华为云码道 CodeArts 代码智能体

最佳实践

文档版本 01

发布日期 2026-08-03

华为云计算技术有限公司

商标声明


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 基于自定义规则的 AI 绘图逻辑构建实践

应用场景

在AI辅助绘图的实际应用中，由于缺乏统一的规则约束，AI在理解用户提示词、构建画面结构以及处理元素之间逻辑关系方面仍存在较大的不确定性。这种随机性通常会导致生成结果与用户预期存在偏差，影响使用效率和创作精度。因此，通常需要用户不断优化提示词，并通过多轮迭代调整，才能逐步接近目标效果，提升AI绘图在实际场景中的可用性与可靠性。

本实践以华为云码道代码智能体IDE工具为例，介绍如何通过配置规则（Rules），对AI绘制sin(x)函数图像的过程进行规范约束，从而实现图像坐标精准、格式统一、结果可控的标准化生成。

操作流程

表 1-1 基于自定义规则的 AI 绘图逻辑构建实践操作流程

序号	步骤	说明
1	准备工作	1. 创建一个项目，用于存放工程中的各类文件。 2. 开启自动批准功能，减少人工干预。
2	创建无规则约束的sin(x)图像	创建一个无规则的sin(x)函数图像。
3	创建有规则约束的sin(x)图像	1. 编写绘图规范规则文件。 2. 根据规则文件生成规则限制的sin(x)图像。

准备工作

在使用华为云码道代码智能体前，您需要先创建一个项目，用于集中存放工程中的各类文件。

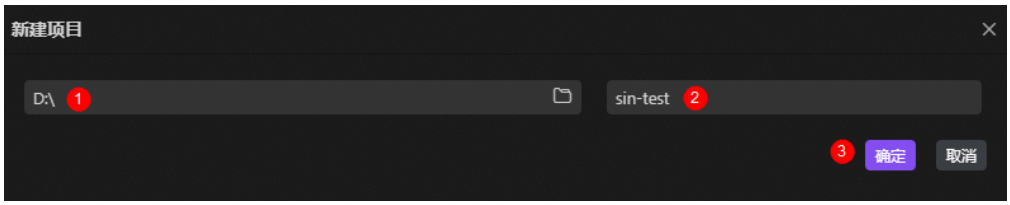
步骤1 参考快速启动中操作，登录华为云码道代码智能体。

步骤2 创建一个项目，用于存放工程中的文件。

1. 在IDE工具顶部菜单栏中，单击“文件(F)”，选择“新建 > 新建项目”，进入新建项目页面。
2. 选择项目存放位置，输入项目名称（如sin-test），单击“确定”。

   项目名称必须以字母开头，可包含字母、数字、中划线或下划线，且总长度不能超过64个字符。

图 1-1 新建项目



项目创建完成后，在“资源管理器”中可以查看到已创建的项目。

图 1-2 查看已创建的项目

步骤3 （可选）选择智能体运行模式并授权自动化操作。

1. 单击华为云码道代码智能体IDE右上角的设置图标  ，进入全局设置页面。
2. 在“对话流 > 智能体 > 终端命令运行模式”中，选择智能体执行终端命令的运行策略，本实践使用默认运行策略，即沙箱运行。
3. 在“对话流 > 智能体 > 自动批准”中，单击  以开启所需的自动批准项目。

   授权自动化操作后，复杂工程级代码生成流程中的各项任务可自动执行。如果您未开启自动化操作授权，在使用华为云码道代码智能体进行编码时，部分操作将需要您手动确认。

注意

开启自动批准存在操作风险，请在开启前充分评估风险，并在安全可信环境中使用。

表 1-2 自动批准参数说明

参数	说明	示例
编辑	允许智能体调用edit、write、deleteFile等工具来编辑您计算机上的文件。	开启
使用浏览器	允许智能体在浏览器中访问网站。	开启

参数	说明	示例
网页抓取工具	允许智能体访问并抓取指定网页的内容。	开启

4. 退出当前页面，完成授权操作。

----结束

创建无规则约束的 sin(x)图像

为了更清晰地展示规则绘图的效果，此处将首先展示**未应用规则**时的绘图结果。

步骤1 在聊天界面的输入框中，输入如下指令，单击发送图标 。

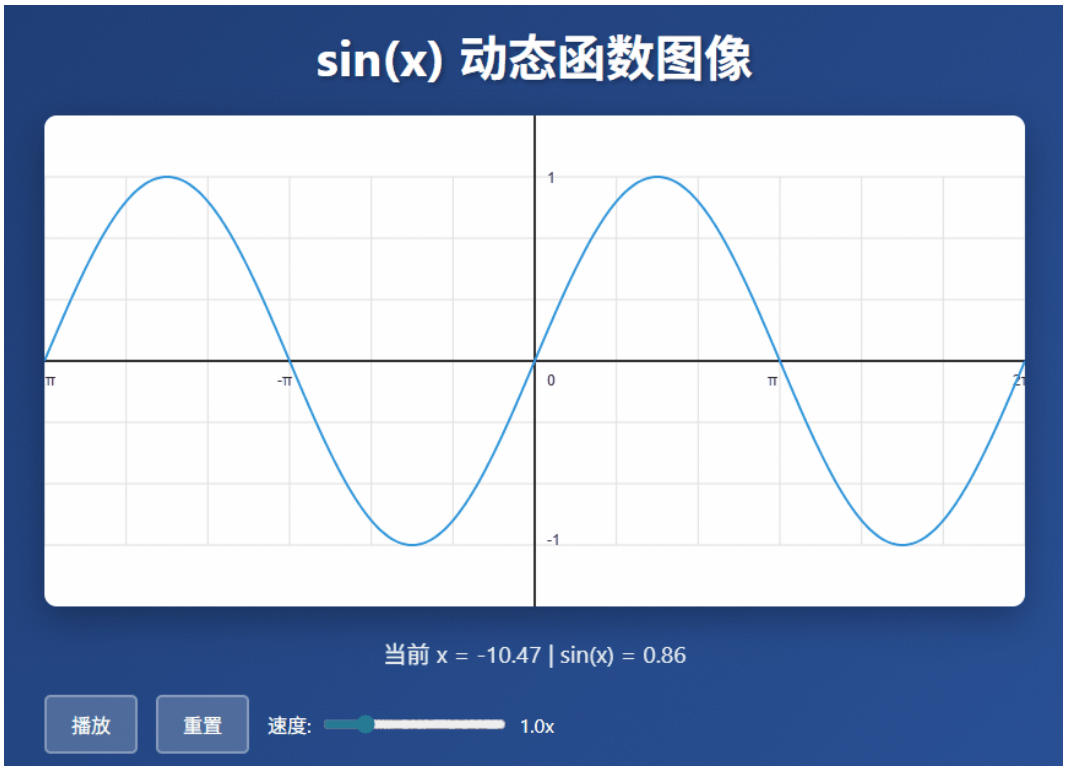
```
在网页中画一个sin(x)的函数动态图像
```

执行完成后，会在资源管理器的SIN-TEST目录下生成一个名为“***.html”的文件（如index.html）。

步骤2 在“index.html”文件上，单击右键选择“在浏览器中预览”，查看图像。

当前展示的效果图仅是示例，请以最终实际生成的效果为准。

图 1-3 无规则的 sin(x)函数图像



----结束

创建有规则约束的 sin(x)图像

上文介绍了无规则约束情况下sin(x)图像的绘制效果。接下来将为您介绍基于绘图规范约束下sin(x)图像的绘图。请参考以下步骤，依次完成项目创建、规则文件编写以及绘图操作。

步骤1 编写绘图规范规则文件。

1. 在IDE工具左上角，选择“文件(F) > 新建 > 新建项目”，创建一个名称为“sin-test-rule”的项目并在新窗口打开。
2. 单击华为云码道代码智能体IDE工具右上角的设置图标，进入全局设置页面。
3. 选择“技能与规则”，在“项目级”页签中单击规则后的。
4. 设置规则名称为“rule-pic”，适用范围保持默认选项“自动应用”，在内容中填写sin(x)的绘图规则，单击“确定”，即可完成规则的创建。

图 1-4 新建 rule-pic 规则


```
# 绘图规范
本文件包含了绘图过程需要遵守的规范

## 1.坐标系规范
- 横轴为x轴
- 纵轴为y轴
- 原点O(0,0)必须清晰标注
- 坐标轴箭头、刻度和单位长度要统一
- 刻度线是细实线
- 轴线是稍粗实线
- 坐标不能和坐标轴交叉

## 2. 角度单位规范
- 教学图像优先使用弧度制
- 需要在坐标轴上标注的刻度：0,$\frac{\pi}{2}$,$\pi$,$\frac{3\pi}{2}$,$2\pi$
```

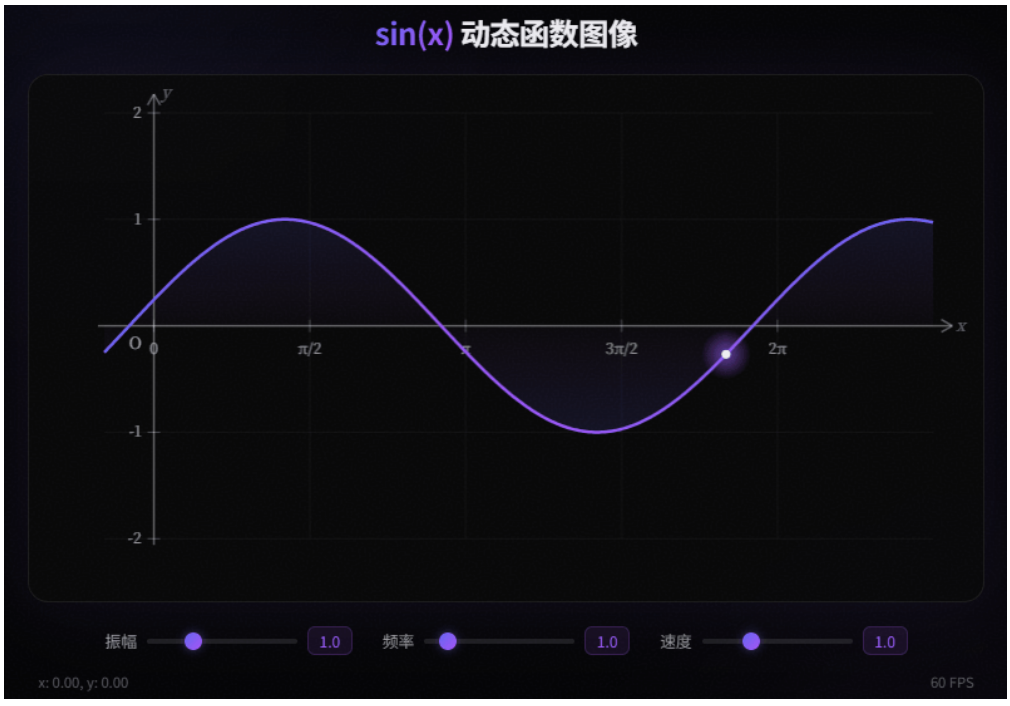
规则创建后，在当前项目（SIN-TEST-RULE）的“.codeartsdoer/rule”目录下，可查看到创建的规则文件rule-pic.mdc。

图 1-5 查看规则文件 rule-pic.mdc

步骤2 根据规则文件生成规则限制的sin(x)图像。

1. 在聊天界面的输入框中，输入如下指令，单击发送图标 。
   ```
   在网页中画一个sin(x)的函数动态图像
   ```
   执行完成后，会在资源管理器的SIN-TEST-RULE目录下生成一个名为“***.html”的文件（如index.html）。
2. 在“index.html”文件上，单击右键选择“在浏览器中预览”，查看图像。
   当前展示的效果图仅是示例，请以最终实际生成的效果为准。

图 1-6 规则下绘制的 sin(x)函数图像



----结束

总结

对比有无规则约束下绘制的sin(x)函数图像，可以明显看到：在引入规则后，函数图像的坐标规范和数学表达准确性都得到了显著提升，有效避免了坐标轴混乱等问题。充

分体现了Rules在统一标准、约束行为、保证输出质量和稳定性方面的重要作用，使结果更规范、可靠、可预期。

本次实践带您了解了**规则**。除规则外，华为云码道代码智能体还提供了**技能**和**MCP**，三者共同构建了智能体的多样化能力。各功能的差异对比如**表1-3**。

表 1-3 技能与其他功能的对比

对比维度	规则（Rule）	技能（Skill）	MCP Server
定义	用于约束智能体的行为方式。	用于描述如何完成特定任务。	提供外部工具的调用能力。
加载方式	在对话开始时加载到上下文中，持续参与推理。	按需加载，减少上下文占用。	不参与推理过程，按需调用外部接口。
使用场景	定义代码规范、输出格式、团队协作约定等。 例如：代码必须符合PEP8规范。	封装测试流程、开发任务、复杂业务逻辑等。 例如：执行UI自动化测试。	连接外部系统，执行具体操作。 例如：控制浏览器操作。
关键区别	控制智能体“应该怎么做”，是行为边界。	解决“如何完成任务”，是流程指导。	提供“调用工具的能力”，是执行能力。

2 使用技能创建器生成 PDF 按章节拆分 Skill

日常办公中，手册、报告、论文、标书等长PDF文档处理较为常见，人工拆分章节、逐一对文件重命名不仅耗时，也影响文档处理效率。为了解决这一痛点，您可以通过华为云码道CodeArts代码智能体，使用技能市场中提供的**技能创建器（Skill Creator）**，快速生成专属的**PDF拆分Skill**。

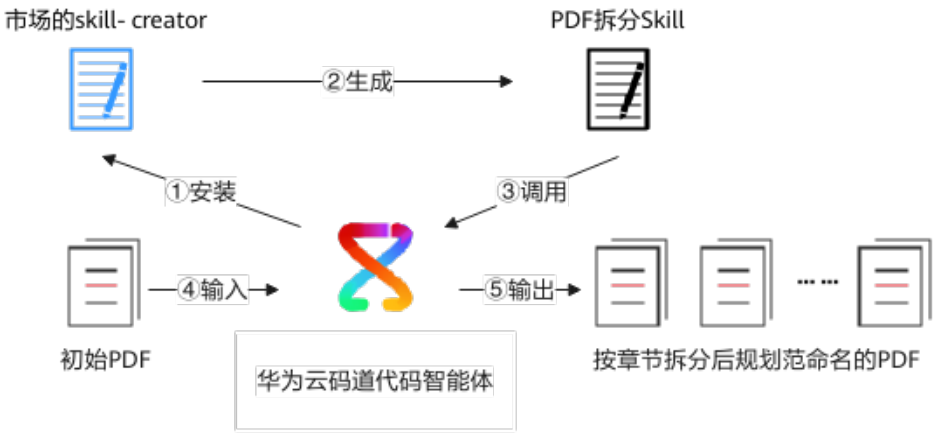
您可以在市场中找到技能创建器，完成安装后进行使用。当您需要一个技能时，可以直接说出“帮忙创建一个Skill”并描述Skill所需要的详细能力，华为云码道代码智能体会为您生成对应的Skill技能包。

PDF拆分Skill可自动识别文档章节标题，按章节智能拆分为多个独立PDF文件，并自动使用章节名称完成命名，全程无需手动分页与重命名，大幅简化长文档处理流程。

- **长文档拆解**：把几百页的手册/报告，拆成单章文件，方便单独发、单独看。
- **规范归档**：拆分完直接按章节名命名，不用手动改名，一键入库。
- **定向分发**：只发某一章内容，不用发整个大文件，更安全、更省流量。

整个流程如图2-1所示。

图 2-1 实践流程



准备工作

在使用华为云码道代码智能体前，您需要先创建一个项目，用于集中存放工程中的各类文件。项目创建完成后，建议开启自动批准功能，AI将自动处理相关操作，无需人工干预。

步骤1 参考快速启动中操作，登录华为云码道代码智能体。

步骤2 创建一个项目，用于存放工程中的文件。

1. 在IDE工具顶部菜单栏中，单击“文件(F)”，选择“新建 > 新建项目”，进入新建项目页面。
2. 选择项目存放位置，输入项目名称（例如pdf-skill-demo），单击“确定”。

   项目名称必须以字母开头，可包含字母、数字、中划线或下划线，且总长度不能超过64个字符。

   项目创建完成后，在“资源管理器”中可以查看到已创建的PDF-SKILL-DEMO项目。

步骤3 （可选）选择智能体运行模式并授权自动化操作。

1. 单击华为云码道代码智能体IDE右上角的设置图标  ，进入全局设置页面。
2. 在“对话流 > 智能体 > 终端命令运行模式”中，选择智能体执行终端命令的运行策略，本实践使用默认运行策略，即沙箱运行。
3. 在“对话流 > 智能体 > 自动批准”中，单击  以开启所需的自动批准项目。

   授权自动化操作后，复杂工程级代码生成流程中的各项任务可自动执行。如果您未开启自动化操作授权，在使用华为云码道代码智能体进行编码时，部分操作将需要您手动确认。

> **⚠ 注意**
>
> 开启自动批准存在操作风险，请在开启前充分评估风险，并在安全可信环境中使用。

表 2-1 自动批准参数说明

参数	说明	示例
编辑	允许智能体调用edit、write、deleteFile等工具来编辑您计算机上的文件。	开启
使用浏览器	允许智能体在浏览器中访问网站。	开启
网页抓取工具	允许智能体访问并抓取指定网页的内容。	开启

4. 退出当前页面，完成授权操作。

----结束

在市场中安装技能创建器（skill-creator）

步骤1 单击华为云码道代码智能体IDE右上角的设置图标，进入全局设置页面。

步骤2 在左侧导航栏中单击“技能与规则”，进入技能与规则页面。

步骤3 单击“市场”，进入“市场”页签，查找“技能创建器”。

步骤4 在指定技能后，单击添加图标，安装位置选择“项目级”，单击“确定”。

安装成功后，该技能后的变为“已安装”。

----结束

生成 PDF 拆分 Skill

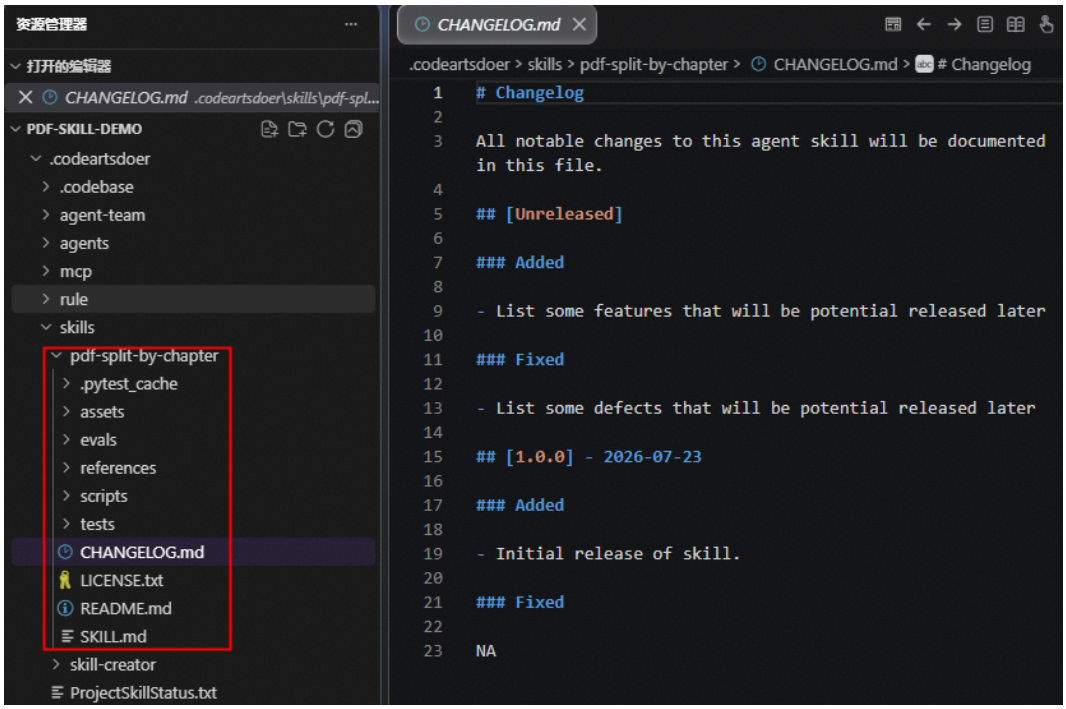
步骤1 在华为云码道代码智能体聊天界面的输入框中，输入如下提示词，单击发送图标。

```
调用skill-creator，生成专业PDF按章节拆分Skill，严格按以下规则执行：
【触发规则】
- 只要用户说“PDF按章节拆分”“按标题拆PDF并命名”“拆分PDF章节”，开始启用
【执行规则】
- 自动识别PDF里的章节标题和章节边界
- 按章节把原PDF拆成多个独立PDF
- 每个拆分后的文件，用对应的章节名称命名
【交付要求】不丢页、不错拆，做完返回所有拆分好的文件
```

步骤2 确认创建信息，选择继续创建，单击“提交”，等待技能创建成功。

技能创建成功后，在资源管理器的PDF-SKILL-DEMO目录下会生成“pdf-split-by-chapters”的技能包。

图 2-2 查看生成的技能包



步骤3 将待拆分的PDF文件放置于当前项目目录下，在聊天界面的输入框中，输入如下提示词。

调用pdf-chapter-splitter，完成华为云码道代码智能体用户指南PDF按章节拆分

单击发送图标，华为云码道代码智能体会帮助您完成PDF文件的拆分。

图 2-3 拆分完成

图 2-4 在对应目录下查看拆分结果

说明

当前展示的效果图仅是示例，请以最终实际生成的效果为准。

----结束

提示词优化

在上一节中，生成Skill的提示词给出基础核心功能，只能完成简单拆分，缺少统一的章节识别规则、命名规则与异常情况处理。当PDF文档内容较为复杂或需要批量处理时，常会出现识别准确率下降、处理时间延长以及命名异常等问题。

您可以进一步对提示词进行优化，比如明确章节识别样式、增加前置文件校验规则，拆分要求与拆分后文件名规范，补充异常提示，让Skill的执行效果更加贴合业务，并且能够长期稳定可靠的运行。

优化后的提示词

```
调用skill-creator，生成专业PDF按章节拆分Skill，严格按以下规则执行：
【触发规则】
- 正向触发：用户提出PDF按章节拆分、按标题拆分并自动命名相关需求时，启用本技能。
- 排他约束：本技能不执行PDF合并、加密解密、文本提取、格式转换等其他操作，杜绝功能冲突。
【标准化处理流程】
1. PDF文件预处理：检测PDF文件状态，识别加密文件，异常则给出明确提示并主动弹窗提示用户。
2. 章节识别解析：识别层级标题，支持“第X章”“1.1”“第一章”“1”等常见格式，自动剔除封面、目录页
3. 精准文档拆分：按照章节边界完成分页拆分，保证章节内容完整连贯，无缺页、漏页、内容截断、页面错乱问题。
4. 拆分后文件命名：以对应章节名称作为拆分后文件名称，自动清除文件名内/ \ : * ? " < > |等系统禁用特殊字符，兼容Windows、Mac系统
5. 结果回执：拆分完成后同步输出拆分明细单，清晰标注各文件对应章节名称与起止页码，有序返回全部拆分完成文件。
【质量要求】
保障拆分结果准确无误，文件命名无乱码，全面兼容市面主流常规PDF格式，满足日常办公及批量生产使用需求。
```

提示词优化技巧总结

Skill执行偏差、文件命名异常等问题，多源于提示词语义模糊、规则缺失等。在撰写提示词时，请遵循以下实用技巧以确保规范。这能全面优化技能执行效果，减少运行偏差，显著提升PDF章节拆分功能的实用性与稳定性。

表 2-2 提示词优化技巧

技巧点	好的样例	坏的样例
明确章节识别规则	识别层级标题，支持“第X章”“1.1”“第一章”“1”等常见格式，自动剔除封面、目录页	自动识别PDF里的章节
明确排他规则，避免大模型误触发其他操作	不处理合并、加密、提取文本等其他PDF操作，避免冲突	未提供该内容
考虑异常场景	• 文件名去掉特殊符号（/ \ : * ? " < > |），避免报错 • 识别不到章节标题时，提示“未检测到章节，并标注对应页码”	未提供该内容

3 检查点会话回退最佳实践

软件系统完成基础核心版本开发后，通常会在迭代中持续新增各类扩展功能，以不断丰富产品能力、满足多样化业务场景。但部分非核心扩展功能在实际验证后，会出现与业务需求不匹配、增加系统性能损耗、导致交互冗余复杂等问题，影响整体稳定性与使用体验。在不破坏核心业务、不残留冗余代码与配置的前提下，通过会话级检查点机制实现系统状态自动存档与精准回滚，可一键还原至初始基础版本，高效解决无效迭代带来的问题。

本实践通过**轻量化咖啡点单系统**，演示检查点（Checkpoint）在会话编辑过程中的状态存档、精准回滚、界面还原能力。

准备工作

在使用华为云码道代码智能体前，您需要先创建一个项目，用于集中存放工程中的各类文件。

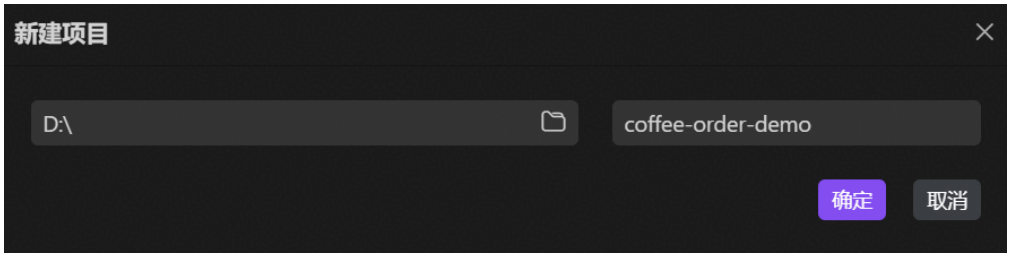
步骤1 参考快速启动中操作，登录华为云码道代码智能体。

步骤2 创建一个项目，用于存放工程中的文件。

1. 在IDE工具顶部菜单栏中，单击“文件(F)”，选择“新建 > 新建项目”，进入新建项目页面。
2. 选择项目存放位置，输入项目名称（如coffee-order-demo），单击“确定”。

   项目名称必须以字母开头，可包含字母、数字、中划线或下划线，且总长度不能超过64个字符。

   图 3-1 新建项目

   项目创建完成后，在“资源管理器”中可以查看到已创建的“COFFEE-ORDER-DEMO”项目。

步骤3 （可选）选择智能体运行模式并授权自动化操作。

1. 单击华为云码道代码智能体IDE右上角的设置图标  ，进入全局设置页面。
2. 在“对话流 > 智能体 > 终端命令运行模式”中，选择智能体执行终端命令的运行策略，本实践使用默认运行策略，即沙箱运行。
3. 在“对话流 > 智能体 > 自动批准”中，单击  以开启所需的自动批准项目。

   授权自动化操作后，复杂工程级代码生成流程中的各项任务可自动执行。如果您未开启自动化操作授权，在使用华为云码道代码智能体进行编码时，部分操作将需要您手动确认。

   ⚠ 注意

   开启自动批准存在操作风险，请在开启前充分评估风险，并在安全可信环境中使用。

   表 3-1 自动批准参数说明

参数	说明	示例
编辑	允许智能体调用edit、write、deleteFile等工具来编辑您计算机上的文件。	开启
使用浏览器	允许智能体在浏览器中访问网站。	开启
网页抓取工具	允许智能体访问并抓取指定网页的内容。	开启

4. 退出当前页面，完成授权操作。

----结束

生成基础咖啡点单系统

点单系统首期采用轻量化建设思路，聚焦核心点餐业务流程，暂不扩展非必需的营销及增值功能，确保核心链路简洁、可用、可靠。

步骤1 在聊天界面的输入框中，输入如下提示词，创建咖啡点单系统。

```
HTML+CSS+JavaScript创建一个基础的咖啡点单系统。
功能要求：
- 菜单展示：静态展示一个包含至少三种咖啡（如：美式、拿铁、卡布奇诺）的列表，每种咖啡标明名称、价格以及通用图标
- 加入购物车：每个咖啡旁边有一个“加入购物车”按钮。
- 订单预览：页面上有一个区域实时显示当前购物车中的商品、数量和计算出的总价。
- 样式：界面要求简洁、干净、温暖色系
```

执行完成后，在资源管理器的“COFFEE-ORDER-DEMO”目录下会生成一个名为“index.html”的文件。

步骤2 在“index.html”文件上，单击鼠标右键选择“在浏览器中预览”，查看网站。

当前展示的效果图仅是示例，请以最终实际生成的效果为准。

图 3-2 咖啡点单系统

----结束

增加个性化推荐功能

团队提出了一个想法：增加一个“每日特调”推荐功能，用于阶段性营销活动，吸引客户下单。这类功能非系统基础核心功能，活动周期结束后将下线并恢复原有界面逻辑。

步骤1 在聊天界面的输入框中，输入如下提示词，优化咖啡点单系统。

```
基于之前的咖啡点单系统，现在为其增加一个“每日特调”推荐功能。
功能要求:
- 在菜单顶部增加一个“今日特调”区域。
- 这个区域会动态展示一款咖啡作为特价推荐，并附带一个简单的推荐理由。
- 例如：“今日特调：冰摇柠檬茶（8折优惠！）”，并附上一个独立的“购买特调”按钮。
```

执行完成后，会同步更新资源管理器中“COFFEE-ORDER-DEMO”目录下的“index.html”文件。

步骤2 在“index.html”文件上，单击鼠标右键选择“在浏览器中预览”，查看网站。

当前展示的效果图仅是示例，请以最终实际生成的效果为准。

图 3-3 增加每日特调推荐

----结束

回退到基础版本

经团队讨论评估，“每日特调”推荐功能模块视觉干扰强、功能重复、日常维护成本高，对现阶段网站弊大于利。因此暂不实现该功能，回归简洁基础的核心点单版本。

步骤1 在聊天界面中，回退对话。

将鼠标悬浮在待回退版本对话上，单击  ，弹出恢复确认对话框，显示待恢复的文件名。

图 3-4 单击回退图标

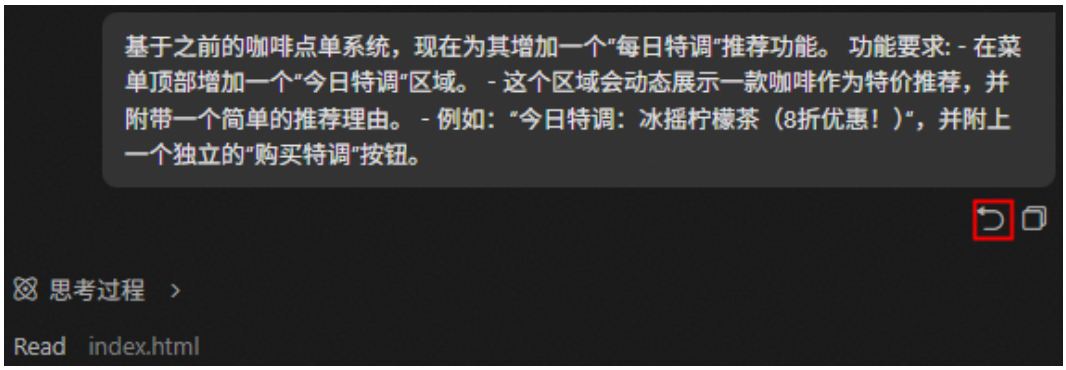


图 3-5 恢复确认对话框

步骤2 单击“确定”，回退对话。

文件会回退至历史代码状态，且当前对话中该时间点及之后的对话消息消失。咖啡点单系统同步回退到基础版本。

----结束

总结

通过上面的实践，可以清晰地看到检查点的核心价值：

- **版本安全兜底**：为华为云码道代码智能体生成的基础系统提供可还原的基准版本。
- **无效迭代快速撤销**：追加功能不符合预期时，无需手动删除代码，一键回退。
- **降低开发试错成本**：允许自由尝试扩展功能，不破坏初始稳定系统。
- **提升运维效率**：无需重新生成项目，快速恢复到最符合需求的版本。

4 井字棋游戏本地开发及静态产物验证实践

前端开发中，会面临TypeScript类型不规范、界面样式调试耗时、工程配置易出错等问题。本实践以Next.js+React+TypeScript井字棋单页应用本地开发场景为例，帮助开发者依托华为云码道代码智能体完成项目初始化、代码生成、本地功能全量验证，提升前端研发效率与代码质量。

准备工作

在使用华为云码道代码智能体前，您需要先创建一个项目，用于集中存放工程中的各类文件。项目创建完成后，建议开启自动批准功能，AI将自动处理相关操作，无需人工干预。

步骤1 参考**快速启动**中操作，登录华为云码道代码智能体。

步骤2 创建一个项目，用于存放工程中的文件。

1. 在IDE工具顶部菜单栏中，单击“文件(F)”，选择“新建 > 新建项目”，进入新建项目页面。
2. 选择项目存放位置，输入项目名称（例如Tic-Tac-Toe_Demo），单击“确定”。

   项目名称必须以字母开头，可包含字母、数字、中划线或下划线，且总长度不能超过64个字符。

   项目创建完成后，在“资源管理器”中可以查看到已创建的TIC-TAC-TOE_DEMO项目。

步骤3 （可选）选择智能体运行模式并授权自动化操作。

1. 单击华为云码道代码智能体IDE右上角的设置图标  ，进入全局设置页面。
2. 在“对话流 > 智能体 > 终端命令运行模式”中，选择智能体执行终端命令的运行策略，本实践依赖较多，推荐选择“手动模式”。
3. 在“对话流 > 智能体 > 自动批准”中，单击  以开启所需的自动批准项目。

   授权自动化操作后，复杂工程级代码生成流程中的各项任务可自动执行。如果您未开启自动化操作授权，在使用华为云码道代码智能体进行编码时，部分操作将需要您手动确认。

> **⚠ 注意**
>
> 开启自动批准存在操作风险，请在开启前充分评估风险，并在安全可信环境中使用。

表 4-1 自动批准参数说明

参数	说明	示例
编辑	允许智能体调用edit、write、deleteFile等工具来编辑您计算机上的文件。	开启
使用浏览器	允许智能体在浏览器中访问网站。	开启
网页抓取工具	允许智能体访问并抓取指定网页的内容。	开启

4. 退出当前页面，完成授权操作。

----结束

生成井字棋核心代码与界面

步骤1 在华为云码道代码智能体聊天界面的输入框中，输入如下提示词，并单击发送图标 。

```
基于React+TypeScript+Next.js 14 App Router编写井字棋游戏页面，要求：
1. 游戏棋盘整体在页面水平、垂直居中；
2. 页面添加标题和游戏玩法说明；
3. X、O使用不同颜色区分；
4. 代码结构规范，可直接运行。
```

步骤2 任务执行的过程中，所有终端操作及高风险命令的运行，需要手动确认运行，等待项目创建成功。

当前展示的效果图仅是示例，请以最终实际生成的效果为准。

图 4-1 项目创建过程

步骤3 单击“全部接受”，接受当前文件修改。

----结束

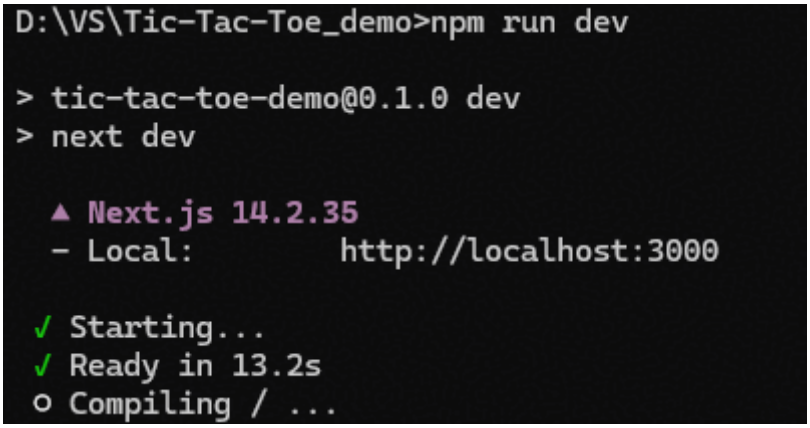
开发环境调试

代码生成完成后，可以先在开发环境完成全量功能验证，确认功能正常后再进行打包配置，提前规避逻辑问题。

步骤1 启动Next.js本地开发调试服务。

1. 本地设备上打开命令提示符窗口，以Windows 11操作系统为例。同时按下“Win+R”，在打开的“运行”对话框中输入cmd，按“Enter”。
2. 切换到当前项目所在的目录，以如下目录为例。
   ```
   D:
   cd \VS\Tic-Tac-Toe_Demo
   ```
3. 执行启动命令，启动Next.js本地开发调试服务，等待编译就绪。
   ```
   npm run dev
   ```


图 4-2 服务启动完成



步骤2 浏览器打开网址http://localhost:3000，进行逐项功能验证：

- 界面：标题、玩法说明、棋盘布局展示正常，X/O颜色区分清晰。
- 交互：点击格子可正常落子，已占用格子无法重复点击。
- 逻辑：横、竖、斜线连线可正常判定获胜，棋盘填满后正确判定平局。

----结束

静态打包与验证

开发环境验证通过后，再配置打包规则、执行编译，并最终验证静态资源。

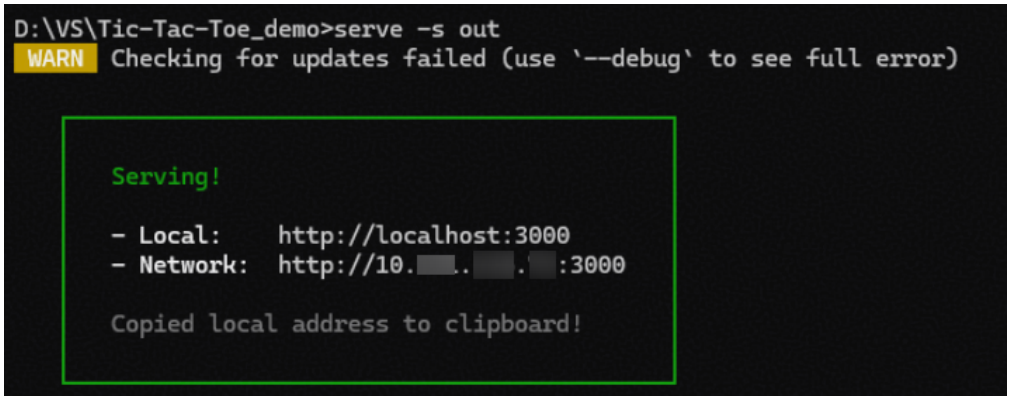
步骤1 在华为云码道代码智能体聊天界面的输入框中，输入如下打包规则，并单击发送图标  。

```
修改next.config.mjs，开启静态导出，打包输出目录设为out，确保代码无报错、可正常打包。
```

步骤2 打包构建项目并预览生产静态资源。

1. 在本地设备上打开命令提示符窗口，以Windows 11操作系统为例。同时按下“Win+R”，在打开的“运行”对话框中输入cmd，按“Enter”。
2. 切换到当前项目所在的目录，以如下目录为例。
   ```
   D:
   cd \VS\Tic-Tac-Toe_Demo
   ```
3. 执行启动命令，进行生产环境打包构建，等待全量编译优化完成。
   ```
   npm run build
   ```
4. 执行如下命令，全局安装静态文件服务（仅首次安装执行）。
   ```
   npm install -g serve
   ```
5. 执行如下命令，启动静态服务。
   ```
   serve -s out
   ```


图 4-3 本地终端执行启动静态服务命令



步骤3 根据输出地址访问页面，再次完整校验界面、交互、胜负判断等功能，确保静态产物与开发环境验证结果表现一致。

当前展示的效果图仅是示例，请以最终实际生成的效果为准。

----结束

常见问题及解决方案

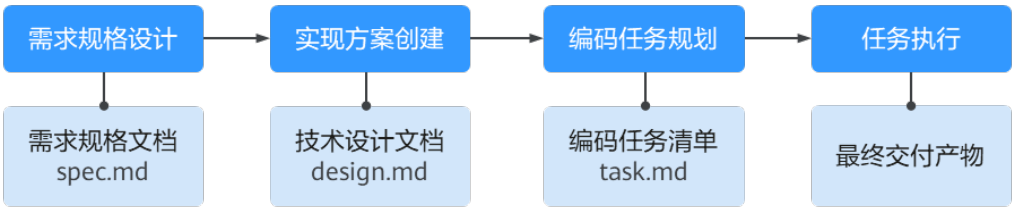
问题现象	根因分析	解决方案
执行npm run dev后，页面依然无法访问	端口占用	指定新端口，例如被占用的是3000，指定成3001。执行npm run dev -- -p 3001（新端口）。
开发环境功能正常，打包后out目录缺失	未正确配置静态导出	重新执行打包配置提示词，修正next.config.mjs文件。 修改next.config.mjs，开启静态导出，打包输出目录设为out，确保代码无报错、可正常打包。

5 SDD 开发模式标准工作流与斜杠命令实践

华为云码道代码智能体的规范驱动开发模式（SDD）是一种以规格说明为核心的软件开发方法。开发者提供需求描述后，智能体依次生成规格说明、设计方案与任务规划，让开发者和智能体在编码前就任务目标及实施方案达成共识，最终产出高质量代码。

华为云码道代码智能体的SDD开发模式采用如图5-1所示的四阶段递进式流程。

图 5-1 SDD 开发模式流程及产物



在此基础上，华为云码道代码智能体提供了两种使用方式：

- 标准SDD工作流：在智能体对话切换到规范开发（Spec-Driven）模式后，由智能体主导推进，按如图5-1所示的顺序执行，每阶段产物人工确认过后进入下一环节。
- 斜杠命令模式：通过如表5-1所示的斜杠命令，由开发者自主调度，可灵活跳过或重复任意阶段。

表 5-1 SDD 斜杠命令说明

命令	功能	输入	输出
/sdd-new	创建需求规格	需求描述	spec.md
/sdd-design	创建技术设计	spec.md所在文件夹名称	design.md
/sdd-tasks	创建编码任务	spec.md和design.md所在文件夹名称	tasks.md

命令	功能	输入	输出
/sdd-apply	执行编码实现	tasks.md所在文件夹名称	代码文件

本章节将以“为Web应用开发用户注册与登录功能”为例，详细对比两种方式的差异与适用场景。

准备工作

在使用华为云码道代码智能体前，您需要先创建一个项目，用于集中存放工程中的各类文件。项目创建完成后，建议开启自动批准功能，AI将自动处理相关操作，无需人工干预。

步骤1 参考**快速启动**中操作，登录华为云码道代码智能体。

步骤2 创建一个项目，用于存放工程中的文件。

1. 在IDE工具顶部菜单栏中，单击“文件(F)”，选择“新建 > 新建项目”，进入新建项目页面。
2. 选择项目存放位置，输入项目名称（例如SDD_Demo），单击“确定”。

   项目名称必须以字母开头，可包含字母、数字、中划线或下划线，且总长度不能超过64个字符。

   项目创建完成后，在“资源管理器”中可以查看到已创建的SDD_DEMO项目。

步骤3 （可选）选择智能体运行模式并授权自动化操作。

1. 单击华为云码道代码智能体IDE右上角的设置图标，进入全局设置页面。
2. 在“对话流 > 智能体 > 终端命令运行模式”中，选择智能体执行终端命令的运行策略，本实践使用默认运行策略，即沙箱运行。
3. 在“对话流 > 智能体 > 自动批准”中，单击以开启所需的自动批准项目。

   授权自动化操作后，复杂工程级代码生成流程中的各项任务可自动执行。如果您未开启自动化操作授权，在使用华为云码道代码智能体进行编码时，部分操作将需要您手动确认。

⚠ 注意

开启自动批准存在操作风险，请在开启前充分评估风险，并在安全可信环境中使用。

表 5-2 自动批准参数说明

参数	说明	示例
编辑	允许智能体调用edit、write、deleteFile等工具来编辑您计算机上的文件。	开启
使用浏览器	允许智能体在浏览器中访问网站。	开启

参数	说明	示例
网页抓取工具	允许智能体访问并抓取指定网页的内容。	开启

4. 退出当前页面，完成授权操作。

----结束

标准 SDD 工作流

标准SDD工作流是华为云码道代码智能体内置的规范驱动模式。用户在聊天界面选择“规范开发 Spec-Driven”后，输入需求描述，智能体会按照预设工作流逐步执行：首先生成需求规格文档spec.md，确认无误后自动进入方案创建，生成design.md，再生成任务清单tasks.md，最后执行编码实现。整个流程由AI自主驱动，用户主要在关键节点进行确认（接受或拒绝生成的文档）。

步骤1 在聊天界面的输入框下方选择“内置 > 智能体”，切换到**智能体**模式。右侧显示当前选用的模型，您可在下拉框中切换不同大语言模型。

说明

如果没有正常显示华为云码道代码智能体的聊天窗口，请在顶部菜单栏的右上方，单击**展开AI侧栏**图标  ，即可打开华为云码道代码智能体。

步骤2 在聊天界面的模式选择中，单击“规范开发”，切换智能体到规范开发模式。

步骤3 在华为云码道代码智能体聊天界面的输入框中，输入如下提示词，并单击发送图标  。

```
请开发一个用户登录接口，支持用户通过用户名和密码进行身份认证，认证成功后返回JWT令牌，登录失败时记录失败次数，连续失败5次后锁定账号15分钟。
```

步骤4 查看生成的需求规格文件spec.md。提出相应的修改意见，华为云码道代码智能体会修改spec.md文件。

```
修改以下问题：
登录成功后除了返回token，还需要返回用户基本信息（昵称、角色）
锁定时间15分钟太长了，改为5分钟
补充接口的限流机制，每分钟同一个IP最多请求10次
响应码不要用200，统一使用业务码20000表示成功
```

步骤5 再次查看修改后的spec.md，确认需求理解准确后，单击“开始实现方案创建”。

步骤6 查看生成的design.md，确认技术方案合理后，单击“开始编码任务规划”。

步骤7 查看生成的tasks.md，确认任务拆解完整后，单击“开始任务执行”。

步骤8 等待任务执行完成后，获得完整的代码实现，同时沉淀了spec.md、design.md和tasks.md三份过程文档。

----结束

斜杠命令模式

华为云码道代码智能体通过底层Skills封装，统一定义了四个标准化斜杠命令。开发者可以根据已有过程交付件情况，按需调用命令，无需重跑全流程。例如，执行“/sdd-

apply”编码中途发现加密库过时，手动修改“design.md”，再依次运行“/sdd-tasks”和“/sdd-apply”。

步骤1 在聊天界面的输入框下方选择“内置 > 智能体”，切换到**智能体**模式。右侧显示当前选用的模型，您可在下拉框中切换不同大语言模型。

说明

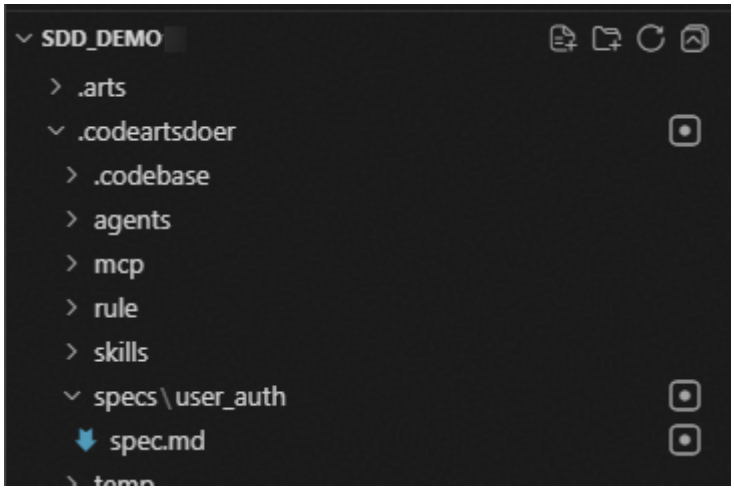
如果没有正常显示华为云码道代码智能体的聊天窗口，请在顶部菜单栏的右上方，单击**展开AI侧栏**图标，即可打开华为云码道代码智能体。

步骤2 在聊天界面的模式选择中，单击“氛围编程”，切换智能体到氛围编程模式。

步骤3 按需调用命令完成SDD开发。

- **/sdd-new：创建需求规格。**
  a. 在对话框输入“/”，在弹出的菜单中选择“/sdd-new”，输出如下需求描述，生成统一格式的spec.md。
  ```
  请帮我开发一个用户注册登录模块，包含手机号/邮箱注册、JWT登录认证、验证码发送、密码重置功能
  ```
  b. 进入**{项目根路径}**/.codeartsdoer/specs/**产物文件夹**目录下，获取生成的spec.md。

  图 5-2 spec.md 所在目录

  c. 您可基于当前需求规格文档进行评审，评审意见可以通过智能体修改或手动修改spec.md文件，直到需求规格文档符合业务预期。
- **/sdd-design：创建技术设计。**
  如果当前您已有spec.md文件，您可直接调用“/sdd-design”，然后输入产物文件夹名称，例如user_auth，生成包含架构、接口描述的design.md。
- **/sdd-tasks：创建编码任务。**
  在对话框输入“/”，在弹出的菜单中选择“/sdd-tasks”，然后输入产物文件夹名称（如user_auth），生成可执行的步骤清单tasks.md。
- **/sdd-apply：执行编码实现。**
  在对话框输入“/”，在弹出的菜单中选择“/sdd-apply”，然后输入产物文件夹名称（如user_auth）。华为云码道代码智能体按tasks.md逐项执行编码任务。

----结束

场景选型

在实际的开发中，标准SDD工作流和斜杠命令两种模式并非互斥，而是可以根据不同的项目阶段和团队角色灵活切换。新项目初期使用标准工作流快速跑通框架，建立完整的SDD文档体系。进入中期迭代切换到斜杠命令模式，利用四个命令进行增量开发和局部重构。

表 5-3 核心差异对比

对比维度	标准SDD工作流	斜杠命令模式
执行方式	对话框输入需求，智能体主导流程串联，用户在关键节点接受或拒绝生成的文档。	对话框调用命令，用户自主推进流程。
灵活性	流程固定，不可跳过。	可任意组合、重复执行单个命令。
适用场景	● 新项目从0到1启动 团队从需求分析、系统设计到SDD文档体系，构建完整的SDD开发流程。 ● 标准化要求高的正式项目 对一致性、可验证性有高要求的场景，所有开发活动均围绕既定规范展开，确保最终在功能、性能等方面与规范高度一致。 ● SDD新手入门 初次接触SDD的开发者，通过华为云码道代码智能体的分步引导快速建立完整链路。 ● 需求相对稳定的项目 需求已经过充分讨论，开发过程中不会频繁调整方向和方案。	● 需求频繁迭代或需求不明确，需反复确认的项目 若产品经理突然修改登录逻辑，只需手动修改spec.md，然后组合执行“/sdd-design”和“/sdd-tasks”，无需重新生成spec.md。 ● 多人异步协作 架构师上午生成design.md并提交Git，开发人员下午拉取代码后，只需在聊天框执行“/sdd-tasks”和“/sdd-apply”即可。 ● 已有部分文档 如果已手动写好spec.md，想直接开始设计，直接敲“/sdd-design”完成后续流程。
适用角色	产品经理快速验证想法。 独立开发者做小型项目。	架构师需把关技术方案。 团队协作开发企业级应用。

6 基于 CodeArts Check MCP 实现代码检查与智能缺陷修复

方案概述

华为云码道代码智能体支持基于MCP对接华为云码道（CodeArts）完成协同研发，可联动华为云码道（CodeArts）创建项目、新建代码仓库、提交推送代码，并在华为云码道（CodeArts）侧配置执行代码检查规则；码道代码智能体依托MCP接收检查结果，统一完成报告解析，随后可在码道代码智能体进行代码修复。

本实践以码道代码智能体为核心，标准化其与华为云码道（CodeArts）的MCP交互流程，打通代码规则检测、报告分析、缺陷修复全链路，实现研发流程统一可控。

6.1 前提条件

在执行本实践前，请先开通华为云码道（CodeArts）套餐、为各个角色配置权限。

本实践以Windows操作系统为例。

约束与限制

表 6-1 实践案例限制

限制类别	具体限制
区域限制	此实践适用范围： 华为云码道（CodeArts）：以“亚太-新加坡”区域为例 华为云码道CodeArts代码智能体：以“中国香港”区域为例
工具限制	此实践仅支持华为云码道代码智能体IDE

限制类别	具体限制
操作系统限制	• Windows操作系统：**推荐使用Windows 11 (x64)**；如果使用Windows 10 (x64)，系统版本需为2019年及以上，建议升级至最新稳定版本。 • macOS操作系统：macOS 11及以上版本，兼容ARM64（Apple Silicon）架构。 • Linux操作系统：Linux（ARM64）架构，glibc版本 ≥ 2.31，glibcxx版本 ≥ 3.4.26。
第三方软件版本	Node.js≥18.0.0

购买华为云码道（CodeArts）套餐

步骤1 请**注册华为账号并开通华为云**。

步骤2 使用华为账号登录**购买CodeArts套餐页面**。

步骤3 完成华为云码道（CodeArts）套餐的参数配置，本实践以**体验版**套餐为例。

表 6-2 购买 CodeArts 套餐

参数	取值	说明
区域	“亚太-新加坡”	CodeArts套餐只在购买时所选择的区域生效，不能跨区域使用。
CodeArts套餐	选择“体验版”	CodeArts套餐提供的套餐版本。
人数	体验版固定为50人	CodeArts套餐中包含的人数，体验版固定为50人。
购买时长	1个月	CodeArts套餐的购买时长。体验版固定为1个月。
自动续费	勾选“自动续费”	该参数为可选参数，请根据需要勾选。如果勾选“自动续费”，资源到期后可以自动进行续费。 • 按月购买：每次续费1个月，次数不限。 • 按年购买：每次续费1年，次数不限。 关于自动续费的更多说明，请参考**自动续费规则说明**。

步骤4 单击“立即开通”，完成购买。

华为账号完成体验版套餐购买后，请继续**设置项目创建者**。

----结束

设置项目创建者

步骤1 华为账号进入华为云码道（CodeArts）首页。

1. 华为账号登录**CodeArts控制台**，单击，选择“亚太-新加坡”。
2. 右上角单击“前往工作台”。

步骤2 在导航栏中单击用户名，选择“租户设置”。

步骤3 左侧导航栏单击“通用设置 > 设置项目创建者”，进入设置项目创建者页面。

步骤4 勾选“设置所有成员都可以创建项目”，选择此选项后，账号中所有IAM用户将拥有创建项目的权限。

设置项目创建者后，请继续**新建CodeArts项目和代码检查规则**。

----结束

6.2 新建 CodeArts 项目和代码检查规则

创建项目和代码仓库

步骤1 具有项目创建权限的账号登录**CodeArts控制台**，选择“亚太-新加坡”。

步骤2 右上角单击“前往工作台”，进入华为云码道（CodeArts）项目首页。

步骤3 单击“新建 > 新建项目”，进入项目的新建页面。

步骤4 在CodeArts首页找到模板“Scrum”，单击“选用”。

步骤5 输入项目名称（项目名称不超过128个字符，本文中使用“req-demo”），单击“确定”。

新建项目成功，将自动跳转至需求管理（CodeArts Req）的“工作项”页面。**项目的创建者默认为项目经理**。

步骤6 项目经理单击左侧导航栏“代码 > 代码托管”，进入代码托管（CodeArts Repo）服务。

步骤7 单击“新建仓库”，进入新建仓库页面。

步骤8 选择“按模板新建”，单击“下一步”。

步骤9 选择模板“Java Web Demo”，单击“下一步”。

步骤10 填写“代码仓库名称”（代码仓库名称需以大小写字母、数字、下划线开头，可包含大小写字母、数字、中划线、下划线、英文句点，但不能以.git、.atom或.结尾，本文中使用“repo-demo”）。

步骤11 勾选可见范围为“公开 > 项目内成员只读”，单击“确定”。

新建代码仓库成功，页面将自动跳转至仓库的“代码”页面。

步骤12 选择“代码 > 分支”页签，单击“新建分支”，在弹框中配置以下信息，单击“确定”。

页面提示操作成功，将自动跳转至“文件”页面，页面中显示分支“develop-dev”。

表 6-3 新建分支参数填写

配置项	示例	说明
基于	master	选择已有的分支。
分支名称	develop-dev	分支的名称，最长200个字节。 • 不支持以“- ”、“.”、“refs/heads/”、“refs/remotes/”开头。 • 不支持空格、“[”、“\”、“<”、“~”、“^”、“:”、“?”、“*”、“!”、“( )”、“'”、“"”、“|”等特殊字符。 • 不支持以“.”、“/”、“.lock”结尾。
关联Req工作项	/	分支关联的工作项，选择已有的工作项。 默认不关联Req工作项

步骤13 单击上方导航栏“设置”，左侧导航栏选择“安全管理 > 权限管理”，打开开关“使用项目权限配置”，继承项目级权限。

----结束

配置代码检查规则

步骤1 项目经理登录CodeArts控制台，选择“亚太-新加坡”。

步骤2 右上角单击“前往工作台”，进入CodeArts项目首页。

步骤3 单击项目卡片“req-demo”，进入“req-demo”项目首页。

步骤4 左侧导航栏选择“代码 > 代码检查”，进入代码检查（CodeArts Check）首页。

步骤5 在“任务”卡片，单击“repo-demo-check”所在行的 ，单击“设置”，进入“repo-demo-check”的“基本信息”页面。

步骤6 在“基本信息”页面设置“默认分支”为“develop-dev”，单击“保存”。

步骤7 在“检查范围”页面设置“版本级检查模式”为“最近一次提交”，单击“保存”。

步骤8 在“集成服务”页面勾选“代码提交时触发”，单击“保存”。

----结束

6.3 开发代码

分支创建与仓库克隆

开发人员需要基于“develop-dev”分支开发新需求的代码。

步骤1 开发人员进入项目“req-demo”，单击导航“代码 > 代码托管”，进入代码托管服务。

步骤2 单击代码仓库“repo-demo”，进入仓库。

步骤3 打开本地Git Bash，执行如下命令，克隆develop-dev分支到本地。

```
git clone -b 分支名 https:example.com
```

其中，*分支名*表示需要克隆的分支名称，*https:example.com*表示代码仓的HTTPS地址。

出现如下回显，表示成功克隆代码仓repo-demo的develop-dev分支到本地。

```
$ git clone -b develop-dev https://test.com/bb5f2eb7ded44e0bbc0ff4bd3d1d244c/repo-demo.git
Cloning into 'repo-demo'...
remote: Enumerating objects: 36, done.
remote: Counting objects: 100% (36/36), done.
remote: Compressing objects: 100% (30/30), done.
remote: Total 36 (delta 0), reused 36 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (36/36), 301.74 KiB | 1.23 MiB/s, done.
```

----结束

代码开发

步骤1 在华为云码道代码智能体IDE工具左上角，选择“文件(F) > 打开项目”，打开已创建的代码仓库“repo-demo”。

步骤2 在华为云码道代码智能体的输入框中，输入如下内容，单击发送图标 。

```
二叉树中的路径被定义为一条节点序列，序列中每对相邻节点之间都存在一条边。同一节点在一条路径序列中，最多出现一次。该路径至少包含一个节点，并且不一定经过根节点，路径和是路径中各节点值的总和。现在请基于这个项目，给一个二叉树的根节点root，返回最大路径和。如果还有不确定的点，可以问我
```

步骤3 等待代码生成后，继续输入如下内容，编写验证脚本，使用java和javac编译运行，单击发送图标 。

```
请写一个验证脚本，使用java+javac编译运行
```

步骤4 等待代码生成后，上方导航栏单击“终端 > 新建终端”，执行如下命令：

```
powershell -ExecutionPolicy Bypass -File run-tests.ps1
```

其中，*run-tests.ps1*需替换为要运行的脚本文件，本文的脚本文件名为run-tests.ps1

出现下述回显，表示代码可正常运行。

```
==========================================
  二叉树最大路径和 - 测试验证
==========================================

[PASS] example1 [1,2,3] => 6
[PASS] example2 [-10,9,20,null,null,15,7] => 42
[PASS] allNegative [-1,-2,-3] => -1
[PASS] leftSkewedTree [5,4,null,3,null,2] => 14
[PASS] negativeBranchesIgnored [2,-1,-2] => 2
[PASS] complexTree => 1

==========================================
  Results: 9 passed, 0 failed
==========================================

所有测试通过!
```

说明

请以实际回显为准。

----结束

6.4 检查代码

提交代码

步骤1 进入项目代码“repo-demo”，鼠标右键“在终端中打开”。在终端执行如下命令，确保是修改的develop-dev分支。

```
git branch
```

出现下述回显，表当前分支正确。

```
* develop-dev
```

步骤2 校验通过后，继续在终端执行如下命令，查看本地代码改动。

```
git status
```

出现下述回显，红色代表未保存的修改文件，确认需要提交的代码。

```
On branch develop-dev
Your branch is up to date with 'origin/develop-dev'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        run-tests.ps1
        src/main/java/com/huawei/codearts/controller/TreeController.java
        src/main/java/com/huawei/codearts/tree/
        src/test/
        target/

nothing added to commit but untracked files present (use "git add" to track)
```

步骤3 继续在终端执行如下命令，将修改加入暂存区。

```
git add .
```

无任何回显，表示成功将修改文件都提交至暂存区。

步骤4 继续在终端执行如下命令本地提交代码。

```
git commit -m "feat: 新增二叉树最大路径代码和测试执行脚本"
```

出现下述回显，表示成功提交代码。

```
[develop-dev e1256e8] feat: 新增二叉树最大路径代码和测试执行脚本
 Committer: chenzhu (F) <test.com>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly:

    git config --global user.name "Your Name"
    git config --global user.email you@example.com

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

 10 files changed, 295 insertions(+)
 create mode 100644 run-tests.ps1
 create mode 100644 src/main/java/com/huawei/codearts/controller/TreeController.java
 create mode 100644 src/main/java/com/huawei/codearts/tree/MaxPathSum.java
 create mode 100644 src/main/java/com/huawei/codearts/tree/MaxPathSumRunner.java
 create mode 100644 src/main/java/com/huawei/codearts/tree/TreeNode.java
 create mode 100644 src/test/java/com/huawei/codearts/tree/MaxPathSumTest.java
 create mode 100644 target/test-classes/com/huawei/codearts/tree/MaxPathSum.class
 create mode 100644 target/test-classes/com/huawei/codearts/tree/MaxPathSumManualTest.class
 create mode 100644 target/test-classes/com/huawei/codearts/tree/MaxPathSumRunner.class
 create mode 100644 target/test-classes/com/huawei/codearts/tree/TreeNode.class
```

步骤5 执行如下命令，推送到远程云端develop-dev分支。

```
git push origin develop-dev
```

出现下述回显，表示本地代码已完整提交到云端仓库“repo-demo”。

```
Enumerating objects: 40, done.
Counting objects: 100% (40/40), done.
Delta compression using up to 16 threads
Compressing objects: 100% (18/18), done.
Writing objects: 100% (32/32), 7.24 KiB | 494.00 KiB/s, done.
Total 32 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Start Git Hooks Checking                                [PASSED]
remote:
remote: To create a merge request for develop-dev, visit:
remote:   https://test.com/codehub/2112135977/newmerge?branchName=develop-
dev&projectNamespace=bb5f2eb7ded44e0bbc0ff4bd3d1d244c/repo-demo
remote:
To https://test.com/bb5f2eb7ded44e0bbc0ff4bd3d1d244c/repo-demo.git
  2f31d8d..e1256e8  develop-dev -> develop-dev
```

提交代码成功后，CodeArts Check将自动触发检查任务。

----结束

查看代码检查结果

步骤1 开发人员登录**CodeArts控制台**，选择“亚太-新加坡”。

步骤2 右上角单击“前往工作台”，进入CodeArts首页。

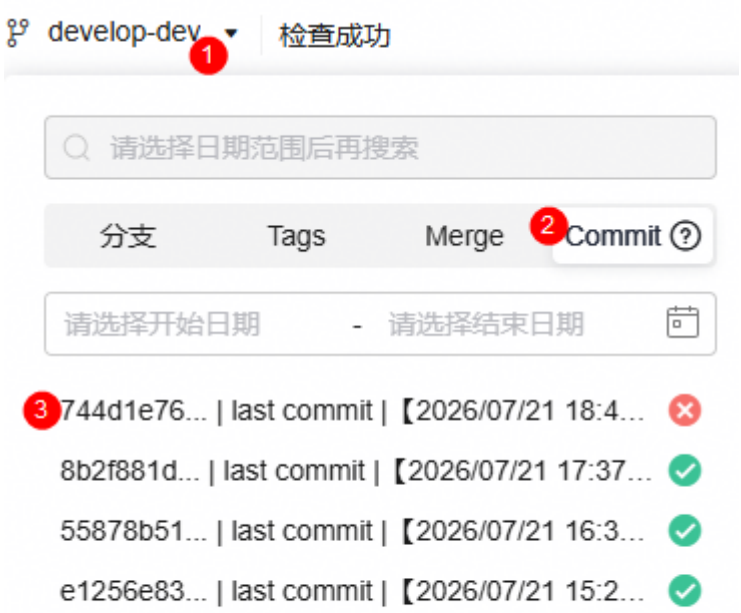
步骤3 单击项目卡片“req-demo”，进入“req-demo”项目首页。

步骤4 左侧导航栏选择“代码 > 代码检查”，进入代码检查（CodeArts Check）首页。

步骤5 在“任务”卡片，单击“repo-demo-check”，进入“repo-demo-check”的“概览”页面。

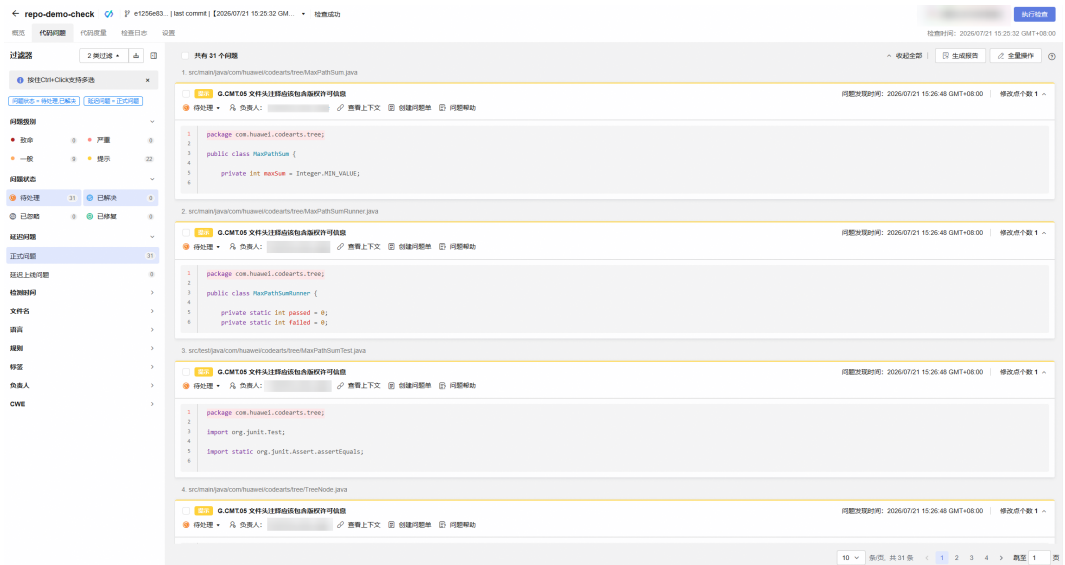
步骤6 单击分支名称，单击“commit”，选择需要查看的commit检查记录。

图 6-1 选择 commit 记录



步骤7 单击上方导航栏的“代码问题”，进入“repo-demo-check”的代码问题页面。如下图所示，最近一次commit的代码共扫描出72个代码问题。

图 6-2 代码问题



----结束

6.5 配置 CodeArts Check MCP

步骤1 在IDE工具左上角，选择“文件(F) > 打开项目”，打开“repo-demo”项目。

步骤2 单击华为云码道代码智能体IDE工具右上角的设置图标  ，进入全局设置页面。

步骤3 在设置页面的左侧导航栏中，选择“MCP工具”，进入MCP服务页面。

步骤4 下载“CodeArts Check MCP”到本地并解压。请参考**官网下载页**。

步骤5 单击“配置MCP”，复制如下配置内容并粘贴到“mcp_settings.json”文件。各参数的说明和取值示例请参考表**表6-4**。

配置完成后按快捷键（Windows/Linux：“Ctrl+S”；macOS:“Command(⌘)+S”）保存，等待工具启动完成。

```
{
  "mcpServers": {
    "CodeArts Check MCP": {
      "disabled": false,
      "timeout": 600000,
      "command": [
        "D:\\Program Files\\nodejs\\node.exe",
        "D:\\codeartscheck-ts-mcp-server-1.0.0\\package\\dist\\index.js"
      ],
      "environment": {
        "region": "ap-southeast-3",
        "ak": "your_ak",
        "sk": "your_sk"
      },
      "type": "local"
    }
  }
}
```

表 6-4 配置参数说明

字段	取值样例	说明
mcpServers	见下方配置	顶层节点，包含所有注册的MCP服务器
CodeArts Check MCP	/	必填，MCP服务器的名称标识，长度不能超过64个字符，用于区分不同的MCP服务器。 **说明** 服务器名称超过64个字符时，可能返回以下报错并导致当前请求不可用。您可以参考**服务器核心功能**中的“Tools”参数的定义。 • type: session.error info • length over limit, maximum is 64
disabled	false	必填，控制MCP服务器禁用状态，默认值为“false”。取值范围如下： • false，启用 • true，禁用
timeout	600000	超时时间（毫秒），超时后，客户端会报错。

字段	取值样例	说明
command	"command": ["D:\\Program Files\\nodejs\\node.exe", "D:\\codeartscheck-ts-mcp-server-1.0.0\\package\\dist\\index.js"],	必填，包含两部分： ● "D:\\Program Files\\nodejs\\node.exe"：运行MCP服务器的可执行文件路径 ● "D:\\codeartscheck-ts-mcp-server-1.0.0\\package\\dist\\index.js"：步骤4解压后，index.js文件的存放地址。

environment	{
	"region": "ap-southeast-3",
	"ak": "your_ak",
	"sk": "your_sk"
	}
		必填，传递给MCP服务器的环境变量集合，包括区域、AK和SK，所有环境变量值均需为字符串类型。
		其中，environment.region参数值需要**固定为“ap-southeast-3”**，environment.ak和environment.sk获取方法如下：
		1. 参考访问密钥新建访问密钥。
		2. 在新建访问密钥页面，勾选协议，单击“下一步”，填写密钥“描述”后，单击“确定”，完成授权。
		创建成功后，请单击“立即下载”，保存密钥信息，否则弹窗关闭后将无法再次获取该密钥信息。
		3. environment.ak和environment.sk的值，通过您下载保存的credentials.csv文件中获取。

type	local	MCP服务器的通信协议类型，取值“local”

步骤6 单击华为云码道代码智能体IDE工具右上角的设置图标  ，进入全局设置页面。

步骤7 在设置页面的左侧导航栏中，选择“MCP工具”，进入MCP服务页面。

步骤8 在“已安装”页签中，如下图所示，MCP服务器“CodeArts Check MCP”已配置且生效。

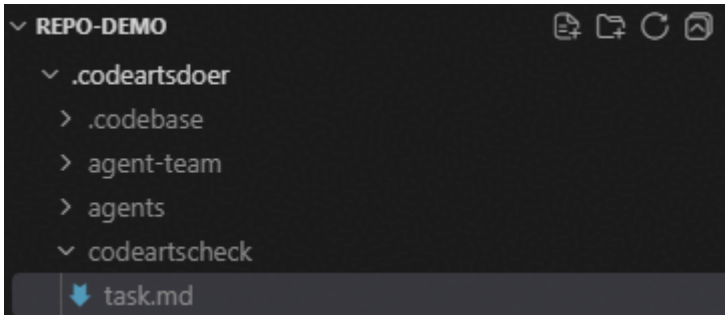
图 6-3 配置生效图

----结束

本地配置检查任务 ID

步骤1 如下图所示，在“.codeartsdoer”下创建文件夹“codeartscheck”后，在“codeartscheck”文件夹下创建“task.md”文件。

图 6-4 新建 task.md 文件



步骤2 登录CodeArts控制台，选择“亚太-新加坡”。

步骤3 右上角单击“前往工作台”，进入CodeArts项目首页。

步骤4 单击项目卡片“req-demo”，进入“req-demo”项目首页。

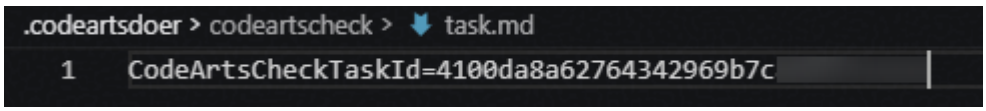
步骤5 左侧导航栏选择“代码 > 代码检查”，进入代码检查（CodeArts Check）首页。

步骤6 在“任务”卡片，单击“repo-demo-check”，进入扫描任务“repo-demo-check”的概览页。

步骤7 复制当前浏览器url的“task”和“detail”中间的字符串，该字符串为此扫描任务的ID。

步骤8 如下图所示，将此任务ID填写到“task.md”文件并保存。

图 6-5 填写 task.md 文件



----结束

6.6 获取扫描报告并修复代码

获取最近一次 commit 的代码安全扫描报告并完成修复

步骤1 在华为云码道代码智能体IDE页面左上方，单击模式切换按钮**Space**，即可从IDE模式切换到码道Agent Space模式。

步骤2 单击“代码开发”，切换到代码开发页签。

步骤3 在聊天界面的输入框下方选择“AgentTeam”，切换到**AgentTeam**模式。右侧显示当前选用的模型，您可在下拉框中切换不同大语言模型。

步骤4 在对话框输入如下指令，单击发送图标 或使用“Enter”快捷键发送。

```
获取最近一次commit的代码安全扫描报告并完成修复
```

码道代码智能体IDE将调用“CodeArts Check MCP”，返回获取的报告路径，用户可在本地查看代码检查报告；内置子智能体将继续修复代码，等待修复完成即可。

----结束

提交修改后的代码

步骤1 在华为云码道代码智能体IDE页面左上方，单击模式切换按钮**IDE**，即可从码道Agent Space模式切换到IDE模式。

步骤2 在终端执行如下命令，查看本地改动，确认修改的文件。

```
git status
```

下述回显为本地修改点。

```
On branch develop-dev
Your branch is up to date with 'origin/develop-dev'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   src/main/java/com/huawei/codearts/controller/TreeController.java
        modified:   src/main/java/com/huawei/codearts/tree/MaxPathSum.java
        modified:   src/main/java/com/huawei/codearts/tree/MaxPathSumRunner.java
        modified:   src/main/java/com/huawei/codearts/tree/TreeNode.java
        modified:   src/test/java/com/huawei/codearts/tree/MaxPathSumTest.java

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .agent-team/
        .arts/
```

步骤3 继续在终端执行如下命令，拉取远程分支最新代码。

```
# 拉取远程所有分支更新
git fetch origin
# 把远程develop-dev最新代码合并到本地当前分支
git pull origin develop-dev
```

- 如果出现如下回显，表示无冲突，可继续执行**步骤4**。

  ```
  nothing added to commit but untracked files present (use "git add" to track)
  From https://test.huawei.com/bb5f2eb7ded44e0bbc0ff4bd3d1d244c/repo-demo
   * branch            develop-dev -> FETCH_HEAD
  Already up to date.
  ```

- 如果出现冲突，终端将提示冲突文件，您可手动打开文件修改冲突内容，修改完后执行如下命令，再继续执行**步骤4**。

  ```
  git add .
  git commit -m "fix: 合并远程代码冲突"
  ```

步骤4 继续在终端执行如下命令，将修改加入暂存区。

```
git add .
```

无任何回显，表示成功将修改文件都提交至暂存区。

步骤5 继续在终端执行如下命令本地提交代码。

```
git commit -m "fix: 修改扫描的代码问题"
```

出现下述回显，表示成功提交代码。

```
[develop-dev 55878b5] fix: 修改扫描的代码问题
 Committer: chenzhu (F) <test.com>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly:

    git config --global user.name "Your Name"
    git config --global user.email you@example.com

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

 9 files changed, 670 insertions(+), 24 deletions(-)
 create mode 100644 .agent-team/ses_07c6a4e4fffe71tWtQ6qZI4mD2/codefix/
```

```
origin_report_4545383827873645935/index.json
 create mode 100644 .agent-team/ses_07c6a4e4fffe71tWtQ6qZI4mD2/codefix/
origin_report_4545383827873645935/subReport_0.md
 create mode 100644 .agent-team/ses_07c6a4e4fffe71tWtQ6qZI4mD2/codefix/
origin_report_4545383827873645935/subReport_1.md
 create mode 100644 .arts/settings.json
```

步骤6 执行如下命令，推送到远程云端develop-dev分支。

```
git push origin develop-dev
```

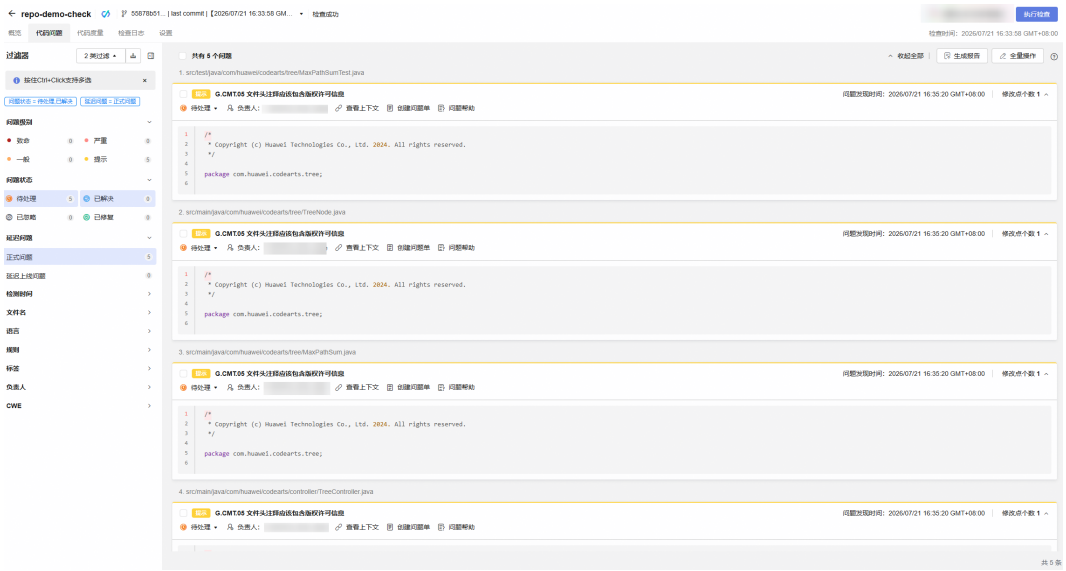
出现下述回显，表示本地代码已完整提交到云端仓库“repo-demo”。

```
Enumerating objects: 47, done.
Counting objects: 100% (47/47), done.
Delta compression using up to 16 threads
Compressing objects: 100% (18/18), done.
Writing objects: 100% (30/30), 6.25 KiB | 112.00 KiB/s, done.
Total 30 (delta 6), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Start Git Hooks Checking                                    [PASSED]
remote:
remote: To create a merge request for develop-dev, visit:
remote:   https://test.com/codehub/2112135977/newmerge?branchName=develop-
dev&projectNamespace=bb5f2eb7ded44e0bbc0ff4bd3d1d244c/repo-demo
remote:
To https://test.com/bb5f2eb7ded44e0bbc0ff4bd3d1d244c/repo-demo.git
   e1256e8..55878b5  develop-dev -> develop-dev
```

提交代码成功后，CodeArts Check将在云端自动触发检查任务。

等待扫描结束后，可参考文档**查看代码检查结果**，查看最新检查报告，如下图所示，剩余5个问题待二次修改。

图 6-6 第二次扫描报告



----结束

二次修改代码

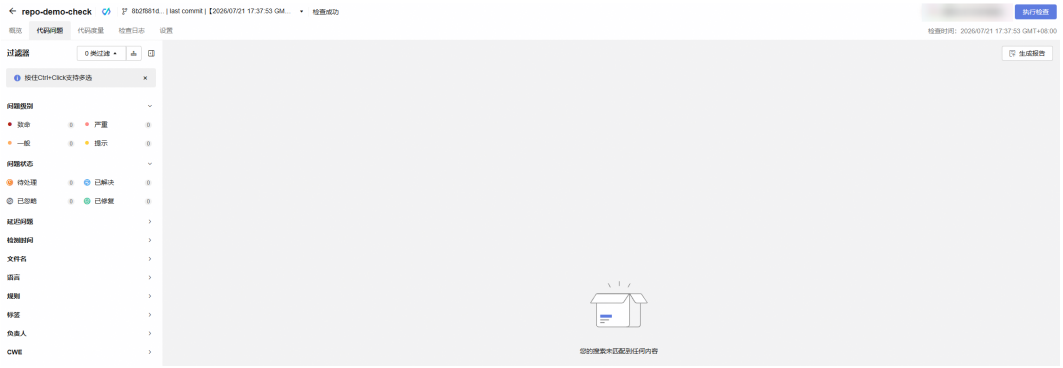
步骤1 参考文档**获取最近一次commit的代码安全扫描报告并完成修复**，码道代码智能体IDE继续获取报告并修改代码。

步骤2 完成二次修改代码后，参考文档**提交修改后的代码**提交本次改动的代码。

提交代码成功后，CodeArts Check将在云端自动触发检查任务。

等待扫描结束后，可参考文档**查看代码检查结果**，查看最新检查报告，如下图所示，所有扫描问题均已修改。

图 6-7 第三次扫描报告



如果报告中仍有问题未修改，请继续参考文档**获取最近一次commit的代码安全扫描报告并完成修复**。

----结束

6.7 新建合并请求

开发人员完成代码开发后，即可提交分支合并请求，Committer审核合并请求后完成合入。

步骤1 开发人员提交合并请求。

1. 登录**CodeArts控制台**，选择“亚太-新加坡”。
2. 右上角单击“前往工作台”，进入CodeArts项目首页。
3. 单击项目卡片“req-demo”，进入“req-demo”项目首页。
4. 左侧导航栏选择“代码 > 代码托管”，进入代码托管首页，单击代码仓库名称“repo-demo”，进入“代码”页。
5. 上方导航栏单击“合并请求”，进入代码合并请求页面。
6. 单击“新建合并请求”，源分支选择“develop-dev”，目标分支选择为“master”，填写“标题”后单击“确认”，完成合并请求创建。

步骤2 Committer审核并合入请求。

1. Committer参考**步骤1.1~步骤1.5**，进入代码仓库“repo-demo”的代码合并请求页面。
2. 单击需要合入的“合并请求”标题，进入合并请求的详情页面。
3. 单击右上角“合入”，即可完成代码合入。

----结束

6.8 相关文档

代码安全扫描报告获取失败

问题现象

用户在获取扫描报告时，IDE返回报错信息“代码检查任务已提交，但在执行阶段因IAM认证信息无效而失败”。

解决方案

这是由于mcp_settings.json文件的environment.ak和environment.sk值不准确。

如果修改mcp_settings.json文件的environment.ak和environment.sk值后，报错仍存在，请通过华为云工单系统**提交工单**获取华为云的客户服务支持。

7 生成通用逻辑代码

一些常见的算法，比如正则表达式、时间处理函数等算法，具有业务逻辑简单但是研发人员编写较为复杂的特性（通常是因为复杂的编码规则，需要人员查阅对应的资料）。

可以使用代码生成，快速生成常见的基础算法，让开发人员专注于处理复杂逻辑。

生成正则表达式/字符串处理函数

步骤1 在IntelliJ IDEA编辑器中编写符合编程语言规范的注释。

```
/*
使用正则表达式，提取输入的字符串中所有数字并返回数组
 */
```

此案例函数逻辑清晰，但是正则的编写对研发人员来说往往需要翻阅资料，较为耗时。

步骤2 将编辑器中光标移动至注释内容最后，按下快捷键“Alt+C”，生成对应处理函数，开发人员在此基础上微调即可满足业务要求。

如果首次生成的代码不完整，可以先按Tab键接受生成的代码，再按快捷键“Alt+C”继续生成代码，直到生成完整代码片段。

----结束

生成日期处理函数

步骤1 在IntelliJ IDEA编辑器中编写符合编程语言规范的注释。

```
/*
生成判断输入的年份是否为闰年
 */
```

步骤2 将编辑器中光标移动至注释内容最后，按下快捷键“Alt+C”，生成对应处理函数，开发人员在此基础上微调即可满足业务要求。

如果首次生成的代码不完整，可以先按Tab键接受生成的代码，再按快捷键“Alt+C”继续生成代码，直到生成完整代码片段。

----结束

8 快速进行仿写

API Controller层的开发基本上和复杂业务逻辑进行了解耦。同时一个业务内的API相似度很高，可以直接使用代码生成，依赖现有的接口去扩展业务接口。

根据注释生成代码

相似结构的接口实现代码：

```
@Override
@OperationLog(objectId = "#roleId", objectType = "AuthRole", details = "")
public RetObj deleteRole(String roleId) {
    try {
        return authRoleService.deleteRole(roleId, DevCloudTokenStore.getUserId().toLowerCase(Locale.ENGLISH));
    } catch (Exception e) {
        return RetObjUtil.returnErrorMessage(e.getMessage());
    }
}

@Override
@OperationLog(objectId = "#params.resourceId", objectType = "AuthResource", details = "")
public RetObj addAuthResoure(AuthResourceParam params){
    try {
        return resourceService.addAuthResoure(params, DevCloudTokenStore.getUserId().toLowerCase(Locale.ENGLISH));
    } catch (Exception e) {
        return RetObjUtil.returnErrorMessage(e.getMessage());
    }
}
```

注释内容：

```
/**
 * 修改权限资源的方法
 * @param params 传入的参数
 * @return 返回修改权限资源后的结果
 */
```

根据上述注释生成的代码：

```
@Override
@OperationLog(objectId = "#params.resourceId", objectType = "AuthResource", details = "")
public RetObj updateAuthResource(AuthResourceParam params) {
```

案例总结

上述项目文件中，已经有结构清晰的上文代码，研发人员在续写的时候，可以通过注释的方式，生成类似结构的代码。

9 编写数据库接口

MyBatis作为常见的数据库框架，经常涉及到大量的接口类生成，并且很多情况下这些接口类都具有类似的格式，因此在上文的基础上生成新的业务接口也是比较常见的代码生成场景。

根据注释生成数据库接口代码

```
// 获取L0的IdList
List<ToolAllListCkBillEntity> getL0IdList(@Param("type") String type, @Param("start_time") String startTime,@Param("end_time") String endTime);
// 获取L1的IdList
List<ToolAllListCkBillEntity> getL1IdList(@Param("type") String type, @Param("start_time") String startTime,@Param("end_time") String endTime);
```

对于类似的业务逻辑，可以直接生成对应的接口。

案例总结

可以通过代码生成能力，快速学习到已有代码的行文风格，并在此基础上快速扩展代码，提升编码效率。

10 查询未知依赖

缺失代码依赖，或者缺少软件包信息，不需要去网站搜索，在IDE内部即可完成查询。

操作流程

步骤1 参考**快速启动**中操作，登录华为云码道代码智能体。

步骤2 进入华为云码道代码智能体的聊天界面。

步骤3 在聊天界面的输入框里输入如下问题描述。

```
我的代码中引入了如下API，请问我需要在pom文件中引入什么依赖?
import org.junit.Before;
import org.junit.Test;
```

步骤4 单击发送图标  。

步骤5 华为云码道代码智能体将会在聊天界面响应结果。

当前展示的效果图仅是示例，请以最终实际生成的效果为准。

图 10-1 响应结果

----结束

11 直接引入组件

对于常见的三方组件，可以使用华为云码道代码智能体提示常见的引入方法和代码模板。

操作流程

步骤1 参考**快速启动**中操作，登录华为云码道代码智能体。

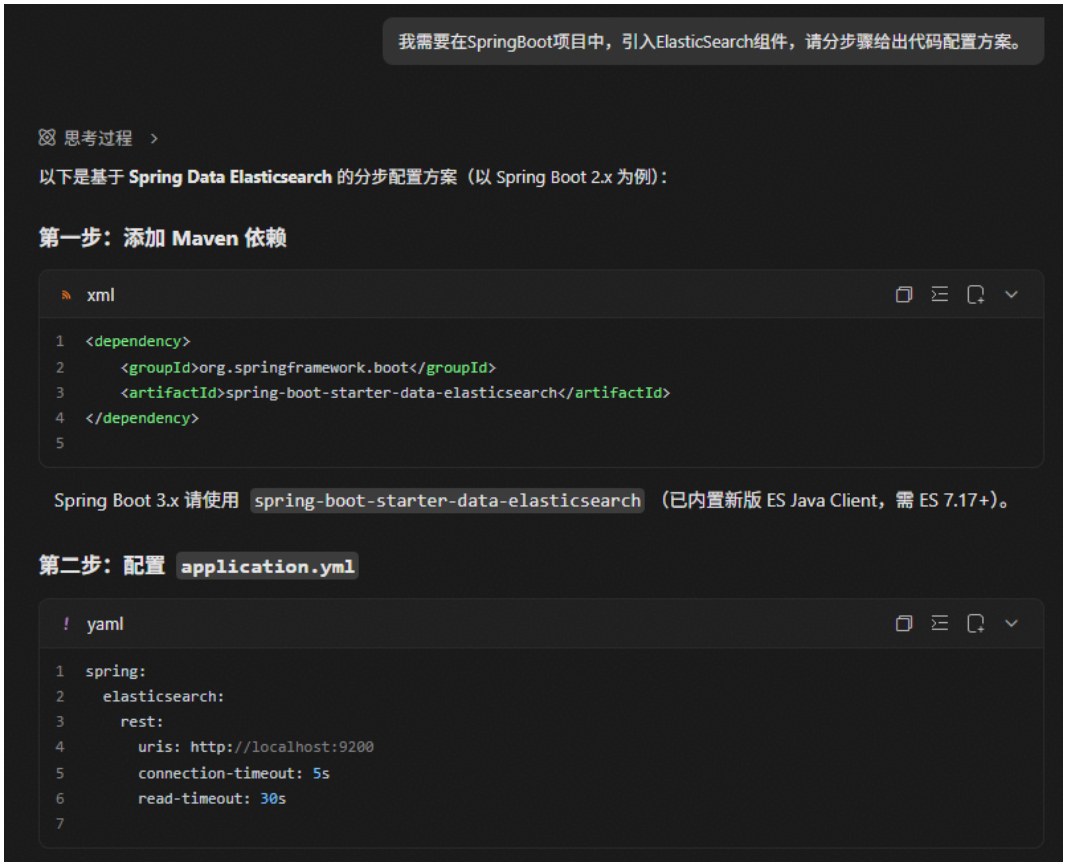
步骤2 在华为云码道代码智能体聊天界面的输入框中输入问题描述，如“我需要在SpringBoot项目中，引入ElasticSearch组件，请分步骤给出代码配置方案。”

步骤3 单击发送图标  。

步骤4 华为云码道代码智能体将会在聊天界面响应结果。

当前展示的效果图仅是示例，请以最终实际生成的效果为准。

图 11-1 响应结果



----结束

12 如何写好一个 Skill

没有Skill的团队，每个人都在重复造轮子。工程师的提示词散落各处，项目经理（Project Manager）的需求表述每次让智能体重新猜，管理层的经验则随着人员流动悄悄蒸发。表面上看，工程师、项目经理、管理层关心的事情完全不同。但在智能体工具的使用层面，大家面对的是同一个困境：经验无法积累，质量依赖个人。

Skill的本质——隐性知识的代码化封装：理解Skill，需要先理解它解决的是一个知识管理问题，而不只是一个AI工具问题。

管理学里有一个经典概念：隐性知识（Tacit Knowledge）。工程师知道“怎么和智能体沟通才能拿到高质量的代码审查”，这是隐性知识——他能做到，但说不清楚，也很难直接教给别人。与之对应的是显性知识——写在文档里、能被读取和传播的知识。

组织的知识管理目标，是尽可能多地把隐性知识转化为显性知识。传统方式是写文档、写SOP、做培训。但这些方式有一个致命的缺陷：文档会过期，SOP会被绕开，培训效果难以量化。更重要的是，文档描述的是人应该怎么做，而不是智能体应该怎么做——在智能体大量介入工作流的今天，这个区别至关重要。

Skill是更彻底的答案。它不是描述最佳实践的文档，而是**将最佳实践直接编码进智能体的执行路径**。当你把“竞品分析的标准框架”写成一个Skill，团队里任何人打开智能体说“帮我分析这个竞品”，智能体都会按照那套经过验证的框架来执行，而不是即兴发挥。

与普通提示词相比，Skill的核心差异有如**表12-1**所示的三个维度。

表 12-1 Skill 与普通提示词对照表

维度	普通提示词	Skill
存储位置	个人笔记/对话历史。	团队共享库，结构化存储。
激活方式	用户手动粘贴。	智能体按需自动加载。
迭代机制	无版本管理，难以跟踪变化。	版本化管理，可回滚。

这个差异意味着：Skill不只是更好用的提示词，它更是一种知识的制度化存储形式，能让个人智慧变成组织能力。

从案例看效果：标准化如何发生

某产品团队有8名项目经理，日常工作中竞品分析报告是高频输出物。在引入Skill之前，8个人的报告风格各异：有人擅长写功能对比，有人专注用户体验，有人把重点放在商业模式——质量参差不齐，管理层每次阅读都要先适应一套新的框架。团队有经验的一位项目经理花了两天时间，把自己总结出的竞品分析框架封装成一个Skill：包含功能矩阵、用户旅程差异、定价策略、市场定位四个维度，每个维度都有标准的分析视角和输出格式。

上线三周之后，变化非常具体：

- 全团队的竞品分析报告开始采用统一框架，管理层的阅读时间从平均20分钟降到8分钟。
- 新来的两名项目经理在第一周就能产出与有经验的项目经理相当质量的分析。
- 项目经理的经验从此不再是“他个人的事”，而是团队的基础能力。

这就是Skill让标准化发生的路径：**不是靠培训和规范约束人，而是把最佳实践直接内置到工具里**，让每次使用都自然地沿着最优路径走。

理解Skill不需要先读代码——它更像是一份交给智能体的“岗位说明书”：告诉它什么情况下上岗、上岗后怎么干。本节拆解Skill的三个核心组成，展示渐进式加载的三层架构。

Skill的三个核心组成：一个最简单的Skill，仅需要name、description与执行指令三个要素。

- name（技能标识符）：name是Skill的唯一ID，也是它在系统和日志里被引用的方式。命名原则很简单，即动词+名词，清晰表达这个Skill做什么。“review-code”、“analyze-competitor”、“generate-prd”是好名字。“helper”、“assistant”、“tool1”是坏名字——模糊的名字在Skill库里会成为维护噩梦。
- description（技能描述）：description是整个Skill里最关键、也最容易被轻视的部分。智能体在决定“要不要激活这个Skill”时，唯一依赖的判断依据就是description。写得模糊等于白写——如果description没能在正确的场景触发Skill，再精妙的执行指令也毫无用处。
- 执行指令：具体执行剧本。指令是Skill被激活之后智能体实际执行的内容。它可以是一段结构化的步骤说明，也可以包含具体的分析框架、输出格式要求、边界条件处理方式。正文指令对智能体的作用，相当于一份详细的操作手册——越具体，执行越稳定。

以一个代码审查Skill为例，完整的三要素大概长这样：

```
yaml
name: review-code

description: |
  当用户提交代码请求审查、或要求对代码质量进行评估时激活。
  适用场景包括：Pull Request 审查、单文件代码质量检查、重构前的质量基线评估。
  不适用于代码调试或功能开发。

instructions: |
  ## 代码审查执行步骤

  ### 1. 首先评估整体结构
  - 模块划分是否清晰
  - 命名规范是否一致
  - 注释覆盖率是否达标

  ### 2. 逐项检查以下维度
  - **可读性**：代码是否容易理解，复杂逻辑是否有注释
```

```
- **安全性**：是否存在常见漏洞（SQL 注入、XSS 等）
- **性能**：是否有明显的性能问题
- **测试覆盖**：关键路径是否有测试

### 3. 输出格式
按以下结构输出审查报告：
- 总体评分（1-10）
- 必须修改项（Blocker）
- 建议改进项（Suggestion）
- 亮点记录（Highlight）
```

渐进式加载的三层架构：Skill的设计里有一个重要的工程决策，不是所有内容都需要始终驻留在智能体的上下文窗口里。一个团队可能有几十个甚至上百个Skill，如果每个Skill的全部内容都常驻上下文，会造成严重的上下文污染和性能损耗。Skill采取渐进式加载的三层架构来解决这一问题。

- 第一层：元数据——只包含name和description，体积极小，始终在智能体的视野内。智能体靠这一层来判断“有没有合适的Skill可以用”。
- 第二层：SKILL.md正文——包含完整的执行指令。只在Skill被激活（即description匹配成功）之后才读入上下文。这一层是Skill的主体，包含了所有的“怎么做”。
- 第三层：捆绑资源——脚本、参考文档、输出模板等附属材料。只在执行过程中需要引用时才按需加载，不会无谓地占用上下文空间。

这个架构的好处是：Skill数量的增加，不会线性增加智能体的上下文负担。一个团队有50个Skill，智能体的常驻上下文里只有50条轻量的description摘要，真正消耗上下文的内容，只有当前任务实际激活的那一两个Skill。

可附带的资源类型与设计原则：除了最基本的三个元素外，Skill的第三层可以携带三类资源，各有适用场景。

- 自动化脚本：适合处理固定的、无需每次判断的机械性操作。比如一个“生成API文档”的Skill，可以附带一个从代码注释自动提取接口信息的脚本，智能体在执行时直接调用，而不是每次手动描述提取逻辑。
- 参考文档：适合需要大量背景知识支撑的Skill。比如“合规检查”Skill可以附带公司的合规手册摘要，“技术方案评审”Skill可以附带架构设计规范。长篇参考文档建议加目录索引，方便智能体定向检索而不是全文扫描。
- 输出模板：适合对输出格式有严格要求的场景。比如周报生成Skill附带固定的Markdown模板，竞品分析Skill附带标准的表格结构。模板能大幅减少智能体在格式上的“自由发挥”，让输出更一致。