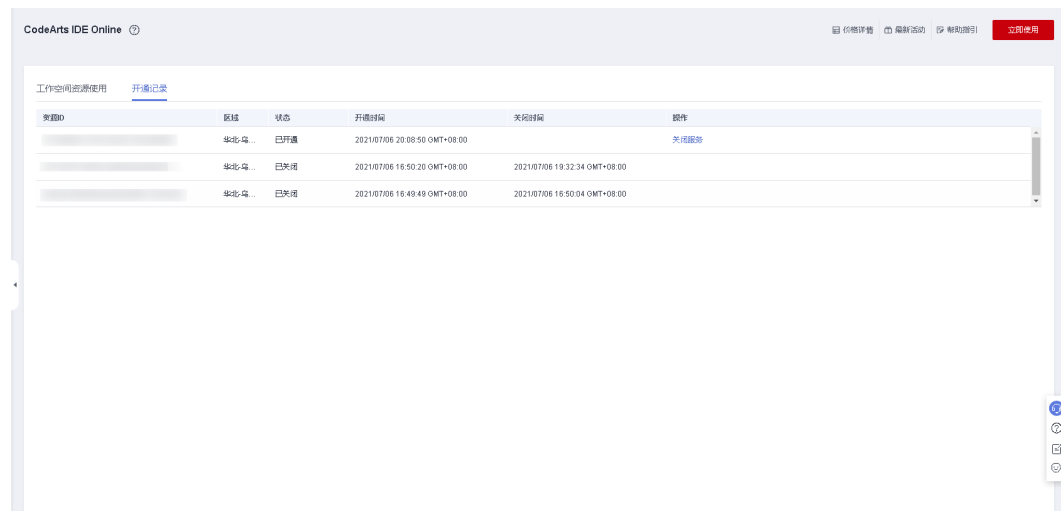
- 步骤1** 使用华为云账户登录[控制台](#)。
- 步骤2** 在左侧导航选择“IDE Online”。
- 步骤3** 单击页面右上角“开通按需”。
- 步骤4** 查看并确认服务开通详情，勾选“我已阅读并同意《CodeArts服务使用声明》”，并单击页面右下角“立即开通”。
- 步骤5** 在弹框中单击“确认”，如[图3-1](#)所示。

图 3-1 确认开通



步骤6 页面右上角提示开通进行中，成功开通后系统自动返回控制台。
在“开通记录”页签可查看服务开通记录，如图3-2所示。

图 3-2 开通记录



开通完成后，单击页面右上角“立即使用”，进入CodeArts IDE Online首页，即可开始使用。

----结束

关闭服务

CodeArts IDE Online服务根据存储资源和计算资源的使用时长计费。若不再使用CodeArts IDE Online服务，为了避免不必要的花费，建议关闭服务。

步骤1 使用华为云账户登录**控制台**，在左侧导航选择“IDE Online”。

步骤2 在“开通记录”页签，单击“关闭服务”，如**图3-3**所示。

图 3-3 关闭服务



资源ID	区域	状态	开通时间	关闭时间	操作
7071	华北-乌...	已开通	2021/07/06 20:08:50 GMT+08:00		关闭服务

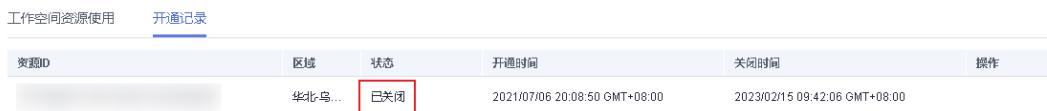
步骤3 根据弹窗提示，输入“释放数据”，单击“确认”，关闭服务，如**图3-4**所示。

图 3-4 确认关闭服务



页面提示“已成功关闭”，开通记录状态显示为“已关闭”，如**图3-5**所示。

图 3-5 服务已关闭



资源ID	区域	状态	开通时间	关闭时间	操作
	华北-乌...	已关闭	2021/07/06 20:08:50 GMT+08:00	2023/02/15 09:42:06 GMT+08:00	

----结束

约束与限制

关闭服务后，系统将停止对服务的使用计费，服务中相关的数据资源将被立即释放，且数据不可恢复，请谨慎操作。

4 登录服务首页

本节介绍登录CodeArts IDE Online首页的操作步骤及首页功能介绍。

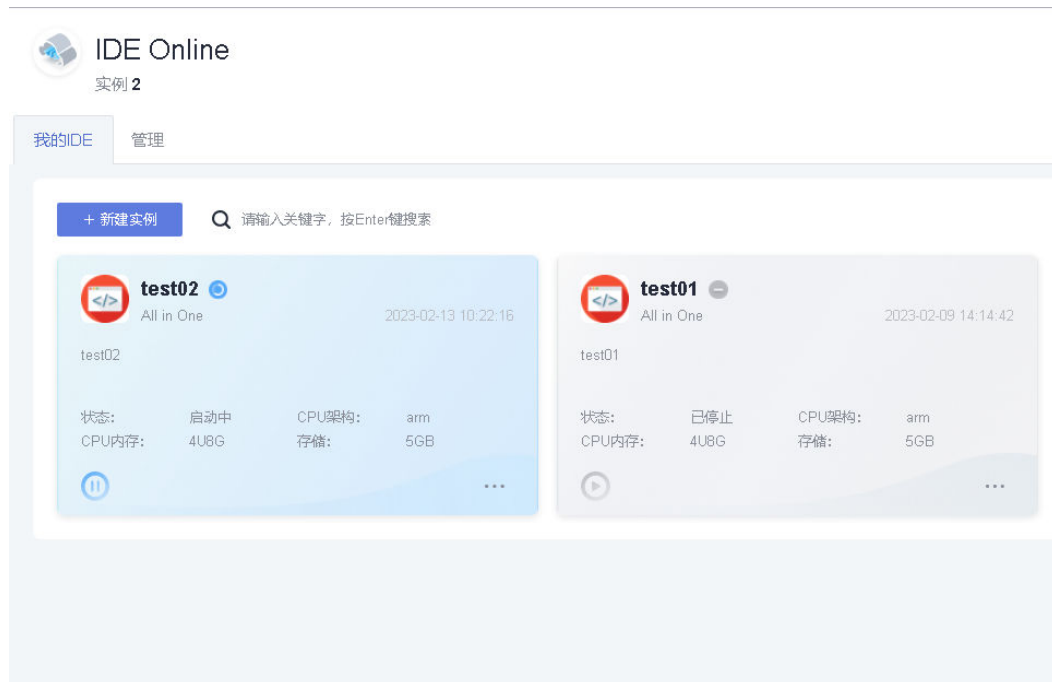
前提条件

已[开通CodeArts IDE Online服务](#)。

操作步骤

- 步骤1** 使用华为云账户访问[华为云首页](#)。
- 步骤2** 在导航栏选择“产品 > 开发与运维 > CodeArts IDE Online”。
- 步骤3** 进入CodeArts IDE Online产品页，单击“立即体验”。
- 步骤4** 进入CodeArts IDE Online服务“我的IDE”页签，显示当前用户的[IDE实例](#)，如[图4-1](#)所示。
初次使用CodeArts IDE Online服务，需[创建IDE实例](#)。

图 4-1 IDE 实例列表



- 页面左侧“免费体验云开发”卡片，可限时免费体验部分IDE实例，但免费体验不能保证最佳使用体验，建议先[开通服务](#)，服务开通后，开始按需计费。
- 页面右侧“插件市场”卡片，可下载开发过程中需要使用的插件。

----结束

5 新建实例

IDE实例：为工程的工作空间，包含计算资源和存储资源，对于线下的开发平台，计算资源属于本地PC，存储资源属于本地硬盘，对于CodeArts IDE Online来说，可以理解为编码、运行、调试的操作都在这个实例里面完成。

本节介绍创建IDE实例（含创建工程）以及修改、删除IDE实例的操作步骤。

前提条件

已开通CodeArts IDE Online服务。

创建 IDE 实例

步骤1 登录服务首页。

步骤2 单击“新建实例”，进入到“基础配置”页面，参考表5-1配置参数。如无法创建实例，需完成开通服务。

表 5-1 基础配置参数说明

参数名	说明
名称	自定义实例名称。 <ul style="list-style-type: none">支持输入数字、字母、“_”、“.”和“-”。以数字、字母开头和结尾。长度为3~100。
描述	自定义对实例的描述。长度不超过100个字符。

参数名	说明
技术栈	请根据实际情况选择技术栈类型。目前支持的类型有： <ul style="list-style-type: none">• All in One（包括以下所有技术栈）• Java（最高支持JDK11.0.17）• C/C++• Python（最高支持版本为3.8）• NodeJS（最高支持版本为16.19.0）• Go（最高支持版本为1.19.4）• Blockchain
CPU架构	根据实际需要选择。 <ul style="list-style-type: none">• 鲲鹏计算：鲲鹏采用精简指令集（RISC）。• X86计算：X86采用复杂指令集（CISC）。
CPU/内存	系统根据所选技术栈默认匹配对应配置。
存储容量	支持三种，请根据实际需要选择。 <ul style="list-style-type: none">• 5GB• 10GB• 20GB
自动休眠	实例无操作超过一段时间后，将会被自动休眠，如需操作实例需重新启动实例。请根据实际需要进行配置。 <ul style="list-style-type: none">• 30分钟• 60分钟• 4小时• 24小时• 永不休眠

步骤3 单击“下一步”，进入“工程配置”页面，参考表5-2配置工程信息。

表 5-2 工程配置参数说明

工程来源	参数名	说明
样例工程	工程名称	自定义工程的名称。 <ul style="list-style-type: none">• 可输入字母、数字、符号“-”和“_”。• 字符长度1~127。
	工程模板	工程文件来源于选择的模板，根据步骤2中选择的“技术栈”类型不同，对应“模板”也不同。请根据需要选择对应的“模板”。
私有仓库	项目名称	工程文件来源于当前用户在需求管理服务中已创建或具有权限的项目。选择工程所属项目的名称即可。

工程来源	参数名	说明
	仓库地址	当前用户所属项目下拥有权限的代码仓库地址，仓库地址获取方式可参考 查看仓库详情 。
	分支	从代码仓库中获取的具有权限的分支。
	名称	工程的名称。默认为代码仓库的名称。
不创建工程		若不创建工程直接进入实例，默认没有相关工程文件在实例内，需要用户在实例内新建或者导入工程。

步骤4 单击“确定”，IDE实例创建完成，系统自动进入IDE工作界面。

----结束

启动 IDE 实例

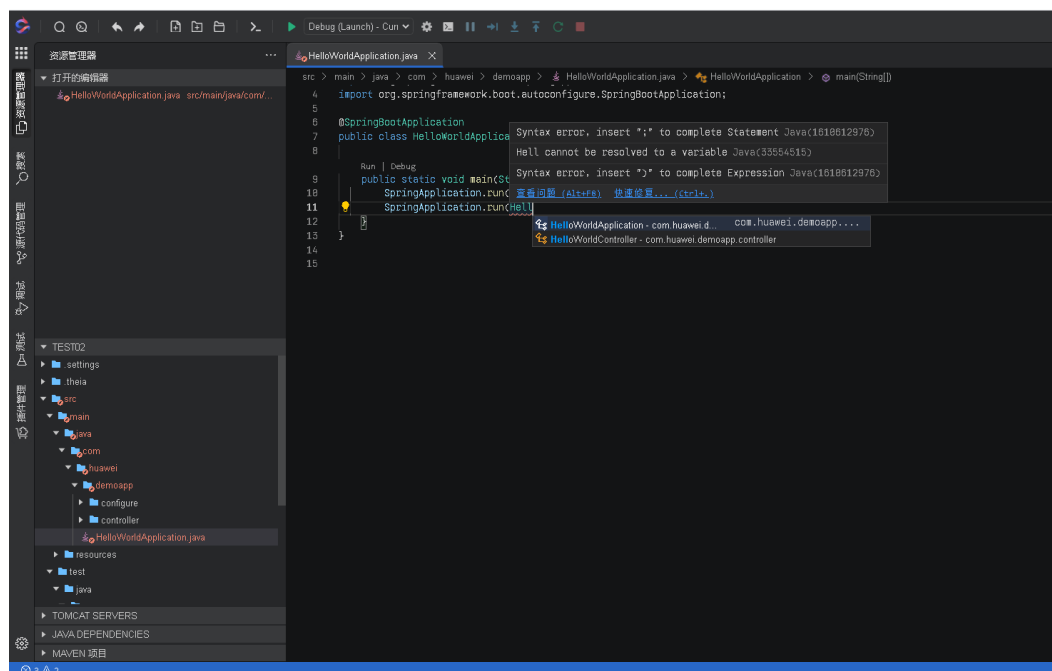
步骤1 单击目标IDE实例所在区域的▶，启动实例，如[图5-1](#)所示。

图 5-1 启动实例



步骤2 启动完成后，进入IDE工作界面，即可在线管理代码，如[图5-2](#)所示。

图 5-2 IDE 工作界面



----结束

修改/删除 IDE 实例

步骤1 单击IDE实例的  按钮，停止实例。


步骤2 单击 ，选择“编辑实例”，可以根据需要修改工作空间名称、描述以及设置自动休眠时间；选择“删除实例”，可删除对应实例，如图5-3所示。

图 5-3 编辑或删除实例



----结束

约束与限制

- IDE实例只有创建者可见。
- IDE实例删除后数据不可恢复。

6 新建/导入项目


本节介绍在IDE实例中新建项目和导入已有项目的操作步骤。

当前支持创建Java项目、NodeJs项目、Python项目、C/C++项目、Go项目和Plugin项目，其中NodeJs项目、Python项目、C/C++项目、Go项目的创建方式，和Java项目一致。本节主要以创建Java项目和Plugin项目为例进行介绍。

- [创建Java项目](#)
- [创建Plugin项目](#)
- [导入项目](#)

创建 Java 项目

步骤1 进入CodeArts IDE Online工作界面。

步骤2 单击左上角，选择“文件 > 新建项目”。

步骤3 在左侧导航栏选择“Java”。

步骤4 根据实际需要选择“类型”。

不同项目参数配置不同，具体根据页面的标签提示填写参数信息，如下所示。

- **创建空白项目**
 - 类型：下拉框选择“Empty Project”类型。
 - 路径：选择或输入将要创建的项目的存放路径。
 - 名称：用户自定义的项目名称。
- **创建Maven项目**
 - 类型：下拉选择“Maven”。
 - 路径：选择或输入将要创建的文件存放路径。
 - ArcheType：下拉框选择“archeType”。支持输入搜索，可以通过关键词搜索并选择目标。
 - ArtifactId：自定义Maven项目的ArtifactId。
 - GroupId：Maven 项目的GroupId。
 - 版本号：Maven 项目的版本号。
 - Package：默认与“GroupId”一致，同时支持自定义。

- **创建Gradle项目**

- 类型：下拉框选择 “Gradle” 。
- 路径：选择或输入将要创建的文件存放路径。若所选路径目录下存在 “pom.xml” 文件且不存在 “build.gradle” 文件，则自动将 “GradleType” 类型选择为 “pom”，并将目标路径拆分填写到对应的输入框中。
- 名称：自定义的项目名称。
- GradleType：支持五种类型：“basic”、“java-application”、“java-library”、“java-gradle-plugin”和“pom”，请根据需要选择对应的类型。

- **创建Template项目**

- 类型：下拉框选择 “Template” 。
- 路径：选择或输入将要创建的文件存放路径。
- 名称：自定义的项目名称。

在下方展示出的卡片中选中想要创建的 “Template” 类型。Template卡片的排列请以页面实际显示为准。


步骤5 单击 “创建”，当输入内容格式全部验证正确后开始创建，此时按钮变为 “正在创建中”。

创建完成后，可选择打开已创建好的项目或者继续停留在当前页面。

----结束

创建 Plugin 项目

步骤1 进入CodeArts IDE Online工作界面。

步骤2 单击左上角，选择 “文件 > 新建项目”。

步骤3 在左侧导航栏选择 “Plugin”，根据以下说明填写参数信息。

- 类型：支持 “Generic”，“Backend” 两种项目类型，请根据实际需要选择目标类型。
- 路径：选择或输入将要创建的文件存放路径。
- 名称：自定义的项目名称。
- 发布者/作者：默认为登录时使用的账号名，支持自定义。
- 许可证：选择license，支持 “MIT”，“BSD” 两种选择，同时支持自定义。
- 版本号：项目的版本号。
- 描述：项目的说明。
- Git初始化：是否需要使用git init命令初始化该插件项目，根据实际需要选择。

步骤4 单击 “创建”，当输入内容格式全部验证正确后开始创建，此时按钮变为 “正在创建中”。


创建完成后，可选择打开已创建好的项目或者继续停留在当前页面。

----结束

导入项目

如在代码仓库中已有项目，可直接导入。

步骤1 进入CodeArts IDE Online工作界面。

步骤2 单击左上角，选择“文件 > 导入项目”。

步骤3 弹出“导入项目”对话框，选择从Git或者归档文件导入项目。

步骤4 参考表6-1填写导入信息。

表 6-1 参数说明

信息项	描述
来源	选择需要导入项目的来源。 <ul style="list-style-type: none">• Git：项目代码来源于代码仓库。• Archive：项目代码来源压缩包。
认证助手	当“来源”是“Git”时需要设置，详情请参考 凭证存储 。根据实际选择“允许”或者“不允许”。
仓库地址	<ul style="list-style-type: none">• 当“来源”是“Git”时：代码仓文件下载地址。地址获取方式参考查看仓库详情。• 当“来源”是“Archive”时：用户归档文件的下载地址。
分支	自定义分支名称。当“来源”是“Git”时需要设置。
路径	根据实际需要选择导入的插件项目需要存放的路径。
用户	默认为登录服务首页时的用户名。
密码	默认为登录服务首页时的密码。

步骤5 单击“导入”，完成项目导入。

----结束

7 基础编辑操作

本节主要介绍CodeArts IDE Online服务工作界面的基本编辑操作，主要包含以下操作：

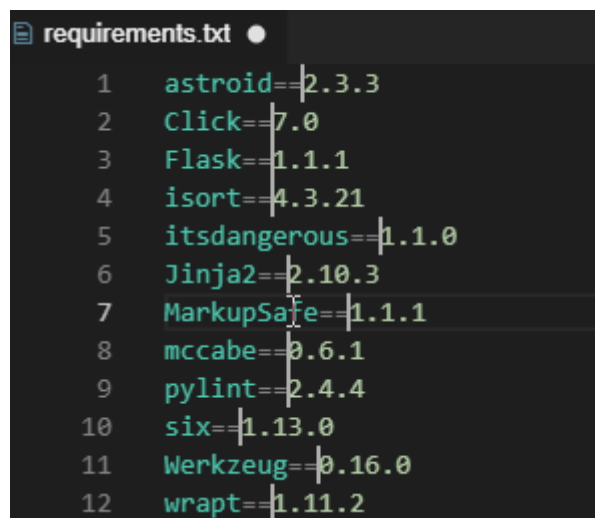
- [多重编辑（多重光标）](#)
- [调整选区](#)
- [选择列](#)
- [保存/自动保存](#)
- [查找与替换](#)
- [全局搜索](#)
- [代码格式化](#)
- [代码折叠](#)

登录工作界面请参考[启动IDE实例](#)。

多重编辑（多重光标）

按住“Alt”键，同时在编辑器中单击需要编辑的位置，添加多个光标进行编辑。也可以用“Ctrl+Alt+↓”或者“Ctrl+Alt+↑”在当前光标的下方和上方添加光标。选中多重光标效果如[图7-1](#)所示。

图 7-1 选中多重光标

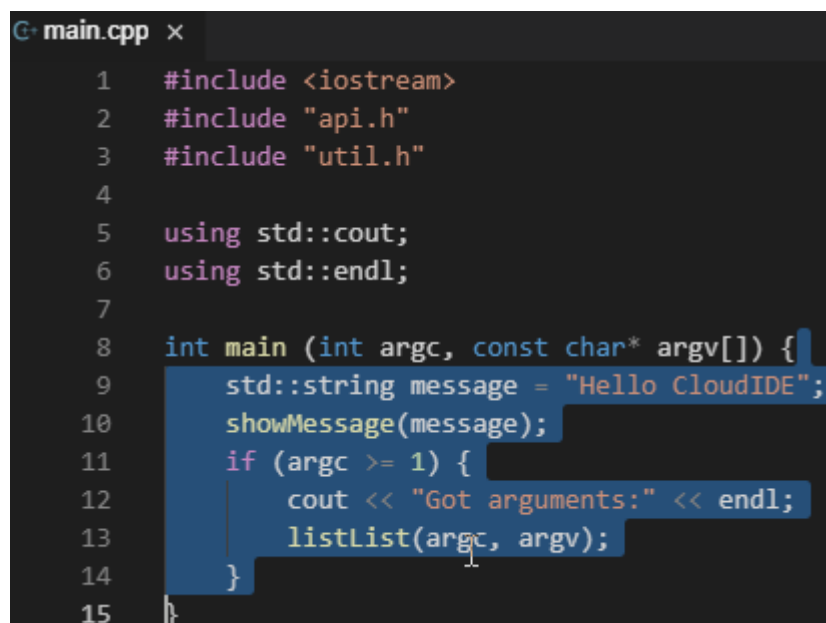


```
requirements.txt ●
1  astroid==2.3.3
2  Click==7.0
3  Flask==1.1.1
4  isort==4.3.21
5  itsdangerous==1.1.0
6  Jinja2==2.10.3
7  MarkupSafe==1.1.1
8  mccabe==0.6.1
9  pylint==2.4.4
10 six==1.13.0
11 Werkzeug==0.16.0
12 wrapt==1.11.2
```


调整选区

根据需要扩大或缩小选区范围，可以用“Shift+Alt+←”和“Shift+Alt+→”来缩小和扩大选区。如[图7-2](#)所示。

图 7-2 选中范围



```
G+ main.cpp x
1  #include <iostream>
2  #include "api.h"
3  #include "util.h"
4
5  using std::cout;
6  using std::endl;
7
8  int main (int argc, const char* argv[]) {
9      std::string message = "Hello CloudIDE";
10     showMessage(message);
11     if (argc >= 1) {
12         cout << "Got arguments:" << endl;
13         listList(argc, argv);
14     }
15 }
```

选择列

按住“Alt+Shift”同时拖动鼠标进行列选择，如[图7-3](#)所示。

图 7-3 选择列

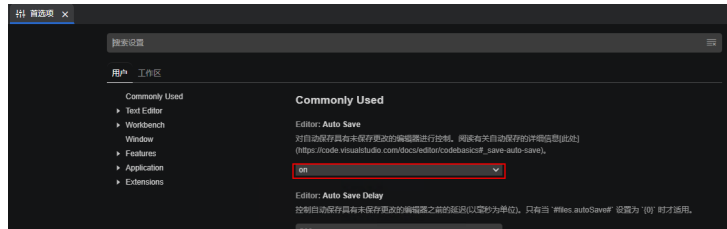



```
requirements.txt x
1  astroid==2.3.3
2  Click==7.0
3  Flask==1.1.1
4  isort==4.3.21
5  itsdangerous==1.1.0
6  Jinja2==2.10.3
7  MarkupSafe==1.1.1
8  mccabe==0.6.1
9  pylint==2.4.4
10 six==1.13.0
11 Werkzeug==0.16.0
12 wrapt==1.11.2
```

保存/自动保存

默认情况下，CodeArts IDE Online的自动保存为开启状态。可通过以下两种方式修改是否自动保存。

- 修改“Editor:Auto Save”配置。
 - a. 单击页面左下角。
 - b. 选择首选项。
 - c. 修改“Editor:Auto Save”配置为“on”或“off”。



- 直接通过菜单进行修改。
 - a. 单击页面左上角。
 - b. 选择“文件 > 自动保存”。

在自动保存关闭状态下，可用“Ctrl+S”快捷键进行保存。

查找与替换

可以快速地查找和替换字段。按下“Ctrl+F”键，输入要查找的文字，找到的匹配项会高亮。若有多个匹配项，可以用“Enter”和“Shift+Enter”在多个匹配项之间进行切换。

- **搜索选中文字**
选中一段文字，按下“Ctrl+F”可搜索选中文字。
- **在选区内查找**
选中一段区域，仅在选中区域中查找。
- **高级搜索与替换**
当前支持三种高级搜索功能，可在搜索框中选择启用以下高级特性。
 - 大小写匹配
 - 全词匹配
 - 正则表达式替换框默认隐藏，可单击搜索框左侧的箭头打开。

全局搜索

支持在当前工作空间中进行全局搜索。单击左侧菜单栏放大镜图标，打开全局搜索视图，在搜索框中输入需要搜索的内容，搜索结果将实时显示在下方，如图7-4所示。

图 7-4 全局搜索

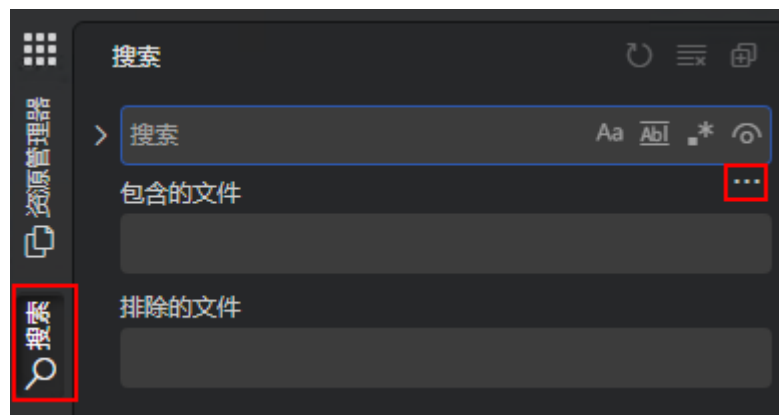


- **高级搜索**

全局搜索同样支持高级搜索，同时，还可以添加文件名作为过滤条件。

单击“...”，打开文件过滤框，在“包含的文件”中可以指定要包含的文件范围，在“排除的文件”中可以指定要排除的文件范围，如图7-5所示。

图 7-5 高级搜索

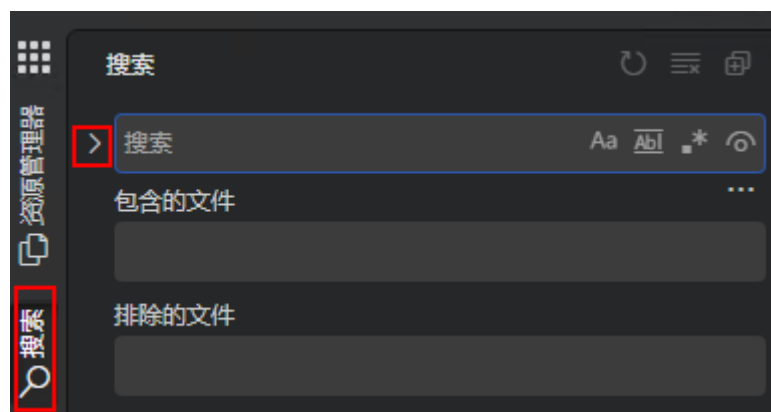


- **搜索与替换**

在全局搜索中，支持替换操作。

默认情况下，替换框呈隐藏状态，可以单击搜索输入框左侧的箭头，展开替换输入框。如图7-6所示。

图 7-6 搜索和替换



代码格式化

支持对代码进行格式化，当前支持两种格式化操作：

- 格式化文档
- 格式化选区

可以用快捷键“Ctrl+Shift+P”，或者用右键菜单使用格式化命令。

根据创建实例时选择的技术栈不同，CodeArts IDE Online支持格式化语言的种类略有不同。默认所有版本都支持JavaScript，TypeScript，JSON和HTML。以下配置可以更好地使用格式化功能：

- Editor.formatOnPaste
- Editor.formatOnSave
- Editor.formatOnSaveTimeout
- Editor.formatOnType

代码折叠

单击编辑器左侧的箭头，可以将代码按区域进行折叠。如图7-7所示。

图 7-7 代码折叠




```
main.cpp x
1  #include <iostream>
2  #include "api.h"
3  #include "util.h"
4
5  using std::cout;
6  using std::endl;
7
8  int main (int argc, const char* argv[]) {
9      std::string message = "Hello CloudIDE";
10     showMessage(message);
11     if (argc >= 1) { ...
14     }
15 }
16
```

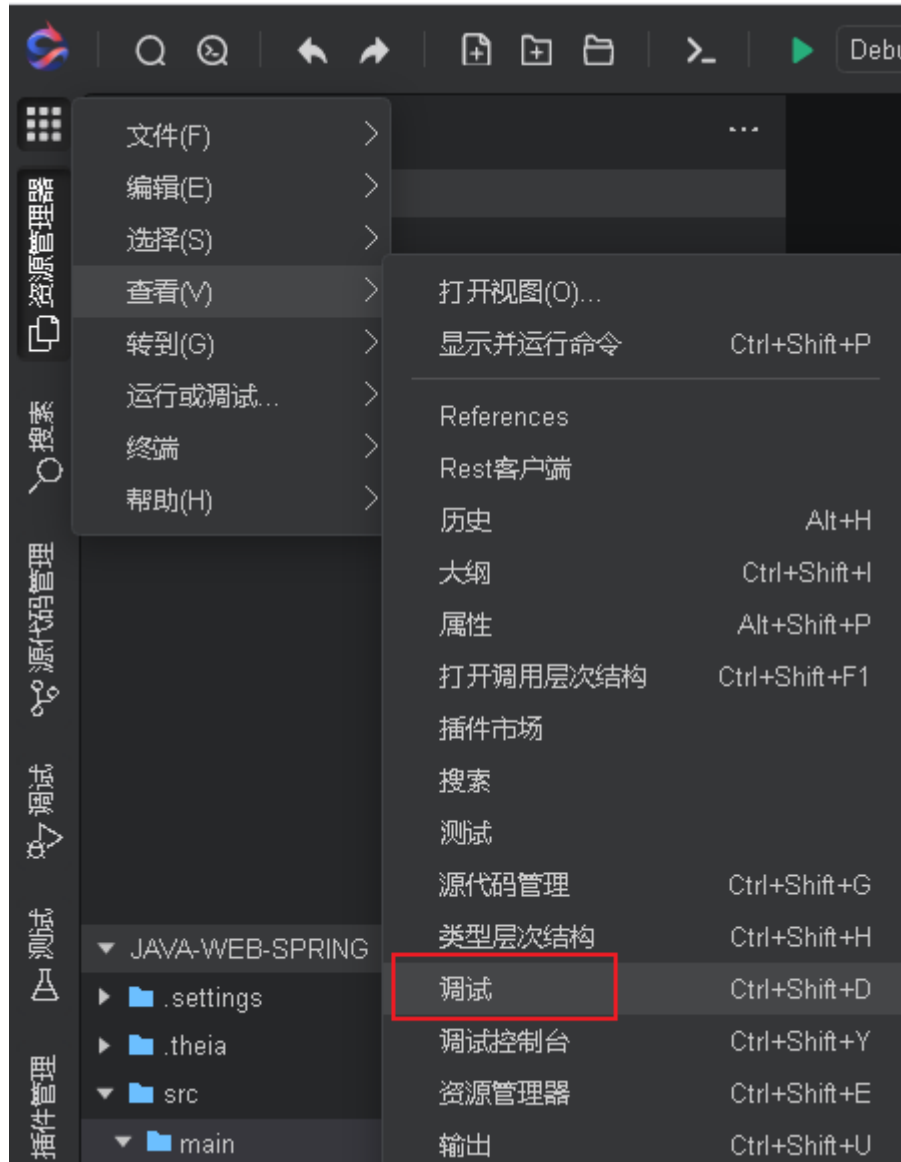
8 运行/调试代码

准备好工程后，可以根据实际需要运行或调试代码。

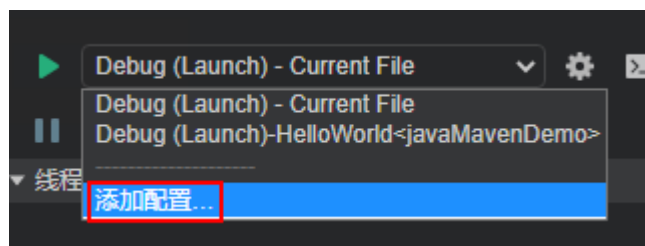
操作步骤

步骤1 进入CodeArts IDE Online代码编辑界面。

步骤2 单击左上角，选择“查看 > 调试”。



步骤3 在调试任务下拉框中选择“新增配置”。



步骤4 选择“Launch Program”。

步骤5 编辑launch.json文件，根据自动补全给出的提示填写相关参数。

步骤6 单击  启动/调试程序。

----结束


9 提交代码

在线修改代码后，CodeArts IDE Online支持通过“SCM”选项提交代码，还可以查看代码提交状态、提交历史等。

如创建IDE实例配置工程时选择“私有仓库”，可将更新后的代码提交到代码仓库。

操作步骤

步骤1 进入CodeArts IDE Online工作界面，在线修改目标文件代码。

步骤2 单击左上角，选择“查看 > 源代码管理”，开始提交代码。

步骤3 代码提交成功后，单击右上角“... > 推送”菜单，将代码推送到远程仓库。

（可选）单击右上角“... > 拉取”菜单，可以拉取最新代码到本地。

（可选）选择“查看 > 历史”菜单，可以查看代码提交的历史记录。

----结束

10 通用操作

10.1 终端基本操作


CodeArts IDE Online集成了在线终端的功能，可以直接在CodeArts IDE Online中打开终端进行操作。基本操作如下：

- [打开终端](#)
- [管理多终端](#)
- [拆分多终端](#)
- [复制黏贴](#)
- [更改样式](#)
- [运行选中命令](#)

登录工作界面请参考[启动IDE实例](#)。

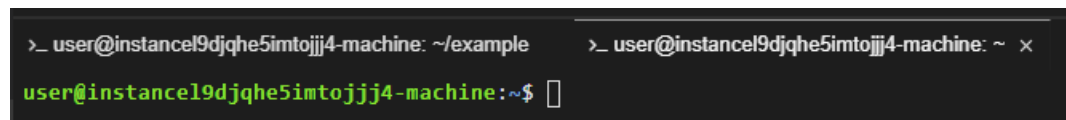
打开终端

可以通过以下两种方式打开终端：

- 使用“Ctrl+`”或“Ctrl+Shift+`”快捷键。
- 单击左上角，选择“终端> 新建终端”。

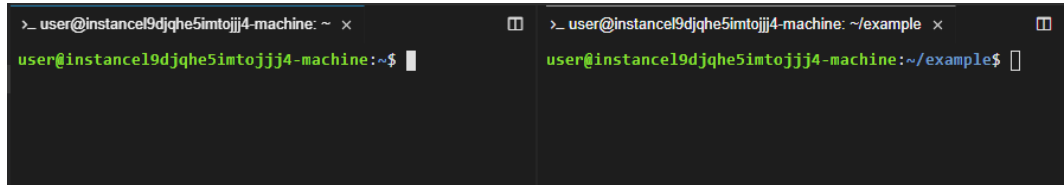
管理多终端

支持建立多个“终端”使用。



拆分多终端

可以单击将“终端”进行拆分；也可拖动“终端”窗口，将其任意组合。



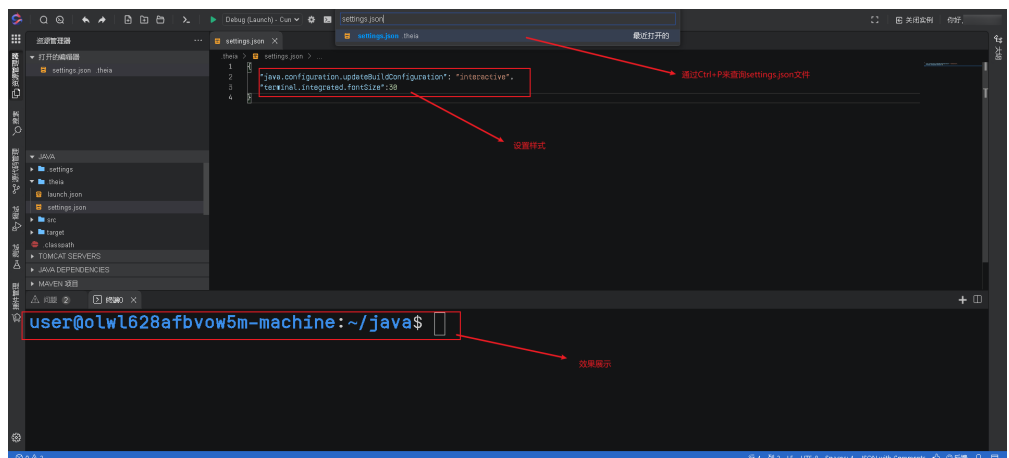
复制黏贴

在“终端”中，可以直接选中文字，使用快捷键“Ctrl+C”和“Ctrl+V”进行复制和粘贴。

更改样式


支持在settings.json文件里自定义Terminal的样式，如：字体大小、行间距等。可参考以下参数项进行配置：

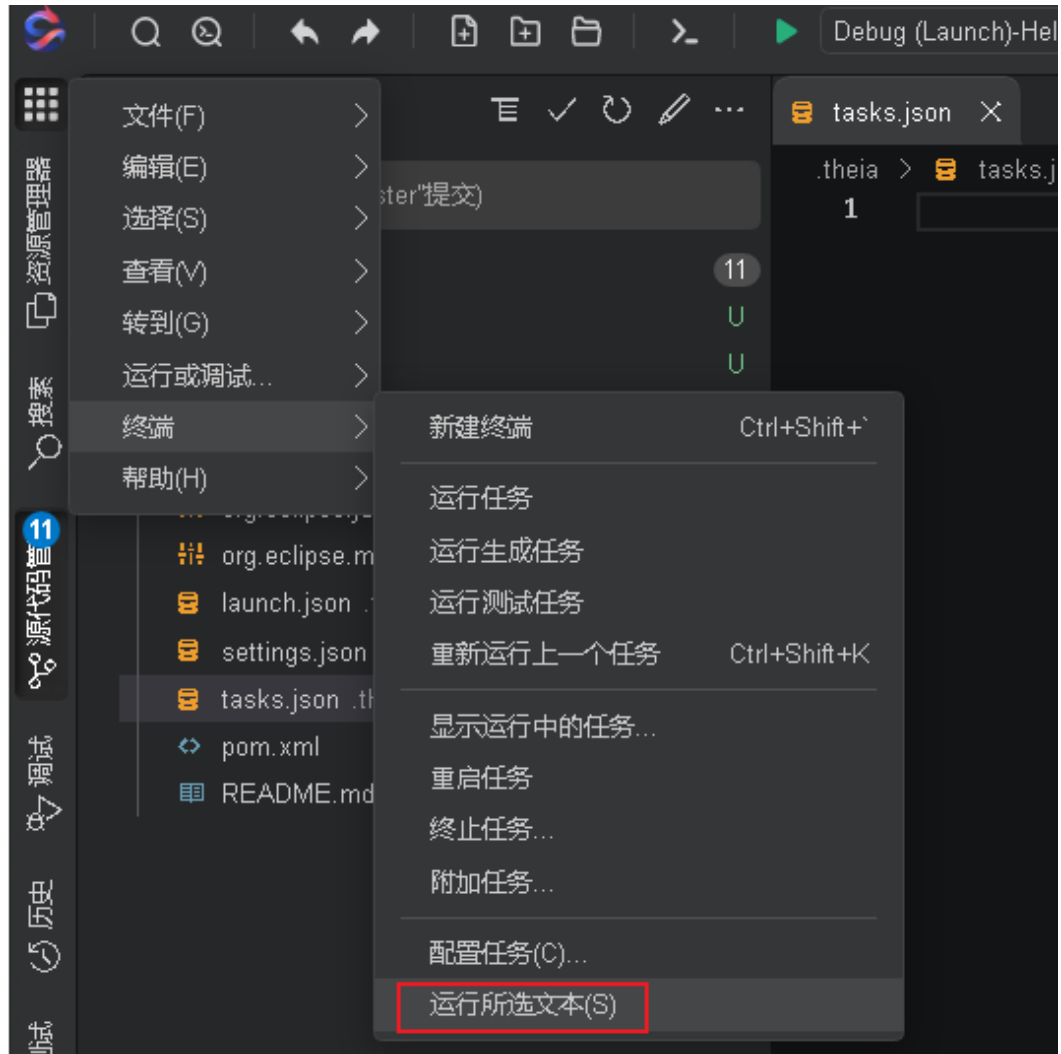
- terminal.integrated.fontFamily
- terminal.integrated.fontSize
- terminal.integrated.fontWeight
- terminal.integrated.fontWeightBold
- terminal.integrated.letterSpacing
- terminal.integrated.lineHeight



运行选中命令

如果需执行代码或文件中出现的命令，可参考以下两种方式操作：

- 选中命令，执行“运行所选文本”命令。
- 选中命令，单击左上角 ，选择“终端 > 运行所选文本”。



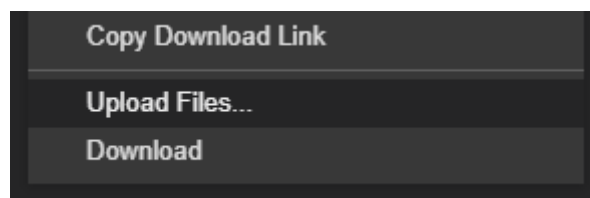
10.2 上传/下载文件

支持将文件上传到工作空间或从工作空间下载到本地。具体操作参考如下：

步骤1 进入CodeArts IDE Online工作界面。

步骤2 可通过以下两种方式进行上传。

- 在“资源管理器”中单击鼠标右键，选择“Upload Files...”或者“Download”，即可上传或下载文件。



- 将文件拖拽到“资源管理器”中，直接上传到指定的目录。

----结束

10.3 自定义命令

如果在进行代码构建、测试、运行、调试前，需要执行自定义运行命令的任务，可通过编辑“tasks.json”文件，添加想要执行的任务。

操作步骤


步骤1 进入CodeArts IDE Online代码编辑界面。

步骤2 在工程根目录下创建“.theia”目录（如已存在请跳过该步骤）。

步骤3 在“.theia”目录下新建“tasks.json”文件。

步骤4 将如下配置信息复制到“tasks.json”文件中，以maven的package命令为例。

```
{
  "version": "2.0.0",
  "tasks": [
    {
      "label": "package",
      "type": "shell",
      "command": "mvn package",
      "group": "build",
      "presentation": {
        "reveal": "always",
        "panel": "new"
      }
    }
  ]
}
```

步骤5 单击左上角。

步骤6 选择“终端> 运行任务”，在命令行运行新定义的命令。

----结束

11 Java 技术栈相关操作

本节主要介绍Java技术栈的相关操作，具体如下：

- [编译Maven项目](#)
- [运行代码](#)
- [添加断点](#)
- [添加调试配置](#)
- [代码调试](#)
- [修改变量](#)
- [JUnit测试](#)

编译 Maven 项目

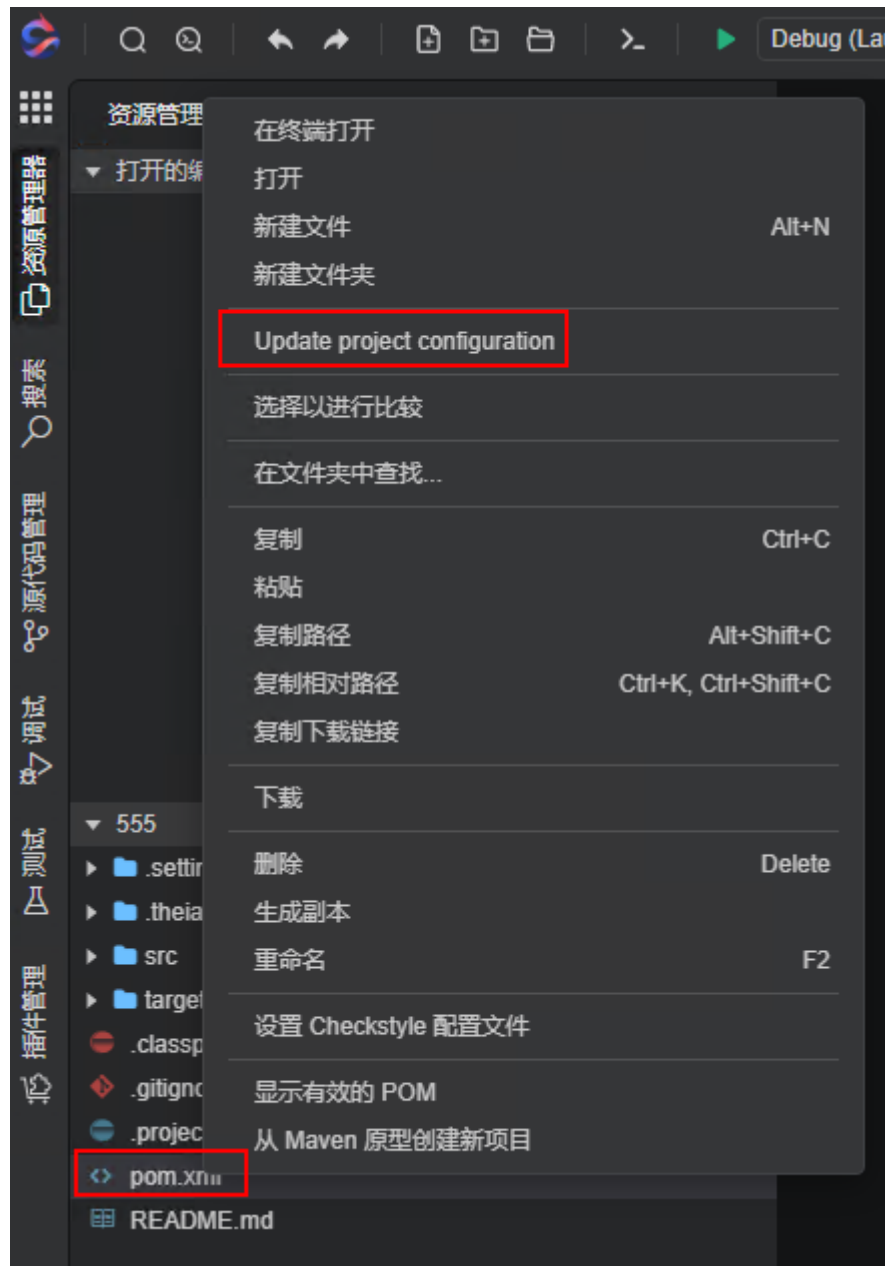
Maven项目导入后IDE会进行自动编译，如果没有触发自动编译可以手工进行编译操作。

步骤1 [进入CodeArts IDE Online代码编辑界面](#)。

步骤2 在“资源管理器”中打开项目。

步骤3 鼠标右键单击项目中的“pom.xml”文件，选择“Update project configuration”，项目会自动刷新并重新编译，如[图11-1](#)所示。

图 11-1 更新项目配置并编译




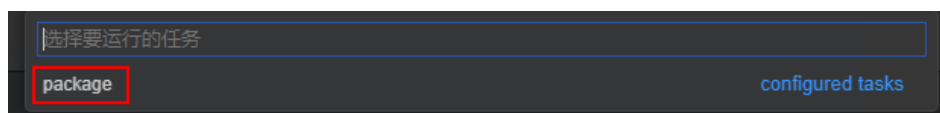
如果无法正常编译，请按照如下步骤使用编译命令进行编译。

1. 在“./theia”目录下，鼠标右键，选择“新建文件”。
2. 在弹出的窗口中填写文件名称“tasks.json”。
3. 将如下内容复制到“tasks.json”文件中，此处以Maven的package命令为例。

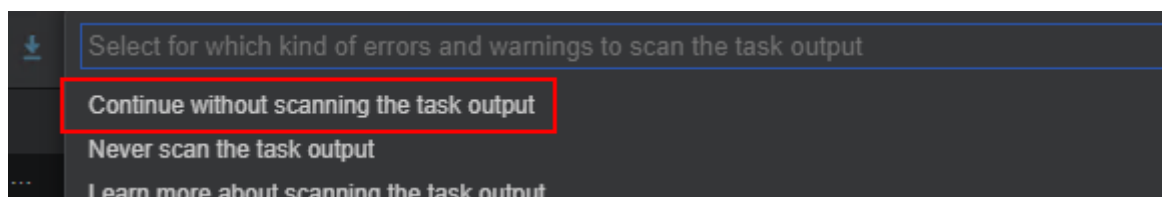
```
{  
  "version": "2.0.0",  
  "tasks": [  
    {  
      "label": "package",  
      "type": "shell",  
      "command": "mvn package",  
      "group": "build",  
      "presentation": {  
        "reveal": "always",
```

```
        "panel": "new"  
    }  
  }  
]  
}
```

4. 单击页面左上角.
5. 选择“终端 > 运行任务”。
6. 在弹出的窗口中选择“package”。



7. 选择“Continue without scanning the task output”，等待项目编译。



----结束

运行代码

CodeArts IDE Online支持直接运行Java的main函数。支持手动编译，或等待自动编译完成。手动编译操作步骤如下：

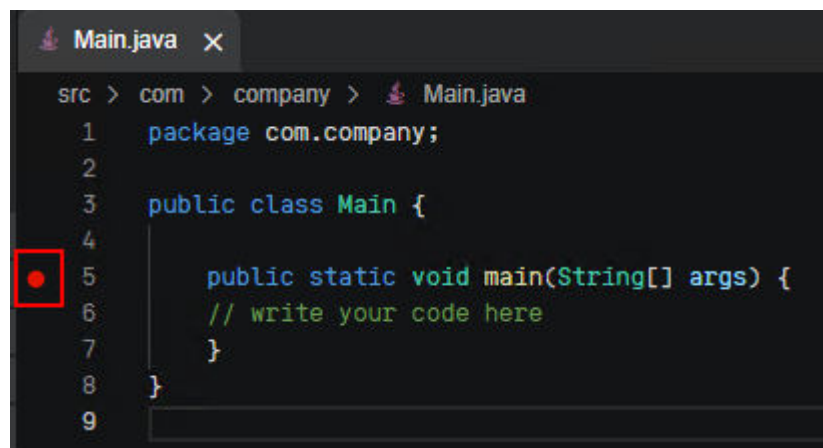
- 步骤1 双击文件名称，打开文件，等待“Run | Debug”按钮显示在main函数上方。
- 步骤2 单击“Run”，运行main函数。

----结束

添加断点

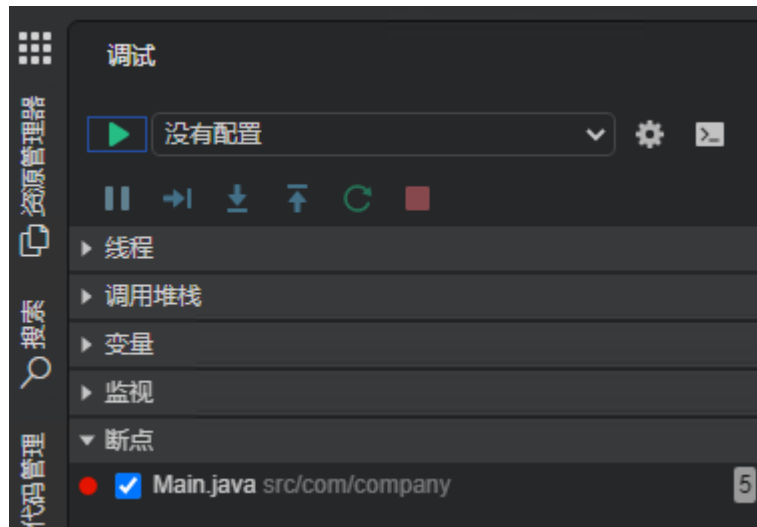
- 步骤1 双击文件名称，打开文件。
- 步骤2 将鼠标移到需要添加断点的行的左侧，显示暗红色小圆点后单击鼠标左键，红点高亮后说明打断点成功，如图11-2所示。

图 11-2 成功打断点



在调试视图中可以查看所有的断点，并可勾选是否启用，如图11-3所示。

图 11-3 查看所有断点



----结束

添加调试配置


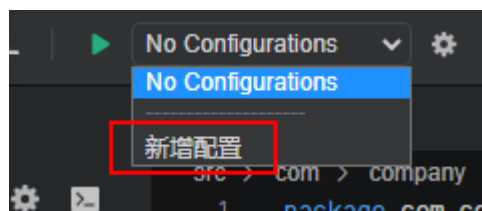

- 步骤1 单击左上角.
- 步骤2 选择“查看 > 调试”，打开调试视图。
- 步骤3 在调试任务下拉框中选择“新增配置”，如图11-4所示。

图 11-4 新增配置



- 步骤4 选择“Launch Program”，根据界面提示在配置文件中设置需要调试的“mainClass”。
- 步骤5 单击调试视图中的，开始调试。

----结束


代码调试

- 步骤1 双击文件名称，打开文件。
- 步骤2 单击main函数上的“Debug”按钮来调试代码。

----结束

修改变量


支持在调试过程中修改变量。

- 步骤1** 单击左上角.
- 步骤2** 选择“查看 > 调试”，打开调试视图。
- 步骤3** 调试并停留在断点处。
- 步骤4** 在“调试控制台”中修改变量值。

----结束

JUnit 测试

CodeArts IDE Online支持配合JUnit测试使用。

- 步骤1** 单击左上角.
- 步骤2** 选择“文件 > 打开”，打开带有JUnit测试用例的工程。
- 步骤3** 鼠标左键双击文件名称，打开测试用例文件，等待“Run Test | Debug Test”出现。
- 步骤4** 单击“Run Test”，等待测试完成。
- 步骤5** 在弹出“Java Test Report”页面中查看错误信息。

报告页面也可以通过单击“Run Test | Debug Test”旁的或打开。

----结束

12 C/C++技术栈相关操作

12.1 代码编辑

CodeArts IDE Online的C/C++插件功能强大，能够帮助用户编写代码，为得到最佳体验，需要为C/C++插件提供头文件的位置。如果头文件放置在以“include”命名的文件夹中，则C/C++插件会自动识别。

其它情形下，需设置“clangd.includePath”选项（请参考[配置头文件路径和宏](#)）。


对于复杂的工程，推荐配置“compile_commands.json”文件（请参考[精确配置代码跳转](#)）。

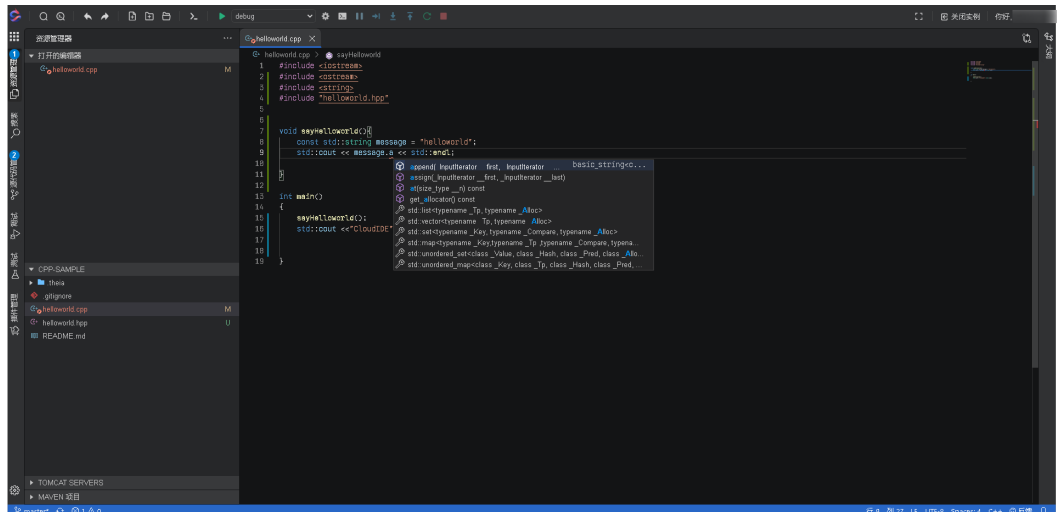
基本编辑操作如下：

- [代码提示](#)
- [代码格式化](#)
- [查找引用](#)
- [配置头文件路径和宏](#)
- [精确配置代码跳转](#)

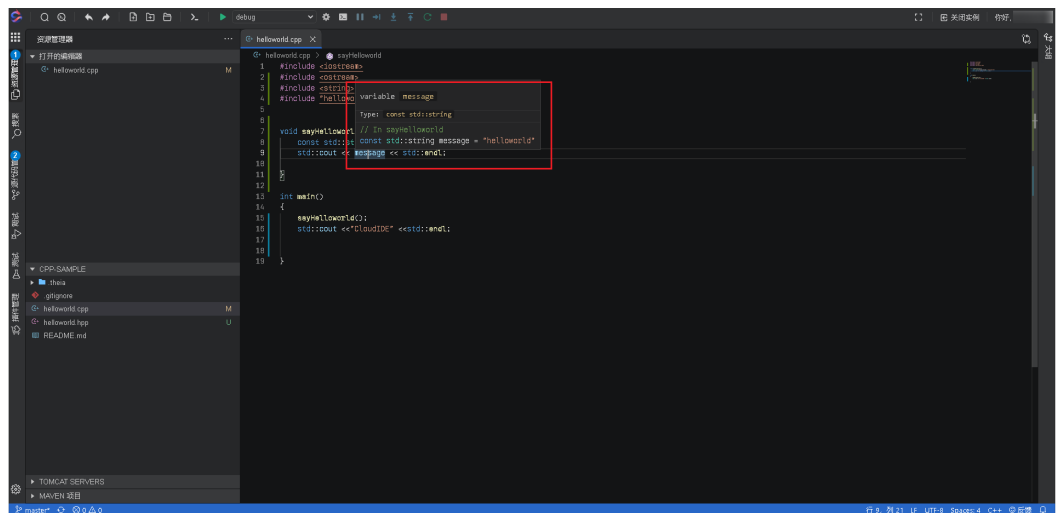
登录工作界面请参考[启动IDE实例](#)。

代码提示

CodeArts IDE Online可以根据输入的内容，实时提示代码详细信息。单击提示中的，还可以查看函数、变量的详细信息。



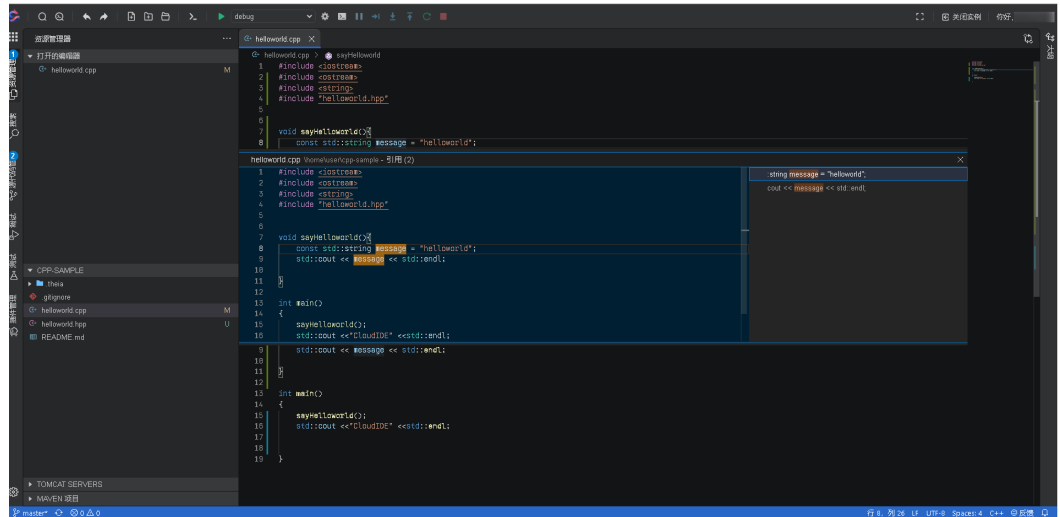
将鼠标悬停在代码变量上，可以显示变量的相关信息。



查找定义

以下三种方式均可以查找符号定义。

- 将光标移到需要查看的符号上，鼠标右键选择“缩略 > Peek Definition”。
- 同时按下“Alt+F12”键。
- 按下“Ctrl”键，鼠标左键单击要查看的符号



代码格式化

可使用以下两种方式进行格式化：

- 使用“Format Document”命令，可对当前编辑器的代码进行格式化。
- 按下“Alt+Shift+F”组合键，可对当前编辑器的代码进行格式化。
- 选中部分代码，使用“Format Selection”命令进行格式化。

Change All Occurrences	Ctrl+F2
Format Document	Alt+Shift+F
Format Selection	Ctrl+K, Ctrl+F
Refactor...	Ctrl+Shift+R

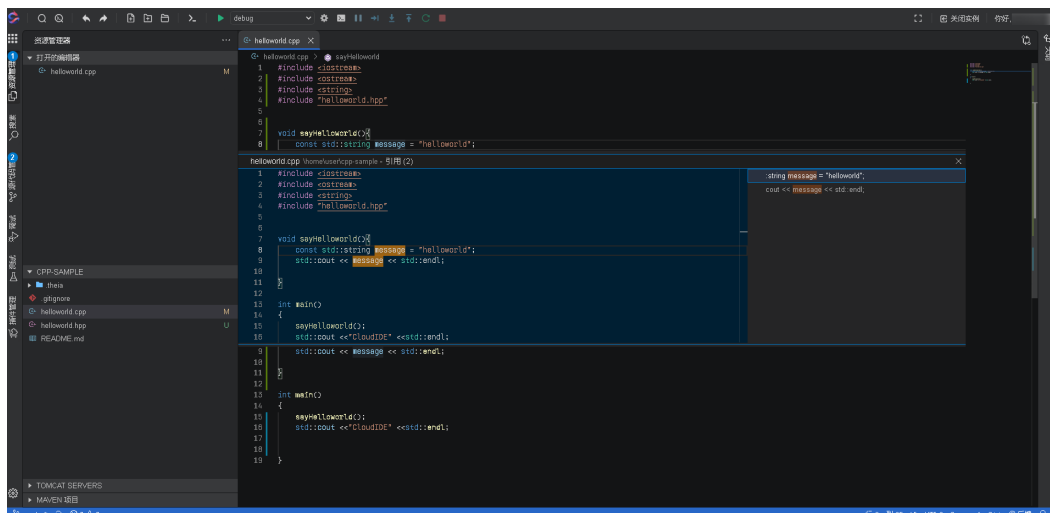
在工程中添加一个“.clang-format”文件，可以定制格式化样例，“.clang-format”文件的配置可参考如下：

```
---
BasedOnStyle: LLVM
---
Language: Cpp
AccessModifierOffset: -4
AlignESCAPEDNEWLINES: DONTALIGN
ALLOWSHORTIFSTATEMENTSONASINGLELINE: TRUE
ALLOWSHORTLOOPSONASINGLELINE: True
BreakBeforeBraces: Custom
BraceWrapping:
  AfterFunction: true
  ColumnLimit: 0
  ConstructorInitializerAllOnOneLineOrOnePerLine: true
IncludeCategories:
- Regex: '^<ext/.*\.h>'
Priority: 2
- Regex: '^<.*\.h>'
Priority: 1
- Regex: '^<.*'
Priority: 2
- Regex: '.*'
Priority: 3
IndentCaseLabels: true
IndentWidth: 4
ObjCBlockIndentWidth: 4
```

```
SortIncludes: false  
SortUsingDeclarations: false  
SpaceAfterCStyleCast: false  
SpacesBeforeTrailingComments: 2  
UseTab: Never  
TabWidth: 4
```


查找引用

将光标移到需要查找的符号上，鼠标右键选择“缩略 > Peek Reference”，可以查看符号引用情况。



配置头文件路径和宏

CodeArts IDE Online可对工程进行自动扫描。但对于需要额外配置头文件路径或者宏的工程，自动扫描与配置可能会失效，此时需要在配置中为工程添加相应配置。

步骤1 单击左上角 ，选择“文件 > 首选项 > 首选项”。

步骤2 添加“clangd.defines”和“clangd.includePath”配置，并保存。

----结束

精确配置代码跳转

对于复杂工程，如果自动扫描结果不能令人满意，则需要配置一个名为“编译数据库”的文件，配合C/C++插件使用。

“编译数据库”是一个JSON格式、名为“compile_commands.json”的文件。当前可以通过以下三种方式生成“compile_commands.json”。CMake和Clang自带支持生成“compile_commands.json”，对于make和ninja管理的工程，则可以用bear工具，具体操作参考如下：

- **CMake**
 - 设置 **CMAKE_EXPORT_COMPILE_COMMANDS**，可以带参数执行。
cmake -DCMAKE_EXPORT_COMPILE_COMMANDS=ON...
 - 或在“CMakeLists.txt”文件中添加如下信息。
set(CMAKE_EXPORT_COMPILE_COMMANDS ON)
- **Clang (version 5.0 and later)**

使用 **-Mj** 参数 (参考)。

- **Bear 与 intercept-build 工具**

Bear 和 **intercept-build** (来自 **scan-build**) 可以帮助生成“**compile_commands.json**”文件。Ubuntu系统可用“**sudo apt-get install bear**”命令安装 **bear**工具。

生成“**compile_commands.json**”文件后，CodeArts IDE Online会询问是否将此文件配置为“**compileCommands**”。

下面以**bear**和**make**为例，介绍详细操作：

步骤1 单击左上角，选择“终端 > 新建终端”。或者使用快捷键“**Ctrl+Shift+`**”打开“终端”。

步骤2 在终端窗口执行“**bear make**”命令进行编译，Makefile的内容如下：

```
default:  
g++ -g -o helloworld src/helloworld.cpp -DVERBOSE
```

步骤3 **bear**在根目录生成了“**compile_commands.json**”文件后，CodeArts IDE Online提示是否将包含此文件的目录，设置为**compileCommands**值，单击“**Yes**”。

步骤4 CodeArts IDE Online提示是否重新加载“**compile_commands.json**”使配置生效，单击“**Reload**”。

----结束

12.2 代码调试

CodeArts IDE Online 支持C/C++工程调试，调试之前需满足以下条件：



- 编译时，已在“**.theia/tasks.json**”中需要加上调试参数“**-g**”，如：“**g++ -g -o hello helloworld.cpp**”，默认已经添加，如没有，需要手动添加。
- 已编译工程，生成可执行文件。

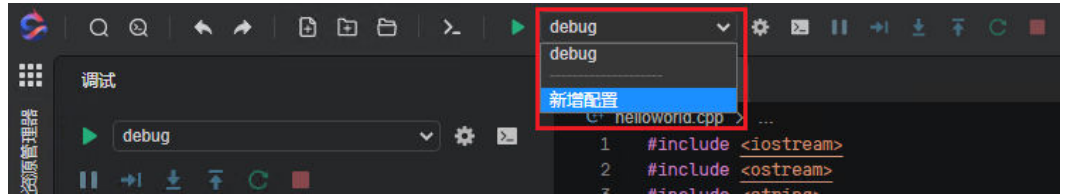
具体相关操作如下：

- [添加调试配置](#)
- [添加断点](#)
- [调试代码](#)
- [查看变量值](#)
- [查看表达式值](#)
- [单文件一键调试](#)

登录工作界面请参考[启动IDE实例](#)。

添加调试配置

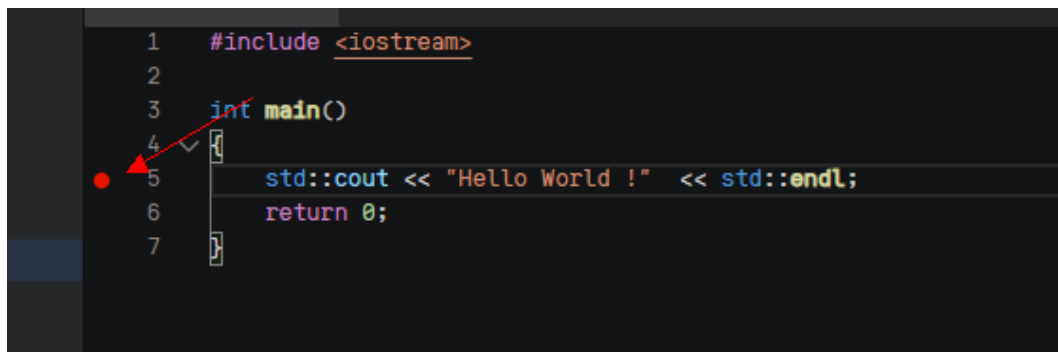
1. 单击左上角，选择“查看 > 调试”，或者单击左侧菜单栏，打开调试视图。
2. 在调试任务下拉框中选择“新增配置”。



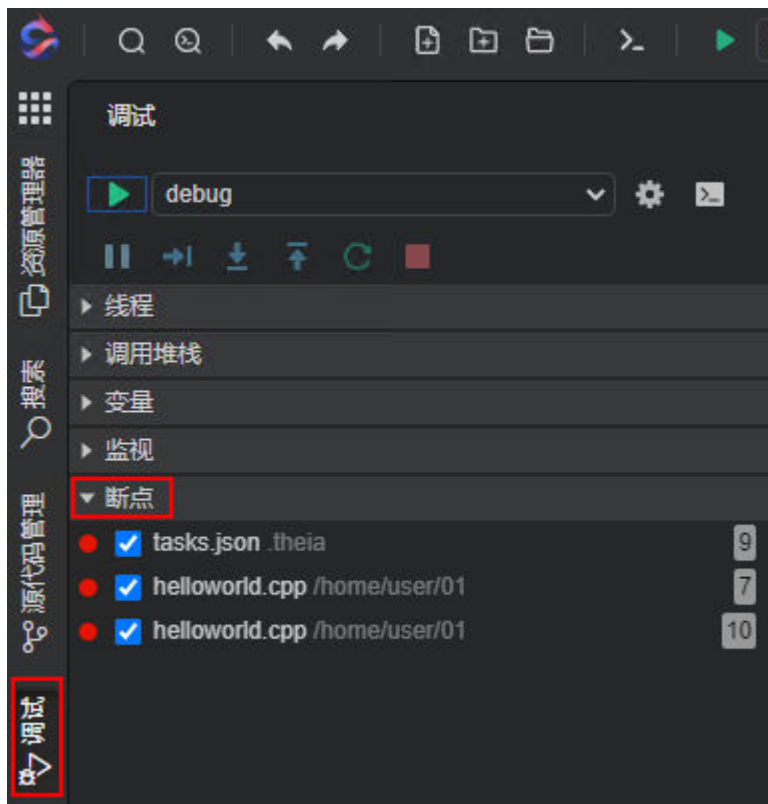
3. 然后选择“GDB: Launch Program”模板。
4. 修改其中的“program”，指向可执行文件。

添加断点

将鼠标移到需要添加断点的行的左侧，显示暗红色小圆点后单击鼠标左键，红点高亮后说明打断点成功。

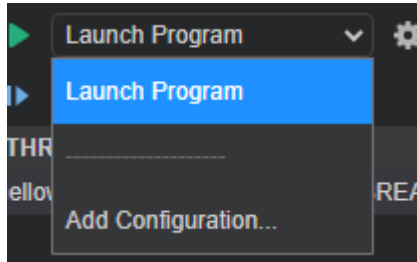



在调试视图中可以查看所有的断点，并可勾选是否启用。



调试代码

步骤1 在调试视图中选择**添加调试配置**时添加的配置。



步骤2 单击 ，或者使用“F5”快捷键，开始调试。

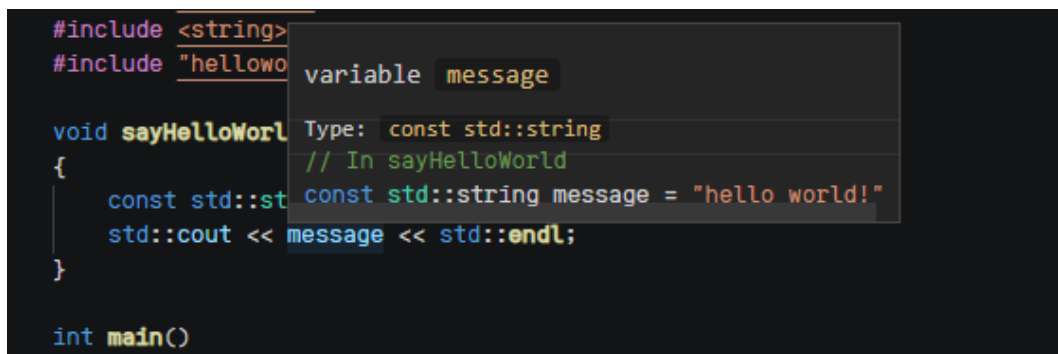
可使用控制按钮进行调试，按钮从左到右分别为“Continue”、“Step Over”、“Step Into”、“Step Out”、“Restart”和“Stop”。



----结束

查看变量值


在调试运行过程中，将鼠标移动到想查看的变量值上，即可查看变量的值。

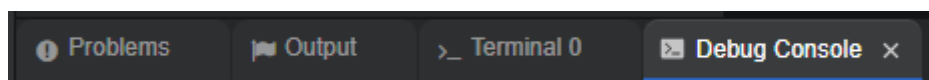


在调试视图中可查看的变量列表。



查看表达式值

1. 单击顶部菜单栏 ，切换到“调试控制台”视图。



2. 在调试控制台底部区域输入表达式。

```
> p message
```

3. 按下键盘回车键，即可在调试控制台区域上方看到表达式的值。

```
p message  
$1 = "hello world!"
```

单文件一键调试

对于单文件或多文件工程，CodeArts IDE Online支持一键编译调试。操作步骤如下：

1. 打开源文件，参考[添加断点](#)为文件添加断点。
2. 选中需要调试的代码，鼠标右键选择“Build and Debug Active File”。
3. 选择执行平台。
 - 默认选择Internal Console。
 - 若程序需要从stdin读取，需要选择Integrated Terminal。

📖 说明

也可以打开.theia/launch.json，把配置中的console项设置为internalConsole或integratedTerminal进行修改。

12.3 常见问题

如何实现每次调试前，代码自动编译？

修改代码后，需重新编译再进行调试。可以新建一个编译工程的Task，并将其label配置为调试配置项的preLaunchTask，以达到调试前自动编译的目的。具体操作如下：

步骤1 进入CodeArts IDE Online工作界面。

步骤2 新建一个名为“build”（也可自定义其它名字）的task。如：在“.theia”目录中新建一个“tasks.json”文件，拷贝如下配置。其中可以把“command”和“args”替换成用户需要使用的编译命令。

```
{  
  "version": "2.0.0",  
  "tasks": [  
    {  
      "label": "build",  
      "type": "shell",  
      "command": "g++",  
      "args": [  
        "-g",  
        "-o",  
        "main",  
        "helloworld.cpp"  
      ]  
    }  
  ]  
}
```

步骤3 为调试配置添加“preLaunchTask”，其值步骤**步骤2**添加的task名一致，如：
"preLaunchTask": "build"

步骤4 CodeArts IDE Online将在执行“preLaunchTask”成功后，运行调试。

----结束

13 Python 技术栈相关操作

本节介绍Python技术栈的相关操作，具体如下：

- [切换Python版本](#)
- [安装PyLint](#)
- [安装依赖库](#)
- [运行/调试代码](#)
- [代码格式检查](#)
- [代码测试](#)

登录工作界面请参考[启动IDE实例](#)。

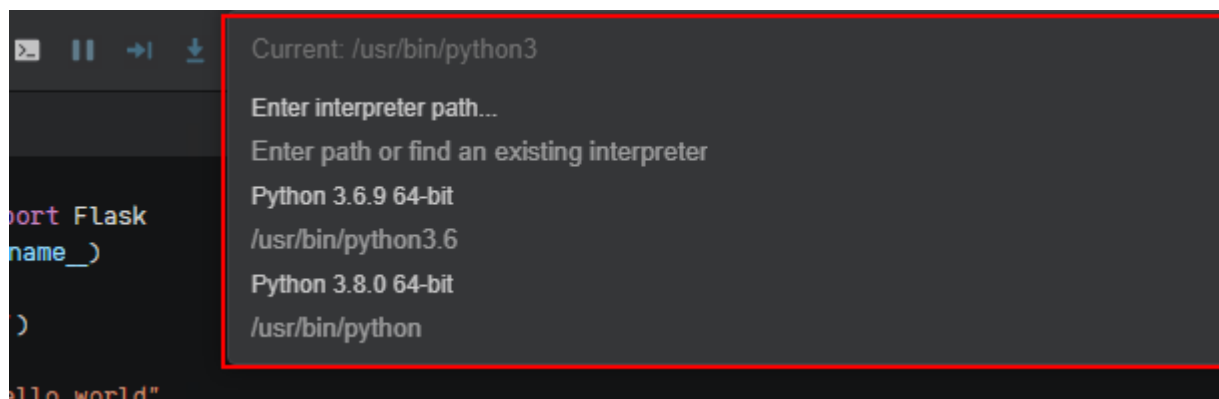
切换 Python 版本

CodeArts IDE Online可以快速切换当前使用的Python版本，具体操作如下：

步骤1 单击页面左下角的Python版本。



步骤2 在页面中上方弹出的下拉框中选择需要更换的Python版本。Python允许多个版本共存，但各版本安装的第三方库不共享，使用时请注意当前的Python版本。




----结束

安装 PyLint

快速安装:


打开CodeArts IDE Online时，在页面右下角的弹框中，单击Install，PyLint会自动安装到当前选择版本的Python中。

手动安装:

1. 单击左上角 ，选择“终端 > 新建终端”，或使用快捷键“Ctrl+Shift+`”打开终端。
2. 执行“python3/python2 -m pip install -U pylint --user”命令安装 Pylint。



安装依赖库

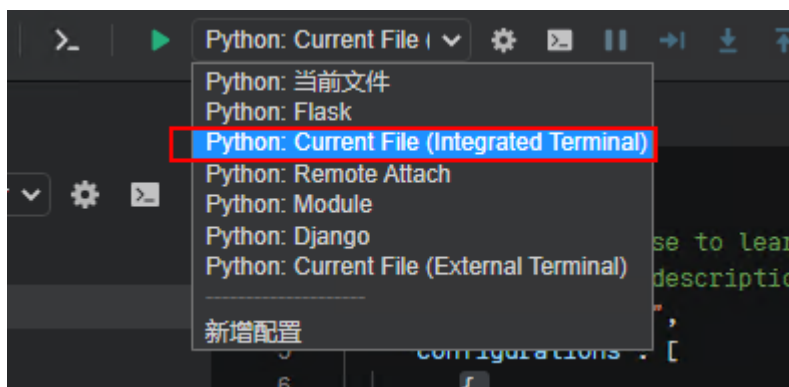
安装依赖需要指定Python，如：Python3使用“python3 -m pip install --user 依赖库”安装。具体操作如下：

1. 单击左上角 ，选择“终端 > 新建终端”，或使用快捷键“Ctrl+Shift+`”打开终端。
2. 在终端窗口执行“python3 -m pip install --user 依赖库”命令，安装依赖库。

运行/调试代码

CodeArts IDE Online可以对当前文件进行调试，具体操作如下：

- 步骤1** 单击左上角 ，选择“查看 > 调试”，可打开调试视图。
- 步骤2** 单击左上角 ，选择“查看 > 调试控制台”，可打开调试控制台。
- 步骤3** 在调试下拉框中选择“Python: Current File (Integrated Terminal)”。



步骤4 选中文件，参考[添加断点](#)，可在文件的代码行添加断点。

步骤5 单击，开始调试代码。

鼠标停留在变量上可查看变量值。


在“调试控制台”中可查看和修改变量值，在“Python Debug Console”中可查看输出结果。

----结束

代码格式检查

前提条件：已[安装PyLint](#)。

具体操作如下：格式检查默认开启，如果需要手动开启，可以执行“Python: Enable Linting”命令。

步骤1 单击左上角，选择“查看 > 显示并运行命令”，或使用快捷键“F1”打开命令窗口。

步骤2 输入lint，找到“Python: 启用 Linting”命令执行。

步骤3 选择“on”开启格式检查。


----结束

代码测试

CodeArts IDE Online支持unittest, pytest和nose对python进行代码测试，其中，pytest同时支持unittest和nose。

前提条件：已[安装PyLint](#)。

具体操作如下：


步骤1 单击左上角，选择“查看 > 显示并运行命令”，或使用快捷键“F1”打开命令搜索框。

步骤2 找到并执行“Python: Configure Tests”命令。

步骤3 在下拉框中选择“pytest”。

步骤4 选择tests目录。

步骤5 单击代码中“Run Test | Debug Test”按钮中的“Run Test”，开始进行代码测试。

单击页面左下角状态栏中的  3，选择“View Test Output”，可查看测试报告。

----结束

14 NodeJs 技术栈相关操作


本节介绍NodeJs技术栈的相关操作，具体如下：

- [安装依赖包](#)
- [查看库包的逻辑实现](#)
- [运行/调试代码](#)

登录工作界面请参考[启动IDE实例](#)。

安装依赖包

在终端窗口中直接运行npm install安装依赖的库包，操作步骤如下：


1. 单击左上角.
2. 选择“终端 > 新建终端”，或使用快捷键“Ctrl+Shift+`”打开终端。
3. 执行“spm install express path”命令安装依赖包。

查看库包的逻辑实现

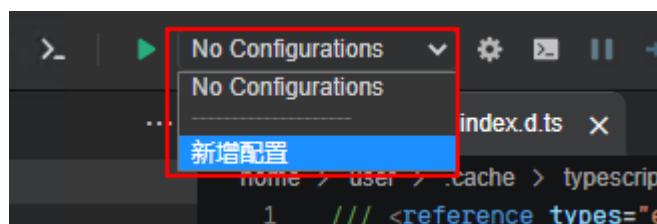
可通过以下两种方式实现：

- 将鼠标移动到引进来的库包名下，按下Ctrl键，光标变成手型时，单击鼠标左键可跳转到对应的逻辑实现。
- 将鼠标移动到引进来的库包名下，单击鼠标右键，选择“转到定义”。

运行/调试代码


步骤1 单击左上角，选择“查看 > 调试”，可打开调试视图。

步骤2 在调试任务下拉框中选择“新增配置”。



步骤3 选择“Node.js: Launch Program”。

步骤4 编辑program运行的文件路径。

步骤5 单击调试视图中的，开始调试代码。

步骤6 查看变量，可使用以下3种方式进行查看。

- 将鼠标停留在变量上直接查看。
- 在调试控制台输入框中输入变量进行查看。
- 在调试视图中VARIABLES状态栏进行查看。

----**结束**

15 Go 技术栈相关操作


本节介绍Go技术栈的相关操作，具体如下：

- [安装依赖包](#)
- [查看库包的逻辑实现](#)
- [运行/调试代码](#)
- [代码格式检查](#)
- [代码测试](#)

登录工作界面请参考[启动IDE实例](#)。

安装依赖包

使用CodeArts IDE Online的Go技术栈，可通过执行**go get**命令安装依赖包，执行命令前需先设置“GOPATH”环境变量，如：`export GOPATH=~/.gotools`，具体操作步骤如下：

- 步骤1** 单击左上角，选择“终端 > 新建终端”，或使用快捷键“Ctrl+Shift+`”打开终端。
- 步骤2** 在终端窗口执行**export GOPATH=~/.gotools**命令，设置“GOPATH”环境变量。
- 步骤3** 在终端窗口执行**go get -u -v github.com/nst/docode**命令，安装依赖包。


----结束

查看库包的逻辑实现

可通过以下两种方式实现：

- 将鼠标移动到引进来的库包名下，按下Ctrl键，光标变成手型时，单击鼠标左键可跳转到对应的逻辑实现。
- 将鼠标移动到引进来的库包名下，单击鼠标右键，选择“转到定义”。

运行/调试代码

- 步骤1** 单击左上角，选择“查看 > 调试”，可打开调试视图。

步骤2 单击调试视图中的，开始调试代码。

步骤3 查看变量，可使用以下3种方式进行查看。

- 将鼠标停留在变量上直接查看。
- 在调试控制台输入框中输入变量进行查看。
- 在调试视图中VARIABLES状态栏进行查看。

----结束

代码格式检查

CodeArts IDE Online默认采用的是go-lint工具进行格式检查。

代码测试

步骤1 鼠标移动到函数名处，单击鼠标右键。

步骤2 选择“Go:Generate Unit Tests For Function”，生成函数的测试用例。

步骤3 生成完毕后，在测试用例的go文件中，单击“run package tests”、“run file tests”、“run test”等链接，可批量测试或单个测试。

----结束

16 CloudShell 相关操作

16.1 约束限制

华为云CloudShell是一款用于管理与运维云资源的网页版Shell工具，通过CloudShell可以完成登录弹性云服务器、连接集群等操作。CloudShell目前支持Windows和Linux系统。

用户在使用CloudShell时，同时打开的实例数量和会话数量有如下限制：

表 16-1 约束限制

场景	实例数量	会话数量
连接云服务器	4	10个/实例
连接CCE集群/容器	15	不涉及
命令行工具	1	不涉及

16.2 文件树操作

CloudShell在命令行环境中提供了文件树操作能力，具体操作如下：

- [新建文件/目录](#)
- [编辑文件](#)
- [文件/目录重命名](#)
- [删除文件/目录](#)
- [下载文件](#)
- [上传文件](#)

相关登录

- [使用CloudShell登录云服务器](#)

- [登录CCE容器](#)
- [使用CloudShell连接集群](#)
- [使用CloudShell登录训练容器](#)

新建文件/目录

支持通过以下两种方式新建文件：

- 选择目录节点，单击鼠标右键，单击“新建文件”。
- 选择目录节点，在键盘上敲击快捷键“Alt+N”。

编辑文件

选择文件节点，双击鼠标左键，在编辑区打开的文件中修改内容并保存，保存的快捷键为“Ctrl+S”。

文件/目录重命名

有以下两种方式可以实现：

- 选择文件/目录节点，单击鼠标右键，单击“重命名”。
- 选择文件/目录节点，在键盘上敲击快捷键“F2”。

删除文件/目录

有以下两种方式可以实现：

- 选择文件/目录节点，单击鼠标右键，单击“删除”。
- 选择文件/目录节点，在键盘上敲击快捷键“Delete”。

下载文件

选择文件节点，单击鼠标右键，单击“下载文件”。单次文件下载，文件大小限制为500MB。

上传文件

有以下两种方式可以实现：

- 选择目录节点，单击鼠标右键，单击“上传文件”。
- 选择文件/目录节点，在键盘上敲击快捷键“Delete”。

约束与限制

- 目录非空时不可删除。
- 单次文件上传，文件容量限制为100MB，超过100MB可能会不稳定，或导致上传失败，建议不要超过这个上限。

16.3 命令行终端操作

相关登录

- [使用CloudShell登录云服务器](#)
- [登录CCE容器](#)
- [使用CloudShell连接集群](#)

快捷键

可使用快捷键编辑输入的命令。下表展示了常用的快捷键操作：

快捷键	功能描述
Ctrl+L	将当前行移到第一行
Ctrl+U	清除当前行
Ctrl+H	向前删除一个字符
Ctrl+A	光标移动到句首
Ctrl+E	光标移动到句末

复制/黏贴

可通过以下两种方式实现：

- 直接通过点击鼠标右键进行复制粘贴。
- 按下键盘“Ctrl+C”（复制）和“Ctrl+V”（黏贴）快捷键实现。

浏览输出历史

对于跨屏内容，支持滚动终端窗口查看历史输出，终端可记录最近1000行输出。

多终端分区布局

支持在同一个页面中创建多个CloudShell终端，并可以直接拖动窗口，随意组合布局。