

应用平台

# 开发中心用户指南

文档版本 09  
发布日期 2024-11-22



版权所有 © 华为云计算技术有限公司 2024。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

<b>1 AppStage 开发中心简介</b>	<b>1</b>
1.1 什么是 AppStage 开发中心	1
1.2 AppStage 开发中心基本概念	2
<b>2 AppStage 开发中心使用前准备</b>	<b>4</b>
<b>3 AppStage 开发中心首页介绍</b>	<b>5</b>
<b>4 AppStage 开发中心使用流程</b>	<b>8</b>
<b>5 为 AppStage 开发中心添加使用人员</b>	<b>11</b>
<b>6 管理 AppStage 开发中心系统配置</b>	<b>13</b>
6.1 集成工具链到开发中心	13
6.2 管理开发中心系统菜单	19
6.3 管理开发中心版本活动	21
6.3.1 添加自定义版本活动	21
6.3.2 修改系统预置版本活动关联的工具链	22
6.3.3 添加版本交付件模板	23
6.3.4 添加版本代码安全检视检查项	25
6.3.5 管理版本标准发布准入配置	26
6.4 工具链同步信息接口规格说明	28
<b>7 创建并管理团队</b>	<b>42</b>
7.1 在 AppStage 开发中心创建开发团队并关联服务	42
7.2 在 AppStage 开发中心为服务关联 CodeArts 项目	43
7.3 为 AppStage 开发中心开发团队添加成员	45
<b>8 创建及规划版本</b>	<b>46</b>
8.1 在 AppStage 开发中心创建版本	46
8.2 在 AppStage 开发中心规划交付件	48
8.3 在 AppStage 开发中心规划补丁版本缺陷	49
8.4 在 AppStage 开发中心规划版本需求	50
<b>9 开发版本</b>	<b>53</b>
9.1 管理 AppStage 开发中心版本交付件文档	53
9.2 在 AppStage 开发中心创建代码仓	54
9.3 下载代码仓库到本地	55

9.4 提交代码到 AppStage 开发中心代码仓.....	59
9.5 在 AppStage 开发中心进行代码安全检视.....	59
<b>10 在 AppStage 开发中心通过流水线编译构建版本.....</b>	<b>61</b>
<b>11 测试版本.....</b>	<b>75</b>
11.1 在 AppStage 开发中心为版本制定测试计划.....	75
11.2 在 AppStage 开发中心创建缺陷.....	76
11.3 在 AppStage 开发中心进行版本测试评估.....	78
<b>12 发布版本.....</b>	<b>82</b>
12.1 在 AppStage 开发中心调测发布版本.....	82
12.2 在 AppStage 开发中心标准发布版本.....	84
<b>13 查看版本发布制品信息.....</b>	<b>88</b>
<b>14 使用 AppStage 开发中心知识库管理团队文档.....</b>	<b>89</b>
<b>15 在 AppStage 开发中心管理插件.....</b>	<b>92</b>
<b>16 使用 AppStage 开发中心效能大盘度量研发效能.....</b>	<b>95</b>

# 1 AppStage 开发中心简介

## 1.1 什么是 AppStage 开发中心

开发中心提供全场景一站式作业平台，承载端到端研发作业流，提供涵盖软件研发全生命周期的研发工具链和研发管理服务。以团队为中心，深度集成第三方工具链能力，基于服务以及版本为维度提供设计、开发、测试、发布全场景一站式研发门户，实现精细化项目管理，掌握和处理项目全量信息，支撑研发角色统一在一站式门户协同工作，提升团队研发效率。

### 开发中心应用场景

- **互联网SaaS开发运营**

在面对市场高速变化、产品盈利窗口窄时，经常由于研发工具难以满足项目实际需求，导致难以及时交付高质量的产品给客户；应用平台提供一站式工具链，覆盖应用开发各个环节，支持云上开发运维一体化，同时提供各种开发插件提升SaaS开发效率，帮助用户业务成功。

- **研发流程和知识能力沉淀**

企业在面对云原生和数字化转型过程中，被动式的工具堆砌，缺少体系化的工具能力和持续的研发知识沉淀，研发工具能力建设影响业务的发展。应用平台开发中心通过一站式工具链实现流程Build In，提供海量研发资产模板，帮助客户实现研发的持续反馈和能力沉淀，提升研发能力和效率。

### 开发中心功能介绍

开发中心的主要功能如[表1-1](#)所示。

表 1-1 开发中心功能介绍

主要功能	功能简介
团队管理	团队和成员管理，团队所属部门、关联服务等信息配置以及成员的新增、修改和删除。
版本管理	需求和业务规划，制定版本计划，跟踪交付任务，跟踪发布上线。

主要功能	功能简介
需求管理	需求管理流程覆盖需求的收集、分析、评审、排序、分发、实现、验证、确认的全过程。
代码管理	提供分布式代码管理和协同开发能力，包括代码托管、代码检查、代码审核、代码追溯、持续集成等功能。
流水线管理	将产品需求纳入流水线进行需求设计、开发、验证，最终通过软件包将实现的用户需求交付给下游。
缺陷管理	从缺陷的发现和提出，到分析定位、实施修复，再到测试和验收，最终导向缺陷的闭环。
知识库	知识库是面向开发团队的知识管理系统，支持知识全生命周期管理，帮助您完成智能、安全的知识作业，提供层级式的目录树、多人在线协同编辑等功能。
开发插件库	提供多种预置的开发插件，同时组织管理员可以上传用户需要的其他本地插件至开发插件库进行灵活管理，也可以为插件分类创建标签，便于快速筛选及查找。
效能管理	提供从需求、缺陷、代码、构建、测试、部署、发布到运营等研发各阶段作业数据的分析洞察能力。

## 1.2 AppStage 开发中心基本概念

基本概念	说明
产品	产品是服务的合集，开发中心基于服务进行版本管理。
服务	服务是实现一系列功能的软件模块单元，是产品的组成要素，一个产品一般由一个或多个服务组成。将一个大型的应用程序可以拆分成一组小型、独立的服务。每个服务都运行在自己的进程中，并通过轻量级的通信机制（如HTTP或消息队列）进行相互通信。每个服务都专注于完成特定的业务功能，并且可以独立地进行开发、部署和扩展。
开发中心用户	开发中心的用户数根据购买规格有相应的用户数量限制，通过组织管理员添加后才能成为开发中心用户，或者申请成为开发中心用户。如果不是开发中心用户，则无法使用开发中心。
交付件	交付件是版本开发过程不同阶段所需或交付的必要文档。在交付件规划页面，系统内置了例如产品需求文档、升级指导书、版本配套表等交付件类型，同时可以添加自定义类型交付件，并为交付件上传文件作为交付件模板。在交付件管理页面，用户可以通过创建在线文档或上传离线文档两种方式创建及管理交付件文档，保证版本开发过程阶段性成果沉淀以及过程可追溯。
版本基线化	版本基线化是指在版本发布前，选择特定流水线最新一次的构建产物作为版本基线，将流水线构建产物与版本关联，版本发布审批通过后，开发中心将版本关联的软件包或镜像包信息通知给部署平台，以便后续部署变更。

基本概念	说明
产能	产能原指在一定时间内，企业所能生产的产品或提供的服务的数量。开发中心中特指版本开发团队每天能提供的开发人员的数量。
Scrum	增量迭代式的软件开发方法，通过最重要的迭代计划会议、每日站会、迭代回顾、验收会议来进行简单高效的管理。
IPD-系统设备类	面向系统设备类产品开发场景的IPD需求管理方法，通过结构化流程、强大的跨项目协作能力来对大型产品开发进行高质高效的管理。
IPD-独立软件类	面向独立软件类开发场景的IPD需求管理方法，通过结构化流程、强大的跨项目协作能力来对大型软件开发进行高质高效的管理。
开发插件	开发插件是一种软件应用程序，用于开发、编译、调试和部署软件。IDE通常包含代码编辑器、编译器、调试器和其他开发工具，以提供一个集成的开发环境，方便开发人员进行软件开发工作。
内建质量	内建质量指在产品或服务设计和开发过程中，将质量要求和标准纳入到产品或服务的设计和开发中，从而在产品或服务最初的构建阶段就实现了质量控制。这样做可以避免在后续阶段出现质量问题，提高产品或服务的可靠性和稳定性。

# 2 AppStage 开发中心使用前准备

使用开发中心前，需要先准备如表2-1所示内容。

表 2-1 准备事项

准备事项	说明
购买AppStage开发中心	首次使用需要先购买开发中心专业版，具体操作请参见 <a href="#">购买AppStage</a> 。
关联组织	首次购买AppStage后，其账号需创建并关联使用AppStage的组织（仅可关联一个组织），才能使用AppStage服务及后续购买AppStage相关产品套餐或增量包等，具体操作请参见 <a href="#">为AppStage关联组织</a> 。关联组织完成后，该账号会自动成为组织管理员，拥有该组织的所有管理权限，同时可以审批其他用户的组织管理员角色权限申请。
配置服务授权	开发中心深度集成CodeArts的需求管理、代码仓管理、流水线管理和缺陷管理等功能，需要关联CodeArts并配置凭证授权，具体操作请参见 <a href="#">配置开发中心服务授权</a> 。
添加部门/成员信息	为已关联的组织添加部门及成员，完善组织架构，具体操作请参见 <a href="#">管理已关联组织的部门及成员</a> 。
录入产品/服务信息	将产品、服务信息录入AppStage系统中，信息录入成功后，AppStage将同步产品、服务信息至开发中心，具体操作请参见 <a href="#">管理产品与服务</a> 。



# 3 AppStage 开发中心首页介绍

## 进入开发中心

步骤1 登录AppStage。

步骤2 选择“开发中心”，如图3-1所示，进入AppStage开发中心。

图 3-1 开发中心入口



----结束

## 开发中心首页介绍


开发中心的用户数根据购买规格有相应的用户数量限制，通过组织管理员添加后才能成为开发中心用户。如果不是开发中心用户，则无法使用开发中心，进入开发中心后可以根据页面提示申请成为开发中心用户。

开发中心用户进入工作台如图3-2所示。


图 3-2 开发中心首页



### 1. 访问入口

提供快捷访问入口， 为菜单访问入口，单击“AppStage”可访问AppStage首页，单击“开发中心”可访问开发中心首页。

### 2. 系统配置

提供系统配置入口及操作指引入口，组织管理员可以在“用户管理”页面添加开发中心用户，在“插件管理”页面对插件进行管理，在工具链中集成并关联工具链。单击可以查看开发中心操作指引。

### 3. 一站式开发功能地图

“一站式开发功能地图”区域展示了开发过程的主要阶段和流程，提供图形化指引。鼠标悬浮到具体的流程节点时，可以查看当前阶段的子流程和参与人员角色。

### 4. 常用功能

常用功能模块的快捷入口，单击功能名称可快捷跳转到相应的功能操作界面。

### 5. 开发插件及示例代码

在该区域可总览常用的开发插件，单击“更多”，可快捷跳转至开发插件库页面，获取更多开发插件及示例代码。

### 6. 我的团队

“我的团队”区域，可快捷管理团队信息，相关操作详见表3-1。

表 3-1 我的团队相关操作

操作	说明
创建团队	<p><b>说明</b></p> <p>仅支持组织管理员创建团队。</p> <ol style="list-style-type: none"> <li>单击“我的团队”区域右上角“创建团队”。</li> <li>在“创建团队”页面，设置团队相关参数，包括团队名称、团队归属部门、关联服务、团队LOGO、团队简介。</li> <li>单击“创建”。</li> </ol> <p>在“我的团队”区域可查看到创建的团队的卡片。</p>

操作	说明
关注团队	<ol style="list-style-type: none"><li>1. 单击“我的团队”区域右上角“关注团队”。</li><li>2. 在“关注团队”页面的“团队列表”中，单击需要添加的团队名称右侧的“关注”。 在“我的团队”列表中可查看到此新关注的团队卡片。</li></ol>
设置默认团队	选中“我的团队”区域内团队卡片左下角的“设为默认”，可将此团队设置为默认团队，默认团队将置顶显示。
移出团队	单击“我的团队”区域内团队卡片右下角的“移出”，可将此团队从我的团队列表移出，我的团队列表中不显示此团队。移出后通过右上角“添加团队”可再次添加到我的团队。

# 4 AppStage 开发中心使用流程

本章节介绍AppStage开发中心的使用流程。

## 前提条件

使用开发中心前需要完成[使用前准备](#)。

## 使用流程

参考[图4-1](#)和[表4-1](#)可帮助您快速上手开发中心的主流程和核心功能。

图 4-1 开发中心使用流程

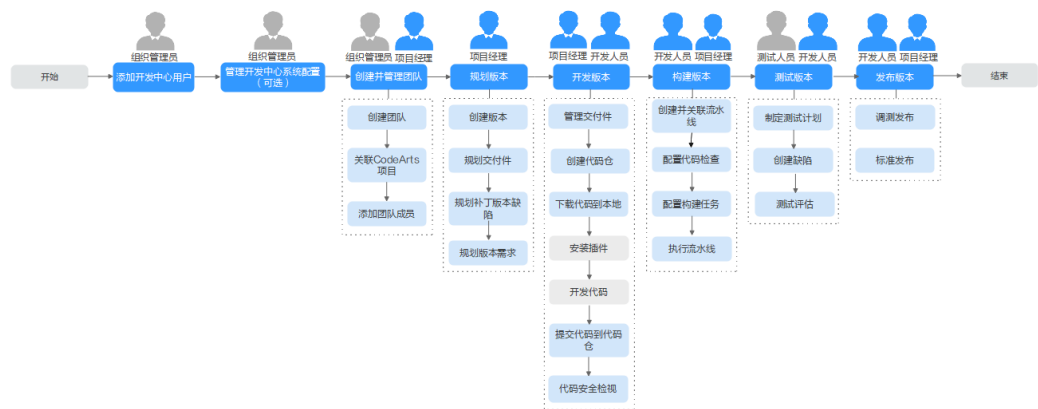


表 4-1 使用流程详解

序号	流程环节	说明
1	<a href="#">添加开发中心用户</a>	开发中心的用户数根据购买规格有相应的用户数量限制，组织管理员通过用户的添加或删除来合理管理开发中心的使用人员。
2	(可选) <a href="#">管理 AppStage 开发中心系统配置</a>	开发中心系统预置了对应的工具链、菜单及版本活动，组织管理员可以自定义添加或修改配置，同时可以预置版本活动中的交付件模板、代码安全检查项和发布准入配置。

序号	流程环节	说明	
3	创建并管理团队	<b>创建团队</b>	在开发中心进行服务开发前，需要组织管理员先创建一个团队，将团队关联需要开发的服务。
		<b>关联CodeArts项目</b>	开发中心默认集成CodeArts工具链，开发中心创建团队后，关联的服务需要对应创建一个CodeArts项目，后续的服务开发管理过程才能正常使用CodeArts的代码托管、流水线、编译构建功能。
		<b>添加团队成员</b>	在团队多用户协作开发过程中，不同角色成员执行各自权限和责任事项，因此需要在团队中添加协作开发的各角色成员（项目经理、开发人员、测试人员等），实现统一有序的成员信息管理。
4	规划版本	<b>创建版本</b>	创建版本的基本配置信息以及选择版本开始和结束时间。
		<b>规划交付件</b>	在版本开发启动初期对产品开发过程所需的交付件进行规划，以辅助开发过程规范有序且可追溯。
		<b>规划补丁版本缺陷</b>	针对补丁类型的版本规划需要解决的缺陷。
		<b>规划版本需求</b>	使用内置的多种开箱即用的场景化需求模型和对象类型录入版本开发相关的需求/任务。
5	开发版本	<b>管理交付件</b>	进行交付件管理等，保证版本开发过程阶段性成果沉淀以及过程可追溯。
		<b>创建代码仓</b>	开发人员进行代码开发前，需要先创建代码仓库用于托管代码。
		<b>下载代码到本地</b>	开发人员进行代码文件的操作前，需要使用Git Bash客户端和TortoiseGit客户端下载代码仓库文件到本地环境中。
		<b>安装插件</b>	开发人员可下载代码开发所需的开发工具，如WiseStudio插件、Nuwa插件等，然后进行安装，具体操作请参见 <a href="#">查看插件详情或下载插件</a> 。
		<b>开发代码</b>	开发代码是开发人员使用编程语言编写程序的过程，包括设计、编写、测试和维护程序的过程。在代码开发过程中，开发人员需要根据需求分析和设计文档，使用编程语言和开发工具编写程序代码，然后进行测试和调试，最终交付给用户使用。代码开发是产品开发的重要环节，决定了产品的质量和可靠性。 开发代码的详细指导请参见《 <a href="#">开发指南</a> 》。
		<b>提交代码到代码仓</b>	开发人员在本地完成业务代码编写后，需要提交代码文件至代码仓库。
6	构建版本	<b>配置流水线</b>	根据业务需要，如开发测试环境应用部署、生产环境应用部署等，对这些自动化任务进行自定义编排，一次配置后就可以一键自动化触发调度执行。
		<b>代码安全检视</b>	进行安全编码TOP问题的信息统计和检查结果录入。
7	测试版本	<b>制定测试计划</b>	明确测试时间、测试范围、测试目标，并管理测试各个阶段的活动。

序号	流程环节	说明	
	<b>缺陷管理</b>	对于版本测试过程发现的缺陷进行记录、跟踪、分析和解决。	
	<b>测试评估</b>	编辑总体测试结论、系统预置的测试类型结论，新增和管理自定义测试类型的测试结论，以及新增和管理版本关键风险问题。	
8	发布版本	<b>调测发布</b>	在版本标准发布（即正式发布）前，可先执行调测发布进行版本调测。
		<b>标准发布</b>	待版本开发及编译构建完成，且发布准入检查项均通过后，可由开发人员、测试人员或项目经理提交标准发布申请，待项目经理审批通过后，将软件版本发布到对应的部署平台。

# 5 为 AppStage 开发中心添加使用人员

开发中心的用户数根据购买规格有相应的用户数量限制，组织管理员通过用户的添加或删除来合理管理开发中心的使用人员。

## 操作须知

- 组织管理员可将组织中的任一用户添加为开发中心用户，已添加的用户不可重复添加。
- 添加至开发中心“用户管理”中的用户称为开发中心用户，未添加至开发中心“用户管理”中的用户称为非开发中心用户。
- 开发中心的“项目经理”、“开发人员”、“测试人员”、“浏览者”角色仅开发中心用户可申请。
- 仅开发中心用户才能被添加为团队空间成员。
- 组织管理将用户从开发中心用户中删除时，会同步删除该用户在开发中心中所有的角色权限。
- 非组织管理员且非开发中心用户访问开发中心时，不可查看团队列表，不可关注团队，需要在开发中心概览页根据页面提示进行申请，申请时可选择任一组织管理员审批（申请电子流将生成待办推送至AppStage首页由审批人进行审批）。
- 开发中心用户总人数只能小于等于总配额，超过用户总配额时，可以联系AppStage开通者在AppStage控制台变更规格，增加开发中心用户数配额。

## 前提条件

- 已获取组织管理员角色权限，权限申请方法请参见[申请权限](#)。
- 组织管理员已在AppStage首页的成员管理页面创建成员账号。组织管理员创建成员账号的方法请参见[创建成员](#)。

## 添加用户

- 步骤1** 在开发中心首页右上角选择“用户管理”。
- 步骤2** 在“用户管理”页面，单击右上角“添加用户”。
- 步骤3** 在“添加用户”页面上方输入框输入用户名或账号名进行搜索，在左侧区域框中勾选一个或多个用户，选中的用户显示在右侧“已选择”区域框。
- 步骤4** 单击“确定”，新添加的用户显示在用户列表中。

### 📖 说明

在用户列表中，可查看已添加的用户账号名、添加人及添加时间等信息。

----结束

## 更多操作

添加用户完成后，还可以执行如表5-1所示的操作。

表 5-1 更多操作

操作	说明
查看用户详情	在用户列表中单击账号名，可查看用户详情，包括基本信息（是否为组织管理员、添加人及添加时间等），以及在开发中心拥有的角色权限相关信息。
删除用户	<b>说明</b> 删除用户将会清空该用户在开发中心中所有的角色权限，请谨慎操作。 1. 在用户列表中“操作”列单击“删除”。 2. 在“删除用户”对话框，单击“确定”。



# 6 管理 AppStage 开发中心系统配置

## 6.1 集成工具链到开发中心

工具链是一系列用于研发流程的工具。不同类型的工具可以作用于研发流程中的不同阶段，通过工具间的配合，形成一整套的研发工具链。研发流程的各个环节有工具协助，可以降低沟通成本，把各个流程集中化管理，有效提升研发效率。

AppStage开发中心支持集成代码检查、缺陷、流水线、发布通知、代码仓、需求、测试、版本活动和其他共9种类型的工具，系统预置了CodeArts的代码检查、缺陷、流水线、代码仓、需求和测试6种类型的6个工具，同时预置了发布通知工具用于对接运维中心，覆盖研发流程中的各个阶段，达到研发团队高效协同的目的。

### 前提条件

已获取组织管理员角色权限，权限申请方法请参见[申请权限](#)。

### 添加并集成工具

**步骤1** 在开发中心首页右上角选择“工具链 > 工具链集成配置”。

**步骤2** 添加工具。

1. 在“工具链集成”页面，单击右上角“添加工具”。
2. 在“基础信息”页面，配置如表6-1所示参数。

表 6-1 添加工具参数说明

参数名称	参数说明
工具LOGO	单击系统默认图片上的“点击修改”，可选择系统预置图片作为工具LOGO，也可单击“上传LOGO”，上传本地图片作为工具LOGO，上传图片支持支持JPG、JPEG、PNG和BMP，最大10MB。
工具名称	自定义工具名称，最大长度为50，禁止输入'; \$<>#=#+%字符，且不能与已有的工具名称重复。

参数名称	参数说明
工具编码（可选）	输入工具编码，工具编码是工具在系统内的唯一标识，不支持重复。以字母开头，支持数字、字母、下划线（_），支持3~64个字符。
工具类型	选择工具类型，支持选择代码检查、缺陷、流水线、发布通知、代码仓、需求、测试、版本活动和其他共9种类型，可多选。
工具描述	输入工具描述，最大长度256。

- 单击“确定”。
- 在“工具类型提示”页面确认工具类型，单击“确认”，工具链添加后不可修改工具链类型。

添加的工具会以“草稿”状态显示在“全部”页签及所选的工具类型页签下，如图6-1所示。

图 6-1 工具类型



### 步骤3 配置工具集成。

在“工具链集成”页面，单击已添加的工具，进入工具集成配置页面，默认显示工具的“基础信息”页签，可以切换页签配置工具集成。

- （可选）认证集成：通过认证集成界面，可更改工具所在应用的认证集成参数，认证集成信息主要用于工具免登录鉴权使用。
  - 当需要对工具进行免登录时，可以开启认证集成开关。
  - 选择认证集成方式：OAuth2、OIDC、SAML、CAS。
  - 根据选择的认证集成方式不同，需要配置不同的参数，参数说明如表6-2所示。配置完成后，单击“保存”。

#### 说明

保存后会生成client\_id和client\_secret，可以用于获取用户访问token，其中client\_secret可单击复制一次，复制后请妥善保管，再次复制须重置后操作。

表 6-2 认证集成配置参数说明

认证集成方式	参数名称	参数说明
OAuth2	回调URL	填写认证回调接口，支持IPV4配置或域名配置，域名地址限制https、http开头，例如https://example.com。

认证集成方式	参数名称	参数说明
	Refresh Token有效期	允许用户在多久时间内不用重新登录应用的时间。 单位秒，默认为2小时，最长180天。
	Access Token有效期	允许用户在多久时间内保持登录应用的时间。 单位秒，默认为2小时，最长24小时。
OIDC	回调URL	填写认证回调接口，支持IPV4配置或域名配置，域名地址限制https、http开头，例如https://example.com。
	管理员登录URL（可选）	管理员登录应用的URL地址。
	退出地址（可选）	统一退出接口被调用时将访问此URL退出应用。
	授权码模式	选择是否开启授权码模式，默认不开启。
	TOKEN签名算法	支持选择：RS256、RS384、RS512。
	Refresh Token有效期	允许用户在多久时间内不用重新登录应用的时间。 单位秒，默认为2小时，最长180天。
	Access Token有效期	允许用户在多久时间内保持登录应用的时间。 单位秒，默认为2小时，最长24小时。
SAML	SP Entity ID	SP唯一标识，对应SP元数据文件中的“Entity ID”的值。
	断言消费地址（ACS URL）	SP回调地址（断言消费服务地址），对应SP元数据文件中“AssertionConsumerService”的值，即当认证成功后响应返回的值。
	Name ID	用户在应用系统中的账号名对应字段，支持选择：邮箱、手机号、用户名、用户id、账号id。

认证集成方式	参数名称	参数说明
	NameID Format (可选)	SP支持的用户名称标识格式。对应SP元数据文件中“NameIDFormat”的值。支持选择： <ul style="list-style-type: none"> <li>urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified</li> <li>urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress</li> <li>urn:oasis:names:tc:SAML:2.0:nameid-format:transient</li> <li>urn:oasis:names:tc:SAML:2.0:nameid-format:persistent</li> </ul>
	Audience URI (可选)	允许使用SAML断言的资源，默认和SP Entity ID相同。
	Single Logout URL (可选)	服务提供商提供会话注销功能，用户在OrgID注销会话后返回绑定的地址。对应SP元数据文件中“SingleLogoutService”的值。“SingleLogoutService”需要支持HTTP Redirect或HTTP POST方式。
	默认Relay State (可选)	使用在IdP发起的认证中，作为默认的一个值。
	支持 ForceAuth (可选)	如果SP要求重新认证，则强制用户再次认证。
	Response签名 (可选)	是否对SAML Response使用IdP的证书签名。
	断言签名 (可选)	断言需使用IdP的证书签名，对应SP元数据文件中“WantAssertionsSigned”值。
	数字签名算法 (可选)	对Response签名或断言签名的签名算法。支持RSA_SHA256、RSA_SHA512、RSA_RIPEMD160，可在下拉框选择。
	数字摘要算法 (可选)	对Response签名或断言签名的摘要算法。支持SHA256、SHA512、RIPEMD160，可在下拉框选择。
	验证请求签名 (可选)	是否对SAML Request签名进行验证，对应SP元数据文件中“AuthnRequestsSigned”值。
	验证签名证书	对SAML Request签名进行验证时，需要输入验证签名证书。

认证集成方式	参数名称	参数说明
CAS	回调URL	填写认证回调接口，支持IPV4配置或域名配置，域名地址限制https、http开头，例如https://example.com。
	管理员登录URL（可选）	管理员登录应用的URL地址。
	退出地址（可选）	统一退出接口被调用时将访问此URL退出应用。

- （可选）同步集成：通过同步集成，可以将开发平台中的应用数据以及版本发布成功信息同步给配置的第三方系统。应用数据包括租户、产品、服务、微服务和版本信息，其中租户信息需要配置认证集成后才能同步。
  - a. 当需要同步数据给第三方系统时，可以开启同步集成开关。  
发布通知类型工具默认开启，且不能关闭。
  - b. 输入签名密钥、加密密钥并选择摘要算法，参数说明如表6-3所示。

表 6-3 同步集成配置参数说明

参数名称	参数说明
签名密钥	用于对回调消息体内容签名。
加密密钥	用于传输ClientSecret的加密公钥。
摘要算法	选择加密算法，如：SHA256或SHA1。 <b>须知</b> SHA1安全强度不高，不建议使用。

- c. 选择需要同步的信息，并开启同步按钮。  
工具类型为发布通知时，可以同步版本发布成功信息，该信息同步按钮默认开启，且不能关闭。
- d. 开启同步按钮后，需要配置对应的接口信息，配置完成后，单击“保存”。

#### 📖 说明

- 配置的接口需要参考[工具链同步信息接口规格说明](#)实现。
- 支持IPV4配置或域名配置，域名地址限制https、http开头，例如https://example.com。
- 同步租户信息接口需填写域名+/sync/v1/tenant部分内容。

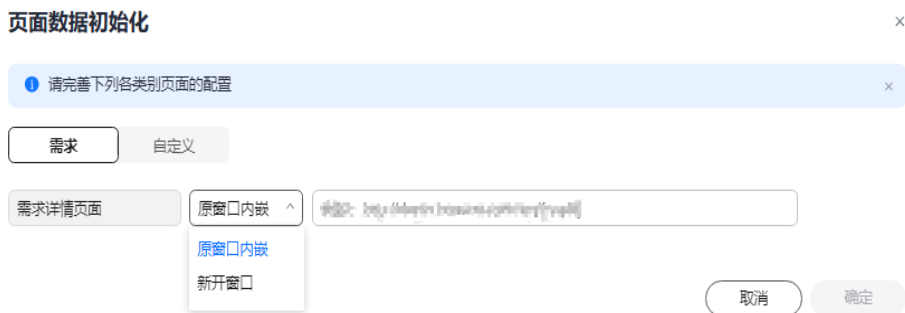
同步集成配置后，同步数据给第三方的日志可在“同步日志”模块查看，可以通过接口类型及同步结果进行筛选。

- 页面集成：通过页面集成配置，可以将第三方系统的页面集成到开发平台。
  - a. 单击“页面数据初始化”。
  - b. 选择页面集成方式并配置页面地址，如图6-2所示。

发布通知及其他类型工具无需进行页面数据初始化，如需添加自定义页面，可单击“新增页面”添加。

其余不同类型工具显示不同的配置页面，可切换类型完成页面配置。如需集成其他页面，可单击“自定义”，切换至“自定义”页签添加其他页面。

图 6-2 页面数据初始化



- c. 配置完成后单击“确定”。
- 接口集成：通过接口集成配置，可以将第三方系统的数据集成到开发平台。
  - a. 配置接口地址。

发布通知类型工具接口在“同步集成”页面配置，测试及其他类型工具无需配置接口集成，其余不同类型工具显示不同的配置接口。

#### 📖 说明

- 配置的接口需要参考[工具链同步信息接口规格说明](#)实现。
  - 支持IPV4配置或域名配置，域名地址限制https、http开头，例如https://example.com。
- b. 配置完成后单击“保存”。

#### 步骤4 发布工具。

单击工具卡片右下角的“发布”，发布该工具。

工具发布后，可以在“工具链 > 菜单管理”页面，添加菜单并为菜单关联该工具集成的页面，具体操作请参见[管理开发中心系统菜单](#)。可以在“工具链 > 版本活动管理”页面，为版本活动关联该工具集成的接口，具体操作请参见[管理开发中心版本活动](#)。

----结束

## 更多操作

您还可以对添加的工具进行如[表6-4](#)所示的操作。

表 6-4 更多操作

操作	说明
查看或修改工具配置信息	单击工具卡片，进入工具详情页面，查看或修改工具配置的基础信息。 <b>说明</b> 仅“草稿”状态的工具支持修改配置信息。
下架工具	已发布工具可以单击工具卡片上的“下架”，下架后工具变为“草稿”状态。
删除工具	“草稿”状态的工具可以单击工具卡片上的“更多 > 删除”，删除该工具。

## 6.2 管理开发中心系统菜单

AppStage开发中心提供菜单管理，可以修改系统菜单关联的工具链或对系统菜单进行上线下线操作，也可以添加菜单并为菜单关联在“工具链集成配置”页面已集成的页面。

### 前提条件

已获取组织管理员角色权限，权限申请方法请参见[申请权限](#)。

### 管理系统菜单

在开发中心首页右上角选择“工具链 > 菜单管理”。

- 修改系统菜单关联的工具链：  
当前仅概览、需求管理、代码仓管理、流水线管理和缺陷管理菜单支持修改关联的工具链。
  - a. 在“菜单管理”页面左侧菜单目录中，单击目标菜单。
  - b. 在目标菜单页面工具链配置模块，单击工具链后的“编辑”。
  - c. 在“修改工具链”页面，选择工具链，可选工具链为对应类型已发布状态的工  
具链，然后单击“确认”。
- 上线/下线系统菜单：  
当前仅需求管理、代码仓管理、缺陷管理及测试（包括子菜单）菜单支持上线/下  
线操作。
  - a. 在“菜单管理”页面左侧菜单目录中，单击目标菜单。
  - b. 在目标菜单页面，单击右上角的“上线”或者“下线”。

### 添加菜单组及子菜单

**步骤1** 在开发中心首页右上角选择“工具链 > 菜单管理”。

**步骤2** 添加菜单组。

1. 在“菜单管理”页面左侧，单击“添加菜单组”，如[图6-3](#)所示。

图 6-3 添加菜单组



2. 修改菜单组图标及名称。
  - （可选）单击图标上的“点击修改”，可以选择菜单组图标或上传图标。
  - 菜单组名称最多支持输入16个字符。
3. 单击“确认”。

**步骤3** 添加子菜单。

1. 在已添加的菜单组页面，单击右上角的“添加子菜单”。
2. 修改子菜单名称，子菜单名称最多支持输入16个字符。
3. 在基本信息模块，授权角色下拉框选择需要为子菜单授权的角色。
4. 在工具链页面配置模块，单击“添加工具链配置”，在“添加工具链”页面，选择该菜单需要关联的工具链页面，单击“确认”。
5. 单击“确认”。在提示页面，选择“上线”，或者“暂不上线”。  
菜单上线后授权角色成员才能查看并操作该菜单页面。

----结束

## 添加菜单

**步骤1** 在开发中心首页右上角选择“工具链 > 菜单管理”。

**步骤2** 在“菜单管理”页面，单击右上角“添加菜单”。

**步骤3** 修改菜单图标及名称。

- （可选）单击图标上的“点击修改”，可以选择菜单图标或上传图标。
- 菜单名称最多支持输入16个字符。

**步骤4** 在基本信息模块，授权角色下拉框选择需要为菜单授权的角色。

**步骤5** 在工具链页面配置模块，单击“添加工具链配置”，在添加工具链页面，选择该菜单需要关联的工具链页面，单击“确认”。

**步骤6** 单击“确认”，在提示页面，选择“上线”，或者“暂不上线”。

菜单上线后授权角色成员才能查看并操作该菜单页面。



----结束

## 更多操作

您还可以对添加的菜单/菜单组/子菜单进行如表6-5所示的操作。



表 6-5 更多操作

操作	说明
上线菜单	单击待上线菜单页面的“上线”。
下线菜单	单击已上线菜单单页面的“下线”。
编辑菜单	1. 在待上线的菜单/子菜单页面，或者菜单组页面，单击“编辑”。 2. 编辑信息，然后单击“确认”。
调整菜单顺序	在“菜单管理”页面左侧菜单目录中，单击菜单后的  并拖拽，调整菜单顺序。 <b>说明</b> <ul style="list-style-type: none"><li>• 添加的菜单组和菜单，依次追加在菜单目录底部。</li><li>• 仅支持同级的菜单调整。</li></ul>
删除菜单	在“菜单管理”页面左侧菜单目录中，单击菜单后的  ，删除菜单。 <b>说明</b> <ul style="list-style-type: none"><li>• 删除菜单组会同步删除该菜单组下的所有子菜单。</li><li>• 仅支持删除已下线的菜单/子菜单，菜单组无限制。</li></ul>

## 6.3 管理开发中心版本活动

### 6.3.1 添加自定义版本活动

开发中心已预置了固定的版本活动，同时支持添加其他版本活动，并为版本活动关联已发布的版本活动类型的工具链。

#### 前提条件

已获取组织管理员角色权限，权限申请方法请参见[申请权限](#)。

#### 添加版本活动

**步骤1** 在开发中心首页右上角选择“工具链 > 版本活动管理”。

**步骤2** 在“版本活动管理”页面，单击右上角“添加活动”。

**步骤3** 在“添加版本活动”页面，配置如[表6-6](#)所示参数。

表 6-6 添加版本活动参数说明

参数名称	参数说明
版本活动名称	自定义版本活动名称，最大长度为32，且不能与已有的版本活动名称重复。

参数名称	参数说明
所属版本活动阶段	选择所属的版本活动阶段，支持选择持续规划、持续开发。
是否必选	默认为必选的版本活动。
是否默认激活	默认为激活状态。
版本活动描述	输入版本活动描述，最多支持输入200个字符。
关联工具链	选择需要关联的工具链，即已添加并且已发布的版本活动类型的工具链，具体操作请参见 <a href="#">集成工具链到开发中心</a> 。



**步骤4** 单击“确认”。

----结束

## 更多操作

您还可以对添加的版本活动进行如[表6-7](#)所示的操作。

**表 6-7** 更多操作

操作	说明
修改版本活动	<ol style="list-style-type: none"><li>单击添加的版本活动卡片，进入版本活动详情页面。</li><li>单击版本活动名称或描述后的，修改基础信息。</li><li>单击已关联的工具链卡片，选择需要关联的工具链，单击“确定”。</li></ol>
删除版本活动	<ol style="list-style-type: none"><li>将鼠标放置在添加的版本活动卡片上，单击。</li><li>在提示页面，输入“确定删除”，单击“确定”。</li></ol>

### 6.3.2 修改系统预置版本活动关联的工具链

持续规划、持续开发阶段预置的版本活动默认关联CodeArts，持续部署发布阶段预置的版本活动默认对接运维中心WiseEye，开发中心支持修改系统预置版本活动关联的工具链。

#### 前提条件

已获取组织管理员角色权限，权限申请方法请参见[申请权限](#)。

#### 修改关联工具链

**步骤1** 在开发中心首页右上角选择“工具链 > 版本活动管理”。

**步骤2** 在“版本活动管理”页面，单击系统预置版本活动，进入版本活动详情页面。

### 步骤3 修改关联工具链。



- 持续规划、持续开发阶段预置的版本活动：单击已关联的工具链卡片，如图6-4所示，显示“添加工具链”页面， 选择工具链，并单击“确定”。

图 6-4 版本活动详情



- 持续部署发布阶段预置的版本活动：单击关联工具链后的  ，选择工具链，可选择多个，然后单击“确定”。

#### 说明

仅支持选择已发布的同类型的工具链，如需求规划可以选择已发布的需求类型的工具链。

----结束

## 6.3.3 添加版本交付件模板

组织管理员可以为交付件规划版本活动添加交付件模板，添加后创建的版本规划交付件时可以使用这些模板添加交付件。

### 前提条件

已获取组织管理员角色权限，权限申请方法请参见[申请权限](#)。

### 添加交付件模板

**步骤1** 在开发中心首页右上角选择“工具链 > 版本活动管理”。

**步骤2** 在“版本活动管理”页面，单击“交付件规划”卡片。

**步骤3** 单击“模板配置”，切换至“模板配置”页签。

**步骤4** 单击“添加交付件”，如**图6-5**所示。

**图 6-5** 添加交付件



**步骤5** 在“添加交付件”页面添加交付件模板，参数说明请参见**表6-8**。

**表 6-8** 添加交付件模板参数说明

参数名称	参数说明
交付件类型	输入交付件的类型。
建议处理角色（可选）	设置建议处理交付件的人员角色，可选择的处理角色包括项目经理、开发人员和测试人员。
上传模板（可选）	单击“添加文件”可选择本地文档模板进行上传。 <b>说明</b> 用户需对自己上传文件的安全风险负责，开发中心不对用户自己上传的文件做任何处理。
交付件说明（可选）	交付件的相关描述或备注说明。

**步骤6** 单击“确定”。

----结束

## 更多操作

组织管理员还可以执行如下**表6-9**所示的操作。

**表 6-9** 更多操作

操作	说明
编辑交付件模板	<ol style="list-style-type: none"> <li>在交付件模板列表中，单击“操作”列的“编辑”。</li> <li>在“编辑交付件”页面，可编辑交付件名称、建议处理角色、模板和交付件说明。</li> <li>单击“确定”。</li> </ol>

操作	说明
删除交付件模板	1. 在交付件模板列表中，单击“操作”列的“删除”。 2. 单击“确认”。

### 6.3.4 添加版本代码安全检视检查项

组织管理员可以为代码安全检视版本活动添加检查项，添加后创建的版本代码安全检视会继承该检查项配置。

#### 前提条件

已获取组织管理员角色权限，权限申请方法请参见[申请权限](#)。

#### 添加检查项

- 步骤1** 在开发中心首页右上角选择“工具链 > 版本活动管理”。
- 步骤2** 在“版本活动管理”页面，单击“代码安全检视”卡片。
- 步骤3** 单击“检查项配置”，切换至“检查项配置”页签。
- 步骤4** 单击“添加检查项”。
- 步骤5** 在“添加检查项”页面添加检查项。
  - 新建问题：输入TOPN问题和TOPN检查项，如[图6-6](#)所示，然后单击“确定”，新增问题及对应检查项。
  - 关联问题：选择已有的TOPN问题，并输入TOPN检查项，然后单击“确定”，为已有问题关联新的检查项。

图 6-6 添加检查项



----结束

## 更多操作

组织管理员还可以执行如下表6-10所示的操作。

表 6-10 更多操作

操作	说明
编辑检查项	<ol style="list-style-type: none"><li>1. 在检查项列表中，单击“操作”列的“编辑”。</li><li>2. 在“编辑检查项”页面，可编辑TOPN问题和TOPN检查项。</li><li>3. 单击“确定”。</li></ol>
删除检查项	<ol style="list-style-type: none"><li>1. 在检查项列表中，单击“操作”列的“删除”。</li><li>2. 单击“确认”。</li></ol>

## 6.3.5 管理版本标准发布准入配置

组织管理员可以为标准发布版本活动配置发布准入检查项，配置后创建的版本标准发布时会继承该准入配置。

### 前提条件

已获取组织管理员角色权限，权限申请方法请参见[申请权限](#)。

## 修改发布准入配置

**步骤1** 在开发中心首页右上角选择“工具链 > 版本活动管理”。

**步骤2** 在“版本活动管理”页面，单击“标准发布（基线版本）”卡片或“标准发布（补丁版本）”卡片。

**步骤3** 单击“发布准入配置”，切换至“发布准入配置”页签。

在发布准入配置的检查项列表中，不同类型版本显示不同的检查项，包括系统预置检查项和自定义检查项。自定义检查项来源于添加的自定义版本活动，并且不同类型版本系统预置的检查项不同。

- 基线类型的版本相应的检查项如下：

- 遗留DI值评估
- 需求状态评估
- 版本发布包检测
- 内建质量
- 测试评估
- 代码安全检视
- 交付件管理

- 补丁类型的版本相应的检查项如下：

- 版本发布包检测
- 内建质量
- 测试评估
- 代码安全检视
- 补丁规划缺陷
- 交付件管理
- 遗留DI值评估

**步骤4** 修改发布准入配置。

- 启用/关闭检查项：单击检查项后的是否启用开关，可以开启或者关闭检查项。关闭后，版本标准发布时不会检查该项。

版本发布包检测默认开启，不允许关闭。

- 编辑检查项：单击单击检查项后的“编辑”，可以修改预置的遗留DI值评估检查项的DI值，以及自定义检查项的名称，如图6-7所示，然后单击“保存”。

图 6-7 编辑检查项



----结束

## 6.4 工具链同步信息接口规格说明

本章节介绍开发中心工具链相关接口规格说明，可以参考对应的接口规格来实现具体的接口，并将接口配置在开发中心工具链集成页面，实现数据同步集成。

### 同步集成接口规格

参考如下接口规格实现接口并配置工具链同步集成，可以将开发中心中的数据（租户、产品、服务、微服务、版本）以及版本发布成功信息同步给配置的第三方系统。

### 同步租户信息

**租户信息同步接口URI:** POST /sync/v1/tenant/produceAPI/tenantSync

请求Body参数:

```
{
  "instanceId": "hua*****23456",           #实例ID
  "orderId": "CS1906666666ABCDE",          #订单号
  "tenantId": "68cbc86*****880d92f36422fa0e", #租户ID
  "tenantCode": "huawei",                   #租户标识
  "name": "huaweitest",                    #租户名称
  "domainName": "https://example.tenantaccount.com", #域名
  "flag": 1,                                #0: 删除, 1: 新增, 2: 修改
  "testFlag": 0,                            #0: 生产正式数据, 1: 调测数据
  "timeStamp": "20220413093539534",        #时间戳
}
```

**租户应用信息同步接口URI:** POST /sync/v1/tenant/produceAPI/applicationSync

请求Body参数:

```
{
  "instanceId": "hua*****23456",           #实例ID
  "appId": "KSIDJF1234456",                #应用的ID
  "tenantId": "68cbc86*****b880d92f36422fa0e", #租户ID
  "clientId": "bc20*****880d92f3",       #客户端id
  "clientSecret": "DSDT*****ECDD#$^#",   #客户端secret
  "flag": 1,                                #0: 删除, 1: 新增, 2: 修改
  "testFlag": 0,                            #0: 生产正式数据, 1: 调测数据
  "timeStamp": "20220413093539534",        #时间戳
}
```

**租户应用授权信息同步接口URI:** POST /sync/v1/tenant/produceAPI/authSync

请求Body参数:

```
{
  "instanceId": "hua*****23456",           #实例ID
  "tenantId": "68cbc86ab*****92f36422fa0e", #租户ID
  "appId": "KSID*****34456",             #应用的ID
  "userList": "[{ userName:'zhangsan01', name:'张三', position:'系统管理员', orgCode:'123456789', role:'admin' enable:'true'}, { userName:'lisi', name:'李四', position:'用户', orgCode:'123456789', role:'user' enable:'true'}]", #授权用户信息, 包含userName:账号名、name:姓名、position:职位、orgCode:部门、role:角色 ( user-普通用户, admin-管理员 )、enable:账号状态 ( true-启用, false-禁用 )
  "currentSyncTime": "20220413093539534", #时间戳
  "flag": 1,                                #0: 删除, 1: 新增, 2: 修改
  "testFlag": 0,                            #0: 生产正式数据, 1: 调测数据
  "timeStamp": "20220413093539534",        #时间戳
}
```



**组织部门信息同步（增量）接口URI:** POST /sync/v1/tenant/produceAPI/singleOrgSync

请求Body参数:

```
{
  "instanceId": "hua*****23456",          #实例ID
  "tenantId": "68cbc*****0d92f36422fa0e", #租户ID
  "orgCode": "10000",                       #部门代码
  "orgName": "开发部",                      #部门名称
  "parentCode": "",                         #上级部门的orgCode, 如果是顶层部门, 这个值为空
  "flag": 1,                                #0: 删除, 1: 新增, 2: 修改
  "testFlag": 0,                            #0: 生产正式数据, 1: 调测数据
  "timeStamp": "20220413093539534",        #时间戳
}
```

**组织部门信息同步（全量）接口URI:** POST /sync/v1/tenant/produceAPI/allOrgSync

请求Body参数:

```
{
  "instanceId": "hua*****t123456",        #实例ID
  "tenantId": "68cbc8*****b880d92f36422fa0e", #租户ID
  "orgInfo": "[{orgCode:"00000001", name:"产品部", parentCode:"265789314"},{orgCode:"00000002",
name:"测试部", parentCode:"265789314"}]", #组织内部门层级信息, 包含部门代码、部门名称、上级部门Code
  "flag": 1,                                #0: 删除, 1: 新增, 2: 修改
  "testFlag": 0,                            #0: 生产正式数据, 1: 调测数据
  "timeStamp": "20220413093539534",        #时间戳
}
```

响应示例:

```
{
  "resultCode": "000000",
  "resultMsg": "Success"
}
```

## 同步产品信息

**接口URI:** POST /sync/v1/product

请求Body参数:

```
{
  "product": {
    "product_id": "string", #产品标识
    "dept_id": "string", #产品关联的部门标识
    "product_name": "string", #产品名称
    "product_code": "string", #产品编码 (产品英文名)
    "status": "string" #状态
  },
  "op_type": "add", #操作类型, add新增、modify修改、delete删除
  "time_stamp": "string" #时间戳
}
```

响应示例:

```
{
  "resultCode": "000000",
  "resultMsg": "Success"
}
```

## 同步服务信息

**接口URI:** POST /sync/v1/service

请求Body参数:

```
{
  "service": {
    "service_id": "string", #产品标识
    "dept_id": "string", #归属的部门标识
    "product_id": "string", #归属的产品标识
    "service_name": "", #服务名称
    "service_code": "", #服务编码（服务英文名）
    "status": "string" #状态
  },
  "op_type": "add", #操作类型，add新增、modify修改、delete删除
  "time_stamp": "string" #时间戳
}
```

响应示例:

```
{
  "resultCode": "000000",
  "resultMsg": "Success"
}
```

## 同步微服务信息

接口URI: POST /sync/v1/micservice

请求Body参数:

```
{
  "micservice": {
    "micservice_id": "string", #微服务标识
    "product_id": "string", #归属的产品标识
    "service_id": "string", #归属的服务标识
    "dept_id": "string", #归属的部门标识
    "micservice_name": "", #微服务名称
    "micservice_code": "", #微服务编码（微服务英文名）
    "status": "string" #状态
  },
  "op_type": "add", #操作类型，add新增、modify修改、delete删除
  "time_stamp": "string" #时间戳
}
```

响应示例:

```
{
  "resultCode": "000000",
  "resultMsg": "Success"
}
```

## 同步版本信息

接口URI: POST /sync/v1/version

请求Body参数:

```
{
  "version": {
    "version_id": "string", #版本标识
    "service_id": "string", #归属的服务标识
    "product_id": "string", #归属的产品标识
    "dept_id": "string", #归属的部门标识
    "version_name": "string", #版本名称
    "status": "string", #状态
    "begin_date": "2024-09-25", #计划开始日期
    "end_date": "2024-09-25" #计划完成日期
  },
}
```

```
"op_type": "add",          #操作类型, add新增、modify修改、delete删除  
"time_stamp": "string"    #时间戳  
}
```

响应示例:

```
{  
  "resultCode": "000000",  
  "resultMsg": "Success"  
}
```

## 发布成功后通知

接口URI: POST /release/v1/notify

请求Body参数:

```
{  
  "service_id": "string",      #服务标识  
  "version_id": "string",      #服务标识  
  "release_type": "string",    #发布类型  
  "release_time": "2024-10-10T11:34:40.531Z", #发布时间  
  "artifacts": [              #发布制品列表  
    {  
      "build_no": "1",          #构建任务编号  
      "name": "javaMavenDemo-1.0.jar", #制品名称  
      "file_size": "2.18KB",     #制品大小  
      "hash_code": [           #哈希码  
        {  
          "hash_type": "SHA256", #哈希算法  
          "hash_value": "0fce55424a0514443f0bd612dcb25abb8de9f06572a2050ba8c6483352d5c5bd" #哈希  
        }  
      ],  
      "upload_target": "artifact", #存放平台类型  
      "upload_location": "string", #存放位置: SWR的组织/OBS桶/服务名  
      "artifact_uri": "/Build-TestBuildCheck-749/20240311.1/" #制品路径  
    }  
  ]  
}
```

## 需求类接口规格

参考如下接口规格实现需求类接口并配置工具链接口集成, 可以将第三方系统的接口数据集成到开发中心。

## 查询项目下需求概览

接口URI: GET /req/v1/statistic

Query参数: service\_id #服务标识

version\_id #版本标识

响应示例:

```
{  
  "closed_num": 1,          #已关闭数量  
  "new_num": 1,            #新建的数量  
  "process_num": 1,        #开发中的数量  
  "rejected_num": 1,       #已拒绝数量  
  "solved_num": 7,         #已解决数量  
  "test_num": 1,           #测试中的数量  
  "total": 12              #总数量  
}
```

## 高级搜索需求列表

接口URI: POST /req/v1/list

请求Body参数:

```
{
  "service_id": "28f61af50fc9452aa0ed5ea25c3cc3d3", #服务标识
  "version_name": "MyDemo", #版本标识
  "req_id": "this is a demo", #需求编号
  "req_type": "string", #需求类型
  "req_title": "string", #需求标题
  "current_handler": "string", #当前处理人
  "status": "string", #需求状态
  "limit": "string", #每页数量, 最小值: 1, 最大值: 100
  "offset": "string" #偏移量, offset是limit的整数倍数, limit=10,offset=0,10,20..., 最小值: 0
}
```

响应示例:

```
{
  "data": [ #需求信息列表
    {
      "req_id": "7894", #需求编号
      "current_handler": "张三", #当前处理人
      "create_time": "2020-09-13 14:40:11", #创建时间
      "create_user": "张三", #创建人
      "expectedEndTime": "2020-09-14", #计划完成时间
      "req_title": "MOCK需求数据", #需求标题
      "priority": "1", #优先级
      "severity": "10", #严重程度
      "status": "5", #需求状态
      "req_type": "7", #需求类型
      "req_description": "需求描述", #需求内容描述
      "version_name": "111" #归属版本
    }
  ],
  "total": 1 #总数量
}
```

## 创建需求

接口URI: POST /req/v1/create

请求Body参数:

```
{
  "req_title": "新需求", #需求标题
  "req_type": "7", #需求类型
  "version_name": "3.3.3.3", #归属版本
  "req_description": "需求描述", #需求内容描述
  "expected_end_time": "2024-09-13", #计划完成时间
  "current_handler": "张三", #当前处理人
  "create_user": "李四", #创建人
  "create_time": "2024-09-13 10:00:00" #创建时间
}
```

响应示例:

```
{
  "req_id": "28f61af50fc9452aa0ed5ea25c3cc3d3", #需求编号
  "req_type": "7", #需求类型
  "req_title": "新需求", #需求标题
  "version_name": "3.3.3.3", #归属版本
  "status": "1", #需求状态
  "req_description": "需求描述", #需求内容描述
  "expected_end_time": "2024-09-13", #计划完成时间
  "current_handler": "张三", #当前处理人
}
```

```
"create_user": "李四",      #创建人  
"create_time": "2024-09-13 10:00:00"  #创建时间  
}
```

## 查询需求详情

接口URI: GET /req/v1/detail/{req\_id}

路径参数: req\_id #需求ID

响应示例:

```
{  
  "req_id": "28f61af50fc9452aa0ed5ea25c3cc3d3", #需求编号  
  "req_type": "7", #需求类型  
  "req_title": "需求", #需求标题  
  "version_name": "5.5.5.5", #归属版本  
  "status": "1", #需求状态  
  "req_description": "需求描述", #需求内容描述  
  "expected_end_time": "2024-09-13", #计划完成时间  
  "current_handler": "张三", #当前处理人  
  "create_user": "李四", #创建人  
  "create_time": "2024-09-13 08:50:52" #创建时间  
}
```

## 更新需求迭代

接口URI: POST /req/v1/update

请求Body参数:

```
{  
  "issue_item_id": "string",  
  "service_id": "string", #服务标识  
  "version_name": "string" #归属版本  
}
```

## 缺陷类接口规格

参考如下接口规格实现缺陷类接口并配置工具链接口集成, 可以将第三方系统的缺陷数据集成到开发中心。

## 查询项目下缺陷概览

接口URI: GET /defect/v1/statistic

Query参数: service\_id #服务标识

version\_id #版本标识

响应示例:

```
{  
  "critical_num": 1, #致命问题数  
  "defect_index": 1, #DI值  
  "normal_num": 3, #一般问题数  
  "serious_num": 2, #严重问题数  
  "tip_num": 1, #提示问题数  
  "total": 7 #总数  
}
```

## 高级搜索缺陷列表

接口URI: POST /defect/v1/list

请求Body参数:

```
{
  "offset": 0,           #偏移量, offset是limit的整数倍数, limit=10,offset=0,10,20..., 最小值: 0
  "limit": 10,          #每页数量, 最小值: 1, 最大值: 100
  "service_id": "28f61af50fc9452aa0ed5ea25c3cc3d3", #服务标识
  "defect_title": "标题", #缺陷标题
  "version_name": "1.1.1.1", #版本标识
  "current_handler": "张三", #当前处理人
  "status": [           #缺陷状态
    "1",
    "2"
  ],
  "defect_id": [        #缺陷编号
    "123456",
    "123457"
  ]
}
```

响应示例:

```
{
  "data": [
    {
      "defect_id": "7894", #缺陷编号
      "current_handler": "张三", #当前处理人
      "created_time": "2020-09-13 14:40:11", #创建时间
      "createUser": "张三", #创建人
      "expectedEndTime": "2020-09-14", #计划完成时间
      "defect_title": "MOCK缺陷数据", #缺陷标题
      "priority": "1", #优先级
      "severity": "10", #严重程度
      "status": "5", #需求状态
      "defect_description": "缺陷描述", #缺陷内容描述
      "version_name": "111" #归属版本
    }
  ],
  "total": 1 #总数量
}
```

## 创建缺陷

接口URI: POST /defect/v1/create

请求Body参数:

```
{
  "defect_title": "缺陷标题", #缺陷标题
  "version_name": "6.6.6.6", #归属版本
  "defect_description": "缺陷描述", #缺陷内容描述
  "expected_end_time": "2024-09-13", #计划完成时间
  "current_handler": "张三", #当前处理人
  "create_user": "李四", #创建人
  "create_time": "2024-09-13 08:53:50" #创建时间
}
```

响应示例:

```
{
  "defect_id": "28f61af50fc9452aa0ed5ea25c3cc3d3", #缺陷编号
  "defect_title": "缺陷标题", #缺陷标题
  "version_name": "6.6.6.6", #缺陷标题
  "priority": "1", #优先级
  "severity": "10", #严重程度
}
```

```
"status": "1", #缺陷状态
"defect_description": "缺陷描述", #缺陷内容描述
"expected_end_time": "2024-09-13", #计划完成时间
"current_handler": "张三", #当前处理人
"create_user": "李四", #创建人
"create_time": "2024-09-13 08:53:50" #创建时间
}
```

## 查询缺陷详情

接口URI: GET /defect/v1/detail/{defect\_id}

路径参数: defect\_id #缺陷标识

响应示例:

```
{
  "defect_id": "28f61af50fc9452aa0ed5ea25c3cc3d3", #缺陷编号
  "defect_title": "缺陷标题", #缺陷标题
  "version_name": "9.9.9.9", #归属版本
  "status": "1", #缺陷状态
  "defect_description": "缺陷描述", #缺陷内容描述
  "expected_end_time": "2024-09-13", #计划完成时间
  "current_handler": "张三", #当前处理人
  "priority": "1", #优先级
  "severity": "10", #严重程度
  "create_user": "李四", #创建人
  "create_time": "2024-09-13 08:50:52" #创建时间
}
```

## 流水线类接口规格

参考如下接口规格实现流水线类接口并配置工具链接口集成, 可以将第三方系统的流水线数据集成到开发中心。

## 启动流水线

接口URI: POST /pipeline/v1/run/{pipeline\_id}

路径参数: pipeline\_id #流水线标识

Query参数: service\_id #服务标识

version\_id #版本标识

请求Body参数:

```
{
  "variables": { #启动参数
    "additionalProp1": "string",
    "additionalProp2": "string",
    "additionalProp3": "string"
  }
}
```

响应示例:

```
{
  "pipeline_run_id": "ad6b7f66283a45be9c1f82b06c831e59" #流水线运行实例ID
}
```

## 停止流水线

接口URI: POST /pipeline/v1/stop/{pipeline\_id}/{pipeline\_run\_id}

路径参数: pipeline\_id #流水线标识  
pipeline\_run\_id #流水线运行实例标识  
Query参数: service\_id #服务标识  
version\_id #版本标识

响应示例:

```
{  
  "success": true  
}
```

## 获取流水线执行详情

接口URI: GET /pipeline/v1/instance/{pipeline\_id}/{pipeline\_run\_id}/detail

路径参数: pipeline\_id #流水线标识  
pipeline\_run\_id #流水线运行实例标识  
Query参数: service\_id #服务标识  
version\_id #版本标识

响应示例:

```
{  
  "pipeline_id": "string",      #流水线ID  
  "pipeline_name": "string",    #流水线名称  
  "executor_id": "string",      #执行人ID  
  "executor_name": "string",    #执行人名称  
  "status": "COMPLETED",      #状态, COMPLETED: 已完成, RUNNING: 运行中, FAILED: 失败,  
  CANCELED: 取消, PAUSED: 暂停, SUSPEND: 挂起, IGNORED: 忽略  
  "trigger_type": "Manual",     #触发类型, Manual: 手动触发, Scheduler: 定时任务触发, RollBack: 回  
  滚执行。WebHook类型, CreateTag: Tag 触发, Note: Gitee仓库评论触发, Issue: Gitee 仓库Issue触发,  
  MR: 合并请求触发, Push: 代码push触发  
  "start_time": 0,              #开始时间  
  "end_time": 0,                #结束时间  
  "sources": {  
    "repo_name": "aaaaa",      #仓库名称  
    "target_branch": "master"  #分支  
  }  
}
```

## 获取流水线详情

接口URI: GET /pipeline/v1/detail/{pipeline\_id}

路径参数: pipeline\_id #流水线标识  
Query参数: service\_id #服务标识  
version\_id #版本标识

响应示例:

```
{  
  "pipeline_id": "9777b6f14dca47ab8e411512b8301698", #流水线标识  
  "pipeline_name": "Mock数据"                          #流水线名称  
}
```



## 获取流水线构建产物

接口URI: GET /pipeline/v1/instance/{pipeline\_id}/{pipeline\_run\_id}/artifacts

路径参数: pipeline\_id #流水线标识

pipeline\_run\_id #流水线运行实例标识

Query参数: service\_id #服务标识

version\_id #版本标识

响应示例:

```
{
  "data": [
    {
      "build_no": "1",          #构建任务编号
      "name": "javaMavenDemo-1.0.jar", #制品名称
      "file_size": "2.18KB",    #制品大小
      "hash_code": [          #哈希码
        {
          "hash_type": "SHA256",    #哈希算法
          "hash_value": "0fce55424a0514443f0bd612dcb25abb8de9f06572a2050ba8c6483352d5c5bd" #哈希值
        }
      ],
      "upload_target": "artifact", #存放平台类型
      "upload_location": "string", #存放位置: SWR的组织/OBS桶/服务名
      "artifact_uri": "/Build-TestBuildCheck-749/20240311.1/" #制品路径
    }
  ]
}
```

## 流水线列表

接口URI: POST /pipeline/v1/list

Query参数: service\_id #服务标识

version\_id #版本标识

请求Body参数:

```
{
  "offset": 0,          #偏移量, offset是limit的整数倍数, limit=10,offset=0,10,20..., 最小值: 0
  "limit": 10,         #每页数量, 最小值: 1, 最大值: 100
}
```

响应示例:

```
{
  "data": [
    {
      "pipeline_id": "string",    #流水线ID
      "pipeline_name": "string",  #流水线名称
      "latest_run": {
        "executor_id": "3fc4481d0e8ec8dd94800a7931b15714ef47728b",
        "executor_name": "XXX",
        "stage_status_list": [
          {
            "name": "Declarative: Tool Install",
            "sequence": 6,
            "status": "COMPLETED",
            "start_time": "2024-09-02 16:22:04",
            "end_time": "2024-09-02 16:22:04"
          }
        ]
      }
    }
  ]
}
```

```
],
  "status": "COMPLETED",          #状态, COMPLETED: 已完成, RUNNING: 运行中, FAILED: 失败,
  CANCELED: 取消, PAUSED: 暂停, SUSPEND: 挂起, IGNORED: 忽略
  "trigger_type": "Manual",        #触发类型, Manual: 手动触发, Scheduler: 定时任务触发, RollBack:
  回滚执行。WebHook类型, CreateTag: Tag 触发, Note: 仓库评论触发, Issue: 仓库Issue触发, MR: 合并请
  求触发, Push: 代码push触发
  "build_params": {
    "target_branch": "master",      #分支
    "repo_id": "x03/JenkinsAdapter", #仓库ID
    "git_url": "https://codehub-dg-g.huawei.com/x03/JenkinsAdapter.git", #扫描的代码仓地址
    "commit_id": "string"
  },
  "start_time": 1725265323466,      #开始时间
  "end_time": 1725265332313        #结束时间
}
}
},
"total": 1          #总数
}
```

## 查询最近执行的流水线列表

接口URI: POST /pipeline/v1/recent-run

Query参数: service\_id #服务标识

version\_id #版本标识

请求Body参数:

```
{
  "offset": 0,          #偏移量, offset是limit的整数倍数, limit=10,offset=0,10,20..., 最小值: 0
  "limit": 10,        #每页数量, 最小值: 1, 最大值: 100
}
```

响应示例:

```
{
  "data": [
    {
      "pipeline_id": "string",      #流水线ID
      "pipeline_name": "string",    #流水线名称
      "latest_run": {
        "executor_id": "3fc4481d0e8ec8dd94800a7931b15714ef47728b",
        "executor_name": "XXX",
        "stage_status_list": [
          {
            "name": "Declarative: Tool Install",
            "sequence": 6,
            "status": "COMPLETED", #状态, COMPLETED: 已完成, RUNNING: 运行中, FAILED: 失败,
            CANCELED: 取消, PAUSED: 暂停, SUSPEND: 挂起, IGNORED: 忽略
            "start_time": "2024-09-02 16:22:04",
            "end_time": "2024-09-02 16:22:04"
          }
        ],
        "status": "COMPLETED",     #状态, COMPLETED: 已完成, RUNNING: 运行中, FAILED: 失败,
        CANCELED: 取消, PAUSED: 暂停, SUSPEND: 挂起, IGNORED: 忽略
        "trigger_type": "Manual",   #触发类型, Manual: 手动触发, Scheduler: 定时任务触发,
        RollBack: 回滚执行。WebHook类型, CreateTag: Tag 触发, Note: 仓库评论触发, Issue: 仓库Issue触发,
        MR: 合并请求触发, Push: 代码push触发
        "build_params": {
          "target_branch": "master", #分支
          "repo_id": "x03/JenkinsAdapter", #仓库ID
          "git_url": "https://codehub-dg-g.huawei.com/x03/JenkinsAdapter.git", #扫描的代码仓地址
          "commit_id": "string"
        },
        "start_time": 1725265323466, #开始时间
        "end_time": 1725265332313    #结束时间
      }
    }
  ]
}
```

```
    }  
  },  
  ],  
  "total": 1      #总数  
}
```

## 代码检查类接口规格

参考如下接口规格实现代码检查类接口并配置工具链接口集成，可以将第三方系统的代码检查数据集成到开发中心。

### 查询代码检查信息

接口URI: GET /pipeline/v1/instance/{pipeline\_id}/{pipeline\_run\_id}/code-check

路径参数: pipeline\_id #流水线标识

pipeline\_run\_id #流水线运行实例标识

Query参数: service\_id #服务标识

version\_id #版本标识

响应示例:

```
{  
  "check_task_id": "28f61af50fc9452aa0ed5ea25c3cc3d3", #代码检查任务标识  
}
```

### 查询代码检查质量概要

接口URI: GET /code-check/v1/summary/{check\_task\_id}

路径参数: check\_task\_id #代码检查任务标识

Query参数: service\_id #服务标识

version\_id #版本标识

响应示例:

```
{  
  "git_url": "string", #扫描的代码仓地址  
  "git_branch": "string", #扫描的代码仓分支  
  "task_name": "string", #代码检查的任务名称  
  "last_check_time": "string", #最后检查时间  
  "is_access": true, #是否通过  
  "critical_count": "string", #致命问题个数  
  "major_count": "string", #严重问题个数  
  "minor_count": "string", #一般问题个数  
  "suggestion_count": "string", #建议问题个数  
  "cyclomatic_complexity_per_method": "string", #平均圈复杂度  
  "maximum_cyclomatic_complexity": "string", #最大圈复杂度  
  "code_duplication_ratio": "string" #代码重复率  
}
```

## 代码仓类接口规格

参考如下接口规格实现代码仓类接口并配置工具链接口集成，可以将第三方系统的代码仓数据集成到开发中心。

## 查询代码仓列表

接口URI: POST /repo/v1/list

请求Body参数:

```
{
  "limit": 10,          #每页数量, 最小值: 1, 最大值: 100
  "offset": 0,         #偏移量, offset是limit的整数倍数, limit=10,offset=0,10,20..., 最小值: 0
  "search": "string"   #代码仓名称, 模糊搜索
}
```

响应示例:

```
{
  "data": [            #仓库信息列表
    {
      "repo_id": "string",    #仓库ID
      "repo_name": "string",  #仓库名称
      "status": "string",     #仓库状态, 0: 仓库正常创建成功, 1: 仓库创建中, 2: 创建失败, 3: 仓库冻结, 4: 仓库已经关闭
      "https_url": "string",  #使用HTTPS克隆仓库时所使用的URL
      "ssh_url": "string",    #使用SSH方式克隆仓库时所使用的URL
      "web_url": "string",    #WEB URL路径, 访问它将跳转至仓库详情页
      "create_name": "string", #创建者
      "update_time": "string", #更新时间
      "create_time": "string", #创建时间
      "description": "string" #描述
    }
  ],
  "total": "string"        #总数量
}
```

## 查询代码仓详情

接口URI: GET /repo/v1/repo/{repo\_id}

路径参数: repo\_id #代码仓标识

响应示例:

```
{
  "repo_id": "string",    #仓库ID
  "repo_name": "string",  #仓库名称
  "status": "string",     #仓库状态, 0: 仓库正常创建成功, 1: 仓库创建中, 2: 创建失败, 3: 仓库冻结, 4: 仓库已经关闭
  "https_url": "string",  #使用HTTPS克隆仓库时所使用的URL
  "ssh_url": "string",    #使用SSH方式克隆仓库时所使用的URL
  "web_url": "string",    #WEB URL路径, 访问它将跳转至仓库详情页
  "create_name": "string", #创建者
  "update_time": "string", #更新时间
  "create_time": "string", #创建时间
  "description": "string" #描述
}
```

## 创建代码仓

接口URI: POST /repo/v1/create

请求Body参数:

```
{
  "repo_name": "string",  #仓库名称
  "description": "string", #仓库描述
  "service_id": "string"  #仓库所属服务
}
```

响应示例:

```
{
  "repo_id": "string",      #仓库ID
  "repo_name": "string",   #仓库名称
  "status": "string",      #仓库状态, 0: 仓库正常创建成功, 1: 仓库创建中, 2: 创建失败, 3: 仓库冻结,
                          4: 仓库已经关闭
  "https_url": "string",   #使用HTTPS克隆仓库时所使用的URL
  "ssh_url": "string",     #使用SSH方式克隆仓库时所使用的URL
  "web_url": "string",     #WEB URL路径, 访问它将跳转至仓库详情页
  "create_name": "string", #创建者
  "update_time": "string", #更新时间
  "create_time": "string", #创建时间
  "description": "string"  #描述
}
```

## 版本活动类接口规格

参考如下接口规格实现版本活动类接口并配置工具链接口集成, 可以将第三方系统的版本活动数据集成到开发中心。

### 查询版本活动状态信息

接口URI: GET /activity/v1/status/{service\_id}/{version\_id}/{activity\_code}

路径参数: activity\_code #版本活动编码

service\_id #服务标识

version\_id #版本标识

响应示例:

```
{
  "status": "init"        #状态, init: 未开始, inProgress: 进行中, abnormal: 异常/告警, completed: 已完成/已关闭
}
```

### 查询版本活动检查信息

接口URI: GET /activity/v1/check/{service\_id}/{version\_id}/{activity\_code}/  
{release\_type}

路径参数: activity\_code #版本活动编码

service\_id #服务标识

version\_id #版本标识

release\_type #发布类型, standard: 标准发布, test: 调测发布

响应示例:

```
{
  "check_result": true,   #检查结果, true: 通过, false: 不通过
  "check_rule": "string", #检查规则
  "result_desc": "string" #检查结果描述
}
```

# 7 创建并管理团队

## 7.1 在 AppStage 开发中心创建开发团队并关联服务

在开发中心进行服务开发前，需要组织管理员创建一个团队，将团队关联需要开发的服务。一个团队只能归属一个部门，可以关联一个或多个服务。

团队创建后，会有单独的团队空间为团队成员提供一个项目开发和共享的工作区域，用于协作、交流和共享资源。在团队空间中，团队成员可以根据职责及权限执行相应的操作，提高团队的工作效率和协作能力。

### 前提条件

- 已获取组织管理员角色权限，权限申请方法请参见[申请权限](#)。
- 创建团队时需绑定团队归属部门并关联服务，已创建部门和服务，具体操作请参见[添加部门](#)和[创建服务](#)。

### 创建团队

**步骤1** 在开发中心首页下方的“我的团队”区域，单击右侧“创建团队”。

**步骤2** 在“创建团队”页面，设置团队相关参数，参数配置请参见[表7-1](#)。

表 7-1 创建团队参数说明

参数名称	参数说明
团队名称	必填项，团队的命名。
团队归属部门	必填项，选择系统中已提前创建的部门。
关联服务	必填项，选择系统中已提前创建并发布的服务。
团队LOGO	必填项，单击系统默认图片上的“点击修改”，可选择本地图片自定义LOGO图片。
团队简介（可选）	非必填项，团队空间的功能描述或其他备注信息。

**步骤3** 单击“创建”。

在“我的团队”区域可查看到创建的团队的卡片。

----结束

## 更多操作

您还可以执行如下表7-2所示的操作。

表 7-2 更多操作

操作	说明
修改团队关联的服务	<ol style="list-style-type: none"><li>1. 在左侧导航栏，选择“团队管理 &gt; 基本信息”。</li><li>2. 在“关联服务”区域，单击“关联服务”。</li><li>3. 在“添加关联服务”对话框中的选择树选择需要关联的服务名称（也可输入关键字进行过滤后选择），可选择一个或多个服务。</li><li>4. 单击“确定”。</li></ol>
编辑团队	<ol style="list-style-type: none"><li>1. 在左侧导航栏，选择“团队管理 &gt; 基本信息”。</li><li>2. 单击“基本信息”区域下的“编辑”。</li><li>3. 编辑团队信息，然后单击“确定”。</li></ol>
删除团队	<ol style="list-style-type: none"><li>1. 在左侧导航栏，选择“团队管理 &gt; 基本信息”。</li><li>2. 单击“删除团队”。</li><li>3. 在提示页面输入框输入“确定删除”，然后单击“确定”。</li></ol>
关注团队	<ol style="list-style-type: none"><li>1. 单击“我的团队”区域右上角“关注团队”。</li><li>2. 在“关注团队”页面的“团队列表”中，单击需要添加的团队名称右侧的“关注”。 在“我的团队”列表中可查看到此新关注的团队卡片。</li></ol>
设置默认团队	选中“我的团队”区域内团队卡片左下角的“默认团队”，可将此团队设置为默认团队，此默认团队将置顶显示在开发中心团队空间中的团队首位。
移出团队	单击“我的团队”区域内团队卡片右下角的“移出”，可将此团队从我的团队列表移出，我的团队列表中不显示此团队。 <b>说明</b> 团队创建人不可移出自己创建的团队。

## 7.2 在 AppStage 开发中心为服务关联 CodeArts 项目

开发中心默认集成CodeArts工具链，创建团队后，组织管理员或项目经理需要为团队关联的服务关联CodeArts项目，后续服务开发过程中才能正常使用CodeArts服务的代码托管、流水线、编译构建功能；同时在服务配置页面，可查看当前服务的基础信息（如所属产品、所属部门、服务状态等）。

## 前提条件

需要具备组织管理员或项目经理角色权限，权限申请方法请参见[申请权限](#)。

## 关联 CodeArts 项目

创建团队时会为团队关联服务，可以在“服务配置”页面的“服务基础信息”区域，查看已关联服务的中文名、英文名、所属产品、归属部门、服务状态等信息。

如果当前服务已关联项目，在“服务配置”页面的“服务扩展信息”区域，可查看关联项目名称、关联项目类型。如果当前服务未关联项目，需将服务关联项目，后续才可正常使用CodeArts服务的相关功能，具体操作如下：

**步骤1** 在开发中心首页下方的“我的团队”区域，选择需要操作的团队卡片，进入该团队空间。

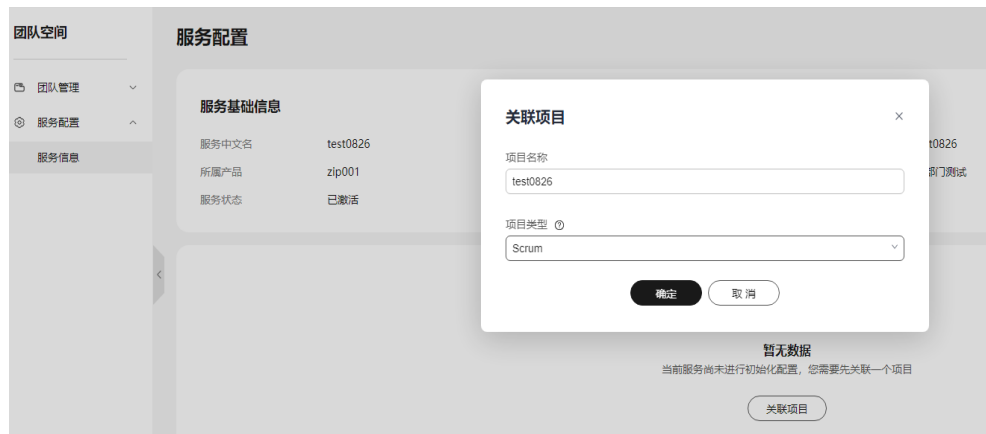
如果该团队关联多个服务，可在顶部切换团队关联的其他服务。也可在顶部切换其他团队。

**步骤2** 在左侧导航栏选择“服务配置”。

**步骤3** 在“服务配置”页面，单击“关联项目”。

**步骤4** 在“关联项目”对话框中的“项目名称”下，输入项目名称（默认为当前服务名称，可自定义修改），根据实际业务需要在“项目类型”下拉框选择项目类型（Scrum、IPD-系统设备类、IPD-独立软件类），如[图7-1](#)所示，然后单击“确定”。

图 7-1 关联项目



**步骤5** 在“关联项目”对话框对将要关联的项目名称进行二次确认，单击“确定”。

### 📖 说明

关联项目成功后不可解除关联。

- 关联项目成功：在“服务配置”页面的“服务扩展信息”区域，可查看关联项目名称、关联项目类型。
- 关联项目失败：系统检测到您尚未进行凭证（AK/SK）授权时，将弹框提示关联失败，并提示您前往AppStage管理控制台进行凭证授权，完成凭证（AK/SK）授权后，返回“服务配置”页面，重新关联项目。

----结束



## 7.3 为 AppStage 开发中心开发团队添加成员

在团队多人员协作开发过程中，不同角色成员执行各自权限和责任事项，因此需要在团队中添加协作开发的各角色成员（如项目经理、开发人员、测试人员等）；当团队中成员发生岗位变动或人员流动时，也可进行相应修改或删除，实现统一有序的成员信息管理。

### 前提条件

- 已[创建开发团队](#)。
- 已[添加开发中心用户](#)。仅开发中心用户才能被添加为团队成员。
- 已获取组织管理员或项目经理角色权限，权限申请方法请参见[申请权限](#)。

### 添加成员

**步骤1** 在开发中心首页下方的“我的团队”区域，选择需要操作的团队卡片，进入该团队空间。

如果该团队关联多个服务，可在顶部切换团队关联的其他服务。也可在顶部切换其他团队。

**步骤2** 在左侧导航栏选择“团队管理 > 基本信息”。

**步骤3** 单击页面下方“团队成员”区域右上角的“添加成员”。

**步骤4** 在“添加角色成员”对话框选择角色、成员及其权限到期日，参数说明请参见[表7-3](#)。

表 7-3 添加角色成员参数说明

参数名称	参数说明
角色	在下拉列表中选择角色名称，可选择开发人员、项目经理、测试人员或浏览者。 其中项目经理角色仅组织管理员才能添加。
成员	输入姓名或账号名搜索，在下拉列表中选择成员，可多次选择添加。
权限到期日	设置权限到期的具体日期（可快捷选择三个月、六个月或360天）。

**步骤5** 单击“确定”。新添加的成员显示在团队成员列表中。

如需移除成员，可以在团队成员列表的“操作”列单击“删除成员”，将不需要的成员移出该团队。

----**结束**

# 8 创建及规划版本

## 8.1 在 AppStage 开发中心创建版本

版本是服务开发的基本单元，是指软件、应用程序或系统有序开发而特定发布的单元。每个版本都有一个唯一的标识符，通常是一个数字或字母组合，用于区分不同的版本。为所要开发的产品/服务创建版本，在版本管理活动中进行流水线和发布流程等有序版本开发过程管理。

### 前提条件

已获取项目经理角色权限，权限申请方法请参见[申请权限](#)。

### 创建版本

- 步骤1** 在开发中心首页下方的“我的团队”区域，选择需要操作的团队卡片，进入该团队空间。  
如果该团队关联多个服务，可在顶部切换团队关联的其他服务。也可在顶部切换其他团队。
- 步骤2** 在左侧导航栏选择“版本管理”。
- 步骤3** 在“版本管理”页面，单击右上角“创建版本”。
- 步骤4** 在“创建版本”页面，设置版本的基本及配置信息，参数说明详见[表8-1](#)。

表 8-1 创建版本参数说明

参数名称		参数说明
基本及配置信息	服务	无需配置，默认为当前服务名称。
	版本号	必填项，根据界面输入框填写规划的版本号（系统将自动合成完整版本号）。示例：23.0.1.100
	版本描述（可选）	非必填项，版本信息相关的简要描述。

参数名称		参数说明
	软件类型	无需配置，默认为“服务软件”。
	版本类型	必填项，可分为两种类型： <ul style="list-style-type: none"> <li>● 基线</li> <li>● 补丁，在版本规划时需规划补丁版本缺陷，详见<a href="#">规划补丁版本缺陷</a>。</li> </ul>
	发布类型	无需配置，默认为“标准发布”。
	关联发布计划	仅服务关联了IPD类型的项目（IPD系统设备类项目、IPD独立软件类项目），且“版本类型”选择“基线”时，需配置此参数。 选择需要关联的发布计划，也可勾选“自动创建新发布计划”，自动创建的新发布计划名称默认为版本号。
	关联迭代	仅服务关联了Scrum类型的项目，且“版本类型”选择“基线”时，需配置此参数。 选择需要关联的迭代，也可勾选“自动创建新迭代”，自动创建的新迭代名称默认为版本号。
	产能	必填项，指版本开发团队每天能提供的开发人员的数量。单位：人/天。
	选择计划时间	版本开始时间
版本发布时间		必填项，版本发布的日期。 自动创建新发布计划或自动创建新迭代时，发布时间默认为设置的版本发布时间。

**步骤5** 单击“提交”。

在版本列表中可查看到此新建版本。

----结束

## 更多操作

创建版本完成后，您还可以执行如下[表8-2](#)所示的操作。

**表 8-2** 更多操作

操作	说明
查看版本详情	在“版本管理”页面的版本列表中，单击版本号可以进入版本详情页面。可以查看版本基本信息并对版本进行持续规划、持续开发、持续部署发布，不同类型版本涉及模块不同。

操作	说明
编辑版本	<p><b>说明</b> 未发布状态的版本支持编辑。</p> <ol style="list-style-type: none"><li>1. 在“版本管理”页面的版本列表中，单击“操作”列的“编辑”。</li><li>2. 在“编辑版本”页面，可编辑版本描述、产能、版本开始时间和版本发布时间。</li><li>3. 单击“提交”。</li></ol>
删除版本	<p><b>说明</b></p> <ul style="list-style-type: none"><li>• 未发布状态的版本可以删除。</li><li>• 删除版本后其名下全部内容将同步被清空，此操作无法恢复，请谨慎操作。</li></ul> <ol style="list-style-type: none"><li>1. 在“版本管理”页面的版本列表中，单击“操作”列的“删除”。</li><li>2. 在“删除版本”对话框，根据界面提示在文本框输入“确定删除”。</li><li>3. 单击“确定”。</li></ol>
归档版本	<p><b>说明</b> 已发布状态的版本可以归档。</p> <ol style="list-style-type: none"><li>1. 在“版本管理”页面的版本列表中，单击“操作”列的“归档”。</li><li>2. 单击“确定”。</li></ol>
移除归档版本	<ol style="list-style-type: none"><li>1. 在“版本管理”页面，单击右上角的“查看归档版本”。</li><li>2. 在归档版本列表中，单击“操作”列的“移除归档”。</li><li>3. 单击“确定”。</li></ol>

## 8.2 在 AppStage 开发中心规划交付件

规划交付件是指在产品版本开发启动初期对产品开发过程所需的交付件进行规划，以辅助开发过程规范有序且可追溯。组织管理员[添加版本交付件模板](#)后，版本管理规划交付件时可以使用这些模板添加交付件，也可以添加自定义类型的交付件。

### 前提条件

- 已[创建版本](#)。
- 已获取项目经理角色权限，权限申请方法请参见[申请权限](#)。

### 添加交付件

**步骤1** 在开发中心首页下方的“我的团队”区域，选择需要操作的团队卡片，进入该团队空间。

如果该团队关联多个服务，可在顶部切换团队关联的其他服务。也可在顶部切换其他团队。

**步骤2** 在左侧导航栏选择“版本管理”。

**步骤3** 在“版本管理”页面的版本列表中，单击版本号，进入版本详情页面。

**步骤4** 在版本详情页面的“版本持续交付”区域，选择“持续规划 > 交付件规划”。

**步骤5** 在“交付件规划”页面，单击右上角“添加交付件”。

**步骤6** 在“添加交付件”页面添加交付件，参数说明请参见表8-3。

表 8-3 添加交付件参数说明

参数名称	参数说明
交付件类型	选择交付件类型，即添加的版本交付件模板类型或者自定义类型，使用模板时每种类型模板可添加一个交付件，自定义类型可以添加多个。
交付件名称	输入交付件的名称。
建议处理角色（可选）	设置建议处理交付件的人员角色，可选择的处理角色包括项目经理、开发人员和测试人员。
上传模板（可选）	单击“添加文件”可选择本地文档模板进行上传。 <b>说明</b> 用户需对自己上传文件的安全风险负责，开发中心不对用户自己上传的文件做任何处理。
交付件说明（可选）	交付件的相关描述或备注说明。

**步骤7** 单击“确定”。

----结束

## 更多操作

项目经理还可以执行如下表8-4所示的操作。

表 8-4 更多操作

操作	说明
编辑交付件	1. 在交付件列表中，单击“操作”列的“编辑”。 2. 在“编辑交付件”页面，可编辑交付件名称、建议处理角色、模板和交付件说明。 3. 单击“确定”。
删除交付件	1. 在交付件列表中，单击“操作”列的“删除”。 2. 单击“确认”。

## 8.3 在 AppStage 开发中心规划补丁版本缺陷

针对创建的版本类型为补丁类型的版本，在版本规划阶段，需要进行交付件规划和补丁规划，以对版本进行细粒度功能分解和开发规划，进而确保开发过程规范有序和高效进行。

## 前提条件

- 已[创建版本](#)。
- 已创建缺陷，创建缺陷的操作方法请参见[创建缺陷](#)。
- 已获取项目经理角色权限，权限申请方法请参见[申请权限](#)。

## 添加缺陷

**步骤1** 在开发中心首页下方的“我的团队”区域，选择需要操作的团队卡片，进入该团队空间。

如果该团队关联多个服务，可在顶部切换团队关联的其他服务。也可在顶部切换其他团队。

**步骤2** 在左侧导航栏选择“版本管理”。

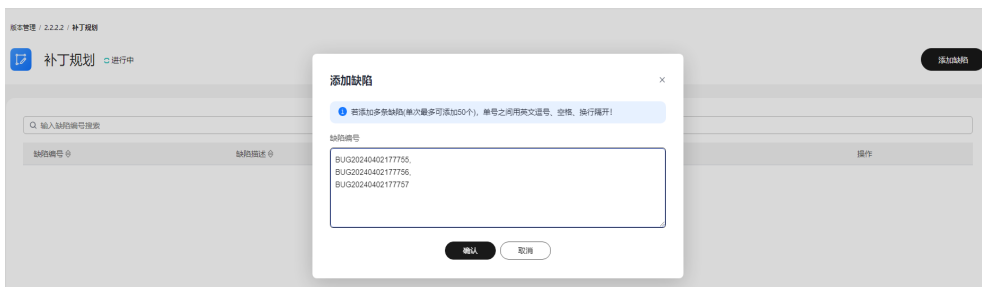
**步骤3** 在“版本管理”页面的版本列表中，单击版本号，进入版本详情页面。

**步骤4** 在版本详情页面的“版本持续交付”区域，选择“持续规划 > 补丁规划”。

**步骤5** 在“补丁规划”页面，单击右上角“添加缺陷”。

**步骤6** 在“添缺陷求”对话框，输入缺陷编号（单次最多可添加50个，单号之间用英文逗号、空格、换行隔开），如[图8-1](#)所示。

图 8-1 添加缺陷



**步骤7** 单击“确认”。

新添加的缺陷编号显示在缺陷列表中。

----结束

## 8.4 在 AppStage 开发中心规划版本需求

需求管理是开发中心集成[需求管理（CodeArts Req）](#)服务的能力，是华为多年研发实践沉淀的需求管理与团队协作服务，内置多种开箱即用的场景化需求模型。用户可以创建版本需求，或者为版本关联需求，完成版本需求规划。

本章节介绍如何使用开发中心集成的CodeArts Req服务能力创建并规划需求，如果用户需要使用其他需求管理工具，可以将工具集成进开发中心，具体操作请参见[集成工具链到开发中心](#)。

- 创建版本需求：不同项目类型，创建需求操作不同。
  - Scrum项目：在开发中心直接创建Story类型需求，创建后数据会同步至CodeArts Req服务中，具体操作请参见[创建Scrum项目版本需求](#)。如果在

[CodeArts Req](#)页面创建需求，数据会同步至开发中心需求管理列表，若需求已设置迭代，数据会按照迭代关联的版本，自动同步至该版本的需求规划列表中。

- IPD系统设备类项目和IPD独立软件类项目：开发中心集成CodeArts Req服务页面，可以通过开发中心的“需求管理”进入并创建需求，操作可以参考[创建IPD系统设备类项目需求](#)和[创建IPD独立软件类项目需求](#)。创建的需求如果已设置发布，数据会按照发布计划关联的版本，自动同步至该版本的需求规划列表中。
- **规划版本需求**：如果在CodeArts Req服务中创建的需求，如果没有设置迭代或发布，可以在开发中心“需求规划”页面为版本添加需求。

## 前提条件

- 已[关联CodeArts](#)。
- 已[创建版本](#)。
- 已获取项目经理、开发人员或测试人员角色权限，权限申请方法请参见[申请权限](#)。

## 创建 Scrum 项目版本需求

**步骤1** 在开发中心首页下方的“我的团队”区域，选择需要操作的团队卡片，进入该团队空间。

如果该团队关联多个服务，可在顶部切换团队关联的其他服务。也可在顶部切换其他团队。

**步骤2** 在左侧导航栏选择“需求管理”，进入“需求管理”页面。

**步骤3** 单击“创建需求”。

**步骤4** 配置如[表8-5](#)所示参数，然后单击“确定”。

表 8-5 创建 Scrum 项目需求参数说明

参数名称	参数说明
需求标题	输入需求标题，长度应在1~255个字符。
需求类型	默认为Story类型需求。
归属版本	选择需求归属的版本，为该版本规划需求。
需求描述（可选）	输入需求描述。
当前处理人	选择需求的处理人。取值范围为该项目的全部成员。

----结束

## 规划版本需求

**步骤1** 在开发中心首页下方的“我的团队”区域，选择需要操作的团队卡片，进入该团队空间。

如果该团队关联多个服务，可在顶部切换团队关联的其他服务。也可在顶部切换其他团队。

**步骤2** 在左侧导航栏选择“版本管理”。

**步骤3** 在“版本管理”页面的版本列表中，单击版本号，进入版本详情页面。

**步骤4** 在版本详情页面的“版本持续交付”区域，选择“持续规划 > 需求规划”。

**步骤5** 在“需求规划”页面，单击“添加需求”。

**步骤6** 勾选需要为版本关联的需求，然后单击“确定”。添加后需求会显示在需求规划列表中。

如果后续需要取消需求与版本的关联，可以在列表中，单击需求所在行操作列的“从此版本移除”，移除该需求。

----结束

## 相关操作

需求创建后，在进行版本开发及管理的过程中处理需求，通过需求状态的流转，完成版本迭代及发布。

- Scrum项目：在“需求管理”页面的需求列表中，单击需求所在行后的“详情”，进入CodeArts Req服务页面，处理该需求，具体操作请参见[在Scrum项目中管理工作项](#)。
- IPD系统设备类项目：请参考CodeArts Req服务文档处理需求，例如在[IPD系统设备类项目中管理原始需求](#)、[在IPD系统设备类项目中管理研发需求](#)等。
- IPD独立软件类项目：请参考CodeArts Req服务文档处理需求，例如在[IPD独立软件类项目中管理原始需求](#)、[在IPD独立软件类项目中管理研发需求](#)等。



# 9 开发版本

## 9.1 管理 AppStage 开发中心版本交付件文档

交付件是版本开发过程不同阶段所需或交付的必要文档。在交付件规划页面，可以使用组织管理员定义的模板添加交付件，或者添加自定义类型交付件。在交付件管理页面，用户可以通过创建在线文档或上传离线文档两种方式创建及管理交付件文档，保证版本开发过程阶段性成果沉淀以及过程可追溯。

### 说明

在线文档功能仅支持白名单用户通过提交工单申请使用。详细操作请参考[提交工单](#)。

### 前提条件

- 已[创建版本](#)。
- 需要具备项目经理、开发人员或测试人员角色权限，权限申请方法请参见[申请权限](#)。

### 通过在线文档方式创建/更新/删除交付件

**步骤1** 在开发中心首页下方的“我的团队”区域，选择需要操作的团队卡片，进入该团队空间。

如果该团队关联多个服务，可在顶部切换团队关联的其他服务。也可在顶部切换其他团队。

**步骤2** 在左侧导航栏选择“版本管理”。

**步骤3** 在“版本管理”页面的版本列表中，单击版本号，进入版本详情页面。

**步骤4** 在版本详情页面的“版本持续交付”区域，选择“持续开发 > 交付件管理”。

**步骤5** 在“交付件管理”页面的交付件列表中，在无文档的交付件所在行的“操作”列单击“创建文档”。

**步骤6** 在“上传文档”对话框，选择“交付方式”为“在线文档”，单击“创建文档”，进入文档编辑页面进行内容写作，文档内容将实时自动保存，创建在线文档成功。

### 📖 说明

如需更换文档交付方式，先删除此在线文档。删除文档后其全部内容将被清空，且内容将无法恢复，请谨慎操作。

**步骤7**（可选）如需更新在线文档，具体操作如下：

1. 返回“交付件管理”页面的交付件列表中，在“交付方式”为“在线文档”的交付件所在行的“操作”列单击“更新文档”。
2. 在文档编辑页面修改内容，文档内容将实时自动保存，更新在线文档成功。

**步骤8**（可选）如需删除在线文档，具体操作如下：

1. 返回“交付件管理”页面的交付件列表中，在“交付方式”为“在线文档”的交付件所在行的“操作”列单击“删除文档”。
2. 在“删除在线文档”对话框，单击“确认”，即可删除不需要的在线文档。

----结束

## 通过离线文档方式上传/更新/删除交付件

### 📖 说明

用户需对自己上传文件的安全风险负责，开发中心不对用户自己上传的文件做任何处理。

**步骤1** 在“交付件管理”页面的交付件列表中，在无文档的交付件所在行的“操作”列单击“创建文档”。

**步骤2** 在“上传文档”对话框，选择“交付方式”为“离线文档”，单击“添加文件”将本地已准备好的文档进行上传（只能上传ZIP、RAR、DOCX、DOC、XLS、XLSX格式的文件，且文件大小不能超过50MB）。

**步骤3** 单击“确定”，上传离线文档成功。

**步骤4**（可选）如需更新离线文档，具体操作如下：

1. 返回“交付件管理”页面的交付件列表中，在“交付方式”为“离线文档”的交付件所在行的“操作”列单击“更新文档”。
2. 在“更新文档”对话框中，鼠标移至已有文档所在行并单击🗑️。
3. 在“删除文件”对话框，单击“确认”，删除旧离线文档。
4. 单击“添加文件”，将本地已准备好的新文档进行上传。
5. 单击“确定”，更新离线文档成功。

**步骤5**（可选）如需删除离线文档，具体操作如下：

1. 返回“交付件管理”页面的交付件列表中，在“交付方式”为“离线文档”的交付件所在行的“操作”列单击“删除文档”。
2. 在“删除文件”对话框，单击“确认”，即可删除不需要的离线文档。

----结束

## 9.2 在 AppStage 开发中心创建代码仓

代码仓管理是开发中心集成[代码托管（CodeArts Repo）](#)服务的能力，是遵循Git的基本运行模式的分布式版本管理平台，其具备安全管控、成员/权限管理、分支保护/合

并、在线编辑、统计服务等功能，旨在解决软件开发者在跨地域协同、多分支并发、代码版本管理、安全性等方面的问题。

本章节介绍如何使用开发中心集成的CodeArts Repo服务能力创建并使用代码仓，如果用户需要使用其他代码托管工具，可以将工具集成进开发中心，具体操作请参见[集成工具链到开发中心](#)。

## 前提条件

- 已[关联CodeArts](#)。
- 已获取项目经理、开发人员角色权限，权限申请方法请参见[申请权限](#)。

## 创建代码仓

**步骤1** 在开发中心首页下方的“我的团队”区域，选择需要操作的团队卡片，进入该团队空间。

如果该团队关联多个服务，可在顶部切换团队关联的其他服务。也可在顶部切换其他团队。

**步骤2** 在左侧导航栏选择“代码仓管理”。

**步骤3** 单击页面右上角“创建仓库”，输入仓库名称及仓库描述，单击“确定”。

创建后仓库列表显示该仓库，状态为“创建中”，待状态变为“使用中”，可以单击该仓库所在行“操作”列的“详情”，进入仓库详情页面，使用该仓库。关于代码托管文档请参见[代码托管（CodeArts Repo）](#)。

----结束

## 9.3 下载代码仓库到本地

开发人员进行代码文件的操作前，需要使用Git Bash客户端和TortoiseGit客户端下载代码仓库文件到本地环境中。

### 前提条件

操作前应确保您的网络可以访问CodeArts的代码托管服务，请参考[验证网络连通性](#)。

### 使用 SSH 协议在 Git Bash 客户端克隆代码

本节内容指导如何使用Git Bash客户端克隆代码托管服务的仓库到本地环境中。

**步骤1** [下载并安装Git Bash客户端](#)。

**步骤2** [配置SSH密钥](#)。

**步骤3** 获取仓库地址。

在仓库主页中，单击“克隆/下载”按钮，获取SSH地址，通过这个地址，可以在本地计算机连接代码托管仓库。



#### 📖 说明

如果您未配置SSH密钥，您可单击上图中“SSH密钥管理”链接进行配置，详情请参考[SSH密钥](#)。

您可在代码托管服务仓库列表中“仓库地址”下获取SSH地址。

#### 步骤4 打开Git Bash客户端。

在本地计算机上新建一个文件夹用于存放代码仓库，在空白处单击鼠标右键，打开Git Bash客户端。

#### 📖 说明

克隆仓库时会自动初始化，无需执行init命令。

#### 步骤5 输入如下命令，克隆代码托管仓库。

```
git clone 仓库地址
```

命令中“仓库地址”即[步骤3](#)中获取的SSH地址。

如果您是第一次克隆仓库，会询问您是否信任远程仓库，输入“yes”即可。

执行成功后，您会看到多出一个与您在代码托管服务新建的仓库同名的文件夹，并且其中有一个隐藏的.git文件夹，则说明克隆仓库成功。

#### 步骤6 此时您位于仓库上层目录，执行如下命令，进入仓库目录。

```
cd 仓库名称
```

进入仓库目录，可以看到此时Git默认为您定位到master分支。

```
Administrator@git-testcce MINGW64 /c/git-test
$ cd test_War_Java_Demo

Administrator@git-testcce MINGW64 /c/git-test/test_War_Java_Demo (master)
$
```

----结束

## 📖 说明

客户端在git clone代码仓库时失败的原因排查：

- 确保您的网络可以访问代码托管服务。  
请在git客户端使用如下测试命令验证网络连通性（其中“\*\*\*\*\*.com”为代码仓库地址）。

```
ssh -vT git@*****.com
```

如果返回内容含有“Could not resolve hostname \*\*\*\*\*.com: Name or service not known”，则您的网络被限制，无法访问代码托管服务，请求助您本地所属网络管理员。
- 请检查建立的SSH密钥配对关系，必要时重新生成密钥并到代码托管控制台进行配置。
- 只有[开启IP白名单](#)的机器才可以在Git客户端克隆。

## 使用 SSH 协议在 TortoiseGit 客户端克隆代码

本节内容指导如何使用TortoiseGit客户端克隆代码托管服务的仓库到本地环境中。

**步骤1** [下载并安装TortoiseGit客户端](#)。

**步骤2** 获取仓库地址。

在仓库主页中，单击“克隆/下载”按钮，获取SSH地址，通过这个地址，可以在本地计算机连接代码托管仓库。

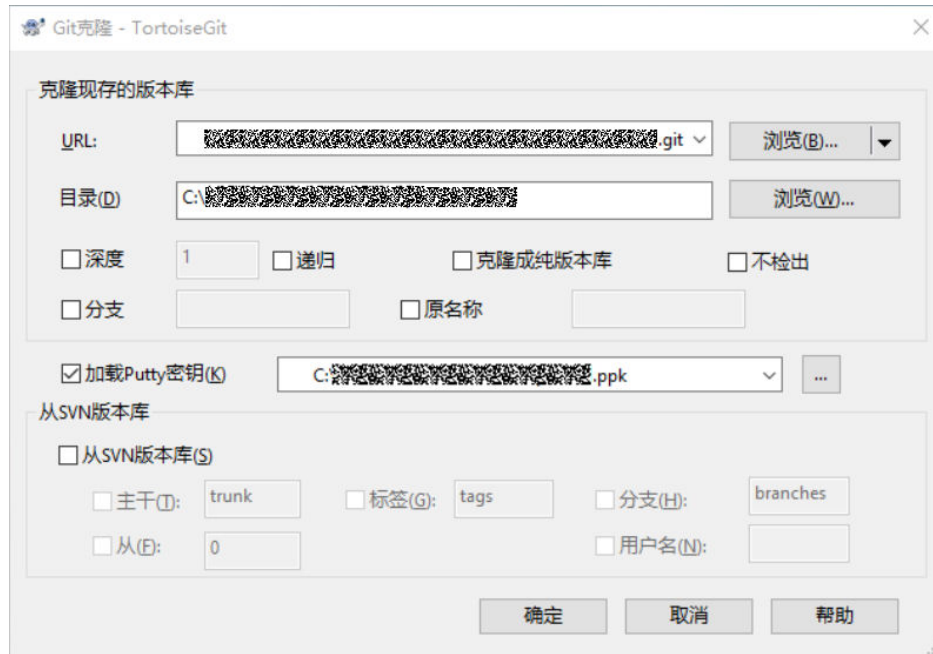
## 📖 说明

您可在代码托管服务仓库列表中“仓库地址”下获取SSH地址。

**步骤3** 进入您的本地仓库目录下，右键选择“Git克隆”菜单选项，如下图所示。

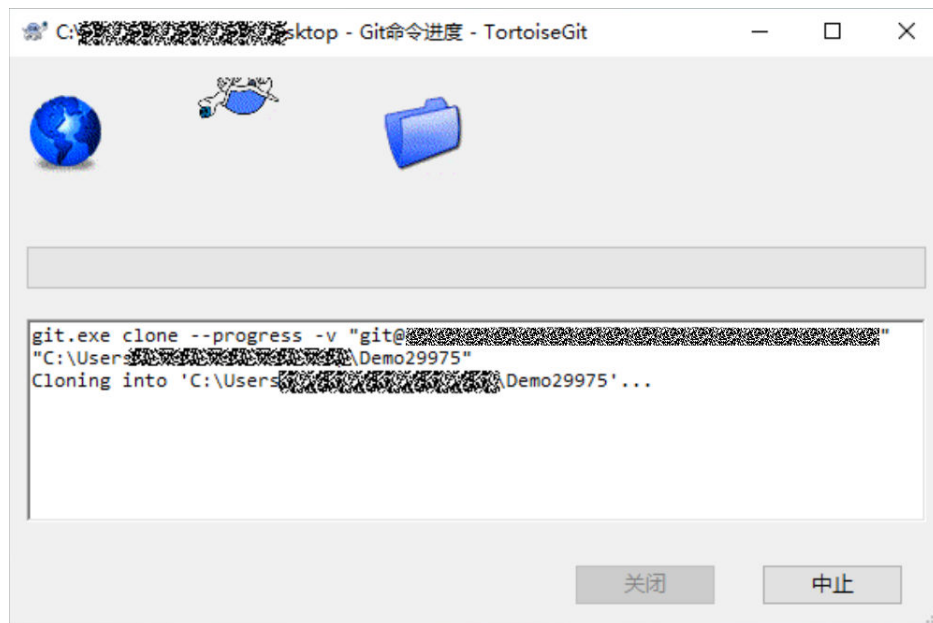


**步骤4** 在弹出的窗口中将上述复制的SSH地址粘贴到URL输入框中，勾选“加载Putty密钥”并选择私钥文件，最后单击“确定”，如下图所示。



**步骤5** 单击“确定”之后即开始克隆仓库，如果您是第一次克隆TortoiseGit客户端会询问您是否信任远程仓库，单击“是”即可。

**步骤6** 克隆用时受仓库大小影响，克隆的动作如下图所示。



----结束

## 使用 SSH 协议在 Linux 或 Mac 中克隆仓库

在配置完Linux Git客户端或Mac Git客户端环境后，Linux或Mac上Git客户端的克隆操作和[使用SSH协议在Git Bash客户端克隆代码](#)的操作完全一样。

## 9.4 提交代码到 AppStage 开发中心代码仓

开发在本地完成业务代码和IaC脚本后，需要提交代码文件至代码仓库。

### 前提条件

- 业务代码开发已完成。
- 已[创建代码仓库](#)。
- 已获取项目经理或开发人员角色权限，权限申请方法请参见[申请权限](#)。

### 操作入口

1. 在开发中心首页下方的“我的团队”区域，选择需要操作的团队卡片，进入该团队空间。  
如果该团队关联多个服务，可在顶部切换团队关联的其他服务。也可在顶部切换其他团队。
2. 在左侧导航栏选择“代码仓管理”。

### 操作指导

使用Git客户端提交修改代码的详细指导请参见[提交代码到代码托管仓库](#)。

## 9.5 在 AppStage 开发中心进行代码安全检视

开发代码完成后，需要进行安全编码TOP问题的信息统计和检查结果录入，对代码安全相关过程进行审视和记录。组织管理员[添加版本代码安全检视检查项](#)后，创建的版本会继承检查项配置。

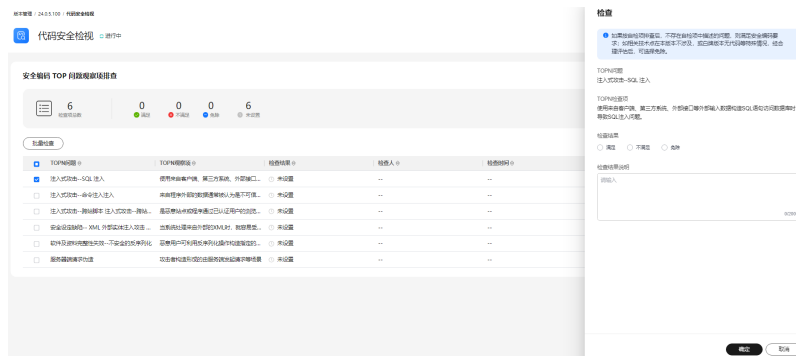
### 前提条件

- 已[创建版本](#)。
- 已获取项目经理或开发人员角色权限，权限申请方法请参见[申请权限](#)。

### 代码安全检视

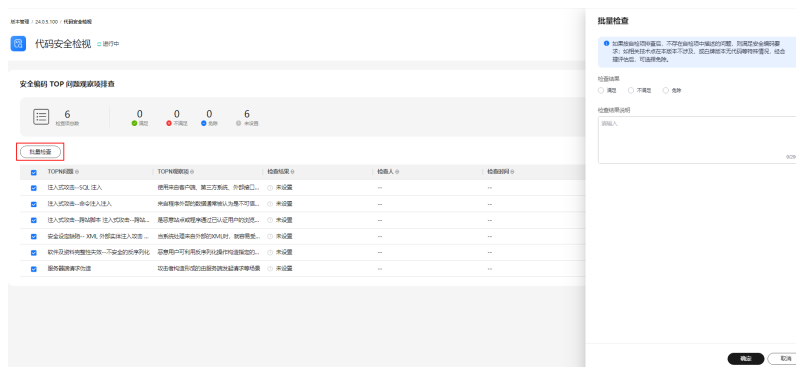
- 步骤1** 在开发中心首页下方的“我的团队”区域，选择需要操作的团队卡片，进入该团队空间。  
如果该团队关联多个服务，可在顶部切换团队关联的其他服务。也可在顶部切换其他团队。
- 步骤2** 在左侧导航栏选择“版本管理”。
- 步骤3** 在“版本管理”页面的版本列表中，单击版本号，进入版本详情页面。
- 步骤4** 在版本详情页面的“版本持续交付”区域，选择“持续开发 > 代码安全检视”。
- 步骤5** 在“代码安全检视”页面可进行如下两种方式的检查：
  - 单项检查：在检查项所在行的“操作”列单击“检查”，在“检查”页面，设置检查结果（满足、不满足或免除）以及检查结果说明，如[图9-1](#)所示。

图 9-1 单项检查



- 批量检查：勾选多条检查项并单击“批量检查”，在“批量检查”页面，设置检查结果（满足、不满足或免除）以及检查结果说明，如图9-2所示。

图 9-2 批量检查



**步骤6** 单击“确定”。检查项列表中显示检查结果、检查人、检查时间和检查结果说明信息。

---结束



# 10 在 AppStage 开发中心通过流水线编译构建版本

流水线管理是开发中心结合[流水线（CodeArts Pipeline）](#)和[编译构建（CodeArts Build）](#)服务的能力，提供可视化、可编排的CI/CD持续交付软件生产线，帮助企业快速转型，实现DevOps持续交付高效自动化，缩短应用TTM（Time to Market）交付周期，提升研发效率。

流水线服务本质上是一个可视化的自动化任务调度平台，需要配合软件开发生产线中编译构建、代码检查、测试计划、部署等服务的自动化任务使用。根据用户需要的场景，如开发测试环境应用部署、生产环境应用部署等，对这些自动化任务进行自定义编排，一次配置后就可以一键自动化触发调度执行，避免频繁低效的手工操作。

关于流水线的详细介绍文档请参考[流水线（CodeArts Pipeline）](#)和[编译构建（CodeArts Build）](#)。

本章节介绍如何使用开发中心集成的CodeArts Pipeline和CodeArts Build服务能力，如果用户需要使用其他流水线管理工具，可以将工具集成进开发中心，具体操作请参见[集成工具链到开发中心](#)。

## 前提条件

- 已[关联CodeArts](#)。
- 已[创建版本](#)。
- 已获取项目经理、开发人员角色权限，权限申请方法请参见[申请权限](#)。

## 创建并关联流水线

**步骤1** 在开发中心首页下方的“我的团队”区域，选择需要操作的团队卡片，进入该团队空间。

如果该团队关联多个服务，可在顶部切换团队关联的其他服务。也可在顶部切换其他团队。

**步骤2** 在左侧导航栏选择“流水线管理”。

**步骤3** 在“流水线管理”页面，选择版本，单击“关联流水线”，在“关联流水线”页面，根据界面提示单击“立即前往”，可以进入CodeArts页面创建流水线，如[图10-1](#)所示。

图 10-1 新建流水线



**步骤4** 在“流水线管理”页面，单击“新建流水线”。

进入“新建流水线 > 基本信息”页面，当前服务即为所属项目，填写流水线名称，根据需要选择流水线源，并配置相关参数，流水线源及相关参数说明如表10-1所示。

表 10-1 参数说明

流水线源	参数项	说明
<b>Repo</b> 提供企业代码托管的全方位服务，为软件开发者提供基于Git的在线代码托管服务。	代码库名称	选择可用的源码仓库。
	默认分支	手工或定时执行流水线时默认使用的分支。
	CodeArts Repo https 授权	配置授权扩展点以提升代码库操作权限，主要用于微服务变更流水线和部分仓库操作插件。
	别名	输入别名，则会生成该仓库对应的相关系统参数（可在参数设置页面查看），否则不会生成。 用户选择Repo流水线源时，如果设置了仓库别名，则会在系统预定义参数页面生成该仓库对应的系统参数，否则不生成。
	描述	对流水线的描述。
<b>码云</b> 连接到码云账号之后可以获取该账号下的仓库、分支等信息。	服务扩展点	选择可用码云服务扩展点，或单击“新建服务扩展点”新建通用Git服务扩展点，具体操作请参见 <a href="#">新建CodeArts服务扩展点</a> 。
	代码库名称	选择可用的源码仓库。
	默认分支	手工或定时执行流水线时默认使用的分支。
	别名	输入别名，则会生成该仓库对应的相关系统参数（可在参数设置页面查看），否则不会生成。 用户选择码云流水线源时，如果设置了仓库别名，则会在系统预定义参数页面生成该仓库对应的系统参数，否则不生成。
	描述	对流水线的描述。

流水线源	参数项	说明
<b>GitHub</b> 连接到Github账号后可以获取该账号下的仓库、分支等信息。	服务扩展点	选择可用GitHub服务扩展点，或单击“新建服务扩展点”新建GitHub服务扩展点，具体操作请参见 <a href="#">新建CodeArts服务扩展点</a> 。
	代码库名称	选择可用的源码仓库。
	默认分支	手工或定时执行流水线时默认使用的分支。
	描述	对流水线的描述。
<b>GitCode</b> 连接到Gitcode账号后可以获取该账号下的仓库、分支等信息。	服务扩展点	选择可用Gitcode服务扩展点，或单击“新建服务扩展点”新建GitCode服务扩展点，具体操作请参见 <a href="#">新建CodeArts服务扩展点</a> 。
	代码库名称	选择可用的源码仓库。
	默认分支	手工或定时执行流水线时默认使用的分支。
	描述	对流水线的描述。
<b>通用Git</b> 通过通用Git服务扩展点可以访问扩展点中配置的仓库。	服务扩展点	选择已有通用git服务扩展点，或单击“新建服务扩展点”新建通用Git服务扩展点，具体操作请参见 <a href="#">新建CodeArts服务扩展点</a> 。
	代码库地址	选择可用的源码仓库。
	默认分支	手工或定时执行流水线时默认使用的分支。
	描述	对流水线的描述。

### 📖 说明

- 其中，GitCode流水线源当前仅支持“北京四”和“广州”局点白名单用户。如果您需要开通此功能，请新建工单或拨打客服热线进行咨询。
- 如果用户新建的流水线任务不需要关联代码仓，流水线源也可以选择“暂不选择”。

**步骤5** 配置基本信息，单击“下一步”。

进入“新建流水线 > 模板选择”页面，请根据需要选择对应模板，在模板基础上快速新建任务；也可以选择“空模板”，即不使用模板，直接新建。如图10-2所示。

图 10-2 选择模板



**步骤6** 选择模板后，单击“确定”，进入“任务编排”页面，根据需要配置流水线，然后单击“保存”。

**步骤7** 在“关联流水线”页面，勾选已创建和配置完成的流水线。

**步骤8** 单击“确定”。所选流水线与当前服务版本绑定。

----结束

## 配置流水线（以添加代码检查任务为例）



**步骤1** 在流水线的“任务编排”页面，单击，新建阶段，如图10-3所示。

图 10-3 新建阶段



**步骤2** 单击 ，在“编辑阶段”页面，修改“阶段名称”为“代码检查”，并将“总是运行”设置为“否”（选择“是”，表示流水线执行时，该阶段下的任务默认选中必须执行且不可取消；选择“否”，表示流水线执行时，该阶段下的任务默认选中但可以取消。），然后单击“确定”。

**步骤3** 在流水线的“代码检查”阶段，单击“新建任务”。

**步骤4** 在“新建任务”页面的右侧区域，选择“代码检查”页签，单击“CodeArtsCheck代码检查”后的“添加”，将此插件添加到任务中，如图10-4所示。

图 10-4 代码检查



**步骤5** 在“任务配置”页面右侧区域“请选择需要调用的任务”处单击“点击创建”，如图10-5所示，进入“新建任务”页面，参照表10-2配置参数信息（此处以创建Repo源码源检查任务为例），配置完成后单击“新建任务”。

图 10-5 创建代码检查任务



表 10-2 参数说明

参数项	描述
归属项目	任务所属项目。默认填写，无需设置。
代码源	选择Repo，从代码托管拉取代码进行检查。
任务名称	代码检查任务名称，可自定义。
仓库	选择需要检查的代码仓库。
分支	填写需要检查的仓库分支名称。
检查语言	选择需要检查的代码语言。

**步骤6** 返回**步骤5**的“任务配置”页面，在“请选择需要调用的任务”下拉列表中选择已配置完成的代码检查任务，“仓库”下拉列表选择**创建代码仓库**中已创建的仓库，其他参数无需配置，保持默认即可。

**步骤7** 单击“确定”，添加代码检查任务完成。

**步骤8** 单击“任务编排”页面右上角“保存”，保存流水线的任务配置。

----结束

## 配置流水线（以添加业务包构建任务为例）

**步骤1** 在流水线的“任务编排”页面，单击 $\oplus$ ，新建阶段。

**步骤2** 单击 $\text{✎}$ ，在“编辑阶段”页面，修改“阶段名称”为“构建阶段”，并将“总是运行”设置为“是”（选择“是”，表示流水线执行时，该阶段下的任务默认选中必须执行且不可取消；选择“否”，表示流水线执行时，该阶段下的任务默认选中但可以取消。），然后单击“确定”。

- 步骤3** 在流水线的“构建阶段”，单击“新建任务”。
- 步骤4** 在“新建任务”页面的右侧区域，选择“构建”页签，单击“Build构建”后的“添加”，将此插件添加到任务中。
- 步骤5** 在“任务配置”页面右侧区域“请选择需要调用的任务”处单击“点击创建”，如图10-6所示。

图 10-6 Build 构建任务配置



- 步骤6** 在“新建构建任务”页面的“基本信息”页面，自定义任务名称（以“adadss-build”为例），归属项目默认为当前服务，源码源选择“Repo”，选择对应的代码仓库（以创建好的“adadss”仓库为例）以及分支“master”，单击“下一步”。
- 步骤7** 在“构建模板”页面，选择“空白构建模板”模板，然后单击“下一步”，进入“构建步骤”的“图形化”页签，该页面的“构建环境配置”和“代码下线配置”无需配置，保持默认配置即可。
- 步骤8** 配置构建步骤“Maven构建”。
  - 返回“构建步骤”页签，单击“点击添加构建步骤”，在右侧“添加步骤”页面搜索“Maven构建”，鼠标移至插件卡片并单击“添加”，如图10-7所示。

图 10-7 Maven 构建



- 执行maven构建命令将业务代码打包，不同项目有所不同。  
参考示例：`mvn -Dassembly clean compile package -Dmaven.test.skip=true -U -T4`

### 📖 说明

若某些jar包依赖不到，可将仓库地址添加在setting配置的公有依赖仓库。

- 步骤9** 配置镜像构建。
  - 登录华为云容器镜像服务控制台，在左侧导航栏选择“我的镜像”，单击右上角“页面上传”，在“页面上传”对话框，创建组织（必须使用华为云账号），单

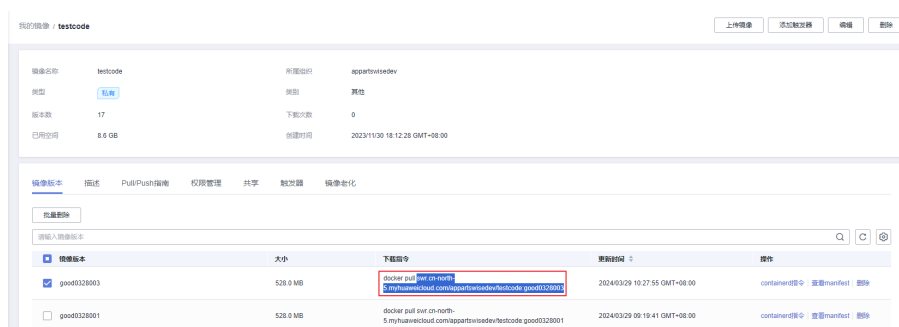
击“选择镜像文件”，选择已经准备好的基础镜像tar包，单击“开始上传”，待任务进度显示“上传完成”，表示镜像文件上传成功。

### 说明

基础镜像tar包仅需上传一次，后续可直接从镜像文件列表中选取。

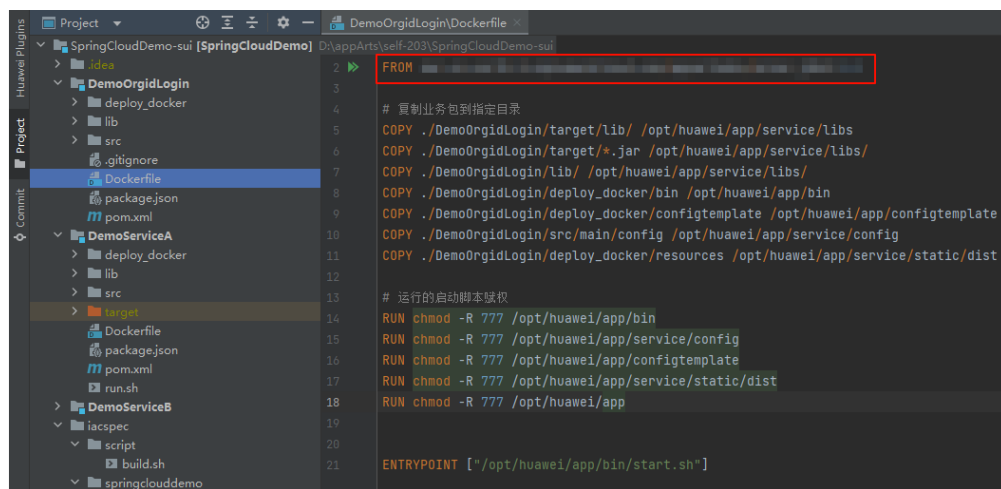
- 在镜像仓库找到**步骤9.1**上传的基础镜像，参照**图10-8**复制镜像地址。

图 10-8 复制镜像地址



- 将**步骤9.2**复制的镜像地址粘贴到Dockerfile文件中第一行FROM命令后，如图**10-9**所示。

图 10-9 将镜像地址粘贴到 Dockerfile 文件中




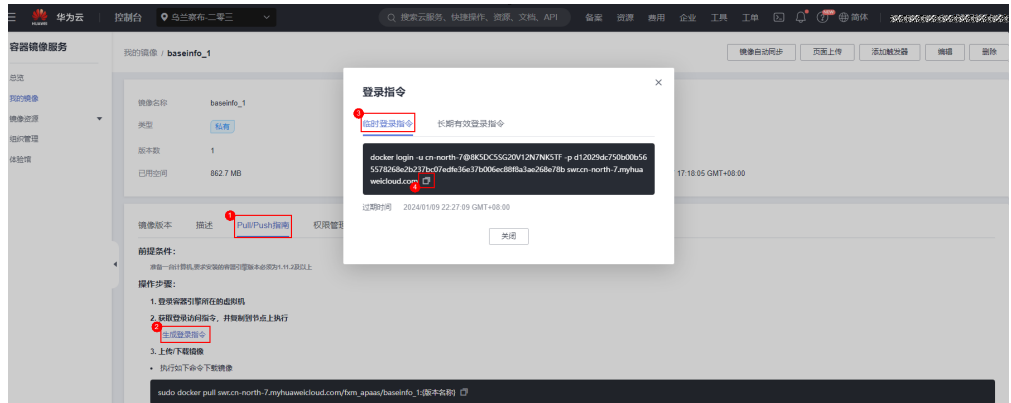
- 返回华为云容器镜像服务控制台镜像列表中单击镜像tar包名称，进入镜像详情页面，选择“Pull/Push指南”页签，单击“操作步骤”中的“生成登录指令”。
- 在“登录指令”对话框中的“临时登录指令”框的指令末尾处单击，复制临时登录指令，如图**10-10**所示。



图 10-10 复制临时登录指令



6. 返回“构建步骤”页面，单击“添加步骤”，在右侧“添加步骤”页面搜索“执行Docker命令”，鼠标移至该插件卡片并单击“添加”。
7. 单击“执行Docker命令”步骤，在右侧“命令”参数下单击“添加”添加docker命令，在“命令”下拉框中选择“login”，参数框中输入步骤9.5获取的临时登录指令中login后的部分，例如“-u cn-north-7@K5X8GVN5B4H4B8KB9SDO -p 1dd00acee886bd05886cded19f5af1cb1e96326e4354fc0f3f493f12a3586518swr.cn-north-7.myhuaweicloud.com”，登录环境以便Dockerfile文件中可正常获取基础镜像。
8. 在“操作”列单击“+”继续添加docker命令，“命令”下拉框中选择“build”，并在参数框中输入命令，指定代码仓中的Dockerfile文件并构建镜像。

例如构建helloworldservice:23.11.02.1镜像，可输入“-t helloworldservice:23.11.02.1 -f ./deploy\_docker/Dockerfile.”，其中“./deploy\_docker/Dockerfile”为Dockerfile文件目录。

### 说明

工作目录一般为根目录。

9. 在“操作”列单击“+”继续添加docker命令，“命令”下拉框中选择“save”，并在参数框中输入命令，将镜像包保存为tar包。  
例如将helloworldservice:23.11.02.1镜像包保存为helloworldservice.23.11.02.1.tar包，可输入“-o helloworldservice.23.11.02.1.tar helloworldservice:23.11.02.1”。至此，添加docker命令完成，如图10-11所示。

图 10-11 执行 docker 命令



**步骤10** 返回“构建步骤”页面，单击“添加步骤”，在右侧“添加步骤”页面搜索“Maven构建”，鼠标移至插件卡片并单击“添加”。执行zip命令，将tar包与业务代码中的package.json文件直接压缩成一个zip包。

如下命令，将helloworldservice.23.11.02.1.tar和package.json文件压缩为helloworldservice\_23.11.02.1.zip包。

```
zip helloworldservice_23.11.02.1.zip helloworldservice.23.11.02.1.tar package.json
```

**步骤11** 继续单击“添加步骤”，添加“上传文件到OBS”插件，在“上传文件到OBS”页面，参照表10-3配置相关参数，配置完成后单击右上角“新建”，配置构建任务完成。

表 10-3 “上传文件到 OBS”（业务包）步骤配置的参数说明

参数名称	参数说明
步骤显示名称	构建步骤的名称，默认为“上传文件到OBS”，保持默认或自定义修改均可。
授权用户	在下拉列表选择： <ul style="list-style-type: none"> <li>当前用户：上传到当前租户的OBS桶。</li> <li>其他用户：可以通过选择IAM账号的方式上传到指定租户的OBS桶。</li> </ul>
IAM账号	“授权用户”选择“其他用户”时需配置此参数，配置步骤如下： <ol style="list-style-type: none"> <li>单击“管理IAM账号”，进入“服务扩展点管理”页面。</li> <li>单击“新建服务扩展点”，选择“IAM账号”，弹出“新建服务扩展点”对话框，填写以下参数，补充租户下用户的AK、SK即可： <ul style="list-style-type: none"> <li>连接名称：服务扩展点的名称。例如“obs权限”。</li> <li>Access Key Id：访问密钥ID（AK），<a href="#">获取访问密钥AK/SK</a>。</li> <li>Secret Access Key：秘密访问密钥（SK），<a href="#">获取访问密钥AK/SK</a>。</li> </ul> </li> <li>信息填写完成，单击“确定”。</li> </ol>
构建产物路径	路径为步骤10的zip包的相对路径或输入*模糊匹配（如“./DemoServiceB/*.zip”）
桶名	用户自定义构建结果上传到的目标OBS桶名。
OBS存储目录	用户自定义构建结果在OBS上的存储目录
OBS存储文件名	非必填项，构建结果在OBS上的存储文件名（不包含目录），留空时可上传多个文件，取构建产物文件名为OBS存储文件名；不为空时只能上传单个文件，如 application.jar。
是否上传文件夹	非必填项，可选择是否开启上传文件夹。
OBS头域	非必填项，上传文件时加入一个或多个自定义的响应头，当用户下载此对象或查询此对象元数据时，加入的自定义响应头会在返回消息的头域中出现。如：“键”填写成“x-frame-options”，“值”填写成“false”，即可禁止OBS中存放的网页被第三方网页嵌入。

**步骤12** 返回**步骤5**的“任务配置”页面，在“请选择需要调用的任务”下拉列表中选择已配置完成的构建任务，“仓库”下拉列表选择**创建代码仓库**中已创建的仓库，其他参数无需配置，保持默认即可。

**步骤13** 单击“确定”，添加业务包构建任务完成。

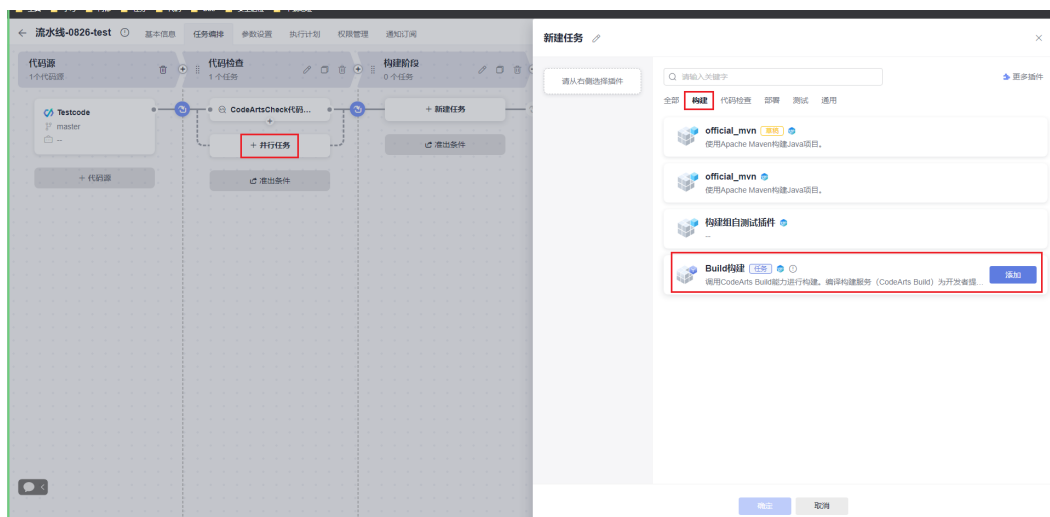
**步骤14** 单击“任务编排”页面右上角“保存”，保存流水线的任务配置。

----结束

## 配置流水线（以添加 IaC 自动部署包构建任务为例）

**步骤1** 在流水线的“任务编排”页面的“构建阶段”，单击 **+ 并行任务**，如**图10-12**所示。

图 10-12 并行任务



**步骤2** 在“新建任务”页面的右侧区域，选择“构建”页签，单击“Build构建”后的“添加”，将此插件添加到任务中。

**步骤3** 在“任务配置”页面右侧区域，设置IaC自动部署包构建任务“名称”（如设置名称为“Build构建\_IaC”），在“请选择需要调用的任务”处单击“点击创建”。

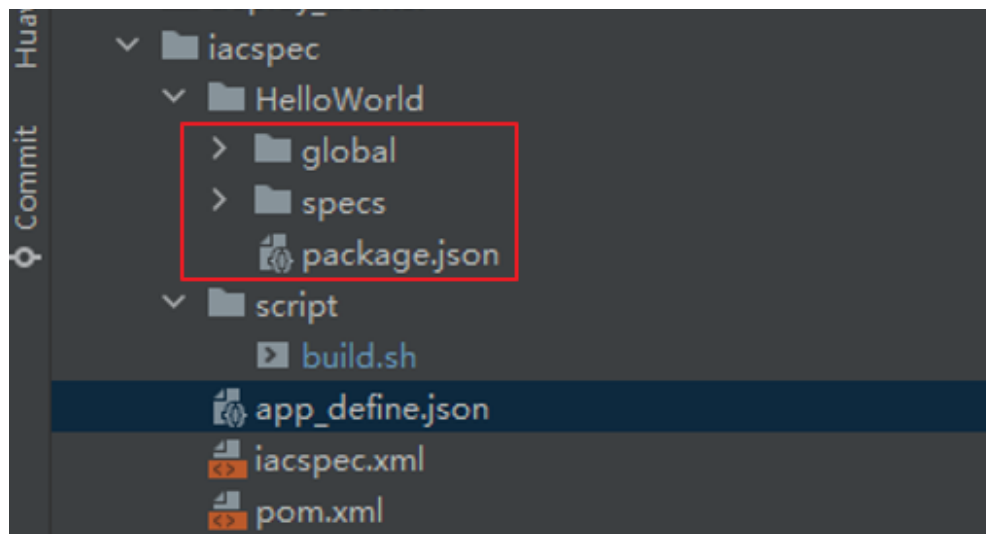
**步骤4** 在“新建构建任务 > 基本信息”页面，自定义任务名称（以“adadss-build-IaC”为例），归属项目默认为当前服务，源码源选择“Repo”，选择对应的代码仓库（例如“adadss”）以及分支“master”，单击“下一步”。

**步骤5** 在“新建构建任务 > 构建模板”页面，选择“空白构建模板”模板，然后单击“下一步”，进入“构建步骤”的“图形化”页签，该页面的“构建环境配置”和“代码下线配置”无需配置，保持默认配置即可。

**步骤6** 返回“构建步骤”页签，单击“点击添加构建步骤”，在右侧“添加步骤”页面搜索“Maven构建”后单击“添加”。

**步骤7** 在右侧“Maven构建”页面的命令窗口，输入如下示例的构建命令，以**图10-13**示例，参考示例命令：`mvn -am -pl iacspec -Dassembly clean compile package -Dmaven.test.skip=true -U -T4`，通过执行指定目录下的pom.xml文件将HelloWorld文件夹下的文件压缩成zip文件。

图 10-13 压缩 zip 包



**说明**

图10-13中“build.sh”、“app\_define.json”、“iacspec.xml”、“pom.xml”这些文件为自定义的脚本文件。

**步骤8** 继续单击“添加步骤”，在右侧“添加步骤”页面搜索“上传文件到OBS”，鼠标移至该插件单击“添加”将其添加到构建步骤中。

**步骤9** 单击“上传文件到OBS”步骤，在“上传文件到OBS”页面，参照表10-4配置相关参数，配置完成后单击“新建”，配置IaC自动部署包构建任务完成。

表 10-4 “上传文件到 OBS”（IaC 自动部署包）步骤配置的参数说明

参数名称	参数说明
步骤显示名称	构建步骤的名称，默认为“上传文件到OBS”，保持默认或自定义修改均可。
授权用户	在下拉列表选择： <ul style="list-style-type: none"> <li>当前用户：上传到当前租户的OBS桶。</li> <li>其他用户：可以通过选择IAM账号的方式上传到指定租户的OBS桶。</li> </ul>
IAM账号	“授权用户”选择“其他用户”时需配置此参数，配置步骤如下： <ol style="list-style-type: none"> <li>单击“管理IAM账号”，进入“服务扩展点管理”页面。</li> <li>单击“新建服务扩展点”，选择“IAM账号”，弹出“新建服务扩展点”对话框，填写以下参数，补充租户下用户的AK、SK即可：                             <ul style="list-style-type: none"> <li>连接名称：服务扩展点的名称。例如“OBS权限”。</li> <li>Access Key Id：访问密钥ID（AK），<a href="#">获取访问密钥AK/SK</a>。</li> <li>Secret Access Key：秘密访问密钥（SK），<a href="#">获取访问密钥AK/SK</a>。</li> </ul> </li> <li>信息填写完成，单击“确定”。</li> </ol>

参数名称	参数说明
构建产物路径	路径为 <b>步骤7</b> 的zip包的相对路径或输入*模糊匹配（如“./DemoServiceB/*.zip”）
桶名	用户自定义构建结果上传到的目标OBS桶名。
OBS存储目录	用户构建结果在OBS上的存储目录。
OBS存储文件名	非必填项，构建结果在OBS上的存储文件名（不包含目录），留空时可上传多个文件，取构建产物文件名为OBS存储文件名；不为空时只能上传单个文件，如 application.jar。
是否上传文件夹	非必填项，可选择是否开启上传文件夹。
OBS头域	非必填项，上传文件时加入一个或多个自定义的响应头，当用户下载此对象或查询此对象元数据时，加入的自定义响应头会在返回消息的头域中出现。如：“键”填写成“x-frame-options”，“值”填写成“false”，即可禁止OBS中存放的网页被第三方网页嵌入。

**步骤10** 返回**步骤3**的“任务配置”页面，在“请选择需要调用的任务”下拉列表中选择已配置完成的IaC自动部署包构建任务，“仓库”下拉列表选择**创建代码仓库**中已创建的仓库，其他参数无需配置，保持默认即可。

**步骤11** 单击“确定”，添加IaC自动部署包构建任务完成。

**步骤12** 单击“任务编排”页面右上角“保存”，保存流水线的任务配置。

----结束

## 执行流水线

**步骤1** 在当前版本的流水线列表中已关联的流水线所在行的“操作”列，单击“执行”。

**步骤2** 在“执行构建”对话框，单击“确定”，流水线开始构建版本的发布软件包。

### 说明

- 在流水线进行中，单击“操作”列“终止”可停止构建。
- 一个版本可多次重复执行发布构建，但以最后一次的构建结果为准。

----结束

## 更多操作

您还可以执行如下**表10-5**所示操作。

表 10-5 更多操作

操作	步骤
查看流水线运行历史	在流水线列表中单击流水线名称，进入“流水线运行历史”页面： <ul style="list-style-type: none"><li>选择“流水线详情”页签，查看流水线编排的详情。</li><li>选择“执行历史”页面，选择时间段，可查看该时间段的流水线执行详情信息。</li></ul>
解除关联流水线	<ol style="list-style-type: none"><li>在流水线列表“操作”列，单击“解除关联”。</li><li>单击“确认”，解除此流水线与当前版本的关联。</li></ol>

# 11 测试版本

## 11.1 在 AppStage 开发中心为版本制定测试计划

测试功能是开发中心集成[测试计划服务（CodeArts TestPlan）](#)的能力，测试计划服务（CodeArts TestPlan）是一款自主研发的一站式测试管理平台，沉淀了华为多年高质量的软件测试工程方法与实践，覆盖测试计划、测试设计、测试用例、测试执行和测试评估等全流程，旨在帮助企业协同、高效、可信地开展测试活动，保障产品高质量上市。

本章节介绍如何使用开发中心集成的CodeArts TestPlan服务能力进行产品测试，如果用户需要使用其他测试工具，可以将工具集成进开发中心，具体操作请参见[集成工具链到开发中心](#)。

### 前提条件

- 已[关联CodeArts](#)。
- 已[创建版本](#)。
- 已获取项目经理或测试人员角色权限，权限申请方法请参见[申请权限](#)。

### 操作步骤

1. 在开发中心首页下方的“我的团队”区域，选择需要操作的团队卡片，进入该团队空间。  
如果该团队关联多个服务，可在顶部切换团队关联的其他服务。也可在顶部切换其他团队。
2. 在左侧导航栏选择“测试”。
3. 选择需要的测试功能进行操作，测试功能的详细介绍如[表11-1](#)所示。

表 11-1 测试功能说明

功能导航	说明	参考文档的链接
测试计划	测试计划明确测试时间、测试范围、测试目标，并管理测试各个阶段的活动。测试计划可以针对某个版本、迭代或专项等。	请参见 <a href="#">测试计划</a> 。
测试设计	测试设计在整个测试活动中起到承前启后的作用，测试设计的目的是明确测试活动的范围、测试目标和测试方法，能够指导测试执行过程的开展，从而规范测试行为。	请参见 <a href="#">测试设计</a> 。
测试用例	测试用例是对一项特定的软件产品进行测试任务的描述，体现在测试方案、方法、技术和策略。其内容包括测试目标、测试环境、输入数据、测试步骤、预期结果、测试脚本等，最终形成文档。	请参见 <a href="#">测试用例</a> 。
测试执行	测试执行阶段中执行测试套件，检查被测系统是否符合测试套件预期结果，记录测试结果，发现产品问题及缺陷并帮助开发人员分析定位问题。	请参见 <a href="#">测试执行</a> 。
测试质量看板	测试质量看板展示了用例库和测试计划对应的需求覆盖率、缺陷数、用例通过率、用例完成率图表，以及用例通过率、用例关联的缺陷等详细信息。	请参见 <a href="#">测试质量看板</a> 。
测试质量评估	测试质量评估涵盖了创建测试报告、总览所选测试计划的相关测试数据情况、查看测试计划下需求、缺陷的遗留和完成情况等明细，以及风险评估等。	请参见 <a href="#">测试质量评估</a> 。

## 11.2 在 AppStage 开发中心创建缺陷

在产品研发过程中，存在各团队、各项目各自为战，产品质量难管控、缺陷修复进度难追踪的问题，严重影响产品交付效率。产品特性和功能在测试验证阶段发现的问题，可以使用缺陷单进行跟踪，对于发现的缺陷进行记录、跟踪、分析和解决，确保产品质量。

缺陷管理是开发中心集成[需求管理（CodeArts Req）](#)服务中“缺陷（Bug）”相关的能力，缺陷管理严格把控缺陷提出、分析、修复、测试、验收、关闭的完整流程，提供跨项目的缺陷作业跟踪追溯能力，实时识别产品缺陷风险，为组织的产品交付质量提供保障。

本章节介绍如何使用开发中心集成的CodeArts Req服务能力创建缺陷，如果用户需要使用其他工具，可以将工具集成进开发中心，具体操作请参见[集成工具链到开发中心](#)。

- Scrum项目：在开发中心直接创建缺陷，创建后数据会同步至CodeArts Req服务中，具体操作请参见[创建Scrum项目版本缺陷](#)。如果在CodeArts Req页面创建缺陷，数据会同步至开发中心缺陷管理列表，若缺陷已设置迭代，数据会按照迭代关联的版本，自动同步至该版本的版本缺陷列表中。



- IPD系统设备类项目和IPD独立软件类项目：开发中心集成CodeArts Req服务页面，可以通过开发中心的“缺陷管理”进入并创建缺陷，操作可以参考[在IPD系统设备类项目中新建缺陷](#)和[在IPD独立软件类项目中新建缺陷](#)。创建的缺陷如果已设置发现发布，数据会按照发现缺陷的发布计划关联的版本，自动同步至该版本的版本缺陷列表中。

## 前提条件

- 已[关联CodeArts](#)。
- 已[创建版本](#)。
- 已获取项目经理、开发人员或测试人员角色权限，权限申请方法请参见[申请权限](#)。

## 创建 Scrum 项目版本缺陷

**步骤1** 在开发中心首页下方的“我的团队”区域，选择需要操作的团队卡片，进入该团队空间。

如果该团队关联多个服务，可在顶部切换团队关联的其他服务。也可在顶部切换其他团队。

**步骤2** 在左侧导航栏选择“缺陷管理”，进入“缺陷管理”页面。

**步骤3** 单击“创建缺陷”。

**步骤4** 配置如[表11-2](#)所示参数，然后单击“确定”。

表 11-2 创建 Scrum 项目缺陷参数说明

参数名称	参数说明
缺陷标题	输入缺陷标题，长度应在1~255个字符。
归属版本	选择缺陷归属的版本。
缺陷描述（可选）	输入缺陷描述。
当前处理人	选择缺陷的处理人。取值范围为该项目的全部成员。

----结束

## 相关操作

缺陷创建后，需要对缺陷进行分析、修复、测试、验收、关闭，完成版本缺陷处理。

- Scrum项目：在“缺陷管理”页面的缺陷列表中，单击缺陷所在行后的“详情”，进入CodeArts Req服务页面，处理该缺陷，具体操作请参见[在工作项详情页中管理工作项](#)。
- IPD系统设备类项目：请参考[在IPD系统设备类项目中管理缺陷](#)处理缺陷。
- IPD独立软件类项目：请参考[在IPD独立软件类项目中管理缺陷](#)处理缺陷。

## 11.3 在 AppStage 开发中心进行版本测试评估

在测试评估页面可以编辑总体测试结论、系统预置的测试类型结论，新增和管理自定义测试类型的测试结论，以及新增和管理版本关键风险问题。

### 前提条件

- 已[创建版本](#)。
- 已获取项目经理或测试人员角色权限，权限申请方法请参见[申请权限](#)。

### 查看版本缺陷统计

**步骤1** 在开发中心首页下方的“我的团队”区域，选择需要操作的团队卡片，进入该团队空间。

如果该团队关联多个服务，可在顶部切换团队关联的其他服务。也可在顶部切换其他团队。

**步骤2** 在左侧导航栏选择“版本管理”。

**步骤3** 在“版本管理”页面的版本列表中，单击版本号，进入版本详情页面。

**步骤4** 在版本详情页面的“版本持续交付”区域，选择“持续开发 > 缺陷管理”。

**步骤5** 在“缺陷管理”页面，选择“版本缺陷统计”页签，可查看缺陷DI（ $DI值 = 关键缺陷数 * 10 + 严重缺陷数 * 3 + 一般缺陷数 * 1 + 提示缺陷数 * 0.1$ ）、缺陷数统计、缺陷状态统计（新建、进行中、测试中等）、未关闭缺陷级别统计、责任人统计等不同维度的缺陷统计。

#### 说明

- IPD类型项目当前仅统计修复状态的缺陷数据，需要在缺陷的修复阶段为缺陷绑定修复发布；Scrum类型项目会统计所有状态缺陷，需要为缺陷绑定对应的迭代。
- 统计数据更新会有5~10分钟延迟。

----结束

### 编辑总体测试结论

**步骤1** 在开发中心首页下方的“我的团队”区域，选择需要操作的团队卡片，进入该团队空间。

如果该团队关联多个服务，可在顶部切换团队关联的其他服务。也可在顶部切换其他团队。

**步骤2** 在左侧导航栏选择“版本管理”。

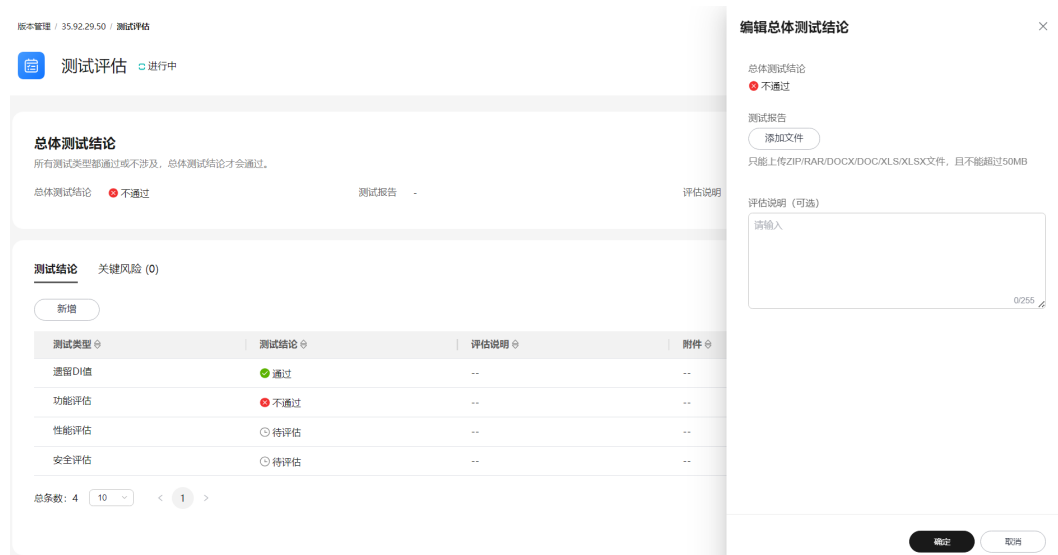
**步骤3** 在“版本管理”页面的版本列表中，单击版本号，进入版本详情页面。

**步骤4** 在版本详情页面的“版本持续交付”区域，选择“持续开发 > 测试评估”，进入“测试评估”页面。

**步骤5** 在“测试评估”页面的“总体测试结论”区域，单击右侧“编辑”。

**步骤6** 在“编辑总体测试结论”页面，可添加测试报告文件以及编辑评估说明（其中的“总体测试结论”由所有测试类型的结论决定，不可编辑，所有单项测试类型都通过或不涉及，总体测试结论才会通过），如图11-1所示，编辑完成后单击“确定”即可。

图 11-1 编辑总体测试结论



----结束

## 管理单项测试结论


**步骤1** 在“测试评估”页面选择“测试结论”页签，系统预置了四个测试类型：遗留DI值、功能评估、性能评估和安全评估。

**步骤2** 单击“新增”，在“新增测试结论”页面，设置测试结论相关参数，如图11-2所示，参数说明请参见表11-3。

图 11-2 编辑测试结论



表 11-3 测试结论参数说明

参数名	参数说明
测试类型	输入测试类型。
测试结论	根据实际情况可设置为：通过、不通过或不涉及。
评估说明（可选）	测试评估的相关说明。
附件（可选）	单击“添加文件”，最多可上传一个附件文件辅助测试评估的说明，且只能上传ZIP、RAR、DOCX、DOC、XLS、XLSX格式文件，文件不能超过50MB。 <b>说明</b> <ul style="list-style-type: none"><li>• 用户需对自己上传文件的安全风险负责，开发中心不对用户自己上传的文件做任何处理。</li><li>• 添加文件后，如需变更文件，可光标移至文件，在文件右侧单击, 将旧文件删除后，再单击“添加文件”重新上传新文件。</li></ul>

**步骤3** 设置完成后单击“确定”。

如需编辑或删除单项测试结论，可以单击测试结论列表“操作”列的“编辑”或“删除”。系统预置的四个测试类型（遗留DI值、功能评估、性能评估、安全评估）不可删除。

----结束

## 管理关键风险

**步骤1** 在“测试评估”页面选择“关键风险”页签，然后单击“新增”。

**步骤2** 在“新增关键风险”页面，如图11-3所示，参数说明请参见表11-4。

图 11-3 新增关键风险

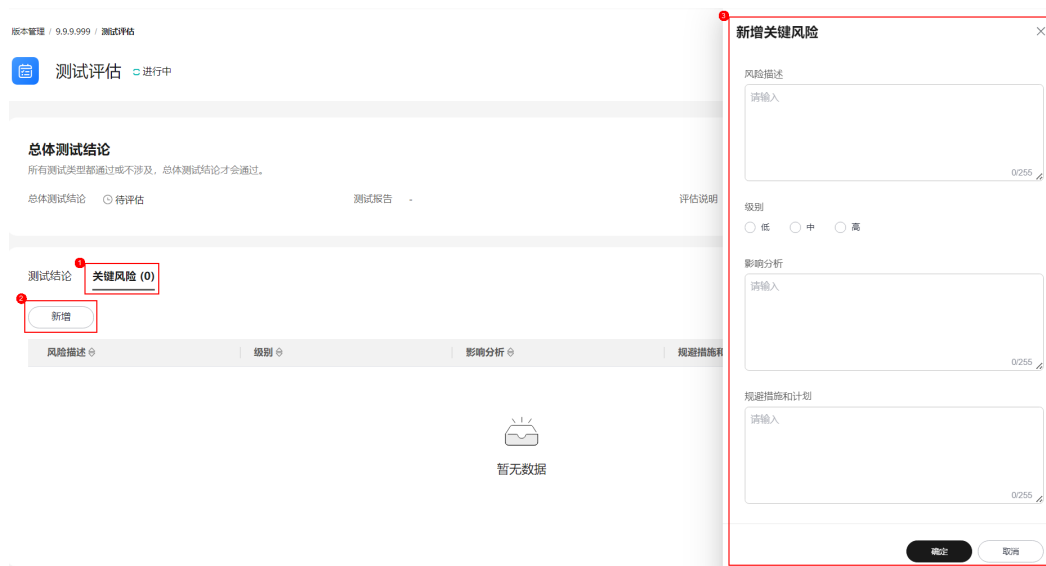


表 11-4 关键风险参数说明

参数名	参数说明
风险描述	风险问题的描述。
级别	区分三个级别：低、中、高
影响分析	风险问题的相关影响分析。
规避措施和计划	规避该风险问题的相关措施和计划。

**步骤3** 单击“确定”。新增的关键风险显示在风险列表中。

如需编辑或删除关键风险，可以单击关键风险列表“操作”列的“编辑”或“删除”。

----结束

# 12 发布版本

## 12.1 在 AppStage 开发中心调测发布版本

在版本标准发布（即正式发布）前，可先执行调测发布进行环境调测。

### 前提条件

- 已[创建版本](#)。
- 已获取项目经理、开发人员或测试人员角色权限，权限申请方法请参见[申请权限](#)。
- 发布软件包到运维中心前，需要在运维中心录入并绑定镜像仓，否则会发布失败，具体操作请参见[管理SWR](#)。

### 创建调测发布活动

**步骤1** 在开发中心首页下方的“我的团队”区域，选择需要操作的团队卡片，进入该团队空间。

如果该团队关联多个服务，可在顶部切换团队关联的其他服务。也可在顶部切换其他团队。

**步骤2** 在左侧导航栏选择“版本管理”。

**步骤3** 在“版本管理”页面的版本列表中，单击版本号，进入版本详情页面。

**步骤4** 在版本详情页面的“版本持续交付”区域，选择“持续部署发布 > 调测发布”。

**步骤5** 在“调测发布”页面单击“创建活动”，在“创建调测发布活动”页面，设置版本名称、备注（可选）。

#### 说明

调测发布的版本名称与标准发布的主版本名称前三段一致，仅需定义第四段编号，且第四段编号不能与标准发布的主版本名称第四段编号相同。

**步骤6** 单击“确定”，在“调测发布”页面的发布版本号列表中可查看到新创建的调测发布版本号，且其状态为“草稿”，可继续执行如下操作：

- 单击调测发布版本号所在行“操作”列“发布”，进入“添加发布软件包”页面添加基线化软件包，具体操作请参见[添加版本发布软件包](#)。

- 单击调测发布版本号所在行“操作”列“删除”，在“删除调测发布活动”对话框单击“确定”，删除不需要的调测发布版本号。

#### 说明

当调测发布版本号列表中的版本号“状态”均为“审批通过”后，才可创建新的调测发布版本活动（调测发布版本号），否则不能新创建。

----结束

## 添加版本发布软件包

由项目经理或开发人员添加版本发布软件包（已完成基线化的软件包），即在版本发布前，选择特定流水线最新一次的构建产物作为版本基线，将流水线构建产物与版本关联，版本发布审批通过后，开发中心将版本关联的软件包或镜像包信息通知给部署平台，以便后续部署变更。如果版本未关联流水线，单击左上角“流水线管理”，在“流水线管理”页面为当前版本关联并执行流水线，具体操作请参考[通过流水线编译构建版本](#)。

**步骤1** 在“调测发布 > 添加发布软件包”页面，单击“添加基线化软件包”（最多可添加20个基线化软件包），弹出“添加基线化软件包”页面。

**步骤2** 在“选择流水线”下拉列表选择流水线，确认最近一次发布构建信息。

**步骤3** 单击右下角“基线化”。开发中心后台将记录流水线发布构建的软件包的存储路径，用于版本发布时获取。

在“版本基线化”页面“版本基线化信息”区域，可查看相应流水线详情，在“软件包列表”中可查看流水线构建的软件包相关信息（如文件位置、文件大小等）。

**步骤4** 单击右下角“下一步”，进入调测发布申请页面[申请调测发布](#)。

----结束

## 申请调测发布

可由开发人员、测试人员或项目经理提交版本调测发布申请。

**步骤1** 在调测发布申请页面的“发布信息”区域，选择区域，（可选）单击“添加文件”上传附件，填写发布内容。

#### 说明

- 用户需对自己上传文件的安全风险负责，开发中心不对用户自己上传的文件做任何处理。
- 上传的附件只能为ZIP/RAR/DOCX/DOC/XLS/XLSX文件，且不能超过50MB。
- 在“发布软件包信息”中如无发布软件包，则根据界面提示单击“基线化”跳转至“流水线管理”页面执行该版本流水线基线化生成发布软件包。

**步骤2** 在“审核信息”区域的“项目经理”下拉列表中选择审批发布的项目经理。

**步骤3** 单击“提交”，“申请发布调测版本”页面的左上角将提示“您所提交的发布申请正在审批中，请耐心等待”。该版本发布申请提交将生成一条待办信息至项目经理的AppStage首页的“我的待办”中，由其单击待办链接跳转至调测发布审批页面完成审批操作，具体操作请参见[审批调测发布](#)。

待审批人审批后，刷新“申请发布调测版本”页面，可根据不同的审批结果进行后续操作：

- 如果审批通过，“申请发布调测版本”页面左上角将提示“恭喜！您所提交的发布申请已通过审批”。  
同时当前页面底部“软件包发布结果”可查看发布结果（单击“查看详情”可查看发布信息详情）：
  - 发布中：部分软件包发布中。
  - 发布成功：发布成功的版本软件包将发布到对应的部署平台（例如AppStage运维中心的“软件仓库 > 软件包管理”）。
  - 发布失败：部分软件包发布失败，可单击“重试”进行发布重试。
- 如果审批未通过，“申请发布调测版本”页面左上角将提示“您所提交的发布申请未通过审批（审批意见：xxx），请修改后重新提交”，则重新执行[步骤1~步骤3](#)。

----结束

## 审批调测发布

开发人员、测试人员或项目经理提交版本调测发布申请后，生成的待办通知到相应审批人的AppStage首页的“我的待办”区域，由审批人单击此待办信息的链接跳转至调测发布审批页面进行审批。

**步骤1** 审批人（项目经理）登录[AppStage](#)。

**步骤2** 在“我的待办”列表中，单击版本调测发布审批的描述，跳转至“调测发布审批”页面。

**步骤3** 查看版本发布信息，在“审批信息”区域，勾选审批结果为“通过”或“不通过”以及输入审批意见，单击“提交”。

- 勾选“通过”：申请人刷新“申请发布调测版本”页面后，“申请发布调测版本”页面左上角将提示申请人“恭喜！您所提交的发布申请已通过审批”。版本软件包将发布到对应的部署平台（例如AppStage运维中心的“软件仓库 > 软件包管理”）。
- 勾选“不通过”：申请人刷新“申请发布调测版本”页面后，“申请发布调测版本”页面左上角将提示申请人“您所提交的发布申请未通过审批（审批意见：xxx），请修改后重新提交”。

----结束

## 12.2 在 AppStage 开发中心标准发布版本

待版本开发及编译构建完成，且发布准入检查项均通过后，可由开发人员、测试人员或项目经理提交标准发布申请，待项目经理审批通过后，将软件发布到对应的部署平台。

### 前提条件

- 已[创建版本](#)。
- 已获取项目经理、开发人员或测试人员角色权限，权限申请方法请参见[申请权限](#)。
- 发布软件包到运维中心前，需要在运维中心录入并绑定镜像仓，否则会发布失败，具体操作请参见[管理SWR](#)。



## 版本基线化

由项目经理或开发人员进行版本基线化。版本基线化是指在版本发布前，选择特定流水线最近一次的构建产物作为版本基线，将流水线构建产物与版本关联，版本发布审批通过后，开发中心将版本关联的软件包或镜像包信息通知给部署平台，以便后续部署变更。

- 步骤1** 在开发中心首页下方的“我的团队”区域，选择需要操作的团队卡片，进入该团队空间。  
如果该团队关联多个服务，可在顶部切换团队关联的其他服务。也可在顶部切换其他团队。
- 步骤2** 在左侧导航栏选择“版本管理”。
- 步骤3** 在“版本管理”页面的版本列表中，单击版本号，进入版本详情页面。
- 步骤4** 在版本详情页面的“版本持续交付”区域，选择“持续开发 > 版本基线化”。
- 步骤5** 在“版本基线化”页面，单击“添加基线化软件包”（最多可添加20个基线化软件包），弹出“添加基线化软件包”页面。
- 步骤6** 在“选择流水线”下拉列表选择流水线，确认最近一次发布构建信息。
- 步骤7** 单击右下角“基线化”。开发中心后台将记录流水线发布构建的软件包的存储路径，用于版本发布时获取。
- 步骤8** 在版本详情页面的“版本持续交付”区域，选择“持续开发 > 内建质量”。
- 步骤9** 在“内建质量”页面，可查看到当前版本基线化后的流水线代码检查执行结果，如图 12-1 所示，单击代码检查任务名称右侧“查看详情”，进入“检查任务详情”页面可查看更多检查详情信息。

图 12-1 内建质量



内建质量数据来源于版本基线化关联的流水线构建记录中的代码检查数据，通过在代码检查过程中不断地检查和改进，以管控代码质量。

----结束

## 检查标准发布准入信息

- 步骤1** 在开发中心首页下方的“我的团队”区域，选择需要操作的团队卡片，进入该团队空间。

如果该团队关联多个服务，可在顶部切换团队关联的其他服务。也可在顶部切换其他团队。

**步骤2** 在左侧导航栏选择“版本管理”。

**步骤3** 在“版本管理”页面的版本列表中，单击版本号，进入版本详情页面。

**步骤4** 在版本详情页面的“版本持续交付”区域，选择“持续部署发布 > 标准发布”。

在“标准发布检查结果”列表中，不同类型的版本显示相应的检查项的检查规则、检查结论和检查结果，检查项数据来源于组织管理员在版本活动页面设置的标准发布准入配置，具体操作请参见[管理版本标准发布准入配置](#)。

**步骤5** 检查项均检查通过后，可以单击“下一步”，进入“标准发布”页面，继续[申请标准发布](#)。

----结束

## 申请标准发布

可由开发人员、测试人员或项目经理提交版本标准发布申请。

**步骤1** 在“标准发布”页面，“审核信息”区域的“项目经理”下拉列表中选择审批发布的项目经理。

**步骤2** 在“发布信息”区域，选择运维中心站点，（可选）单击“添加文件”上传附件，填写发布内容。

### 说明

- 用户需对自己上传文件的安全风险负责，开发中心不对用户自己上传的文件做任何处理。
- 上传的附件只能为ZIP/RAR/DOCX/DOC/XLS/XLSX文件，且不能超过50MB。
- 在“发布软件包信息”中如无发布软件包，则根据界面提示单击“基线化”跳转至“流水线管理”页面执行该版本流水线基线化生成发布软件包。

**步骤3** 单击“提交”。

“标准发布”页面的左上角将提示“您所提交的发布申请正在审批中，请耐心等待”。该版本发布申请提交将生成一条待办信息至项目经理的AppStage首页的“我的待办”中，由其单击待办链接跳转至标准发布审批页面完成审批操作，具体操作请参见[审批标准发布](#)。

待审批人审批后，刷新“标准发布”页面，可根据不同的审批结果进行后续操作：

- 如果审批通过，“标准发布”页面左上角将提示“恭喜！您所提交的发布申请已通过审批”。  
同时当前页面底部“软件包发布结果”可查看发布结果（单击“查看详情”可查看发布信息详情）：
  - 发布中：部分软件包发布中。
  - 发布成功：发布成功的版本软件包将发布到对应的部署平台（例如AppStage运维中心的“软件仓库 > 软件包管理”）。
  - 发布失败：部分软件包发布失败，可单击“重试”进行发布重试。
- 如果审批未通过，“标准发布”页面左上角将提示“您所提交的发布申请未通过审批（审批意见：xxx），请修改后重新提交”。

----结束

## 审批标准发布

开发人员、测试人员或项目经理提交版本标准发布申请后，生成的待办通知到相应审批人的AppStage首页的“我的待办”区域，由审批人单击此待办信息的链接跳转至标准发布审批页面进行审批。

**步骤1** 审批人（项目经理）登录AppStage。

**步骤2** 在“我的待办”列表中，单击版本标准发布审批的描述，跳转至“标准发布审批”页面。

**步骤3** 在“标准发布审批”页面，查看版本交付概况信息和版本发布信息，在“审批信息”区域，勾选审批结果为“通过”或“不通过”以及输入审批意见，单击“提交”。

- 勾选“通过”：申请人刷新“标准发布 > 发布申请”页面后，页面左上角将提示申请人“恭喜！您所提交的发布申请已通过审批”。版本软件包将发布到对应的部署平台（例如AppStage运维中心的“软件仓库 > 软件包管理”）。
- 勾选“不通过”：申请人刷新“标准发布 > 发布申请”页面后，页面左上角将提示申请人“您所提交的发布申请未通过审批（审批意见：xxx），请修改后重新提交”。

----结束

# 13 查看版本发布制品信息

版本发布会将版本关联的软件包或镜像包发布保存在OBS或SWR中，用户可以在“制品管理”页面查看制品信息。

## 查看版本发布制品信息

**步骤1** 在开发中心首页下方的“我的团队”区域，选择需要操作的团队卡片，进入该团队空间。

如果该团队关联多个服务，可在顶部切换团队关联的其他服务。也可在顶部切换其他团队。

**步骤2** 在左侧导航栏选择“制品管理”。

**步骤3** 在“制品管理”页面查看制品信息。

可以输入版本号或者制品名称搜索需要查看的制品，也可以单击目标制品所在行“操作”列的“查看详情”，如图13-1所示，查看制品详情。


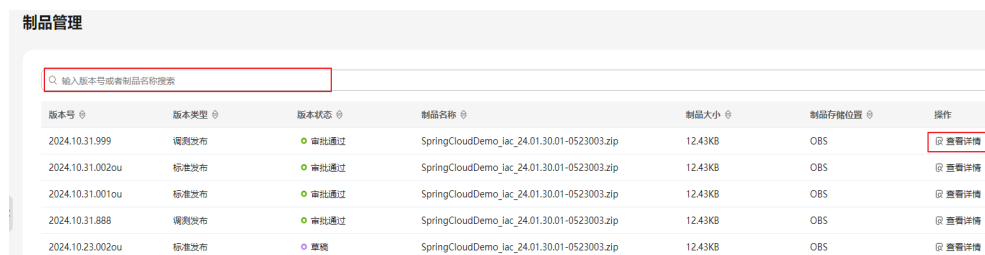
“制品详情”页面还可以查看制品存储路径、关联流水线名称、SHA256等信息，其中SHA256的值可以单击  复制。

图 13-1 制品管理



制品管理

输入版本号或者制品名称搜索

版本号	版本类型	版本状态	制品名称	制品大小	制品存储位置	操作
2024.10.31.999	镜像发布	审核通过	SpringCloudDemo_lac_24.01.30.01-0523003.zip	12.43KB	OBS	查看详情
2024.10.31.002ou	标准发布	审核通过	SpringCloudDemo_lac_24.01.30.01-0523003.zip	12.43KB	OBS	查看详情
2024.10.31.001ou	标准发布	审核通过	SpringCloudDemo_lac_24.01.30.01-0523003.zip	12.43KB	OBS	查看详情
2024.10.31.888	镜像发布	审核通过	SpringCloudDemo_lac_24.01.30.01-0523003.zip	12.43KB	OBS	查看详情
2024.10.23.002ou	标准发布	草稿	SpringCloudDemo_lac_24.01.30.01-0523003.zip	12.43KB	OBS	查看详情

---结束

# 14 使用 AppStage 开发中心知识库管理团队文档

AppStage开发中心知识库是面向开发团队的知识管理系统，支持知识全生命周期管理，帮助您完成智能、安全的知识作业，提供层级式的目录树、多人在线协同编辑等功能。

## 说明

目前此功能仅支持白名单用户通过提交工单申请使用。详细操作请参考[提交工单](#)。

## 前提条件

已获取项目经理、开发人员或测试人员角色权限，权限申请方法请参见[申请权限](#)。

## 使用限制

知识库文档层级最多5层，每层文档最多100个，每个服务最多创建10000个文档。



## 新建文档

**步骤1** 在开发中心首页下方的“我的团队”区域，选择需要操作的团队卡片，进入该团队空间。

如果该团队关联多个服务，可在顶部切换团队关联的其他服务。也可在顶部切换其他团队。

**步骤2** 在左侧导航栏选择“知识库”。

**步骤3** 按照如下方式新建文档，可以单击目录树上方的“我创建的”

- 新增父文档：在知识库页面左侧目录树，单击目录后的，单击“新建文档”。
- 新建子文档：在知识库页面左侧目录树，单击其他已创建文档后的，单击“新建文档”。

----结束

## 编辑文档


**步骤1** 在开发中心首页下方的“我的团队”区域，选择需要操作的团队卡片，进入该团队空间。

如果该团队关联多个服务，可在顶部切换团队关联的其他服务。也可在顶部切换其他团队。

**步骤2** 在左侧导航栏选择“知识库”。

**步骤3** 在知识库页面左侧目录树，选择需要编辑的文档。

**步骤4** 单击文档编辑区域右上方的“编辑文档”，开始编辑。

- 如需插入组件，可以单击“插入”，选择需要的组件拖入文档编辑区域待插入位置。
- 编辑的文档会自动保存。
- 可以单击，进入全屏页面编辑文档。


----结束

## 删除文档

**步骤1** 在开发中心首页下方的“我的团队”区域，选择需要操作的团队卡片，进入该团队空间。

如果该团队关联多个服务，可在顶部切换团队关联的其他服务。也可在顶部切换其他团队。

**步骤2** 在左侧导航栏选择“知识库”。

**步骤3** 在知识库页面左侧目录树，单击文档后的，单击“删除”。

**已删除文档会放入回收站。**

### 说明

如果待删除文档下存在子文档，需删除子文档后该文档才可删除。

----结束

## 删除回收站文档

**步骤1** 在开发中心首页下方的“我的团队”区域，选择需要操作的团队卡片，进入该团队空间。

如果该团队关联多个服务，可在顶部切换团队关联的其他服务。也可在顶部切换其他团队。

**步骤2** 在左侧导航栏选择“知识库”。

**步骤3** 单击知识库页面左侧目录树下方的“回收站”。

**步骤4** 在回收站列表，单击文档后的“删除”。

**步骤5** 在弹出的“删除文档”页面，输入“确定删除”。

**步骤6** 单击“确定”。

### 📖 说明

- 从回收站删除后文档不可恢复，请谨慎操作。

----结束

## 从回收站恢复文档

**步骤1** 在开发中心首页下方的“我的团队”区域，选择需要操作的团队卡片，进入该团队空间。

如果该团队关联多个服务，可在顶部切换团队关联的其他服务。也可在顶部切换其他团队。

**步骤2** 在左侧导航栏选择“知识库”。

**步骤3** 单击知识库页面左侧目录树下方的“回收站”。



**步骤4** 在回收站列表，单击文档后的“恢复”。

----结束

## 更多操作

您还可以对文档进行如[表14-1](#)所示的操作。

表 14-1 更多操作

操作	说明
查看我创建的文档	单击知识库页面左侧目录树上方的“我创建的”。
重命名文档	<ol style="list-style-type: none"><li>1. 在知识库页面左侧目录树，单击文档后的 ，单击“重命名”。</li><li>2. 重新输入文档名称后，单击 。</li></ol>

# 15 在 AppStage 开发中心管理插件

用户在代码开发中时常用到多种插件，除开发中心系统预置插件外，用户还可将自己需要的其他本地插件上传至开发中心进行灵活管理，也可以为插件分类创建标签，便于快速筛选及查找。

组织管理员可以在“插件管理”页面添加或管理插件，其他成员可以在开发插件库中查看插件详情或下载插件。

## 前提条件

如需管理插件，需要获取组织管理员角色权限，权限申请方法请参见[申请权限](#)。

## 添加插件

- 步骤1** 在开发中心首页右上角选择“插件管理”。
- 步骤2** 在“插件管理”页面，单击右上角“添加插件”。
- 步骤3** 在“添加插件”页面，配置如[表15-1](#)所示参数。

表 15-1 添加插件参数说明

参数名称	参数说明
插件LOGO	单击系统默认图片上的“点击修改”，可选择本地图片自定义LOGO图片。
插件名称	自定义插件名称。
插件描述（可选）	输入插件的功能或作用等相关描述。
详情链接	插件所在的具体链接地址。
是否支持下载	可设置为以下两种： <ul style="list-style-type: none"><li>支持：支持下载插件。</li><li>不支持：不支持下载插件。</li></ul>



参数名称	参数说明
上传下载包	只能上传ZIP/RAR文件，且不能超过100MB。 <b>说明</b> 用户需对自己上传文件的安全风险负责，开发中心不对用户自己上传的文件做任何处理。
插件标签（可选）	对插件的某项特征进行的抽象分类和概括，其值（标签值）具备可分类性，用来描述或标记标签的关键词或短语，帮助用户快速地找到相关的插件信息。

**步骤4** 单击“确定”。

新添加的插件显示在“插件管理”页面的插件列表中，且状态为“未启用”。如需显示在“开发插件库”页面，需在“状态”列设置为“启用”。

---结束

## 管理插件

添加插件完成后，还可执行如表15-2所示操作对插件进行管理。

表 15-2 管理插件

操作	说明
编辑插件	<ol style="list-style-type: none"> <li>在“插件管理”页面的插件列表中，单击“操作”列“编辑”。</li> <li>在“编辑插件”页面，参照表15-1可编辑参数配置。</li> <li>单击“确定”。</li> </ol>
删除插件	<p><b>说明</b></p> <ul style="list-style-type: none"> <li>删除插件将清空该插件所有内容，且此操作无法恢复，请谨慎操作。</li> <li>仅支持删除未启用的插件，如需删除，需将插件状态设置为未启用。</li> </ul> <ol style="list-style-type: none"> <li>在“插件管理”页面的插件列表中，单击“操作”列“删除”。</li> <li>在“删除插件”对话框，输入“确定删除”。</li> <li>单击“确定”。</li> </ol>
上移插件	在“插件管理”页面的插件列表中，单击“操作”列“更多 > 上移”，上移该插件在插件列表中的位置。
下移插件	在“插件管理”页面的插件列表中，单击“操作”列“更多 > 下移”，下移该插件在插件列表中的位置。

## 管理标签


**步骤1** 在“插件管理”页面，单击右上角“标签管理”。

**步骤2** 在“标签管理”页面，单击“新增标签”，在标签列表中可自定义标签名称，然后在“操作”列单击“保存”。

**步骤3** 单击“标签管理”页面右下角“关闭”即可。

----结束

## 查看插件详情或下载插件

在开发中心左上角单击，在菜单列表中选择“开发插件库”，进入“开发插件库”页面。

- 单击相应插件卡片右下角的“查看详情”，获取相应开发工具、插件、软件包或详情信息。
- 单击相应插件卡片右下角的“下载”，下载插件到本地。


### 说明

仅支持下载在“插件管理”页面上传的支持下载的插件。

# 16 使用 AppStage 开发中心效能大盘度量研发效能

效能大盘是AppStage集成[看板 \(CodeArts Board\)](#)的能力，为企业管理者、项目经理、团队Leader、开发者提供面向DevSecOps领域端到端的研发效能度量能力，提供从需求、缺陷、代码、构建、测试、部署、发布到运营等研发各阶段作业数据的分析洞察能力，覆盖交付质量、交付效率、交付能力、交付成本、交付价值，同时集成了华为先进的方法论和优秀实践，助力企业数字化转型和数据驱动运营及治理，提升企业软件能力可信和研发效能。

## 操作步骤

在开发中心左上角单击，在菜单列表中选择“效能大盘”，进入“效能大盘”页面。

查看研发效能度量数据，具体操作指导请参考[看板 CodeArts Board](#)。