

测试计划

# 用户指南

文档版本 03  
发布日期 2024-10-15



版权所有 © 华为技术有限公司 2025。保留一切权利。

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

## 商标声明



HUAWEI和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

## 注意

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

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

# 安全声明

## 漏洞处理流程

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

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

如企业客户须获取漏洞信息，请参见如下网址：

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

# 目录

<b>1 测试计划服务(CodeArts TestPlan)使用流程</b> .....	<b>1</b>
<b>2 开通 CodeArts TestPlan 服务</b> .....	<b>3</b>
<b>3 访问 CodeArts TestPlan 服务首页</b> .....	<b>7</b>
<b>4 配置 CodeArts TestPlan 测试计划</b> .....	<b>8</b>
<b>5 配置 CodeArts TestPlan 测试版本</b> .....	<b>13</b>
<b>6 配置 CodeArts TestPlan 测试用例</b> .....	<b>17</b>
6.1 CodeArts TestPlan 生成测试用例方式介绍.....	17
6.2 通过思维导图创建测试用例.....	18
6.2.1 基于 CodeArts TestPlan 思维导图设计测试用例方式.....	18
6.2.2 创建 CodeArts TestPlan 思维导图并生成常规用例.....	21
6.2.3 创建 CodeArts TestPlan 思维导图并生成组合用例.....	31
6.2.4 管理 CodeArts TestPlan 思维导图.....	40
6.3 创建手工测试用例.....	47
6.4 创建接口自动化测试用例.....	48
6.4.1 CodeArts TestPlan 接口自动化用例使用流程.....	48
6.4.2 创建 CodeArts TestPlan 接口自动化测试用例模板.....	49
6.4.3 使用自定义 URL 请求添加 CodeArts TestPlan 接口脚本.....	50
6.4.3.1 添加并设置 CodeArts TestPlan 接口脚本的 URL 请求.....	50
6.4.3.2 设置 CodeArts TestPlan 接口脚本的检查点.....	55
6.4.3.3 设置 CodeArts TestPlan 接口脚本的响应提取.....	68
6.4.4 使用 Postman 快速添加 CodeArts TestPlan 接口脚本.....	69
6.4.5 使用 cURL 快速添加 CodeArts TestPlan 接口脚本.....	70
6.4.6 使用关键字库快速添加 CodeArts TestPlan 接口脚本.....	71
6.4.6.1 CodeArts TestPlan 中关键字库的介绍.....	71
6.4.6.2 将测试步骤保存为 CodeArts TestPlan 接口脚本关键字.....	73
6.4.6.3 将测试步骤保存为 CodeArts TestPlan 组合脚本关键字.....	77
6.4.6.4 刷新 CodeArts TestPlan 关键字.....	79
6.4.7 添加 CodeArts TestPlan 接口脚本的逻辑控制步骤.....	81
6.4.8 设置 CodeArts TestPlan 接口脚本的测试用例参数.....	83
6.4.9 设置 CodeArts TestPlan 接口脚本的环境参数.....	87
6.4.10 导入 CodeArts TestPlan 接口自动化用例测试数据集.....	92

6.4.11 接口自动化用例内置函数说明.....	93
6.4.11.1 在 CodeArts TestPlan 中调用二元加法运算函数.....	93
6.4.11.2 在 CodeArts TestPlan 中调用二元减法运算函数.....	94
6.4.11.3 在 CodeArts TestPlan 中调用二元乘法运算函数.....	96
6.4.11.4 在 CodeArts TestPlan 中调用二元除法运算函数.....	97
6.4.11.5 在 CodeArts TestPlan 中调用获取当前时间戳函数.....	100
6.4.11.6 在 CodeArts TestPlan 中调用获取指定时间戳函数.....	101
6.4.11.7 在 CodeArts TestPlan 中调用日期转时间戳函数.....	103
6.4.11.8 在 CodeArts TestPlan 中调用时间戳转日期函数.....	104
6.4.11.9 在 CodeArts TestPlan 中调用时间戳加减运算函数.....	108
6.4.11.10 在 CodeArts TestPlan 中调用生成 Base64 编码函数.....	110
6.4.11.11 在 CodeArts TestPlan 中调用生成 SHA512 编码函数.....	111
6.4.11.12 在 CodeArts TestPlan 中调用生成 SHA256 编码函数.....	113
6.4.11.13 在 CodeArts TestPlan 中调用生成 MD5 哈希值函数.....	114
6.4.11.14 在 CodeArts TestPlan 中调用生成指定范围随机数函数.....	116
6.4.11.15 在 CodeArts TestPlan 中调用生成指定长度随机字符串函数.....	117
6.4.11.16 在 CodeArts TestPlan 中调用指定范围随机生成小数函数.....	119
6.4.11.17 在 CodeArts TestPlan 中调用生成 UUID 函数.....	121
6.4.11.18 在 CodeArts TestPlan 中调用倒序索引提取数组函数.....	122
6.4.11.19 在 CodeArts TestPlan 中调用倒序索引提取数组元素的值函数.....	123
6.4.11.20 在 CodeArts TestPlan 中调用大写转小写函数.....	123
6.4.11.21 在 CodeArts TestPlan 中调用小写转大写函数.....	125
6.4.11.22 在 CodeArts TestPlan 中调用字符串拼接函数.....	126
6.4.11.23 在 CodeArts TestPlan 中调用字符串切割函数.....	128
6.4.11.24 在 CodeArts TestPlan 中调用获取字符串长度函数.....	129
6.4.12 接口自动化用例系统关键字说明.....	131
6.4.12.1 系统关键字概述与一览表.....	131
6.4.12.2 在 CodeArts TestPlan 接口脚本中调用 GetIAMToken 关键字.....	133
6.4.12.3 在 CodeArts TestPlan 接口脚本中调用 MySQLQuery 关键字.....	136
6.4.12.4 在 CodeArts TestPlan 接口脚本中调用 MySQLUpdate 关键字.....	138
6.4.12.5 在 CodeArts TestPlan 接口脚本中调用 MySQLInsert 关键字.....	139
6.4.12.6 在 CodeArts TestPlan 接口脚本中调用 MySQLDelete 关键字.....	140
6.4.12.7 在 CodeArts TestPlan 接口脚本中调用 OpenGaussQuery 关键字.....	141
6.4.12.8 在 CodeArts TestPlan 接口脚本中调用 OpenGaussUpdate 关键字.....	143
6.4.12.9 在 CodeArts TestPlan 接口脚本中调用 OpenGaussInsert 关键字.....	144
6.4.12.10 在 CodeArts TestPlan 接口脚本中调用 OpenGaussDelete 关键字.....	145
6.4.12.11 在 CodeArts TestPlan 接口脚本中调用 PostgreSQLQuery 关键字.....	146
6.4.12.12 在 CodeArts TestPlan 接口脚本中调用 PostgreSQLUpdate 关键字.....	148
6.4.12.13 在 CodeArts TestPlan 接口脚本中调用 PostgreSQLInsert 关键字.....	149
6.4.12.14 在 CodeArts TestPlan 接口脚本中调用 PostgreSQLDelete 关键字.....	150
6.4.12.15 在 CodeArts TestPlan 接口脚本中调用 MongoDBQuery 关键字.....	152
6.4.12.16 在 CodeArts TestPlan 接口脚本中调用 MongoDBInsert 关键字.....	154

6.4.12.17 在 CodeArts TestPlan 接口脚本中调用 MongoDBUpdate 关键字.....	155
6.4.12.18 在 CodeArts TestPlan 接口脚本中调用 MongoDBDelete 关键字.....	156
6.4.12.19 在 CodeArts TestPlan 接口脚本中调用 RedisGet 关键字.....	157
6.4.12.20 在 CodeArts TestPlan 接口脚本中调用 RedisSet 关键字.....	159
6.4.12.21 在 CodeArts TestPlan 接口脚本中调用 OBSWrite 关键字.....	160
6.4.12.22 在 CodeArts TestPlan 接口脚本中调用 OBSDelete 关键字.....	161
6.4.12.23 在 CodeArts TestPlan 接口脚本中调用 OBSQuery 关键字.....	163
6.4.12.24 在 CodeArts TestPlan 接口脚本中调用 KafkaProducer 关键字.....	164
6.4.12.25 在 CodeArts TestPlan 接口脚本中调用 KafkaConsumer 关键字.....	165
6.4.12.26 在 CodeArts TestPlan 接口脚本中调用 TCP 关键字.....	167
6.4.12.27 在 CodeArts TestPlan 接口脚本中调用 UDP 关键字.....	169
6.4.12.28 在 CodeArts TestPlan 接口脚本中调用 WSConnect 关键字.....	170
6.4.12.29 在 CodeArts TestPlan 接口脚本中调用 WSRequest 关键字.....	171
6.4.12.30 在 CodeArts TestPlan 接口脚本中调用 WSWriteOnly 关键字.....	172
6.4.12.31 在 CodeArts TestPlan 接口脚本中调用 WSReadOnly 关键字.....	173
6.4.12.32 在 CodeArts TestPlan 接口脚本中调用 WSDisConnect 关键字.....	173
6.4.12.33 在 CodeArts TestPlan 接口脚本中调用 DubboClient 关键字.....	174
6.5 执行 CodeArts TestPlan 测试用例.....	178
6.6 管理 CodeArts TestPlan 测试用例.....	180
<b>7 创建与执行 CodeArts TestPlan 测试套件.....</b>	<b>191</b>
7.1 创建 CodeArts TestPlan 测试套件.....	191
7.2 执行 CodeArts TestPlan 测试套件.....	194
<b>8 查看与评估 CodeArts TestPlan 测试质量.....</b>	<b>197</b>
8.1 查看 CodeArts TestPlan 测试质量看板.....	197
8.2 评估 CodeArts TestPlan 测试质量.....	201
<b>9 配置 CodeArts TestPlan 全局配置项.....</b>	<b>207</b>
9.1 配置 CodeArts TestPlan 发送消息通知.....	207
9.2 配置 CodeArts TestPlan 功能开关.....	208
9.3 管理 CodeArts TestPlan 项目成员.....	209
9.4 设置 CodeArts TestPlan 测试用例字段.....	209
9.5 设置 CodeArts TestPlan 测试套件的状态和结果.....	210
9.6 设置 CodeArts TestPlan 测试报告背景图和 Logo.....	210
9.7 设置 CodeArts TestPlan 接口自动化用例请求超时时间、资源池和 DNS 映射.....	211

# 1 测试计划服务(CodeArts TestPlan)使用流程

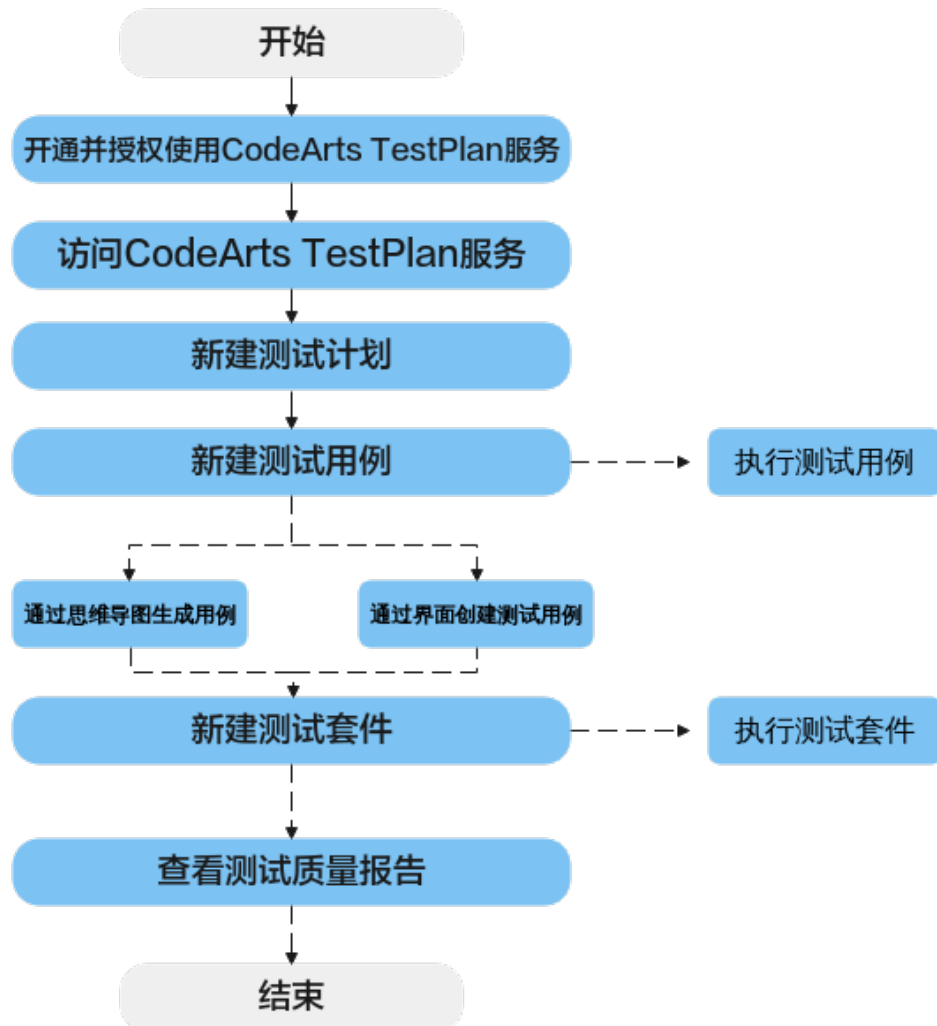
---

测试计划（CodeArts TestPlan）是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。

在[软件开发生产线](#)解决方案中，测试计划服务属于其中一个子服务，具体位置可参考[CodeArts产品构成](#)。

## 测试计划服务基本操作流程

图 1-1 CodeArts TestPlan 基本操作流程





# 2 开通 CodeArts TestPlan 服务

## 前提条件

已注册华为云并实名认证，如果还没有华为账号，请参考以下步骤创建。

1. 打开[华为云网站](#)。
2. 单击“注册”，根据提示信息完成注册。  
注册成功后，系统会自动跳转至您的个人信息界面。
3. 参考[实名认证](#)完成个人或企业账号实名认证。

## 开通 CodeArts TestPlan

可参考[购买CodeArts套餐](#)。

## 添加成员并授权成员角色

项目级权限操作路径：

**步骤1** 单击目标项目名称，进入项目（若无项目，请[新建项目](#)）。

**步骤2** 单击菜单“**设置 > 权限管理**”，进入项目权限管理界面。

---结束

测试计划服务各用户角色类型及对应的操作权限说明如下：

操作/角色	项目经理	测试经理	产品经理	系统工程师	Committer	开发人员	测试人员	运维经理	参与者 浏览者
用例	所有角色都具有创建用例、修改用例、删除用例、执行用例、停止用例、导出用例到文件、查看用例、从文件导入用例、从版本导入用例、规范检查权限。除了产品经理的其他角色具有合并至基线权限。					所有角色都具有创建用例、修改用例、删除用例、执行用例、停止用例、导出用例到文件、查看用例、从文件导入用例、从版本导入用例、规范检查权限。无合并到基线权限		所有角色都具有导出用例到文件、查看用例权限。	
测试套	所有角色都具有创建测试套件、修改测试套件、删除测试套件、查看测试套件、执行测试套件、停止测试套件权限。							所有角色都具有查看测试套件权限。	
测试版本	所有角色都具有查看测试版本权限。除了产品经理的其他角色都具有创建测试版本、修改测试版本、删除测试版本权限。					所有角色都具有查看测试版本权限。			
测试计划	所有角色都具有创建测试计划、修改测试计划、删除测试计划、查看测试计划权限。							所有角色都具有查看测试计划权限。	
测试质量看板	所有角色都具有创建测试质量看板、修改测试质量看板、删除测试质量看板、查看测试质量看板权限。							所有角色都具有查看测试质量看板权限。	
测试质量评估	所有角色都具有创建测试质量评估、修改测试质量评估、删除测试质量评估、查看测试质量评估、下载测试质量评估权限。							所有角色都具有查看测试质量评估、下载测试质量评估权限。	
用例回收站	所有角色都具有在回收站删除、恢复、查看权限。							所有角色都具有在回收站查看权限。	
测试设置	所有角色都具有查看测试设置权限。除了产品经理、开发人员、测试人员其他角色都具有创建、修改、删除权限。							所有角色都具有查看测试设置权限。	

操作/角色	项目经理	项目 经理	测试 经理	产品 经理	系统 工程 师	Co mm itte r	开发 人员	测试 人员	运维 经理	参与 者 浏览 者
接口 测试- 关键字	所有角色都具有创建关键字、查看关键字、删除关键字、编辑关键字权限。								所有角色都具有查看关键字权限。	
全局 变量	所有角色都具有创建全局变量、查看全局变量、删除全局变量、编辑全局变量权限。								所有角色都具有查看全局变量权限。	
测试 设计- 思维 导图	所有角色都具有查看思维导图、创建思维导图、删除（个人创建的思维导图）、修改（个人的思维导图）权限。 除了产品经理、开发人员、测试人员的其他角色都具有删除（全部思维导图）、修改（全部思维导图）权限。								所有角色都具有查看思维导图权限。	
测试 设计- 思维 导图 备份	所有角色都具有查看思维导图备份、创建思维导图备份、删除（个人的思维导图备份）、恢复（个人的思维导图备份）权限。 除了产品经理、开发人员、测试人员的其他角色都具有删除（全部思维导图备份）、恢复（全部思维导图备份）权限。								所有角色都具有查看思维导图备份权限。	
测试 设计- 思维 导图 模板	所有角色都具有查看思维导图模板、创建思维导图模板、删除（个人的思维导图模板）、修改（个人的思维导图模板）权限。 除了产品经理、开发人员、测试人员的其他角色都具有删除（全部思维导图模板）、修改（全部思维导图模板）权限。								所有角色都具有查看思维导图模板权限。	
测试 设计- 回收 站	所有角色都具有在回收站查看删除的思维导图、删除（个人的思维导图）、恢复（个人的思维导图）权限。								所有角色都具有在回收站查看思维导图、删除（个人的思维导图）权限。	
接口	所有角色都具有创建接口、查看接口、修改接口、删除接口、拷贝接口、导入接口权限。								所有角色都具有查看接口权限。 项目参与者具有创建接口、修改接口、删除接口、拷贝接口、导入接口权限。	

操作/ 角色	项目 管理员	项目 经理	测试 经理	产品 经理	系统 工程 师	Co mm itte r	开发 人员	测试 人员	运维 经理	参与 者 浏览 者
<b>规则</b>	所有角色都具有创建规则、查看规则、修改规则、删除规则权限。								所有角色都具有查看规则权限。 项目参与者具有创建规则、修改规则、删除规则权限。	
<b>实例</b>	所有角色都具有部署实例、开始实例、停止实例、更新实例、查看实例权限。 除了测试人员的其他角色都具有删除实例权限。								所有角色都具有查看实例权限。 项目参与者具有部署实例、开始实例、停止实例、更新实例、查看实例权限。	

### 说明

如需要修改当前角色的权限，请联系项目管理员进行操作权限的编辑。

自定义角色无预置权限，可以联系项目管理员添加当前角色类型对应资源的相应操作权限。

# 3 访问 CodeArts TestPlan 服务首页

进入测试服务有两种方式：[项目首页入口](#)和[导航入口](#)。

## 项目首页入口

通过项目首页入口，进入测试服务，默认展示“测试计划”页面。通过导航可以切换至“测试设计”、“测试用例”、“测试执行”、“测试质量看板”、“测试质量评估”页面，对当前项目的测试活动进行管理。

**步骤1** 登录软件开发生产线首页。

**步骤2** 鼠标悬浮在项目卡片上，在卡片底部单击“测试”，进入测试计划服务，默认展示“测试计划”页面。


----结束

## 导航入口

通过导航入口，进入测试服务，默认展示测试计划首页。页面展示当前登录用户的统计数据，包括测试计划、测试用例、测试套件、缺陷等。同时也可以在该页面选择项目后进入项目，对所选项目的测试活动进行管理。

**步骤1** 登录软件开发生产线首页。

**步骤2** 可以选择以下两处导航进入测试计划服务。

- 单击页面上方导航“服务 > 测试计划”。
- 单击页面左上角 ，在页面左侧滑出的窗口中单击“测试计划”。

----结束

# 4 配置 CodeArts TestPlan 测试计划

标准的测试过程包括测试计划、测试设计、测试执行、测试报告几个主要阶段。

- 在测试计划和测试设计阶段，要明确测试范围和测试目标、制定测试策略、准备测试工具和测试环境、建立测试模型、设计测试用例、开发自动化测试脚本。
- 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。

## 新建测试计划

### 说明

Scrum项目Epic/Feature/Story类型的工作项及看板项目默认的“需求”类型工作项才可以作为需求被添加至测试计划。

**步骤1** 登录软件开发生产线首页，搜索目标项目并单击项目名称，进入项目。

**步骤2** 单击导航栏“测试 > 测试计划”。

**步骤3** 单击页面上方“新建计划”，进入“新建测试计划”页面。

**步骤4** 配置以下信息：名称、待测版本号（可选）、选择处理者、计划周期、关联迭代（可选）、描述（可选），单击“下一步”。

配置项	是否必填	描述
名称	是	测试计划的名称。可以描述计划对应的测试场景或功能。 名称长度为3~128个字符且不能以空格开头，只能包含中文、英文字母、数字和符号(空格, _/ *&`^~; ; :: () ( ) {} = +, x ¥ …… —— ’ !@#%\$. \ [ ] < > ? ! "、 ‘ “ ” ? 一。 , 《 》 [ ] \) !
版本号	否	为测试计划设置版本号。 版本号长度为3~32个字符（可以输入数字，字母，中文，中划线，下划线，英文句点）。
处理者	是	将项目中的成员设置为当前测试计划的处理者。
计划周期	是	选择测试计划的开始日期和截止日期。

配置项	是否必填	描述
关联迭代	否	为测试计划关联当前项目中的迭代。
描述	否	对计划的简要描述，不超过1000个字节。

**步骤5** 按需要勾选执行方式、添加需求，单击“保存”完成测试计划的创建。

- 此处选择的执行方式，后续可在测试计划中修改。
- 选择执行方式后，将在测试用例、测试执行页面中生成对应名称的菜单，管理手工测试、接口自动化用例和套件，并且在质量报告中体现对应执行方式的预置统计报表。

----结束

## 编辑测试计划

在测试计划列表中，单击需要编辑的测试计划名称，页面右侧将滑出编辑窗口，可对测试计划进行编辑。

- 页签“详情”中，可以编辑测试计划（可以编辑测试计划的名称、编辑描述、选择执行方式、编辑基本信息），编辑完毕单击页面右上方“保存”。
- 页签“需求”中，可以添加、移除当前测试计划范围的需求，操作方式与新建测试计划里添加、移除需求相关步骤相同。
- 页签“测试用例”中，可以查看该计划内的测试用例，也可以添加测试计划所属版本的测试用例。
- 页签“操作历史”中，可以查看测试计划的编辑历史。

## 设计测试计划

设计测试计划就是根据测试计划确定的测试需求设计测试用例、开发自动化测试脚本、准备测试数据。

**步骤1** 登录软件开发生产线首页，搜索目标项目并单击项目名称，进入项目。

**步骤2** 单击导航栏“测试 > 测试计划”。

**步骤3** 在列表选择需要设计的测试计划。

- 鼠标放在“设计”上，查看测试计划的设计进展：用例数、需求总数、已覆盖需求数。  
当用例数>0时，“设计”之上的圆点由灰色变为蓝色，该测试计划处于设计中，例如下图。
- 单击“设计”，进入“测试用例”页面。

图 4-1 进入测试用例页面



**步骤4** 在“手工测试”页签中，单击页面右侧“导入”，在下拉菜单中单击“添加已有用例”。

**说明**

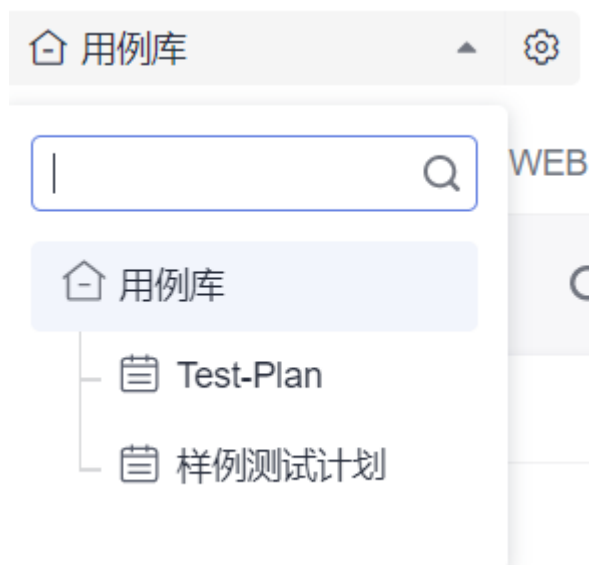
若尚未创建测试用例，或需要创建新的测试用例，可单击“新建用例”来创建测试用例，详细操作请参考[新建手工测试用例](#)，新建的用例同时也被添加至测试用例库中。

**步骤5** 在弹框中勾选测试用例，单击“确定”完成测试用例的添加。

**步骤6** 单击“接口自动化”页签，可向测试计划中添加或新建接口自动化测试用例。添加用例的方法与**步骤4**相同，创建用例的方法请参照[新建接口自动化测试用例](#)。

**步骤7** 单击页面左上方测试计划名称，可以切换测试计划，也可以查看全局用例库。

全局用例库展示了当前版本下的所有测试用例（包括属于或者不属于测试计划的），可根据需要维护全局用例库。



----结束




## 执行测试计划

**步骤1** 返回“测试计划”页面，在列表选择需要执行的测试计划。

- 鼠标放在“执行”上，查看测试计划的执行进展：测试进度、已执行用例、通过率、已完成缺陷/总缺陷。  
当执行用例数>0时，“执行”之上的圆点由灰色变为蓝色，该测试计划处于测试中。
- 单击“执行”，进入“测试执行”页面。



**步骤2** 在“手工测试”页签中，可以通过测试套件执行多个已创建的测试用例，单击操作栏的图标 ，执行手工测试套件，如果没有测试套件请参考[新建手工测试套件](#)。

**步骤3** 单击“接口自动化”页签，可执行接口自动化套件，详细操作请参考[执行接口自动化套件](#)。

**步骤4** 用户将测试计划中的所有测试用例状态手动设置为“完成”后，测试计划的状态自动更新为“完成”。

### 说明

用户在测试计划页面中单击“执行”，进入“测试执行”，可以通过创建测试套件执行测试计划中的用例。

----结束

## 管理与度量测试计划

**步骤1** 登录软件开发生产线首页，搜索目标项目并单击项目名称，进入项目。

**步骤2** 单击导航栏“测试 > 测试计划”。

**步骤3** 在列表选择需要查看报告的测试计划，单击“报告”。



**步骤4** 查看测试计划的质量报告。

- 页面展示测试计划当前的需求覆盖率、缺陷、用例通过率、用例完成率，并分析记录测试风险。
- “手工测试”、“接口自动化”部分，按执行方式统计测试用例执行情况和缺陷数量。
- 单击左下方“点击添加报表”，可以在页面中添加更多报表，也可以通过单击右上角“新建报告”来添加测试报告。
- 单击页面左上方测试计划名称，可以切换测试计划，也可以查看全局用例库的质量报告。

----结束

# 5 配置 CodeArts TestPlan 测试版本

CodeArts TestPlan支持对测试用例进行基线化管理，提供了“产品基线库、版本分支、测试计划”三层用例管理和评估能力，支撑团队间作业的高效协同，实现亿级测试资产的管理与跨版本复用，满足复杂产品多版本并行研发交付的场景。

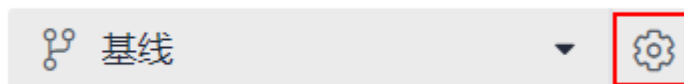
- 版本：每个版本记录了某个特定软件版本的测试用例情况、用例执行情况、测试套情况，测试套执行情况、质量报告等信息。
- 版本和测试计划的关系：一个版本包含一个用例库和多个测试计划。
- 基线：基线是经过正式评审后的特殊版本，是后续测试活动的基础，基线中的用例一般比较稳定，是长期积累的测试资产。
- 测试用例分层管理：是指可以从“产品基线库-版本分支-测试计划”的三层用例分层结构对用例进行管理。在用例管理中，支持从版本合并用例到基线，支持从基线导入用例到版本，也支持版本间用例合并的冲突管理等，可以满足多版本并行测试和多人协同测试，解决公司测试资产积累和复用问题。

## 新建版本

**步骤1** 登录服务首页，搜索目标项目并单击项目名称，进入项目。

**步骤2** 单击导航栏“测试> 测试用例”。

**步骤3** 单击“基线”右侧进入“版本管理”页面。




**步骤4** 在“版本管理”页面单击“+添加”。


**步骤5** 输入版本名称，单击“保存”。

----结束


## 编辑版本

在“版本管理”页面中，鼠标悬浮在需要编辑的测试版本上，将显示图标，单击该图标即可对版本名称进行编辑。

## 删除版本

在“版本管理”页面中，在对应版本的“操作”列中，单击即可对该版本进行删除。

## 查看用例变更

在“版本管理”页面中，在对应版本的“操作”列中，单击可以查看对应版本下的用例变更历史。

## 管理版本

管理版本是根据测试策略将基线中的用例导入到版本，或者将版本中的用例合并到基线的过程。

- 从其他版本导入测试用例

**步骤1** 登录服务首页，搜索目标项目并单击项目名称，进入项目。

**步骤2** 单击导航栏“测试>测试用例”。

**步骤3** 在页面左上角版本下拉框中选择一个版本。



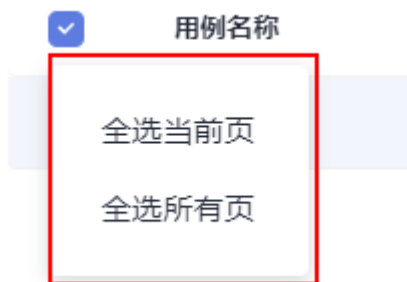
**步骤4** 单击“手工测试”页签，单击页面右侧“导入”，在下拉列表中单击“从版本导入”。

### 说明

若尚未创建测试用例，或需要创建新的测试用例，可单击“新建用例”来创建测试用例，详细操作请参考[创建手工测试用例](#)。

**步骤5** 在弹框中勾选需要导入的用例并选择覆盖规则，单击“确定”，完成导入其他版本的用例。

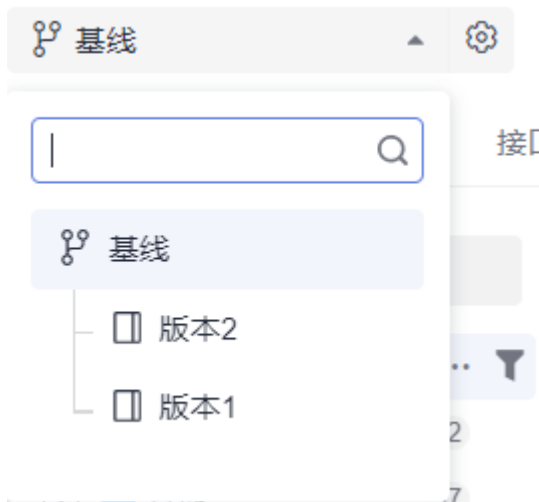
鼠标悬浮在“用例名称”左侧的勾选框上，可以选择将当前页下的用例或对应版本的所有用例导入到当前版本。



----结束

**合并版本中测试用例到基线（测试用例页面）**

- 步骤1** 登录服务首页，搜索目标项目并单击项目名称，进入项目。
- 步骤2** 单击导航栏“测试>测试用例”。
- 步骤3** 在页面左上角版本下拉框中选择一个版本。



- 步骤4** 选择执行方式的页签，单击页面右侧“合并到基线”。

**说明**

若尚未创建测试用例，或需要创建新的测试用例，可单击“新建用例”来创建测试用例。

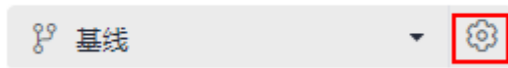
- 步骤5** 在弹框中勾选需要合并的测试用例并选择覆盖规则，单击“确定”完成测试用例的合并。


鼠标悬浮在“用例名称”左侧的勾选框上，可以合并当前页的用例或当前版本的全部手工测试用例到基线。

----结束

**合并版本中全部测试用例到基线（通过版本管理）**

- 步骤1** 登录服务首页，搜索目标项目并单击项目名称，进入项目。
- 步骤2** 单击导航栏“测试>测试用例”。
- 步骤3** 单击“基线”右侧⚙️进入“版本管理”。



**步骤4** 在“版本管理”页面中，选择想要合并用例到基线的版本，单击页面右侧 。

**步骤5** 在弹框中选择规则，单击“确定”完成测试用例的合并。

----结束

## 度量版本

**步骤1** 登录服务首页，搜索目标项目并单击项目名称，进入项目。

**步骤2** 单击导航栏“测试 > 测试质量看板”，进入“测试质量看板”页面。

**步骤3** 查看版本的测试质量看板。

在页面左上角版本下拉框中可以切换版本，再次单击测试计划下拉列表，可以切换测试计划，查看测试质量看板。



----结束

# 6 配置 CodeArts TestPlan 测试用例

## 6.1 CodeArts TestPlan 生成测试用例方式介绍

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。

CodeArts TestPlan内置手工测试、接口自动化测试、性能自动化测试以及自定义自动化测试，用户可以在对应执行方式下创建用例。

用户可以通过设计思维导图的方式生成测试用例，也可以在测试用例界面直接创建测试用例。以下是不同测试用例的相关概念：

- 用例库：用于管理当前项目以及对应版本中所有类型的用例，可参考[用例库](#)。
- 手工测试用例：基于某种测试场景，创建测试步骤，对各个步骤分布设置预期结果，所有步骤执行完成后可以设置用例的结果及完成状态。
- 接口自动化用例：通过自动化执行预定义的接口来验证系统的功能和性能，基于接口设计规范，通过模拟用户操作和输入数据，检查系统是否能正确地处理请求和返回正确的响应。

### 用例库

在用例库中可以查看、管理及使用项目中当前版本的所有测试用例。

- 可以添加用例库的用例至测试计划中。
- 在测试计划中仅能查看、管理及使用当前测试计划中的测试用例。
- 在测试计划中创建的用例会出现在用例库中。

查看用例库方式如下：

**步骤1** 登录软件开发生产线首页，搜索目标项目并单击项目名称，进入项目。

**步骤2** 单击导航栏“测试 > 测试用例”，默认显示“用例库”。

用户可以在用例库所在的下拉列表中，选择测试计划，查看对应测试计划中的用例详情。

----结束

## 基于特性目录生成测试用例

用户可以在“特性目录”中，创建自定义的特性文件夹，在该文件夹下创建测试用例。

**步骤1** 登录软件开发生产线首页，搜索目标项目并单击项目名称，进入项目。

**步骤2** 单击导航栏“测试 > 测试用例”。

**步骤3** 单击页面左侧“特性目录”。

- 所有的用例都默认归属在特性目录中，特性目录中默认存在一个子目录“其他”，单击特性目录右侧的...可以新建子目录。单击新建的子目录旁...可将该子目录进行删除、重命名，也可以在新建的子目录中新建测试用例、新建子目录。

**步骤4** 单击目标文件夹，选择用例的执行方式页签，单击“新建用例”，创建成功的测试用例将保存在该文件夹下。

----结束

## 基于需求目录生成测试用例

当项目已关联相关需求，用户可以在“需求目录”中的对应需求下，创建测试用例。

**步骤1** 登录软件开发生产线首页，搜索目标项目并单击项目名称，进入项目。

**步骤2** 单击导航栏“测试 > 测试用例”。

**步骤3** 单击页面左侧“需求目录”。项目已关联的需求都默认归属在“需求目录”下。

在创建的项目中添加需求，可参考[新建工作项](#)。

**步骤4** 单击“需求目录”下的某个需求，可以查看与该需求关联的所有用例，也可以选择用例的执行方式页签，单击“新建用例”，新建成功的用例默认关联所选需求。

**步骤5** 单击需求名称右侧的...，可以查看需求详情，也可以新建与所选需求关联的测试用例。

----结束

## 6.2 通过思维导图创建测试用例

### 6.2.1 基于 CodeArts TestPlan 思维导图设计测试用例方式

CodeArts TestPlan提供了多维度测试策略和设计模板，应用启发式测试策略和设计模型，提供“需求-场景-测试点-测试用例”四层测试分解设计能力，启发测试人员发散性思维，将脑海中的测试模型图形化地表达出来，支撑所有利益干系人高效沟通与交流。通过该特性，能够持续优化测试完备度，帮助测试人员在执行过程中减少产品测试场景的遗漏。

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。

借助思维导图可以进行启发式测试设计并把设计过程可视化。



## 基于需求设计思维导图


测试计划服务支持为需求新建思维导图，从需求出发，分解测试场景、分析测试点、输出测试方案和测试用例。

**步骤1** 搜索目标项目并单击项目名称，进入项目。

**步骤2** 单击左侧导航栏的“测试 > 测试设计”。

**步骤3** 单击页面左侧“需求目录”。项目已关联的需求都默认归属在“需求目录”下。

在创建的项目中添加需求，可参考[新建工作项](#)。

**步骤4** 在“需求目录”下的搜索框内，输入需要目标需求的关键字，单击，目标需求将显示在需求目录下。

**步骤5** 在页面左侧“需求目录”中选中一个需求，单击页面左上角“普通新建”或“模板新建”。

**步骤6** 进入设计思维导图页面，后续操作请参考[创建CodeArts TestPlan思维导图并生成常规用例](#)。


----结束

## 基于特性设计思维导图

测试设计服务支持为特性新建思维导图，从特性出发，分解测试场景、分析测试点、输出测试方案和测试用例。

**步骤1** 登录软件开发生产线首页，搜索目标项目并单击项目名称，进入项目。

**步骤2** 单击导航栏“测试 > 测试设计”。


**步骤3** 选择“特性目录”，在“特性目录”下的搜索框内，输入需要目标特性的关键字，单击。

**步骤4** 选中一个特性目录，单击页面“普通新建”或“模板新建”。

**步骤5** 进入设计思维导图页面，后续操作请参考[创建CodeArts TestPlan思维导图并生成常规用例](#)。

----结束

### 修改思维导图所属特性目录

**步骤1** 在测试设计列表中，鼠标悬浮在思维导图所在行的勾选框前，当显示时，可以拖拽导图至左侧的“特性目录”，修改测试设计归属目录。



**步骤2** 拖拽后的思维导图将显示在新归属的目录下。

用户可以全选或勾选测试设计列表中的思维导图，单击“修改所属特性”，在弹框中选择目标归属目录，可以批量修改思维导图所属的特性目录。

#### 📖 说明

思维导图只支持修改所属特性目录，不支持修改所属需求目录。

#### ---结束

### 为用例关联需求

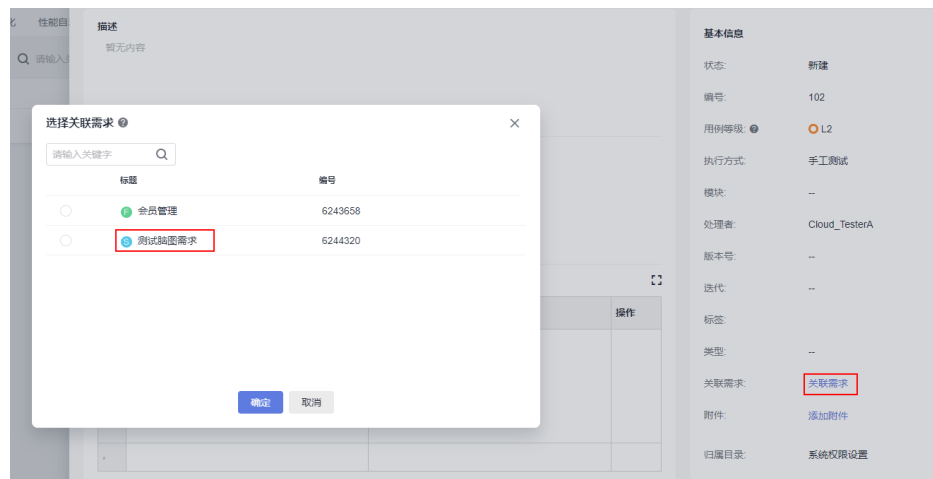
在特性目录中设计并归档的用例，如果未与需求关联，通过以下步骤可以为用例关联需求。

**步骤1** 进入基于特性目录生成的思维导图。

**步骤2** 找到已完成归档且需要关联需求的用例，单击🔗。

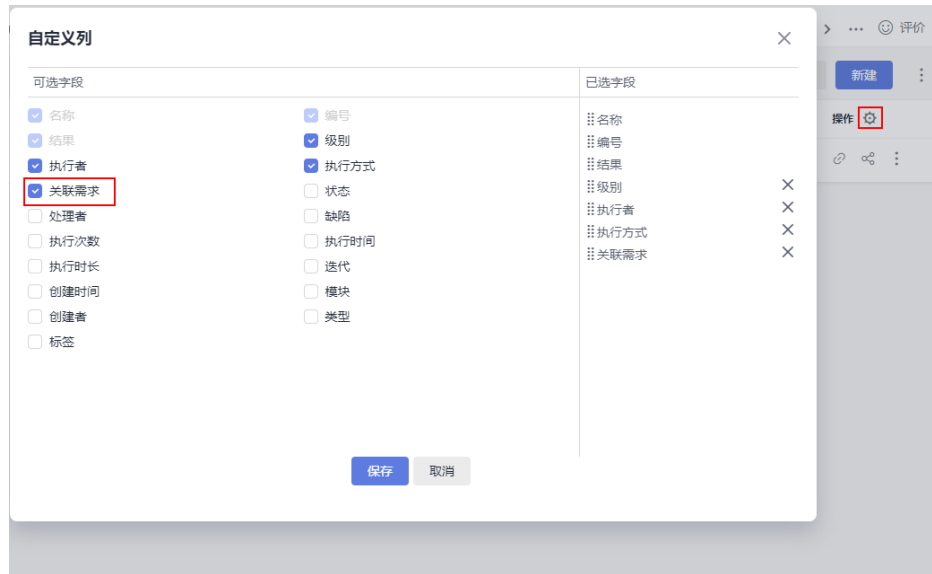
**步骤3** 在页面右侧单击“关联需求”，在弹框中勾选需求，单击“确定”

**步骤4** 单击右上角“保存”。



**步骤5** 单击导航栏“测试 > 测试用例”，在页面中找到用例，可看到列表中对“关联需求”一列中的值为“已关联”。

如果列表中没有“关联需求”列，请单击右上方的齿轮，进行自定义列设置，勾选“关联需求”，单击“确定”。



----结束

## 6.2.2 创建 CodeArts TestPlan 思维导图并生成常规用例

思维导图又称脑图，用于规划测试方案、设计测试场景、定义测试点、编排测试步骤、生成测试用例等。在测试计划服务的“测试设计”页面中可以使用思维导图功能。

### 前提条件

用户需具有创建思维导图的权限（即在项目内的角色：除了浏览者、参与者、运维经理的其他角色），请参考[添加成员并授权成员角色](#)。

### 约束与限制

思维导图中单节点的描述最大不超过2000个字符。

### 新建思维导图

#### 普通新建


**步骤1** 搜索目标项目并单击项目名称，进入项目。

**步骤2** 单击导航栏“测试 > 测试设计”。

**步骤3** 单击页面左上角“普通新建”。

页面跳转至新创建的思维导图页面，页面正中显示为根节点，根节点的名称自动填充为“思维导图”。双击根节点可修改根节点名称。

**步骤4** 用户可以单击页面左上方，在弹框中输入思维导图的名称，单击“确定”。

**步骤5** 单击页面左上角，页面跳转回测试设计列表。列表中显示已创建的思维导图名称。

----结束

#### 模板新建

- 步骤1** 登录软件开发生产线首页，搜索目标项目并单击项目名称，进入项目。
- 步骤2** 单击导航栏“测试 > 测试设计”。
- 步骤3** 单击页面左上角“普通新建”右侧的▾，单击“模板新建”。
- 步骤4** 根据需要选择模板，单击“预览”可以查看该思维导图详情，单击“立即使用”进入导图。
- 步骤5** 页面跳转至到新创建的思维导图页面，显示所选模板详情。
- 步骤6** 单击页面左上角←，页面跳转回测试设计列表。
- 步骤7** 在测试设计列表中，单击操作列中的\*\*\*，在下拉栏中单击“新标签页打开”，可以分多个浏览标签页打开多个脑图。

----结束

### 重命名

- 步骤1** 在测试设计列表中，单击操作列中的\*\*\*，在下拉栏中单击“重命名”。
- 步骤2** 弹框中将显示思维导图上一个旧名称，用户可以输入不超过500字符的新名称。
- 步骤3** 单击“确定”，完成对思维导图的重命名。

----结束

## 绘制思维导图

完成新建思维导图操作后，在“测试设计”页面单击需要编辑的思维导图名称，即可进行思维导图的绘制。

### 新增节点

新增节点包括新增邻节点（即同级节点）与新增子节点。选中思维导图中任一节点，可根据需要选择新增节点的类型。根节点只能新增子节点，其它节点可以新增邻节点和子节点。

- 新增子节点
  - 进入已创建的思维导图，选中任意节点，通过以下方式新增子节点。
    - 单击鼠标右键，在页面下拉列表中选择“新增子节点(Ins)”。
    - 单击键盘上的“Insert”或“Tab”。



- 新增邻节点
 

进入已创建的思维导图，选中除根节点外的任意节点，通过以下方式新增邻节点。


  - 单击鼠标右键，在页面下拉列表中选择“新增邻节点（Enter）”。
  - 单击键盘上的“Enter”。



### 删除节点



在思维导图中可删除根节点以外的任意节点，若被删除的节点含有子节点，那么子节点也会随之删除。

进入已创建的思维导图，选中除根节点外的任意节点，通过以下方式删除节点。

- 敲击鼠标右键，在页面下拉列表中选择“删除节点（Del）”。
- 单击页面右上方 ，并在下拉列表中选择“删除当前节点”。
- 使用快捷键：敲击键盘上的“Delete”。

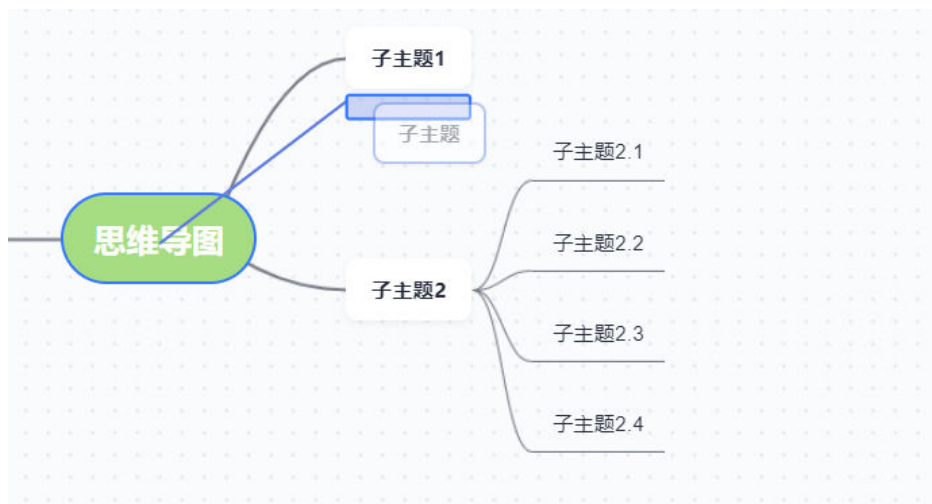
### 移动节点

- 上移/下移

进入已创建的思维导图，选中待移动节点，单击思维导图上方工具栏中  或 ，可在同级之间上下移动节点。

- 拖拽节点

进入已创建的思维导图，选中待移动节点，鼠标左键单击节点不松开，拖拽到需要的位置，松开鼠标左键。



## 场景设计

完成新建思维导图后，可以针对此需求进行场景设计。

**步骤1** 进入已创建的思维导图。

**步骤2** 为根节点新增一个子节点（可以为任意非根节点添加场景，）。

**步骤3** 输入场景的描述，本章节以“注册会员”为例。

**步骤4** 选中**步骤2**中创建的节点，单击思维导图上方工具栏中 **SC** 。

当被选中的节点前出现 **SC**，表示添加场景成功。



---结束

## 测试点设计

完成新建思维导图后，可以针对此需求进行测试点设计。

**步骤1** 进入已创建的思维导图。

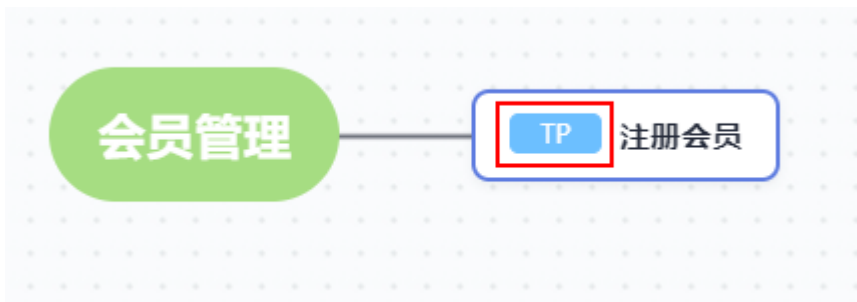
**步骤2** 为根节点新增一个子节点（可以为任意非根节点添加测试点）。

**步骤3** 输入子节点名称，本章节以“注册会员”为例。

测试点的名称应符合1到128位长度，可以包含中文、大小写字母、数字、特殊字符（- \_/!\*&'^~;; :: () ( ) {}=+,x¥……——’ !@#%&.'[]<>?! "、‘“”？—。 ， 《 》 【 】）的组合格式。

**步骤4** 选中**步骤2**中创建的节点，单击思维导图上方工具栏中 **TP** 。

当被选中的节点前出现 **TP**，表示添加测试点成功。



----结束

## 用例设计

完成测试点设计后，可以进一步细化成用例。

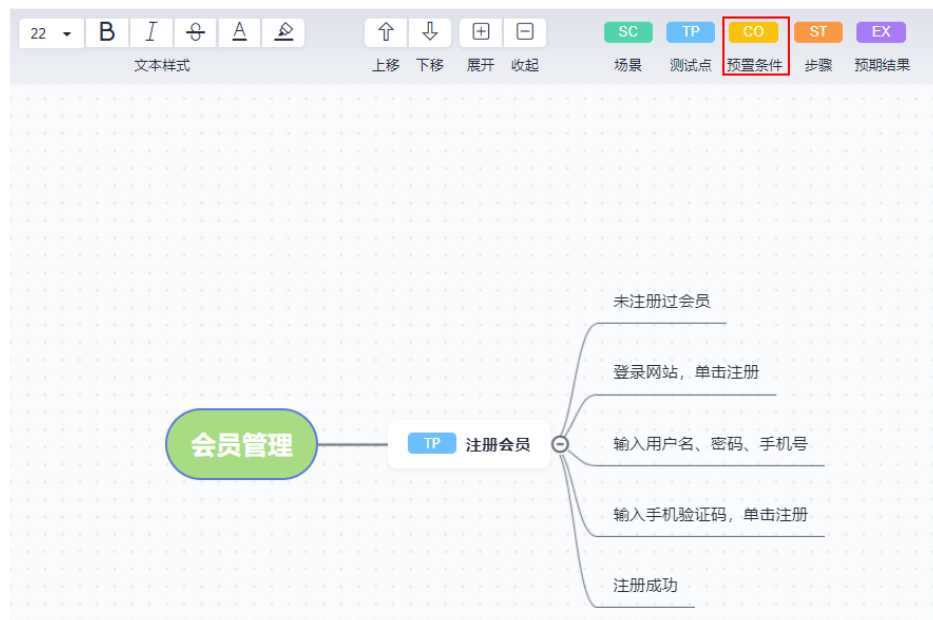
用例包括预置条件、步骤、预期结果三个部分，进行用例设计时分别设置这三部分内容。

**步骤1** 进入已创建的思维导图。

**步骤2** 在“注册会员”节点下按需新建子节点预置条件、步骤、预期结果，根据需要新建子节点。

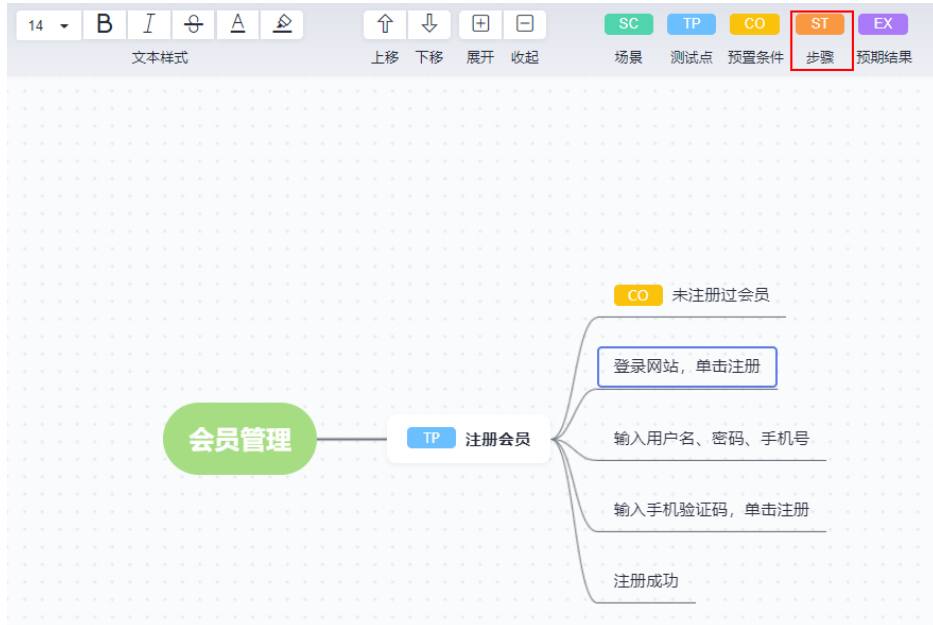
**步骤3** 选中作为预置条件的节点，单击思维导图上方工具栏中 **CO**。

当节点中出现 **CO** 时，说明设置成功。



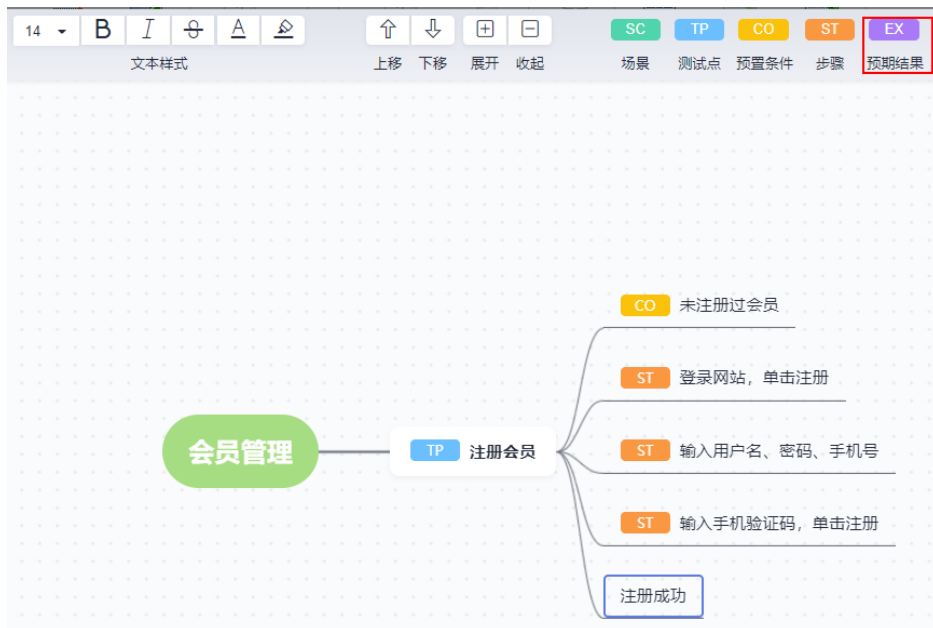
**步骤4** 选中作为步骤的节点，单击思维导图上方工具栏中 **ST**。

当节点中出现 **ST** 时，说明设置成功。



**步骤5** 选中作为预期结果的节点，单击思维导图上方工具栏中 **EX**。

当节点中出现 **EX** 时，说明设置成功。



----结束

## 添加标签

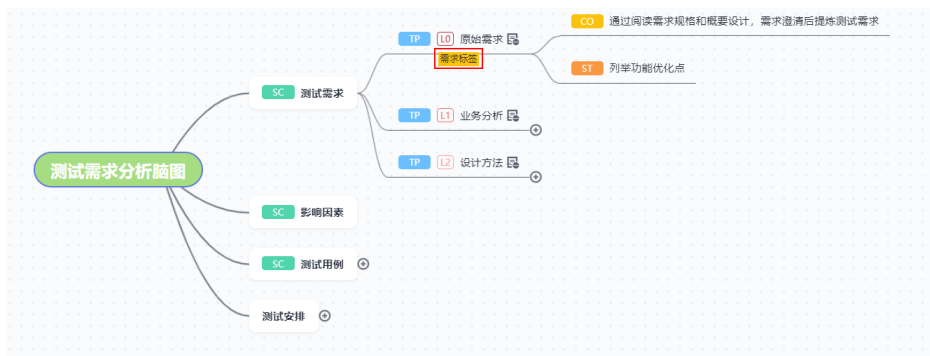
在测试设计过程中，可以根据需要为节点添加标签，作为测试设计的辅助工具。

**步骤1** 进入思维导图，选择一个节点，鼠标右键单击该节点，在下拉列表中选择“添加标签”。



**步骤2** 根据需要在节点下方的输入框中输入标签内容，单击屏幕任意位置，即可保存。

**步骤3** 保存成功后，节点下方将出现黄色背景的标签。




----结束

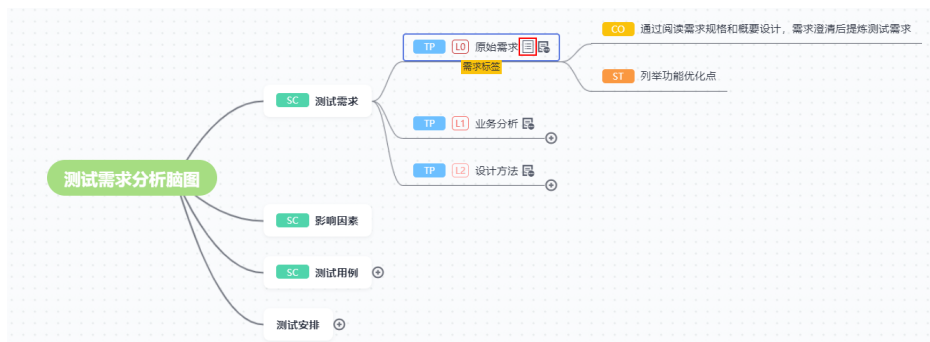
## 添加备注

在测试设计过程中，可以根据需要为节点添加备注，作为测试设计的辅助工具。

**步骤1** 进入思维导图，选择一个节点，单击页面上方“备注”。

**步骤2** 在页面右侧滑出的窗口中输入备注信息，单击屏幕任意位置，即可保存。

**步骤3** 节点中将出现  图标，表示节点成功添加了备注信息。



----结束

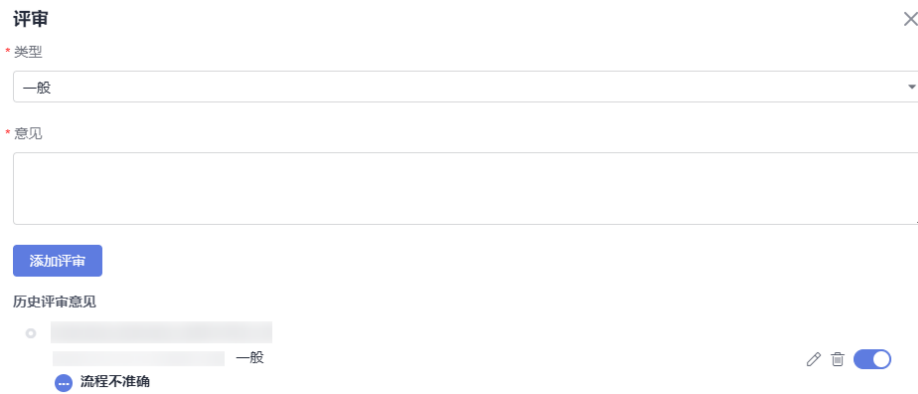
## 在线评审


在设计思维导图中，可以对任一节点进行在线评审，添加评审意见。

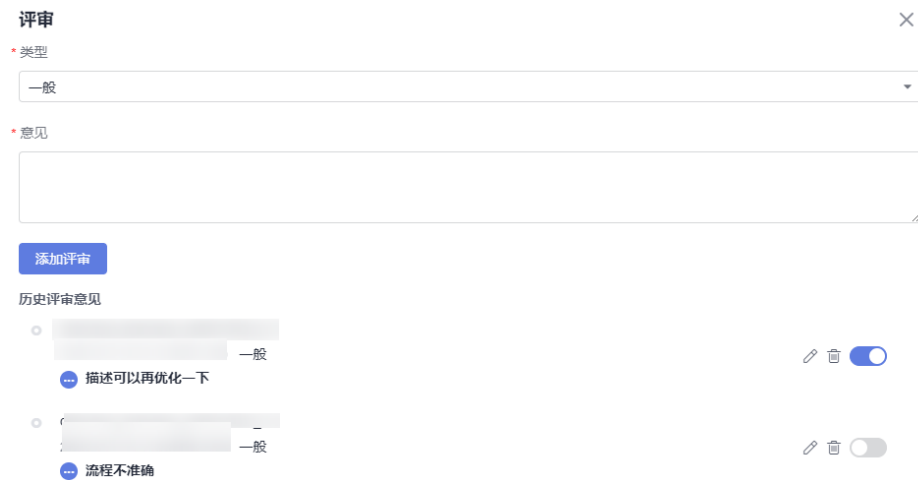
**步骤1** 进入待评审的思维导图，选择一个用例节点，单击鼠标右键，在下拉栏中选择“添加评审”。

**步骤2** 在弹框中选择类型，并输入意见，单击“添加评审”。

**步骤3** 添加成功，在弹框中的“历史评审意见”中可查看到评审记录。



**步骤4** 单击弹框中评审意见右侧的  可以修改评审意见的类型、意见。  
单击“评审意见”右侧的开关按钮，可以将已确认好的评审关闭。




----结束


## 生成用例

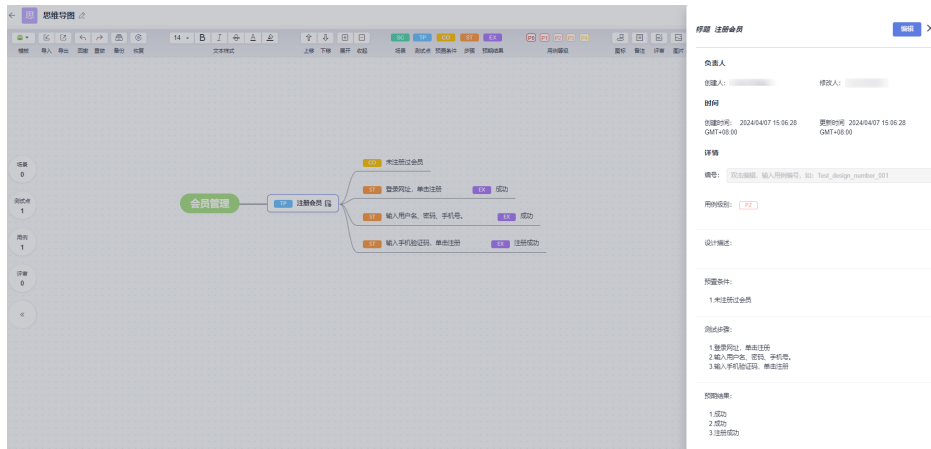
完成用例设计后，可将思维导图生成一条真正的测试用例。

### 生成单个用例

**步骤1** 在思维导图中，选中待生成用例的TP节点，鼠标右键单击该节点，在下拉列表中选择“生成用例”。

**步骤2** 当节点中出现时，说明操作成功，此时生成的是草稿用例。

单击，页面右侧将滑出用例详情。



用户可以根据需要设置是否按一个步骤对应一个预期结果，请参考[功能开关](#)。

### 📖 说明

1. 生成测试步骤时，只读取打上TP标签节点的第一层子节点，顺序根据ST的标签按照从上到下，依次递增。
2. 若ST标签没有设置EX预期结果，对应序号的预期结果为空。
3. 如果在测试步骤节点下标注了多个预期结果，只读取第一个。

### ----结束


### 批量生成用例

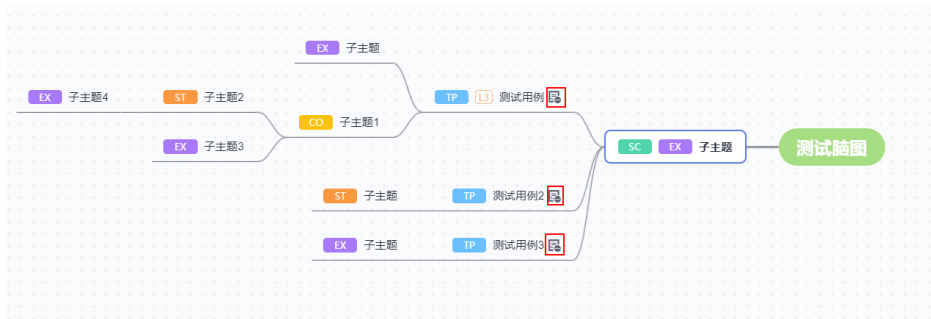
当一个场景下设置了多个测试点，可以通过场景批量生成草稿用例。

**步骤1** 在思维导图中，选中含有多个测试点的场景。

**步骤2** 鼠标右键单击该节点，在下拉列表中选择“生成用例”。

### ----结束

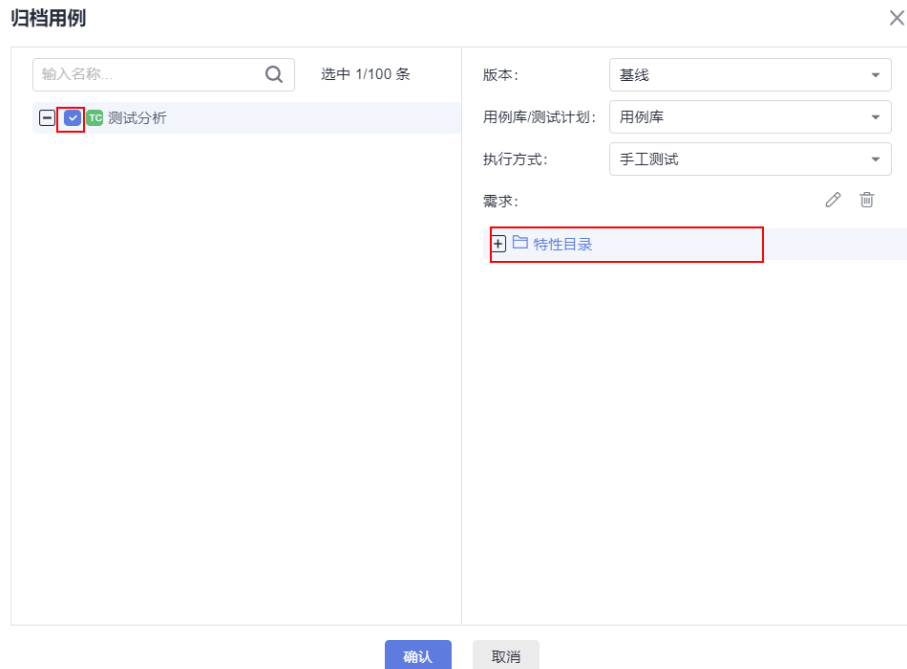
此场景节点包含的全部测试点节点均出现，变成草稿用例。





## 归档用例

将生成的用例归档后，在测试用例页面可找到对应用例记录。

- 步骤1** 在思维导图中，选中已生成用例的节点，鼠标右键单击该节点，在下拉列表中选择“归档用例”。
- 步骤2** 在弹框中，在左侧勾选需要归档的用例，单击页面右侧的下拉列表，选择版本、需要存放的用例库/测试计划、执行方式，关联需求并选择特性目录，单击“确认”。



- 步骤3** 当节点中出现  时，说明操作成功，在“测试用例”页面可搜到该用例。  
单击  ，页面将跳转至测试用例详情页。



----结束

### 6.2.3 创建 CodeArts TestPlan 思维导图并生成组合用例

数据因子组合技术是解决多个测试因子如何组合成测试用例的技术，能够解决依赖手工组合生成用例时因子组合覆盖率不全、测试设计的功能不够丰富等遗漏问题。

测试因子是在测试过程中，该测试受到哪几种因素影响，比如环境、测试的方式以及测试难易程度等，有几种影响因素就有几个测试因子。因子组合的目的是为了在创建用例时能够覆盖多个测试因素，有助于避免测试遗漏。

#### 前置条件


需要创建带有“测试点”的节点，只有测试点可以进行组合。例如：测试手机的基本功能可以作为一个测试点。




#### 通过添加动作因子、数据因子生成组合用例

在测试设计中，可以为节点添加“动作因子”、“数据因子”、“有效值”、“无效值”。

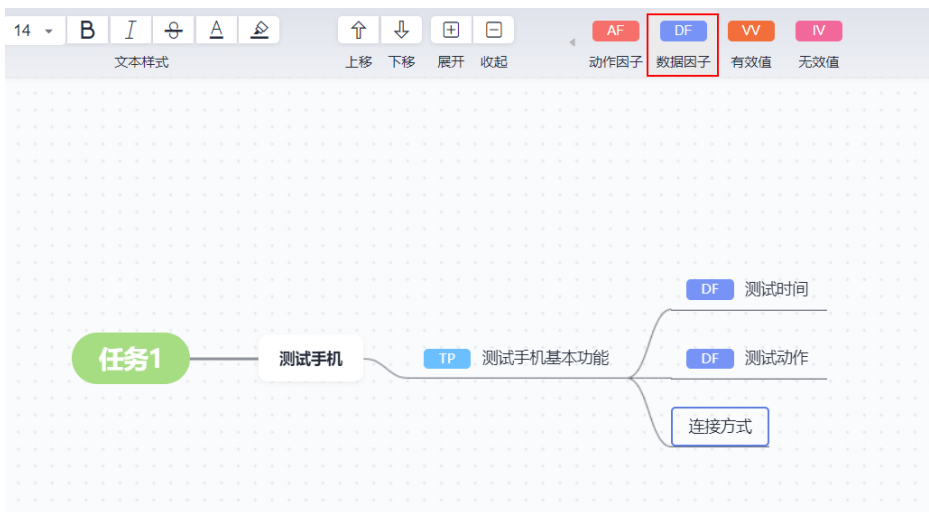
**步骤1** 进入已创建的思维导图。

**步骤2** 在顶部操作栏，单击“预期结果”右侧的 ，将显示“动作因子”、“数据因子”、“有效值”、“无效值”。

**步骤3** 选中需要添加因子的节点（可以添加动作因子或数据因子，思维导图里动作因子和数据因子的最大添加数各为100个）。

单击思维导图上方工具栏中 。例如：在测试手机的基本功能中，用户可以配置“使用手机，在\${测试时间}，使用\${连接方式}，进行\${测试动作}”作为一个测试流

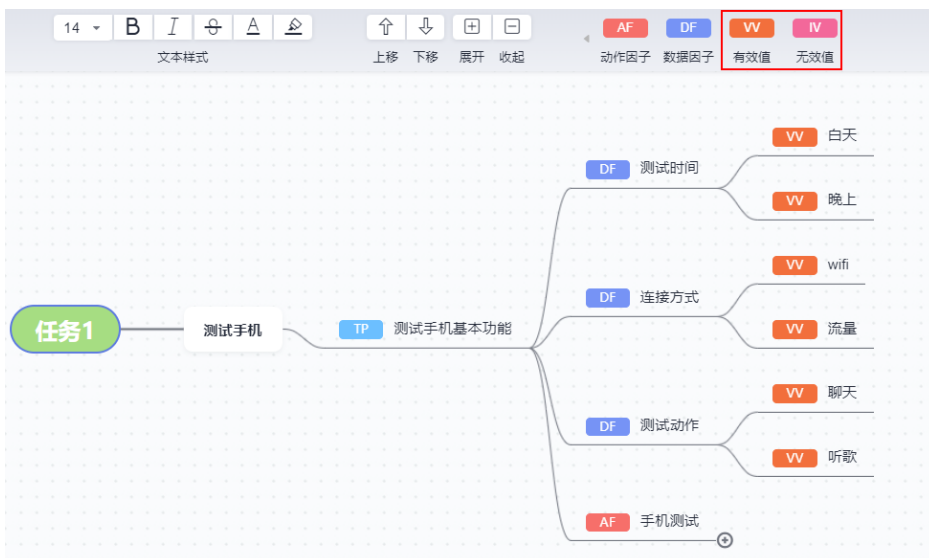
程，用户可以将\${测试时间}、\${连接方式}、\${测试动作}设置为数据因子，方便覆盖各个场景的组合，避免手工组合的遗漏。



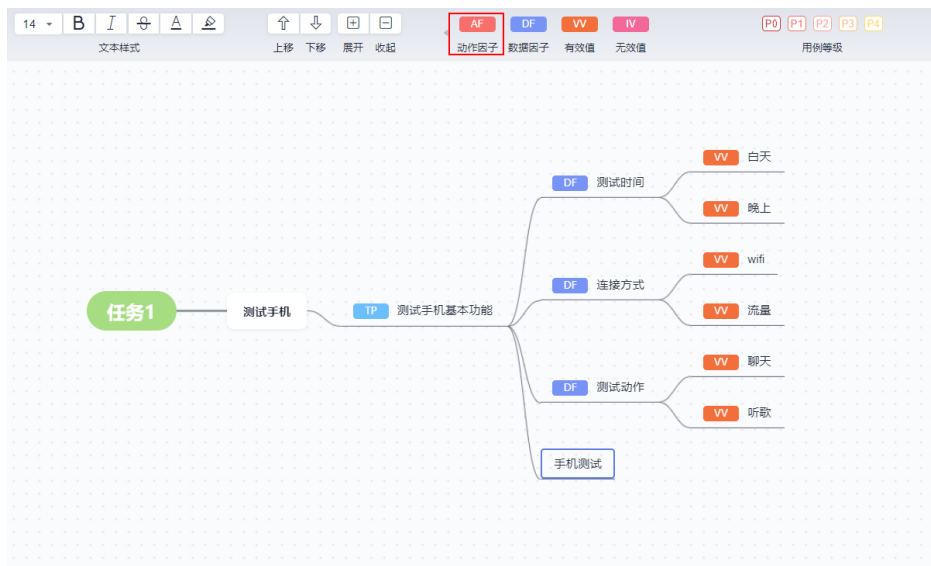
**步骤4** 添加有效值或无效值：为已添加数据因子的节点新增一个子节点，选中该子节点单击思维导图上方工具栏中 **WV** 或 **IV**。

- 有效值：用户待测的系统中数据因子的正常取值。
- 无效值：在数据组合中，无效值表示待测系统中的异常值，或者系统数据因子里允许范围之外的值。一个好的测试模型，除了需要测试系统中有效的组合之外，还需要验证系统中的无效值，以确保程序正确处理错误。任何一条测试用例中只允许出现一个无效值，因为任一无效值都会使系统异常，所以一条测试用例出现多个无效值是无意义的。

例如：用户可以为\${测试时间}、\${连接方式}、\${测试动作}并配置有效值或无效值。



**步骤5** 添加动作因子：选中目标子节点，单击思维导图上方工具栏中 **AF**。例如：可以将手机测试设置为一个动作因子。



**步骤6** 选中待生成用例的TP节点，鼠标右键单击该节点，在下拉列表中选择“组合创建用例”。

**说明**


读取打上“DF”节点的父节点，按照从上到下的顺序显示“DF”标签的有效值、无效值等相关信息。

**步骤7** 在“组合创建用例”的页面中，默认勾选思维导图中已设置的“动作因子”与“数据因子”信息。

**步骤8** 单击“下一步”。

**步骤9** 选择组合算法的卡片，单击“选用并生成预览”。组合算法介绍请参见下表：

数据组合覆盖类型	描述
全组合 ( AC,All Combinations )	每个测试因子所有取值进行全组合。AC是覆盖最全面的覆盖方式。
N-wise(主要介绍 pair-wise)	覆盖任意N个输入的组合方式，当N等于参数个数的时候就是全组合。 科学试验数据证明：当N=2时，生成的测试数据效率最高，因此当N=2时， pair-wise方法应用最多。
PairWise	与N-Wire同类型，适用于测试组合中有侧重点的场景，N取值为2。
TripleWise	与N-Wire同类型，适用于测试组合中有侧重点的场景，N取值为3。相比于PairWise，覆盖组合更多。

数据组合覆盖类型	描述
基本选择组合(BC, Basic Choice)	<p>适用于测试比对单因素变化场景，基于原始生成的一个组合，每次更换一个数据因子的一个值，生成不同组合。</p> <p>以基本组合为基础，通过修改对应数据因子的取值创建新的组合。</p> 
单一选择组合 (EC, Each Choice)	<p>每一个测试因子的每一个取值在所有测试因子组合中至少出现一次。</p>


若需要更换组合算法，可以单击“更换算法”重新选择。

### 须知

如果您回到思维导图中，对数据因子的节点进行修改或删除后，再次单击“组合创建用例”，需要单击“刷新预览”，同步变更后的信息。

**步骤10** (可选) 用户可以在生成组合用例前，在弹框中勾选“数据约束”，设置算法为数据因子添加约束条件，可参考以下步骤：

1. 勾选“数据约束”后，单击“新增”。最多添加20个约束。
2. 用户根据需要选择运算关系，变量名称选择要添加约束的数据因子，选择操作符，在下取值拉栏中选择对应数据因子的有效值或无效值。

单击图标可以清空已设置的取值。

3. 在条件表达式中可以查看配置的算法，单击“约束结果预览”查看生成结果。
4. 单击“确定”，完成因子组合约束配置。
5. 用户更新数据约束后，单击“刷新预览”。

### 举例：

Type: Primary、Logical、Single、Span、Stripe、Mirror、RAID-5

Size: 10、100、500、1000、5000、10000、40000

Format method: quick、slow

File system1: FAT、FAT32、NTFS

File system2: FAT、FAT32、NTFS

Cluster size: 512、1024、2048、4096、8192、16384、32768、65536

Compression: on、off



- **非条件约束**

非条件约束表示参数及参数取值之间的约束。

[Size] > 10000，表示用例集中Size只能取40000。

[File system1] <>[File system2]，表示用例集中File system1的值与File system2的值不能相同。

[Size] > 10000 and [Compression] ="on"，表示用例集中Size中只能取40000并且Compression只能为on。

- **IF条件约束**

使用 IF expr1 THEN expr2的形式进行配置，其中expr的形式为“[数据因子] 操作符 [取值]”。

如增加约束 IF [File system1] = "FAT" THEN [Cluster Size] <=4096; 预期结果中当File system1取值为“FAT”时，Cluster Size只能取512, 1024, 2048, 4096。

数据约束 ×

关系	(	变量名称	操作符	取值	)	操作
IF		File system1	=	FAT		+ 删除
THEN		Cluster size	<=	4096		+ 删除

条件表达式

IF [File system1] = "FAT" THEN [Cluster size] <= 4096;

约束结果预览

```

        },
        {
          "Format method": "slow",
          "Cluster size": "512",
          "Type": "Logical",
          "Compression": "ON",
          "Size": "10",
          "File system2": "NTFS",
          "File system1": "FAT"
        }
      ],
      {
        "Format method": "slow"
      }
    ]
  }
  
```

确定

- **运用AND/OR/NOT关系**

当算法更加复杂时，可以通过And、Or、Not来构造表达式，构造时需要加括号，否则约束解析将失败。

如增加约束IF ([File system1] = "NTPS" OR ([File system2] = "NTPS" AND [Cluster Size] <=4096)) THEN [Compression] ="off" ;当File system1为NTPS或者File system2为“NTPS”并且Cluster Size在512, 1024, 2048, 4096取一个值时，Compression为off。

数据约束

关系	(	变量名称	操作符	取值	)	操作
IF	(	File system1	=	ntfs		+ 删除
OR	(	File system2	=	NTFS		+ 删除
AND		Cluster size	<=	4096	)	+ 删除
THEN		Compression	=	OFF		+ 删除

条件表达式

```
IF ( [File system1] = "ntfs" OR ( [File system2] = "NTFS" AND [Cluster size] <= 4096 )) THEN [Compression] = "OFF";
```

约束结果预览

```
};
{
  "Format method": "slow",
  "Cluster size": "512",
  "Type": "Logical",
  "Compression": "OFF",
  "Size": "10",
  "File system2": "FAT32",
  "File system1": "ntfs"
},
{
  "Format method": "quick",
```

确定

● 运用IN/LIKE操作符

当需要使某个因子取值某一个值集合中时，可以通过IN操作符来构造约束。如IF [File system1] IN {"FAT","FAT32"} THEN [Cluster Size] <=4096;

当File system1取值为"FAT"时或者"FAT32"时，Cluster Size只能取512, 1024, 2048, 4096，该约束等价于IF ([File system1] = "FAT" OR [File system1] = "FAT32") THEN [Cluster Size] <= 4096

数据约束

关系	(	变量名称	操作符	取值	)	操作
IF	(	File system1	IN	FAT FAT32 输入变量取值		+ 删除
THEN		Cluster size	<=	4096		+ 删除

条件表达式

```
IF [File system1] IN {"FAT","FAT32"} THEN [Cluster size] <= 4096;
```

LIKE表达式用于通配符匹配，支持\*字符、? 一个字符，“FA\*”表示以FA开头的字符，“\*FA”表示以FA结尾的字符，“? A”表示含A的字符。

如IF ([File system1] LIKE "FA\*") THEN [Cluster Size] <= 4096;当File system1取值"FAT"或者"FAT32"时，Size只能取512, 1024, 2048, 4096。

数据约束

关系	(	变量名称	操作符	取值	)	操作
IF	(	File system1	LIKE	FA*		+ 删除
THEN		Cluster size	<=	4096		+ 删除

条件表达式

```
IF ([File system1] LIKE "FA*") THEN [Cluster size] <= 4096;
```

### 说明

1. 约束中区分整形变量及字符串变量，当变量的取值为数字时，则视该变量为数字变量；当变量取值有任一不是数字时，则视该变量为字符串变量。
2. 对两个数据因子使用操作符进行约束时，[数据因子1] 操作符 [数据因子2]，数据因子1必须与数据因子2同类型。

数据约束

关系	(	变量名称	操作符	取值	)	操作
<input type="text"/>	<input type="text"/>	File system1	<=	输入变量取值	<input type="text"/>	+ 删除
<input type="text"/>	<input type="text"/>	File system2	<input type="text"/>	输入变量取值	<input type="text"/>	+ 删除

条件表达式

[File system1] <= [File system2]

3. 整形变量间添加约束不支持LIKE操作符。字符串变量间添加约束不支持>、<、>=、<=操作符。

**步骤11** 在“组合预览结果”列表中，勾选需要创建的数据组合。

鼠标悬停在第一个勾选框，可以全选所有数据或全选当前页数据，单击“下一步”。

组合预览结果 (8/8)

刷新预览

<input checked="" type="checkbox"/>	组合名称	测试时间	连接方式	测试动作
<input type="checkbox"/>	聊天	白天	wifi	聊天
<input type="checkbox"/>	听歌	白天	wifi	听歌
<input checked="" type="checkbox"/>	白天_流量_聊天	白天	流量	聊天
<input checked="" type="checkbox"/>	白天_流量_听歌	白天	流量	听歌
<input checked="" type="checkbox"/>	晚上_wifi_聊天	晚上	wifi	聊天
<input checked="" type="checkbox"/>	晚上_wifi_听歌	晚上	wifi	听歌
<input checked="" type="checkbox"/>	晚上_流量_聊天	晚上	流量	聊天
<input checked="" type="checkbox"/>	晚上_流量_听歌	晚上	流量	听歌


**步骤12** 在“配置用例”页面中，输入以下信息：

- 用例名称：名称默认为TP节点名称\_\${数据因子}的格式。
- 用例编号：用例编号根据填写规则，递增生成。
- 用例等级：选择P0~P4。
- 用例描述：输入1-500字符长度的描述信息。
- 前置条件：输入1-2000字符长度的前置条件信息。


**步骤13** 在编写测试步骤中，可以通过“\$”的方式进行引用数据因子，单击“确定”。




测试步骤

#	步骤描述	预期结果	操作
1	打开手机。选择在\${测试时间}，使用\${连接方式}，进行\${测试动作}。		+ 删除

**步骤14** 单击, 可以查看用例列表。

**步骤15** 用例列表将默认显示生成的测试用例列表, 选择“组合策略列表”, 可以查看使用组合算法的历史。

**步骤16** 单击需要归档用例的操作列图标, 单击“归档”。其他用例操作如下:

- 归档多个用例: 在用例列表中, 勾选多个需要归档的用例, 单击下方“归档”。
- 编辑用例: 单击操作列中的图标, 可以查看、编辑用例详情。
- 删除用例: 单击需要删除用例的操作列图标, 单击“删除”。
- 批量删除: 在用例列表中, 勾选多个需要归档的用例, 单击下方“删除”。
- 搜索用例: 在用例列表的搜索栏中, 输入用例名称的关键词, 单击。
- 过滤用例: 单击用例列表下的下拉栏, 可以根据“所有”、“未归档”、“已归档”进行过滤用例。

**步骤17** 在“归档用例”页面中, 在左侧勾选目标用例, 在右侧选择归档到的版本、测试计划, 选择执行方式和特性目录, 单击“确定”。

**步骤18** 在导航栏单击“测试 > 测试用例”, 选择版本、测试计划、目标用例类型页签, 查看已归档的用例。


----结束

## 通过因子库生成组合用例

### 创建因子

**步骤1** 单击导航栏“测试 > 测试设计”。

**步骤2** 单击页面右上方“测试因子中心”。

**步骤3** 根目录下无法新建因子, 需要创建文件夹: 单击因子库根目录的图标, 单击“新建文件夹”。

**步骤4** 输入文件夹名称: 输入应符合非空字符且长度在[1,500]之间的文件夹名称。






**步骤5** 单击创建完成的文件名称, 单击“新建因子”。

**步骤6** 配置以下信息后, 单击“确定”。

配置项	是否必填	描述
因子名称	是	输入因子名称, 不超过500个字符。
因子类型	是	选择“数据因子”或“动作因子”。
因子描述	否	对因子进行简要描述, 请输入不超过500个字符。
前置条件	是	因子类型选择动作因子后, 输入前置条件。
测试步骤	是	因子类型选择动作因子后, 可以输入步骤描述、预置结果, 不超过2000个字符。单击操作列中的+, 可以添加多个测试步骤。

配置项	是否必填	描述
数据类型	否	默认为String类型。
有效值/无效值	否	因子类型选择数据因子后，可以输入有效值或无效值。单击“增加选项”，可以添加多个有效值或无效值。
备注	否	输入备注信息，请输入不超过500个字符。

**步骤7** 新建成功的因子显示在文件夹列表中，可以对因子进行如下操作：

- 复制因子：单击需要复制因子所在操作列的  图标，在页面中可以修改因子的相关信息，单击“复制”。
- 编辑因子：单击需要复制因子所在操作列的  ，在页面中可以修改因子的相关信息。
- 删除因子：单击需要复制因子所在操作列的  ，单击“删除”。
- 批量删除因子：勾选需要删除的因子，在下方单击“删除”。
- 导出因子：勾选需要导出的因子，在下方单击“导出”，可将所选因子信息表格保存到本地。
- 过滤因子：单击新建因子右侧的下拉栏，可以选择全部、我创建的、动作因子、数据因子进行过滤。
- 搜索因子：在上边搜索栏内，输入因子名称的关键字。
- 编辑因子列表的表头：单击因子列表操作列的  图标，在弹框中勾选需要显示的列名称，在右侧已选列可以调整显示顺序。
- 导入：单击右上方“导入”，在弹框中单击“下载模板”，将模板表格下载到本地。编辑完成后，在弹框中单击  选择本地文件，单击“确定”。

----结束

### 思维导图关联因子库

**步骤1** 单击导航栏“测试 > 测试设计”。

**步骤2** 新建或选择目标思维导图。

**步骤3** 选中需要关联因子的节点，右键单击该节点，在下拉栏中单击“关联因子库”。

#### 须知

需要选择带有测试点的节点进行关联，否则无法关联因子库。

**步骤4** 在关联因子库页面中，勾选需要添加的因子，单击“确定”，添加成功的因子将显示在思维导图中。










**步骤5** 生成用例的后续操作可参考[步骤6](#)。














----结束

## 6.2.4 管理 CodeArts TestPlan 思维导图

### 思维导图基础操作

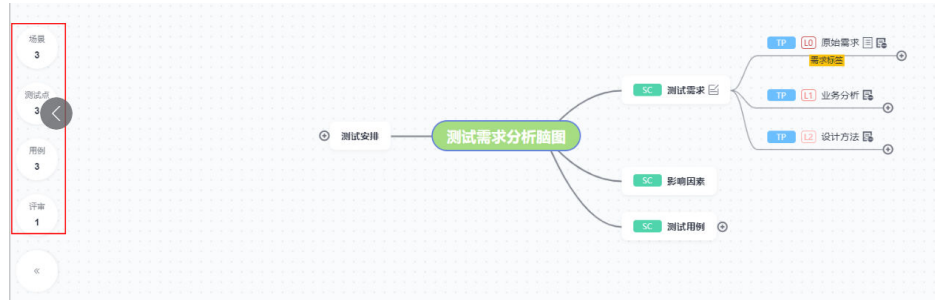
进入思维导图后，除了编辑节点，还可以进行以下操作：

操作项	说明
重命名	单击思维导图左上方  ，在弹框中输入新名称，可以对当前思维导图重命名。
删除思维导图	在思维导图页面右上方单击  ，在下拉列表中选择“删除思维导图”，单击“确定”，删除当前思维导图。 被删除的思维导图将移入回收站中。
管理回收站 (在思维导图页面)	在思维导图右上方单击  ，在下拉列表中选择“回收站”，在弹框中可查看已被删除的思维导图列表。 对于在回收站中的思维导图，可以完成如下操作： <ul style="list-style-type: none"> <li>单击 ，单击“确定”，恢复对应行中的思维导图。 <b>说明</b> <ul style="list-style-type: none"> <li>恢复思维导图，会替换当前页面中的内容，请谨慎操作。</li> <li>回收站记录一旦被恢复，该条记录将从列表中移除。</li> <li>由于回收站恢复会覆盖当前思维导图内容，建议新建空白思维导图进行回收站恢复。</li> </ul> </li> <li>单击 ，查看对应行中思维导图的详情。</li> <li>单击 ，单击“确定”，彻底删除对应行中的思维导图，删除后不可恢复，请谨慎操作。</li> </ul>
回收站 (在测试设计页面)	在测试设计页面左下角单击“回收站”，在弹框中可查看已被删除的思维导图列表。 对于在回收站中的思维导图，可以完成如下操作： <ul style="list-style-type: none"> <li>单击 ，单击“确定”，恢复对应行中的思维导图。 <b>说明</b> <ul style="list-style-type: none"> <li>从测试设计页面的回收站恢复思维导图，会在目录下新建一个思维导图。</li> <li>回收站记录一旦被恢复，该条记录将从列表中移除。</li> </ul> </li> <li>单击 ，查看对应行中思维导图的详情。</li> <li>单击 ，单击“确定”，彻底删除对应行中的思维导图，删除后不可恢复，请谨慎操作。</li> </ul>

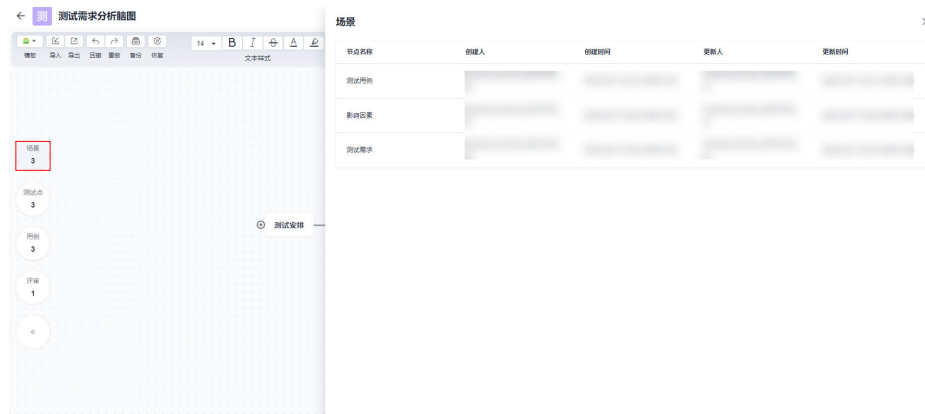
操作项	说明
导入	<p>在思维导图上方工具栏中单击 ，单击“确定”。在弹窗中选中本地已存在的“.xmind”类型文件，可将选中的文件导入至当前页面中。</p> <p><b>说明</b></p> <ul style="list-style-type: none"> <li>支持导入不超过20MB的文件。</li> <li>支持导入脑图类压缩文件个数不超过100个。</li> <li>导入的内容会替换当前页面中的思维导图，请谨慎操作。</li> <li>导入脑图成功后不会改变根节点。</li> </ul>
导出	<p>在思维导图上方工具栏中单击 ，选择导出的格式，支持PDF、PNG格式，单击“确定”，可将页面中的思维导图导出到本地。</p>
备份	<p>在思维导图上方工具栏中单击 ，在弹框中输入名称和描述，单击“确定”，可为页面中的思维导图创建一个备份。</p>
恢复	<p>在思维导图上方工具栏中单击 ，在弹框中选择一个备份，单击 ，在弹框中单击“确定”，可将已备份的思维导图恢复到当前页面中。</p> <p><b>说明</b></p> <ul style="list-style-type: none"> <li>恢复思维导图，会替换当前页面中的内容，请谨慎操作。</li> <li>执行恢复思维导图之前，会自动备份当前思维导图的内容。</li> </ul>
撤销	<p>在思维导图上方工具栏中单击 ，可撤销页面中的最后一步操作。</p>
重做	<p>在思维导图上方工具栏中单击 ，可恢复最近被撤销的一步操作。</p>
全部展开	<p>选中一个带  的节点，思维导图上方工具栏中单击 ，可将该节点下所有子节点全部展开，展开后节点旁的图标变为 。</p>
全部收缩	<p>选中一个带  的节点，思维导图上方工具栏中单击 ，可将该节点下所有子节点全部收缩，收缩后节点旁的图标变为 。</p>

## 节点统计

进入思维导图中，页面左侧将显示对于该图中各类型节点的统计值，包括场景、测试点、用例以及评审的总数。



单击任意统计数值，页面右侧将滑出该数值对应的节点列表信息。



在列表中单击任意节点名称，即可在思维导图中定位到对应的节点，查看节点详情。

## 管理思维导图模板

### 保存模板

已创建的思维导图，可以保存为模板。

**步骤1** 进入已创建的思维导图，根据需要完成思维导图的编辑。

**步骤2** 单击思维导图上方工具栏中“模板”，在下拉栏中选择“保存模板”。

**步骤3** 在弹框中输入名称，单击“确定”。

**步骤4** 单击左上角 ←，返回测试设计列表，单击页面左上角“模板新建”，选择“自定义模板”页签，即可在弹框中看到刚保存的模板。




----结束

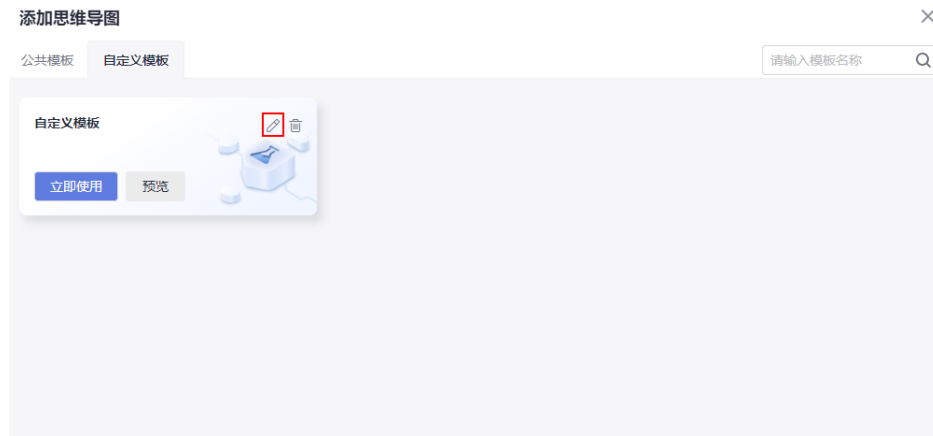
### 编辑模板


成员可以对自己保存的模板进行编辑。



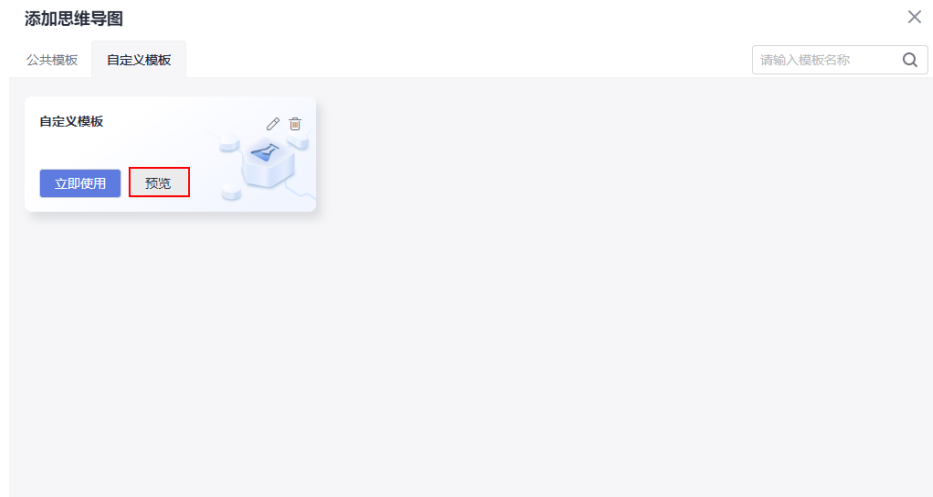
**步骤1** 进入“测试设计”页面，单击页面左上角“模板新建”。

**步骤2** 单击“自定义模板”页签，在弹框中选择一个模板，单击模板对应的。




**步骤3** 根据需要编辑模板，编辑完成单击页面左上角。

**步骤4** 单击页面左上角“模板新建”，选择“自定义模板”页签，在弹框中找到编辑完成的模板，单击“预览”，即可查看修改后的模板详情。



----结束

## 搜索节点

在思维导图右上角的搜索框中输入关键字，并单击，即可在思维导图中找到含有关键字的节点，搜索到的节点会被蓝框标出。

如果匹配多个搜索结果，可在搜索框的右侧看到匹配数目，以及当前节点在第几匹配项。单击上下箭头可以调整前后匹配项。




## 调整外观


思维导图的默认为居中视图，即根节点在中央，子节点分布在根节点两侧。



单击思维导图右下角工具栏，可以调整思维导图的外观。

- 单击 ，调整为左视图，即子节点在根节点的左侧。



- 单击 ，调整为右视图，即子节点在根节点的右侧。



- 单击 **☰**，调整为默认居中视图。
- 单击 **—** 或 **+**，可以将思维导图缩小或放大。

## 快捷键

测试设计支持通过以下快捷键编辑思维导图。

操作	快捷键
视图居中	End
移动视图	W/A/S/D
展开/收缩	F
放大/缩小	+/-
编辑节点	F2或Space
编辑换行	Alt+L/Shift+Enter/Ctrl+Enter
复制	Ctrl+C
粘贴	Ctrl+V
剪切	Ctrl+X
回撤	Ctrl+Z
重做	Ctrl+Y
添加同级节点	Enter
添加子节点	Ins或Tab
删除节点	Del
选中根节点	Ctrl+Home或Home

操作	快捷键
选中父节点	Backspace
移动选中节点	↑/←/↓/→
移动节点	Ctrl+↑/←/↓/→
添加场景	Alt+C
添加测试点	Alt+P
添加预置条件	Alt+O
添加步骤	Alt+T
添加预期结果	Alt+X
添加动作因子	Alt+Shift+A
添加数据因子	Alt+Shift+D
添加用例等级	Ctrl+0 / 1 / 2 / 3 / 4
添加图片	Ctrl+I
添加文件	Ctrl+D
设置标签	F3
查看快捷键	Ctrl+Shift+L

单击思维导图左下角“快捷键”，可以查看快捷键列表。



## 6.3 创建手工测试用例

手工测试用例用于管理测试场景、测试步骤，在测试计划服务的“测试用例 > 手工测试”页面中可以新建手工测试用例。创建测试用例需要具有创建用例的权限（即在项目内的角色为：除了运维经理、浏览者与参与者的其他系统角色）。

### 新建手工测试用例

- 步骤1** 登录软件开发生产线首页，搜索目标项目并单击项目名称，进入项目。
- 步骤2** 单击导航栏“测试 > 测试用例”。
- 步骤3** 默认显示“基线”和“用例库”，在“手工测试”页签，单击页面左上方“新建用例”，进入新建页面。
- 步骤4** 根据实际情况输入名称、描述、前置条件、测试步骤等信息，单击“保存”完成手工测试用例的创建。

手工测试用例主要配置项如下：

配置项	描述
名称	用例的名称（必填），名称长度为1-128个字符。可以描述用例对应的测试场景或功能。
描述	对用例的简要描述。可以输入当前用例的测试范围、测试目标、测试策略、测试方法、测试工具、测试数据、测试指标、测试环境等信息。
前置条件	描述执行当前用例所需要的必备前提条件。
测试步骤	录入具体的步骤描述和对应的预期结果。
标签	根据需要对当前用例设置标签，以空格分隔，每个用例最多可关联10个标签。
编号	用户可以自定义用例编号，若不输入，则服务自动生成编号 用户在测试用例列表界面，也可以修改编号，编号的字符长度为3~128字符。
用例等级	请根据用例对应场景或功能的重要程度选择用例等级，默认等级为2级。 <ul style="list-style-type: none"><li>● 0级：最基本的功能验证，用例不宜过多，各模块尽量保证在10-20个，占比5%左右。</li><li>● 1级：基本功能验证，可用于继承特性的基本特性验证、迭代验收前的基本功能验证等，占比20%左右。</li><li>● 2级：重要特性验证，可用于测试版本（非回归版本）中功能测试，占比60%左右。</li><li>● 3级：一般功能/非重要功能验证，包括对基本/重要功能的异常测试，占比15%~20%左右。</li><li>● 4级：特殊输入、场景、阈值条件的用例，该级别用例不宜过多，占比0%~5%左右。</li></ul>

配置项	描述
模块	选择当前用例对应的模块，模块列表来源于项目设置，请参考 <a href="#">模块设置</a> 。
版本号	选择当前用例对应的版本号。
迭代	根据当前用例需要在哪个迭代进行测试而选择对应迭代。
处理者	指定完成对应测试任务的人员。
关联需求	建议把测试用例和测试需求双向关联。如果当前用例用于测试某一特定需求且该需求已录入“需求管理”服务，单击“选择关联需求”可以将用例和需求工作项进行关联。
归属目录	与特性目录相关联，默认归属到其他目录。
添加附件	与当前用例相关的附件。可以关联“文档”中已存在的文档，也可以从本地上传附件。
自定义字段	用户在“测试设置”中设置的自定义字段，请参见 <a href="#">设置CodeArts TestPlan测试用例字段</a> 。

---结束

## 编辑用例详情

以Scrum项目为例，当前迭代已在需求管理录入相关需求。

测试工作开始前，测试人员先在用例库中创建相应手工测试用例。根据前端需求，创建相应的手工测试用例，并将该手工测试用例与前端需求进行关联。

**步骤1** 完成[新建手工测试用例](#)操作后，在“测试用例 > 手工测试”页面单击需要编辑的用例名称，默认进入测试用例“详情”页面。

**步骤2** 根据需要编辑测试详情。

- 在“测试步骤”下的表格中，分别单击“步骤描述”、“预期结果”列的空白处，根据需要输入对应内容。单击表格中“操作”一栏的+，添加一个步骤，并按照需要填写步骤描述与预期结果。
- 单击页面右侧“关联需求”，在弹框中勾选所要关联的需求，单击“确定”完成关联。

----结束

## 6.4 创建接口自动化测试用例

### 6.4.1 CodeArts TestPlan 接口自动化用例使用流程

接口自动化用例提供了丰富的用户界面，支持可视化地编排自动化测试步骤以及丰富的检查点和相应提取配置，可以在简单易用的可视化接口管理页面上，快速生成与接口核心逻辑相匹配的测试用例，实现“零代码”快速生成测试用例。接口自动化用例具备多测试步骤的逻辑编排能力，支持通过拖拽来安排接口测试的顺序，生成自动化

测试策略，完成场景级接口自动化脚本生成，测试人员可以从重复工作中释放出来，进行探索性测试等更具有创造性和高价值的测试活动。

接口自动化用例包含用例基本信息和脚本两部分：

- 基本信息用于管理和描述测试用例，包含名称（必填）、类型、模块、版本号、迭代、关联需求、编号、标签、用例等级、处理者、归属目录、描述、前置条件、测试步骤、预期结果。
- 脚本用于定义自动化测试步骤，包含测试步骤、逻辑控制、测试参数。

接口自动化用例的使用流程包括三个阶段：准备阶段、测试阶段、销毁阶段。

准备阶段对应页面中的“前置步骤”，实现测试前置条件的准备；测试阶段对应“测试步骤”，实现接口的功能测试；销毁阶段对应“后置步骤”，实现准备阶段和测试阶段测试数据的释放或恢复。

- 准备阶段（可选）：前置步骤。
  - 在此阶段中准备测试阶段需要的前置条件的数据，如果没有前置条件，可以忽略此阶段。
  - 在准备阶段通过调用接口的方式初始化前置条件，如果前置条件的数据需要在测试阶段中引用，可以使用参数传递将数据参数化后供测试阶段引用，详见[设置CodeArts TestPlan接口脚本的响应提取](#)。
- 测试阶段：测试步骤。

定义接口核心测试步骤，测试阶段中的测试步骤可以引用准备阶段提取的参数。
- 销毁阶段（可选）：后置步骤。
  - 为了不影响其它测试或者下一次测试，建议在每次测试结束后清理测试环境数据，恢复测试环境的初始状态，销毁准备阶段创建的数据。
  - 如果没有数据需要销毁，可以忽略此阶段。通过调用接口的方式销毁数据，销毁阶段的测试步骤可以引用准备阶段提取的参数。

## 6.4.2 创建 CodeArts TestPlan 接口自动化测试用例模板

### 新建接口自动化测试用例

**步骤1** 登录软件开发生产线首页，搜索目标项目并单击项目名称，进入项目。

**步骤2** 单击导航栏“测试 > 测试用例”。

**步骤3** 选择“接口自动化”页签，单击页面左上方“新建用例”，进入新建页面。

**步骤4** 输入用例名称，根据需要配置其它信息，单击“保存”完成新建用例。或者单击“保存并编写脚本”，进入“脚本”页面，完成脚本编辑。

接口自动化用例主要配置项与手工测试用例相似，详情请参考[新建手工测试用例](#)。

#### 说明

用例名称应符合1到128位长度，可以包含中文、大小写字母、数字、特殊字符（-/\_/\*&`^~;:;::(){}+=,x¥……——’!@#%&.'[]<>?·! ”、‘ “ ” ? 一。 ， 《 》 【 】 ）的组合格式。

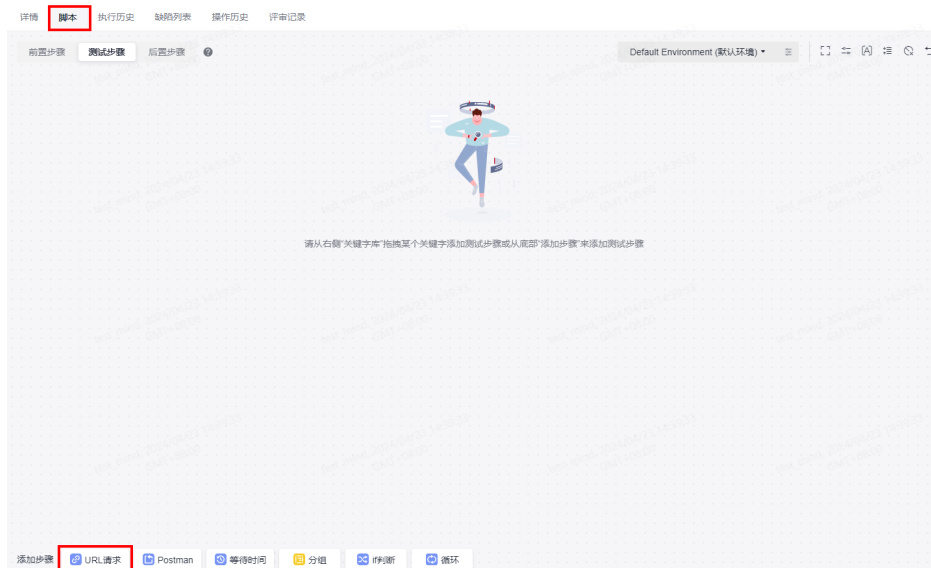
----结束

## 6.4.3 使用自定义 URL 请求添加 CodeArts TestPlan 接口脚本

### 6.4.3.1 添加并设置 CodeArts TestPlan 接口脚本的 URL 请求

**步骤1** 完成[创建CodeArts TestPlan接口自动化测试用例模板](#)操作后，在“测试用例 > 接口自动化”页面单击需要编辑的用例名称。

**步骤2** 选择“脚本”页签，单击“URL请求”，生成一个测试步骤。



用户可以导入被测接口的Swagger描述文件生成脚本模板，基于脚本模板编排测试用例，详细操作请参考[将测试步骤保存为CodeArts TestPlan接口脚本关键字](#)。

选择脚本模板，拖拽脚本模板卡片、或单击脚本模板卡片上的+，将脚本添加至测试步骤。



**步骤3** 根据需要编辑URL请求，包括[添加并设置CodeArts TestPlan接口脚本的URL请求](#)、[设置CodeArts TestPlan接口脚本的检查点](#)、[设置CodeArts TestPlan接口脚本的响应提取](#)，可以在“请求”输入需要测试的环境地址，可以是域名或ip，当用户从本地导入swagger或postman，“请求”这里的地址会自动生成。

**步骤4** （可选）重复[步骤2~步骤3](#)，添加前置步骤与后置步骤。

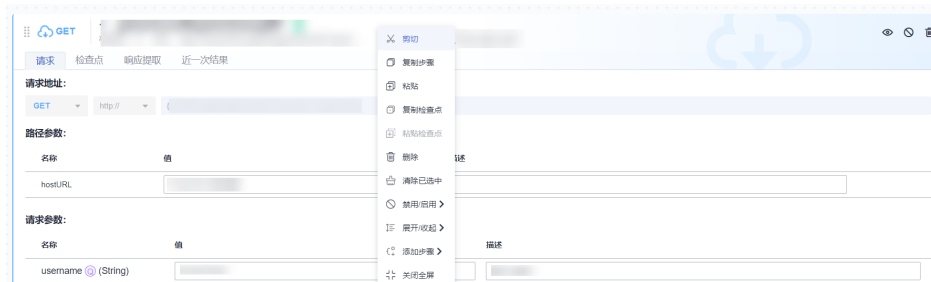
**步骤5** 编辑完毕，单击页面右上角“保存”，完成测试脚本的编写。

----结束



## 说明

- 接口自动化用例支持在请求URL路径、请求头、请求体、检查点参数、URL响应中使用内置函数。内置函数详情请参考[接口自动化用例内置函数说明](#)。
- 接口自动化用例支持在编辑时，可以在测试步骤标题区域使用右键，对测试步骤进行剪切、复制、粘贴、删除等操作；存在多个测试步骤时，支持通过“Ctrl+左键”多选后，批量执行右键操作；其中，“复制”或“剪切”测试步骤后，支持当前页签/跨页签/跨用例“粘贴”测试步骤。



- 需要注意的是，在标题区域内的右键响应，均为对测试步骤的操作。因此在编辑测试步骤标题区域文本框时，浏览器右键被系统右键覆盖，浏览器右键不生效。如需在测试步骤标题区域的文本框中对文本进行复制粘贴操作，请使用快捷键的方式进行复制（Ctrl+C）和粘贴（Ctrl+V）操作。

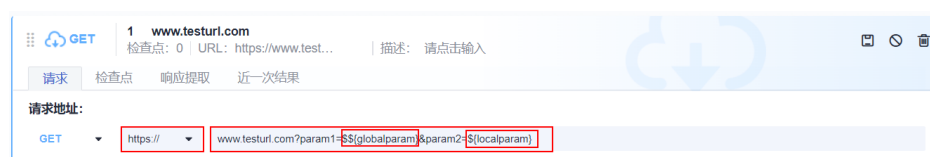
## 请求 URL 与 URL 参数

在接口自动化用例的“脚本”页签中填写需要请求的URL、支持HTTP/HTTPS请求。

接口自动化可以支持以下类型URL请求方式。其中，新建的URL的请求方式默认为GET方式。



请求方式	说明
GET	从API检索数据。
POST	上传文件和添加新数据。
PUT	替换现有数据。
DELETE	删除现有数据。
HEAD	获取响应的HTTP头。
OPTIONS	预检请求。
PATCH	更新一些现有数据的字段。

在请求URL中支持使用环境参数、局部参数以及响应提取参数，使用方法请参考[设置 CodeArts TestPlan接口脚本的测试用例参数](#)。



## 请求头

接口自动化预置HTTP协议的常用请求头，在请求头模块，输入要填写的请求头信息。

请求头支持表单、文本两种格式。页面中默认使用表单形式，通过   图标进行表单和文本的切换。

请求头: 

名称	值	描述
Content-Type	application/json	
Accept	application/json,charset=UTF-8	

- 表单：在请求头“名称”列选择或输入请求头名称，在“值”列选择/输入内容。

请求头: 

名称	值	描述
Content-Type	application/json	
Accept	application/json,charset=UTF-8	


 Accept-Charset  
 Accept-Encoding  
 Accept-Language  
 Accept-Datetime

 Postman
  等待时间
  分组
  if判断
  循环

- 文本：需要按照“key:value”的格式填写请求头。  
与表单模式不同的是，文本模式仅支持对“请求头名称”和“值”进行配置操作，但不支持对“描述”进行配置操作。

填写请求头时需要关注以下两点约束条件：

- 请求头的总数不能超过30。
- 请求头的长度不能超过10000。

请求头: 

```
Content-Type:application/json
Accept:application/json,charset=UTF-8
```

测试计划服务中预置了多个常用的HTTP请求头，详情如下表：

请求头名称	说明
Accept	能接受的回应内容类型 ( Content-Types ) ， 如： text/plain等。
Accept-Charset	能接受的字符集， 如： utf-8等。
Accept-Encoding	能接受的编码方式列表： compress   gzip   identity。
Accept-Language	能接受的回应内容的自然语言列表， 如： en-US等。
Accept-Datetime	能接受的按照时间来表示的版本。

请求头名称	说明
Access-Control-Request-Method	用于预检请求让服务器知道哪些HTTP方法的实际请求时将被使用。
Access-Control-Request-Headers	用于预检请求让服务器知道哪些HTTP头的实际请求时将被使用。
Authorization	用于超文本传输协议认证的认证信息。
Cache-Control	用来指定在请求/响应链中的所有缓存机制都必须遵守的指令。
Connection	浏览器优先使用的连接类型。
Cookie	由服务器通过Set- Cookie发送的一个超文本传输协议Cookie。
Content-Length	以八位字节数组（8位的字节）表示的请求体的长度。
Content-MD5	请求体内容的二进制MD5散列值，以Base64编码的结果。
Content-Type	请求体的多媒体类型（用于POST和PUT请求中），如：application/json。
Date	发送消息的日期和时间。
Expect	表明客户端要求服务器做出特定的行为。
Forwarded	包含来自被改变或丢失时代理参与请求的路径代理服务器的面向客户端的信息。  用于调试，统计和生成依赖于位置的内容，并且通过设计它可以显示隐私敏感信息，例如客户端的IP地址。因此，在部署此标头时必须牢记用户的隐私。
From	发起请求的用户的邮件地址。
Host	服务器的域名（用于虚拟主机），以及服务器所监听的传输控制协议端口号。如果所请求的端口是对应的服务的标准端口，则端口号可省略。  自超文件传输协议版本1.1（HTTP/1.1）开始是必需字段，在服务中如果URL中填写的域名为IP地址会自动添加该字段，否则请将被测应用的IP与端口信息填写在该字段。
If-Match	仅当客户端提供的实体与服务器上对应的实体相匹配时，才进行对应的操作。主要用作像PUT这样的方法中，仅在用户从上次更新某个资源以来，该资源未被修改的情况下，才更新该资源。
If-Modified-Since	允许在对应的内容未被修改的情况下返回304未修改（304 Not Modified）。

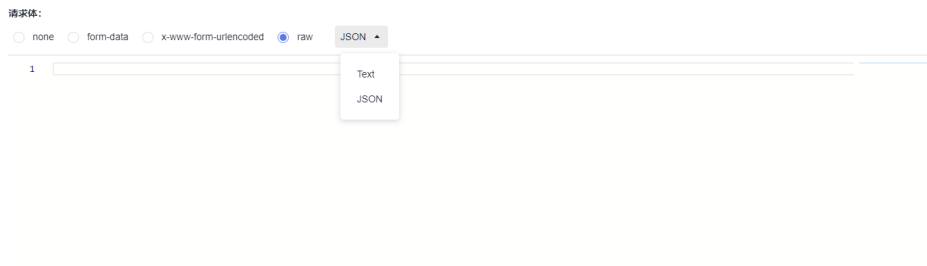
请求头名称	说明
If-None-Match	允许在对应的内容未被修改的情况下返回304未修改（ 304 Not Modified ），参考超文本传输协议的实体标记。 典型用法中，URL被请求时，Web服务器会返回资源和其相应的ETag值，且会被放置在HTTP的ETag字段中，然后，客户端可以决定是否缓存该资源和相应的ETag。以后，如果客户端想再次请求相同的URL，将会发送一个包含已保存的ETag和If-None-Match字段的请求。
If-Range	如果实体未被修改过，则向发送方发送其所缺少的那一部分或多个部分，否则，发送整个新的实体。
If-Unmodified-Since	仅当实体自某个特定时间以来未被修改的情况下，才发送回应。
Max-Forwards	限制消息可被代理及网关转发的次数。
Origin	发起一个针对跨来源资源共享的请求。要求服务器在回应中加入一个“访问控制-允许来源”（ Access-Control-Allow-Origin ）字段。
Pragma	与具体的实现相关，这些字段可能在请求/回应链中的任何时候产生多种效果。
Proxy-Authorization	用来向代理进行认证的认证信息。
Range	仅请求某个实体的一部分，字节偏移从0开始。
Referer	表示浏览器访问的前一个页面上的某个链接将浏览器带到了当前所请求的这个页面。
TE	浏览器预期接受的传输编码方式：可使用回应协议头Transfer-Encoding字段中的值；还可用“trailers”（与“分块”传输方式相关）值来表明浏览器希望在最后一个尺寸为0的块之后还接收到一些额外的字段。
User-Agent	浏览器的浏览器身份标识字符串。
Upgrade	要求服务器升级到另一个协议。
Via	向服务器告知请求是由哪些代理发出的。
Warning	一个一般性的警告，告知在实体内容体中可能存在错误。

## 请求体

请求体是接口请求需要传递的消息（报文），支持文本、JSON请求体、表单参数三种格式。

当请求消息方式选择POST、PUT、DELETE、OPTIONS、PATCH、HEAD时，页面中会出现请求体；请求消息方式为GET时不显示请求体。

- 文本：可以输入标准json串，与选择JSON类型请求体使用方式一致。



- 表单参数：支持文本和文件两种类型。

- 文本类型：设置参数名和值。



- 文件类型：设置参数名，并通过上传文件为参数赋值，文件类型不限。



### 6.4.3.2 设置 CodeArts TestPlan 接口脚本的检查点

#### 设置建议

推荐用户设置检查点。针对接口类请求，请提供判断响应码的检查点。  
当检查点设置为空，不论接口的响应码是什么，都会判断结果为成功。

#### 检查点说明

测试检查点又名断言，通过检查接口响应是否符合预期，判定系统是否满足预期。  
在接口自动化用例详情中，测试步骤的“检查点”页签中可以定义测试检查点。



检查点内容包括属性、比较符和目标值定义。

字段	说明
重试次数	如果检查点失败，重新执行该测试步骤的次数，重试次数范围：0~5的正整数。
重试间隔	如果检查点失败，每次重试的间隔时长，单位为ms，重试间隔范围：0~10,000的正整数。
来源	被检测字段的来源，如响应体（JSON）、响应头、响应码、变量。
属性	<p>在属性中，支持输入\$调用全局变量、局部变量、内置函数。</p> <ul style="list-style-type: none"> <li>若来源是响应码，属性可为空。详细介绍请参见<a href="#">响应码检查</a>。</li> <li>若来源是响应头，属性为响应头中字段的名称。详细介绍请参见<a href="#">响应头检查</a>。</li> <li>若来源是响应体（JSON），属性有两种填写方法：               <ol style="list-style-type: none"> <li>普通提取表达式（非“\$”开头），例如“item.name”。取字段中的值，支持嵌套取值。详细介绍请参见<a href="#">响应体（JSON）检查</a>。 从响应体中提取数组时，下标可以是数字，也可以是“key:value”表达式，详细介绍请参见<a href="#">示例：根据给定的key:value从响应体中获取字符串</a>。</li> <li>JsonPath表达式（“\$.”或“\$[”开头），例如“\$.store.book[0].title”。详细介绍请参见<a href="#">示例：根据JsonPath从响应体中获取数据</a>。</li> </ol> </li> <li>若来源是变量，属性为全局变量，局部变量，响应提取后的变量。详细介绍请参见<a href="#">变量检查</a>。</li> </ul>
高级提取类型	<p>可选项，使用高级提取类型，辅助提取检查点信息，若选择不涉及，则视为不使用额外的方式匹配。</p> <p>目前有两种方式：</p> <ul style="list-style-type: none"> <li>字符串提取，也就是字符串的截取，详见<a href="#">字符串提取说明</a>。</li> <li>正则表达式，即使用正则方式对来源字符串进行过滤，详见<a href="#">正则表达式说明</a>。</li> </ul> <p>高级提取类型优先使用字符串提取功能，若不能满足需求可考虑使用正则表达式。</p>
类型值	高级提取类型中所需的参数。
比较符	支持数字、字符串、JSON对象比较、类型比较等多种比较符，详细介绍请参见 <a href="#">比较符说明</a> 。
目标值	检查点期望值。目标值支持使用内置参数，请参考 <a href="#">内置参数</a> 了解如何使用内置参数。

例如，检查响应体（JSON格式）中“item.name”字段的第零位（首位）到第四位之间是不是petty，参数配置如下：

字段	值
来源	响应体 (JSON)
属性	item.name
高级提取类型	字符串提取
类型值	0、5
比较符	等于(字符串)
目标值	petty

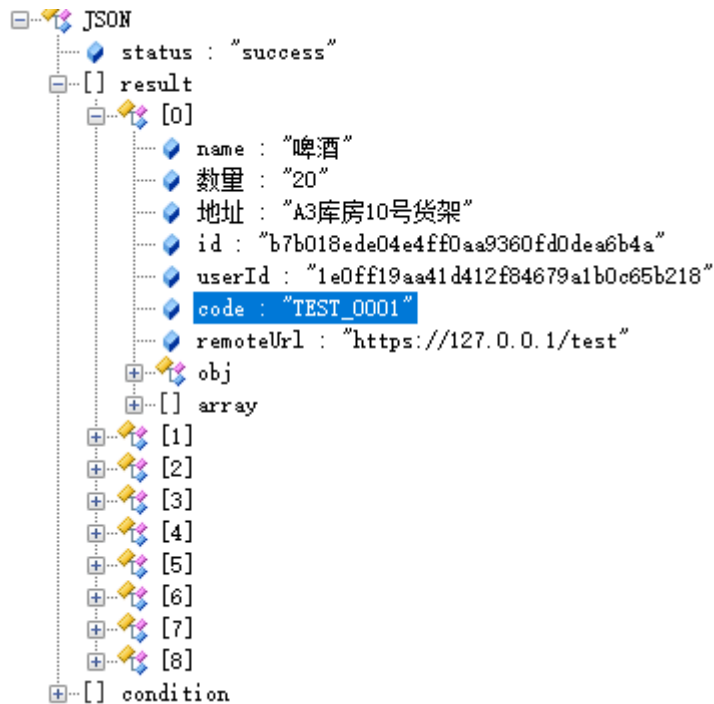
## 字符串提取说明

在设置检查点或者响应提取时，若使用设置属性栏的表达式无法满足需求，则可以在“高级提取类型”处选择“字符串提取”功能。

使用字符串截取功能，“类型值”输入框有两个：

- 第一个框表示起始索引。首位用“0”表示，起始索引包含在截取的字符串中。
- 第二个框表示结束索引。结束索引不包含在截取的字符串中。

以下面响应体为例：



- 若需要在响应体中提取“result”数组第一个元素的“code”的5到9位，与目标值比较，则可以在“检查点”中进行如下配置。



- 若需要提取响应体中提取“result”数组的第一个元素的“code”的5到9位，赋值给变量“code”，则可以在“响应提取”中进行如下配置。

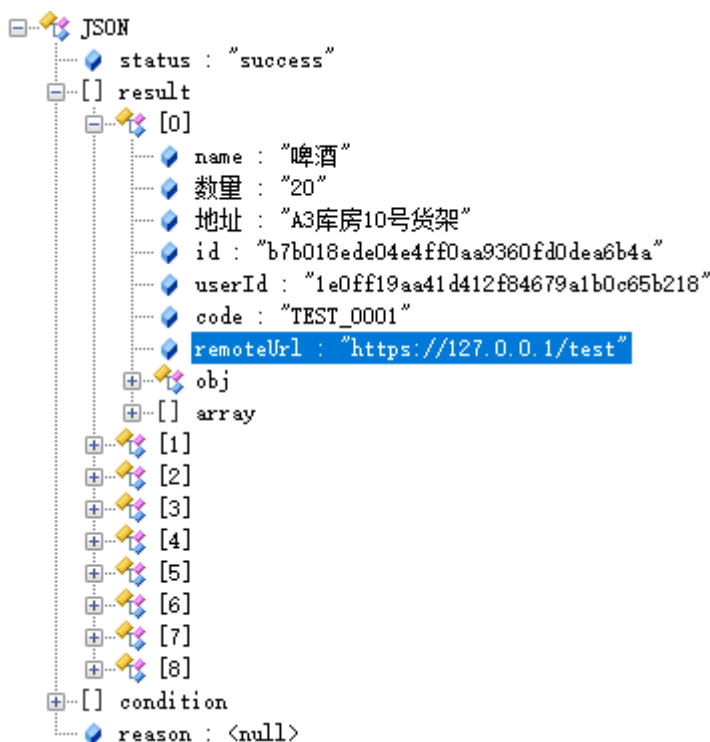


## 正则表达式说明

在设置检查点或者响应提取时，若使用设置属性栏的表达式无法满足需求，或者配合高级类型提取下的字符串提取功能也不能满足需求时，则可以在“高级提取类型”处选择“正则表达式”来提取。

正则表达式引擎使用Java正则表达式引擎。

以下面响应体为例：



- 若检查点验证响应体中“result”数组的第一个元素的“remoteUrl”，使用正则匹配得到IP值和目标值比较，则配置如下。其中，正则表达式类型值设置为“((2[0-4]\d|25[0-5])|[01]?\d\d?)\.){3}(2[0-4]\d|25[0-5])|[01]?\d\d?”。



- 若响应提取响应体中“result”数组的第一个元素的“remoteUrl”，使用正则匹配得到IP值赋值给变量“remotep”，则配置如下。其中，正则表达式类型值设置为“((2[0-4]\d|25[0-5])|[01]?\d\d?)\.){3}(2[0-4]\d|25[0-5])|[01]?\d\d?”。





## 比较符说明

测试计划服务支持以下比较类型：

比较类型	比较符	是否需要填入值	举例
数字比较	<ul style="list-style-type: none"> <li>等于(数字)</li> <li>不等于(数字)</li> <li>大于等于(数字)</li> <li>小于等于(数字)</li> <li>大于(数字)</li> <li>小于(数字)</li> </ul>	是	<ul style="list-style-type: none"> <li>响应码 <b>等于</b> 200</li> <li>响应码 <b>不等于</b> 200</li> <li>响应码 <b>大于等于</b> 200</li> <li>响应码 <b>小于等于</b> 200</li> <li>响应码 <b>大于</b> 200</li> <li>响应码 <b>小于</b> 200</li> </ul>
字符串比较	<ul style="list-style-type: none"> <li>等于(字符串、区分大小写)</li> <li>不等于(字符串、区分大小写)</li> <li>等于(字符串、不区分大小写)</li> <li>包含(字符串)</li> <li>不包含(字符串)</li> </ul>	是	<ul style="list-style-type: none"> <li>响应体中的属性param1 <b>等于</b> test</li> <li>响应体中的属性param2 <b>不等于</b> test</li> <li>响应体中的属性param3 <b>等于</b> TEST</li> <li>响应体中的属性param4<b>包含</b> tri</li> <li>响应体中的属性param5<b>不包含</b> tri</li> </ul>
正则比较	正则表达式	是	<ul style="list-style-type: none"> <li>响应体中的属性param1 <b>正则表达式</b> <code>^[A-Za-z0-9]{1,32}\$</code></li> </ul>
通用比较	<ul style="list-style-type: none"> <li>是空（该字段是否为空，即是否没有该字段）</li> <li>不是空（该字段是否不为空，即是否有该字段）</li> </ul>	否	<ul style="list-style-type: none"> <li>响应体中的属性param1 <b>是空</b></li> <li>响应体中的属性param2 <b>不是空</b></li> </ul>
Json数组比较	<ul style="list-style-type: none"> <li>JSON数组为空数组（json数组是否为空数组）</li> <li>JSON数组为非空数组（json数组是否为非空数组）</li> </ul>	否	<ul style="list-style-type: none"> <li>响应体中的属性param1 <b>JSON数组为空数组</b></li> <li>响应体中的属性param2 <b>JSON数组为非空数组</b></li> </ul>
	JSON数组的大小（json数组的大小是否为...）	是	<ul style="list-style-type: none"> <li>响应体中的属性param1 <b>JSON数组的大小</b></li> </ul>

比较类型	比较符	是否需要填入值	举例
类型比较	<ul style="list-style-type: none"> <li>是JSON类型（是否是json对象类型）</li> <li>是JSON数组类型（是否是json数组类型）</li> </ul>	否	<ul style="list-style-type: none"> <li>响应体中的属性param1是JSON类型</li> <li>响应体中的属性param2是JSON数组类型</li> </ul>
Json对象比较	JSON等于（json等于）	是	<ul style="list-style-type: none"> <li>响应体中的属性param1JSON等于{"name":"zhangsan"}</li> </ul>

## 响应码检查

将响应码与目标值进行比较，例如：检查响应码是否等于“200”。



## 响应头检查

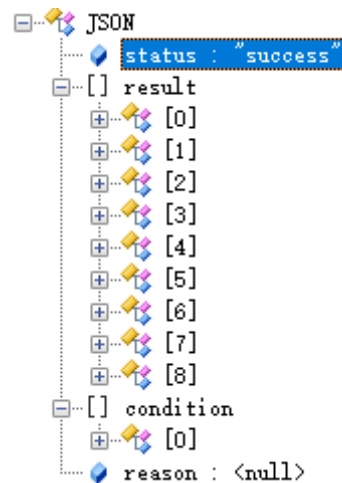
将响应头中字段的值和目标值进行比较，例如：检查响应头中“content-type”的值是否等于“text/plain;charset=UTF-8”。



## 响应体（JSON）检查

1. 检查响应体（JSON）中对象字段的值。例如：

响应体结构如下：

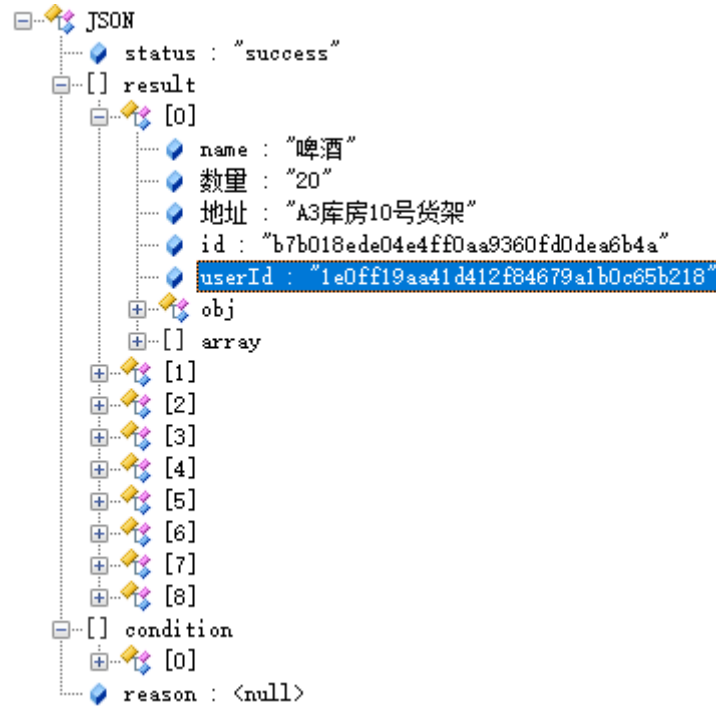


检查响应体对象中字段名为“status”的值，检查点配置如下：



2. 检查响应体 (JSON) 中某个数组的对象的字段值 (数组条件采用下标确定对象, 下标从0开始)。例如：

响应体结构如下：

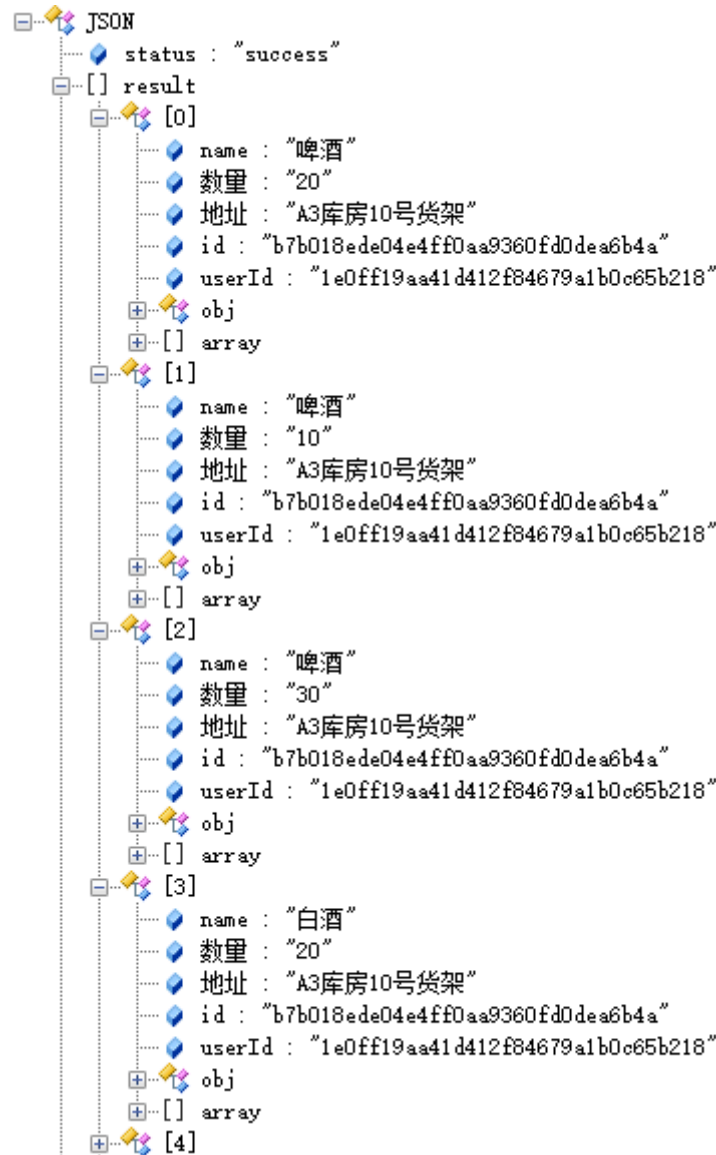


检查响应体“result”数组的第1个元素对象字段名为“userId”的值，检查点配置如下：



3. 检查响应体 (JSON) 中某个数组的对象的字段值 (数组条件采用模糊匹配功能确定对象)。例如：

响应体结构如下：



- a. 检查响应体“result”数组下“name”是“啤酒”的所有对象取第1个之后获取id的值，检查点配置如下：



### 说明

当获取的数组下的对象只有一个，“[0]”可以省略，示例中的表达式可以写为“result[name==啤酒].id”。

- b. 检查响应体“result”数组下“name”是“啤酒”的并且“obj”对象下属性“a”的值是“2”的对象的id值，检查点配置如下：



## 变量检查

检查比较全局变量，局部变量，响应提取后的变量值和目标值。例如：

- 检查全局变量“hostIp”的值是否等于“127.0.0.1”。
- 检查局部变量“local”的值是否等于“text”。
- 检查响应提取后的变量“name”的值是否等于“白酒”。



## 示例：根据给定的 key:value 从响应体中获取字符串

检查体的“属性”字段，可输入下标为“key:value”格式的数组，用以实现根据给定的“key:value”从响应体中获取满足该条件的json串。

“key:value”之间支持以下连接符：

连接符	说明	举例
==	等于（字符串、数值），支持汉字。	name==张三 age==20
!=	不等于（字符串、数值），支持汉字。	name!=张三 age!=20
>	大于（数值）	age>20
>=	大于等于（数值）	age>=20
<	小于（数值）	age<20
<=	小于等于（数值）	age<=20

数组下标支持以“&&”连接的多组“key:value”条件，表示提取的JSON串需要同时满足所有的“key:value”条件。例如：student[age>20&&gender=女]，表示提取条件为年龄大于20岁且性别为女。

数组的搜索条件中支持嵌套数组，在此情况下子嵌套条件中的对象必须全部满足条件，才能使得子条件成立。

### 📖 说明

- 若value值为空对象，则需要使用\$null，即key==\$null。
- 若value值是null串（"null"），则使用null，即key==null。
- 若value值是空串（""），则可以不写，即key==。
- 如果检查点的属性值中value含有特殊字符，需要使用单引号或者双引号作为边界符  
eg: name=="zhangsan(ab)"

以下面的响应体（JSON格式）为例：

```
{
  "status": "success",
  "result": [
    {
      "name": "啤酒",
      "数量": "20",
      "地址": "A3库房10号货架",
      "obj": {
        "a": 1,
        "b": "test",
        "c": "测试"
      },
      "array": [
        {
          "id": 1,
          "name": "aaa"
        },
        {
          "id": 2,
          "name": "bbb"
        }
      ]
    },
    {
      "name": "啤酒",
      "数量": "10",
      "地址": "A3库房10号货架",
      "obj": {
        "a": 1,
        "b": "test",
        "c": "测试"
      },
      "array": [
        {
          "id": 1,
          "name": "aaa"
        },
        {
          "id": 2,
          "name": "bbb"
        }
      ]
    },
    {
      "name": "白酒",
      "数量": "20",
      "地址": "A3库房10号货架",
      "obj": {
        "a": 1,
        "b": "test",
        "c": "测试"
      },
      "array": [
        {
          "id": 1,
          "name": "aaa"
        },
        {
          "id": 2,
          "name": "bbb"
        }
      ]
    },
    {
      "name": "白酒",
      "数量": "30",
      "地址": "A3库房10号货架",
      "obj": {
        "a": 1,
```

```
    "b": "test",
    "c": "测试"
  },
  "array": [
    {
      "id": 3,
      "name": "aaa"
    },
    {
      "id": 4,
      "name": "bbb"
    }
  ]
},
{
  "name": null,
  "数量": "10",
  "地址": "A3库房10号货架",
  "obj": {
    "a": 2,
    "b": "test",
    "c": "测试"
  },
  "array": [
    {
      "id": 5,
      "name": "aaa"
    },
    {
      "id": 6,
      "name": "bbb"
    }
  ]
},
{
  "name": "",
  "数量": "10",
  "地址": "A3库房10号货架",
  "obj": {
    "a": 2,
    "b": "test",
    "c": "测试"
  },
  "array": [
    {
      "id": 5,
      "name": "aaa"
    },
    {
      "id": 6,
      "name": "bbb"
    }
  ]
}
],
"condition": [
  {
    "name": "啤酒",
    "数量": "120",
    "地址": "A1库房15号货架"
  }
],
"reason": null
}
```

从响应体中获取数据与对应的表达式如下表所示：

提取条件	表达式
获取result数组中name是白酒的数据。	result[name==白酒]
获取result数组中name不是啤酒、且数量大于20的数据。	result[name!=啤酒&&数量>20]
获取result数组中name是空对象、且array数组中id小于等于2的数据。	result[name==\$null&&array[id<=2]]
获取result数组中name是null串、且obj对象的属性a等于2的数据。	result[name==null&&obj.a==2]
获取result数组中name是空串（""）的数据。	result[name==]

若需要检查（JSON格式）name是白酒、且数量大于20的数据是不是“[{\"name\": \"白酒\", \"数量\": \"30\", \"地址\": \"A3库房10号货架\", \"obj\": {\"a\": 1, \"b\": \"test\", \"c\": \"测试\"}, \"array\": [{\"id\": 3, \"name\": \"aaa\"}, {\"id\": 4, \"name\": \"bbb\"}]}]”，则检查点各字段的配置可参考下表：

字段	值
来源	响应体（JSON）
属性	result[name!=啤酒&&数量>20]
比较符	等于(字符串)
目标值	[{\"name\": \"白酒\", \"数量\": \"30\", \"地址\": \"A3库房10号货架\", \"obj\": {\"a\": 1, \"b\": \"test\", \"c\": \"测试\"}, \"array\": [{\"id\": 3, \"name\": \"aaa\"}, {\"id\": 4, \"name\": \"bbb\"}]}]

## 示例：根据 JsonPath 从响应体中获取数据

JsonPath详情可参见[官网](#)。

以下面的响应体（JSON格式）为例：

```
{
  "store": {
    "book": [
      {
        "category": "reference",
        "author": "Nigel Rees",
        "title": "Sayings of the Century",
        "price": 8.95
      },
      {
        "category": "fiction",
        "author": "Evelyn Waugh",
        "title": "Sword of Honour",
        "price": 12.99
      },
      {
        "category": "fiction",
        "author": "Herman Melville",
```



```

    "title": "Moby Dick",
    "isbn": "0-553-21311-3",
    "price": 8.99
  },
  {
    "category": "fiction",
    "author": "J. R. R. Tolkien",
    "title": "The Lord of the Rings",
    "isbn": "0-395-19395-8",
    "price": 22.99
  }
],
"bicycle": {
  "color": "red",
  "price": 19.95
}
},
"expensive": 10
}

```

JsonPath表达式可以使用点符号或者括号符号，例如：

- \$.store.book[0].title
- \$('[store]')['book'][0]['title']

JsonPath表达式与其对应的含义如下表所示：

JsonPath表达式	含义描述
\$.store.book[*].author	所有书籍的作者
\$.author	所有作者
\$.store.*	所有的东西，包括书籍和自行车
\$.store..price	所有东西的价格
\$.book[2]	第三本书
\$.book[-2]	倒数第二本书
\$.book[0,1]	前两本书
\$.book[:2]	从索引 0（包括）到索引 2（不包括）的所有书籍
\$.book[1:2]	从索引 1（包括）到索引 2（不包括）的所有书籍
\$.book[-2:]	最后两本书
\$.book[2:]	从末尾开始的两本书
\$.book[?(@.isbn)]	所有有isbn的书
\$.store.book[?(@.price < 10)]	商店里所有价格小于10的书
\$.book[?(@.price <= \$('[expensive'])]	商店里所有不“贵”的书
\$.book[?(@.author =~ /. *REES/i)]	所有作者匹配正则表达式.*REES（忽略大小写）的书籍
\$.*	所有内容

**说明**

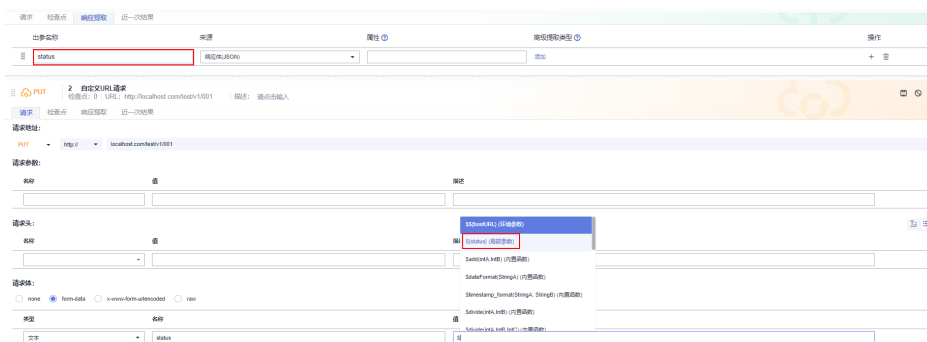
若使用length函数或者size函数，则多次调用时不要使用深度扫描（即“..”符号），例如“\$.book.length()”返回书的数量，此时需要确定路径，改为“\$.store.book.length()”。

### 6.4.3.3 设置 CodeArts TestPlan 接口脚本的响应提取

响应提取是提取接口响应结果的某一部分，命名为参数，供后续测试步骤参数化调用。响应提取需要在前序测试步骤定义，后续测试步骤使用。

- 在前序测试步骤中，在“响应提取”页签创建要传递的参数。响应提取的来源需要用到内置参数，请参考[内置参数](#)了解如何使用内置参数。响应提取同时支持正则表达式的匹配，提取出与给定正则表达式匹配的返回值。
- 在后续测试步骤中，通过`${参数名}`方式引用前序测试步骤创建的响应提取。后续步骤的URL、请求头、请求体中均可以引用此参数。如果在JSON格式的请求体中引用此参数，请在参数外使用英文引号，如：

```
{
  id: "用例ID"
  name: "${name}"
}
```



- 响应提取支持根据给定的“key:value”获取字符串，详细配置可参考[示例：根据给定的key:value从响应体中获取字符串](#)。

字段	说明
出参名称	用于之后使用 <code>\${出参名称}</code> 来引用此参数，名称使用字母数字下划线。
来源	被检测字段的来源，如响应体（JSON）、响应头、响应码。

字段	说明
属性	<p>在属性中，支持输入\$调用全局变量、局部变量、内置函数。</p> <ul style="list-style-type: none"> <li>若来源是响应码，属性为空。详细介绍请参见<a href="#">响应码检查</a>。</li> <li>若来源是响应头，属性为响应头中字段的名称。详细介绍请参见<a href="#">响应头检查</a>。</li> <li>若来源是响应体（JSON），属性有两种填写方法：               <ol style="list-style-type: none"> <li>普通提取表达式（非“\$”开头），例如“item.name”。取字段中的值，支持嵌套取值。详细介绍请参见<a href="#">响应体（JSON）检查</a>。 从响应体中提取数组时，下标可以是数字，也可以是“key:value”表达式，详细介绍请参见<a href="#">示例：根据给定的key:value从响应体中获取字符串</a>。</li> <li>JsonPath表达式（“\$.”或“\$[”开头），例如“\$.store.book[0].title”。详细介绍请参见<a href="#">示例：根据JsonPath从响应体中获取数据</a>。</li> </ol> </li> </ul>
高级提取类型	<p>可选项，使用高级提取类型，辅助提取响应结果信息，若选择不涉及则视为不使用额外的方式匹配。</p> <p>目前有两种方式：</p> <ul style="list-style-type: none"> <li>字符串提取，也就是字符串的截取，详见<a href="#">字符串提取说明</a>。</li> <li>正则表达式，即使用正则方式对来源字符串进行过滤，详见<a href="#">正则表达式说明</a>。</li> </ul> <p>高级提取类型优先使用字符串提取功能，若不能满足需求可考虑使用正则表达式。</p>
赋值给环境动态参数	<p>将响应提取后的值赋值给动态参数，用于后续测试引用该<a href="#">动态参数</a>。</p>

## 6.4.4 使用 Postman 快速添加 CodeArts TestPlan 接口脚本

### 背景信息

接口自动化用例支持通过导入Postman文件生成测试步骤。

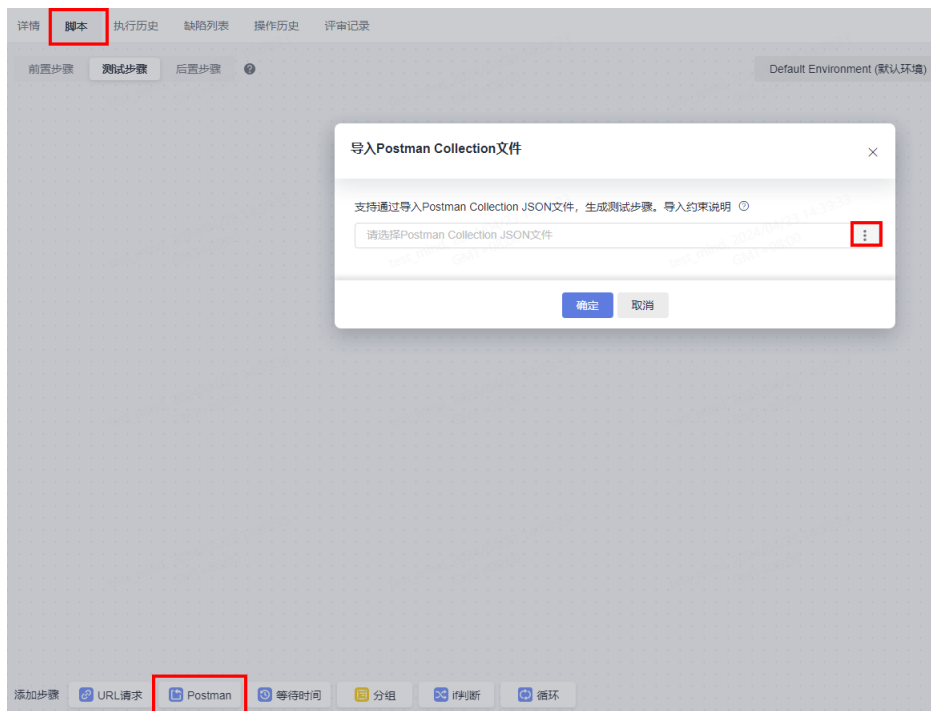
导入Postman需满足以下要求：

- 支持Postman Collection v2.1标准。
- 仅支持Postman请求方法、请求URL、请求头、请求体生成测试步骤。
- Postman请求体导入方式仅支持form-data、x-www-form-urlencoded、raw。
- Postman请求体form-data上传附件需要在测试步骤中单独上传。

### 操作步骤

- 步骤1** 完成[创建CodeArts TestPlan接口自动化测试用例模板](#)操作后，单击用例名称，选择“脚本”页签。

**步骤2** 单击“Postman”，在弹框中单击图标，选择待导入的文件，单击“确定”。



**步骤3** 页面中将显示URL请求，单击右上角“保存”，系统将提示更新成功。

若系统提示保存失败，页面右上角会弹框提示失败原因。

----结束

## 6.4.5 使用 cURL 快速添加 CodeArts TestPlan 接口脚本

### 背景信息

cURL是一个利用URL语法在命令行下工作的文件传输工具。接口自动化支持从Chrome浏览器控制面板中复制接口请求的cURL，一键粘贴自动生成接口测试脚本。

### 操作步骤

**步骤1** 完成[创建CodeArts TestPlan接口自动化测试用例模板](#)操作后，单击用例名称，选择“脚本”页签。

**步骤2** 使用新的浏览器页面打开待测接口所在网页，敲击键盘“F12”打开控制面板。

**步骤3** 选择“Network”页签，在列表中找到接口请求，单击鼠标右键，在弹出的菜单中选择“Copy > Copy as cURL (bash)”。

**步骤4** 返回“脚本”页签，在页面空白处单击鼠标右键，在弹出菜单中选择“添加步骤 > cURL”。



粘贴成功，页面中自动生成测试脚本。



----结束

## 6.4.6 使用关键字库快速添加 CodeArts TestPlan 接口脚本

### 6.4.6.1 CodeArts TestPlan 中关键字库的介绍

#### 背景信息

关键字驱动测试是一种测试自动化的技术，通过提供一组称为关键字的“构建块”创建自动化测试用例。关键字驱动测试可用于组件测试、系统测试等不同的测试级别，其优势体现在易用性、可理解性、可维护性、测试信息的重用、支持测试自动化、节约潜在的成本和进度等方面。

在设计测试用例时，经常会遇到有一些相同的前置步骤或者测试逻辑。如果每一个测试用例中都编写这些步骤，重复工作量很大，并且难以维护。测试关键字可以帮助复用这些测试步骤。

关键字库将接口关键字、组合关键字、系统关键字进行统一管理，打造一站式关键字管理能力，在用例脚本设计时用户体验保持一致。


- 接口关键字中定义单个接口的请求，可以通过导入Swagger文件、保存自定义URL请求等方式生成。

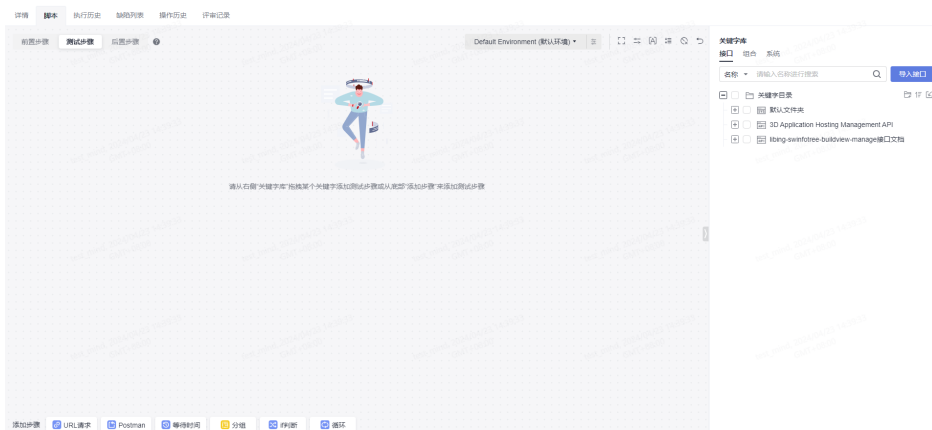
接口关键字请参见[将测试步骤保存为CodeArts TestPlan接口脚本关键字](#)。

- 组合关键字用于将多个步骤封装常用测试逻辑，可以被测试用例调用实现逻辑复用。组合关键字请参见[将测试步骤保存为CodeArts TestPlan组合脚本关键字](#)。
- 系统关键字涵盖认证、协议、中间件、数据库四大类别，覆盖身份认证、复杂协议、数据处理、数据预置、数据验证、接口集成等丰富场景。系统关键字请参见[接口自动化用例系统关键字说明](#)。

## 进入关键字库

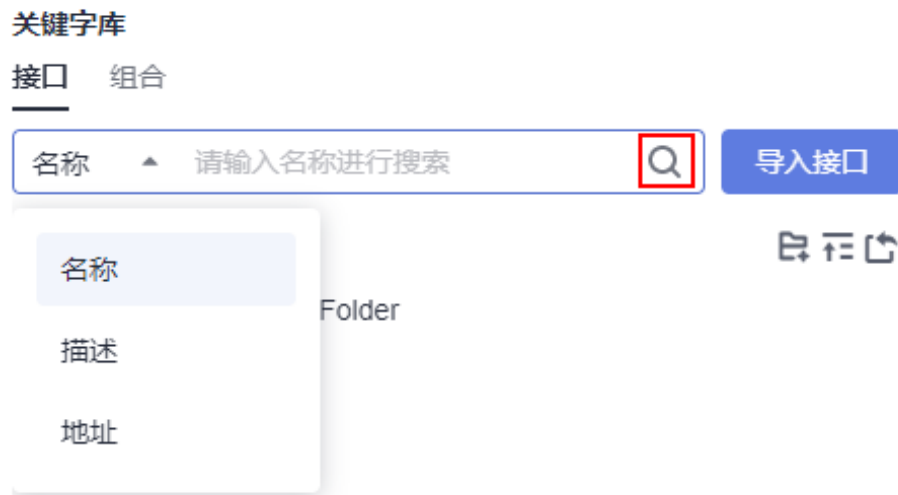
**步骤1** 完成[创建CodeArts TestPlan接口自动化测试用例模板](#)操作后，在“测试用例 > 接口自动化”页面单击需要编辑的用例名称。

**步骤2** 选择“脚本”页签，页面右侧展示“关键字库”（如果没有页面右侧没有显示，单击）。



- Swagger导入关键字放置在关键字库的“接口”页签中，详细操作请参考[将测试步骤保存为CodeArts TestPlan接口脚本关键字](#)。
- 关键字用例和组合关键字放置在关键字库的“组合”页签中，支持从0到1创建新的组合关键字，详细操作请参考[将测试步骤保存为CodeArts TestPlan组合脚本关键字](#)。
- 认证类、数据库操作类、中间件类、协议类等实用关键字放置在“系统”页签中，详情请参考[接口自动化用例关键字](#)。

**步骤3** 单击“接口”页签下的下拉列表，可以按照“名称”、“描述”、“地址”，搜索对应的关键字。



----结束

### 6.4.6.2 将测试步骤保存为 CodeArts TestPlan 接口脚本关键字

#### 背景信息

接口关键字中定义单个接口的请求，可以通过导入Swagger文件、保存自定义URL请求等方式生成。

Swagger是一个定义、开发、调试RESTful等接口的工具，使用Swagger可以标准化地定义接口属性，便于互联互通。接口自动化支持导入Swagger 2.0&3.0格式的接口描述文件、解析接口的定义描述、生成脚本模板，只需要按照模板输入接口参数就可以创建接口自动化用例。

通过导入接口Swagger描述文件可以生成脚本模板，一个脚本模板对应Swagger中的一个接口定义，基于脚本模板可以可视化编排测试用例。

脚本模板和Swagger接口描述中字段的对应关系如下：

- swagger 2.0规范如下：

脚本模板属性	Swagger接口定义属性
名称	默认使用operationId，可在设置中配置使用summary。
路径	schema + :// + basePath + path。
请求参数 hostURL	host
其它请求参数	parameters

- swagger 3.0规范如下：

脚本模板属性	Swagger接口定义属性
名称	默认使用operationId，可在设置中配置使用summary。
路径	url + path。
请求参数 hostURL	servers: - url: https://{hostURL}/variable variables: hostURL: default: test.demo.com
其它请求参数	parameters, requestBody, responses

## 导入 Swagger 文件生成测试脚本

- 步骤1** 完成[创建CodeArts TestPlan接口自动化测试用例模板](#)操作，单击用例名称，选择“脚本”页签。
- 步骤2** 在右侧“关键字库”界面，单击“导入接口”。

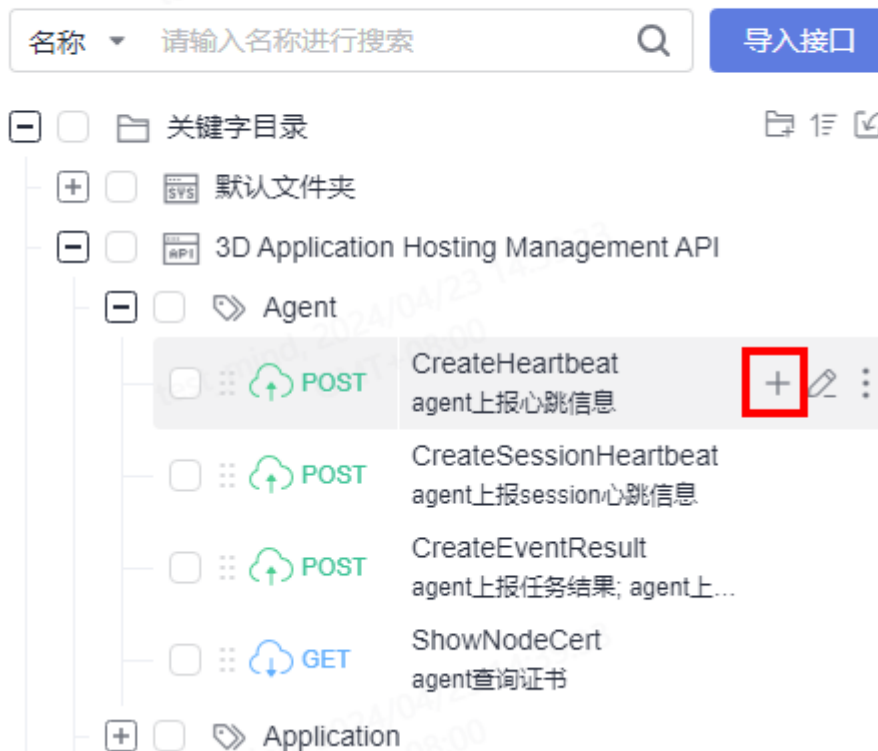


- 步骤3** 单击“点击添加文件或拖拽上传”，选择配置好的Swagger接口文档，单击“确定”。
- 步骤4** 文档导入成功后，系统自动解析生成脚本模板。脚本模板包含了接口的基本描述信息。可以从“关键字库”界面单击+或拖拽接口添加至测试步骤，只需要按模板输入接口参数就可以测试。



### 关键字库

接口 组合 系统

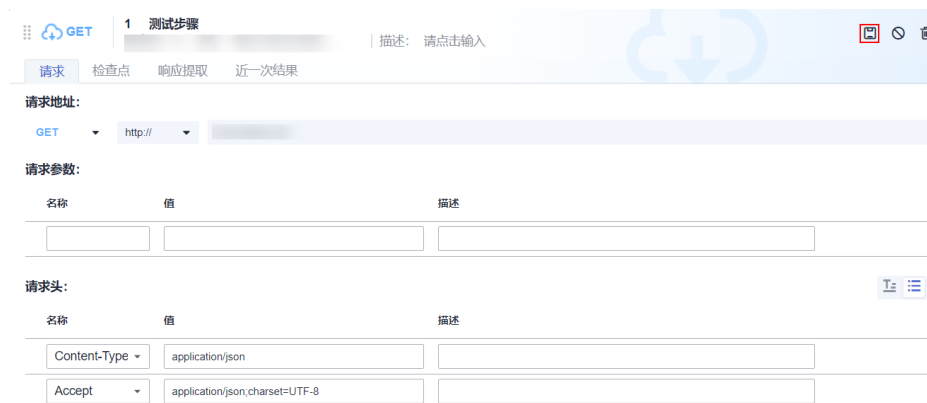


---结束

## 保存自定义接口关键字

**步骤1** 完成[创建CodeArts TestPlan接口自动化测试用例模板](#)操作，单击对应的接口自动化用例名称，选择“脚本”页签。

**步骤2** 完成[使用自定义URL请求添加CodeArts TestPlan接口脚本](#)，选择需要设置为关键字的测试步骤（仅针对自定义URL请求步骤），在该测试步骤页面的右上方单击，保存接口关键字。



**步骤3** 在弹出的页面内，输入“接口关键字名”、“接口关键字描述”并选择该关键字需要储存的目录位置。关键字的默认储存位置为“接口 > 关键字目录”中的默认文件夹。

新建自定义接口关键字

\* 接口关键字名：

请输入接口关键字名

接口关键字描述：

自定义URL请求

\* 存储目录：

- [-] 关键字目录
  - [SYS] 默认文件夹

保存 取消

**步骤4** 单击“保存”完成设置。






----结束

## 管理已添加的关键字

对于已添加的关键字，将鼠标悬浮在关键字区域时，可以选择进行以下操作。



- 单击关键字名称，可以查看关键字的详细信息。


- 单击“关键字目录”右侧的 ，可以创建文件夹。可以将步骤3中设置好的关键字储存在自定义的文件夹内。
- 鼠标悬浮在关键字名称的左侧区域，可以调整关键字的顺序。
- 单击  或鼠标悬浮在关键字拖拽到测试步骤空白页，可以将关键字添加至测试步骤。
- 单击测试关键字任意区域，可以编辑关键字对应用例详情。
- 单击 ，可以查看该关键字的信息。
- 单击 ，可禁用关键字，再次单击可解除禁用。
- 单击 ，可从操作步骤中删除该关键字。

### 6.4.6.3 将测试步骤保存为 CodeArts TestPlan 组合脚本关键字

#### 背景信息

在设计测试用例时，经常会遇到有一些相同的前置步骤或者测试逻辑。如果每一个测试用例中都编写这些步骤，重复工作量很大，并且难以维护。组合关键字用于将多个步骤封装常用测试逻辑，可以被测试用例调用实现逻辑复用。

#### 场景一


**步骤1** 在“脚本”页面右上方单击  图标。

**步骤2** 在页面中输入组合关键字的“名称”、“描述”，选择需要归属的目录，根据需要配置请求参数。勾选已添加的“URL请求”，单击“保存”，完成添加组合关键字。

**步骤3** 单击“关键字库 > 组合”，查看已经保存的组合关键字。

---结束

#### 场景二

**步骤1** 在“关键字库 > 组合”的页面中，单击需要储存的对应文件夹旁边的 。



**步骤2** 在页面中输入组合关键字“名称”、“描述”。

**步骤3** 可以在组合关键字中添加自定义URL请求。



单击“接口”页签，选择需要添加的关键字所在的文件夹，单击需要添加的关键字右侧的 **+** 或鼠标悬浮在关键字区域拖拽至测试步骤区域。

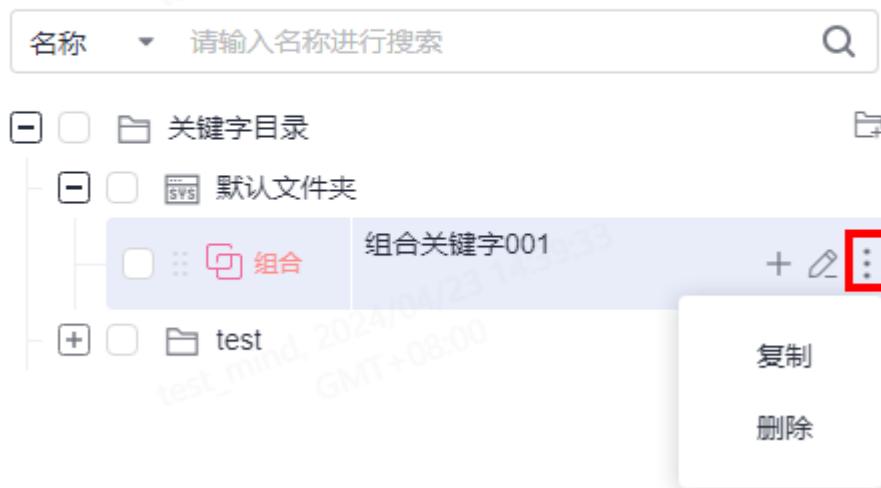
**步骤4** 单击“保存”，完成添加组合关键字。

----结束


对于已添加的组合关键字，将鼠标悬浮在关键字区域时，可以选择进行以下操作。

### 关键字库

接口 组合 系统



- 单击“关键字目录”右侧的 **+**，可以创建文件夹。可以参照**步骤1**在自定义的文件夹内创建组合关键字。
- 单击 **+** 或鼠标悬浮在“组合”拖拽到测试步骤空白页，可以为脚本添加该组合关键字步骤。

- 鼠标悬浮在“组合”，可以调整组合关键字的顺序。
- 单击，可以编辑组合里关键字的信息。
- 单击“标记状态”，可以为组合关键字设置“正常”、“新增”、“更新”其中一种状态。
- 单击“复制”，可以在文件夹下复制组合关键字用例。
- 单击“删除”，可以删除组合关键字。


#### 6.4.6.4 刷新 CodeArts TestPlan 关键字

一个关键字可以重复使用在多个测试用例中，当需要对测试用例中使用到的同一个关键字进行批量修改时，可以使用刷新关键字功能。当前，由导入YAML文件生成的关键字和组合关键字支持刷新关键字功能，自定义URL或者Postman导入的步骤保存的关键字、系统关键字、自定义关键字暂不支持刷新关键字功能。

#### 前提条件

需要刷新的关键字已使用于测试用例或组合关键字中。

#### 操作步骤

- 步骤1** 登录软件开发生产线首页，搜索目标项目并单击项目名称，进入项目。
- 步骤2** 单击导航栏“测试 > 测试用例”。
- 步骤3** 在“接口自动化”页签中单击用例名称，选择“脚本”页签。
- 步骤4** 在右侧“关键字库”界面，单击待刷新的关键字后面的，单击“刷新全部”。



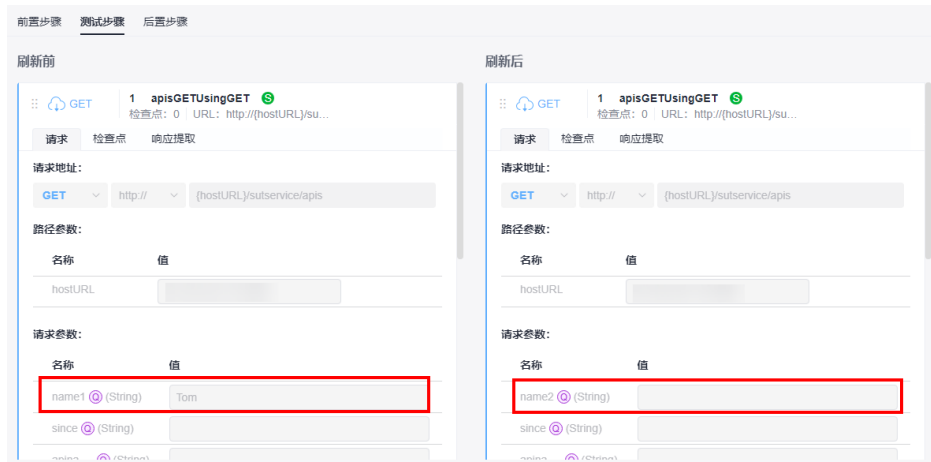
- 步骤5** 在刷新关键字弹框右侧，单击“配置”，配置关键字的刷新规则，支持不同参数间映射继承。

例如YAML文件参数名称“name1”更新为“name2”，如下图：

```
33 operationId: apisGETUsingGET
34 deprecated: true
35 consumes:
36   - application/json
37 produces:
38   - */*
39 parameters:
40   - name: name1
41     in: query
42     description: namespace
43     required: true
44     type: string
45     minimum: 0
46     maximum: 100
```

```
33 operationId: apisGETUsingGET
34 deprecated: true
35 consumes:
36   - application/json
37 produces:
38   - */*
39 parameters:
40   - name: name2
41     in: query
42     description: namespace
43     required: true
44     type: string
45     minimum: 0
46     maximum: 100
```

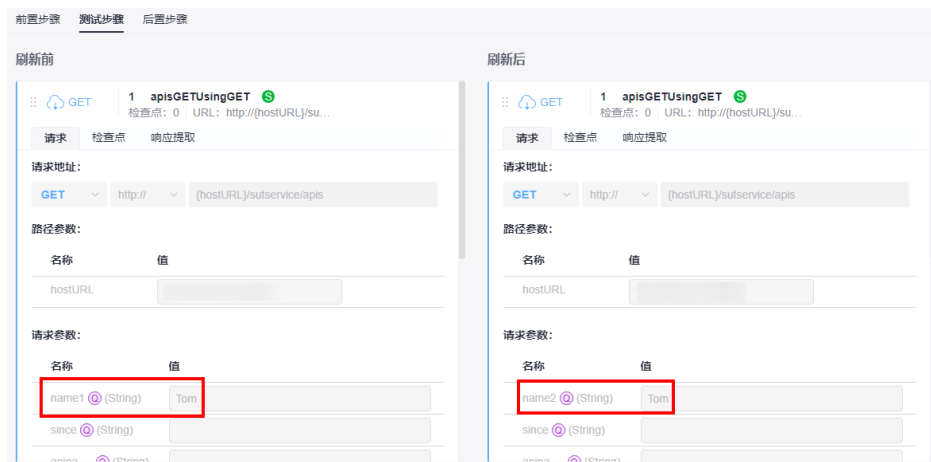
导入更新的YAML文件后，在刷新关键字弹框页面中可以看到，“name1”的值不会自动继承到“name2”下。



需要配置刷新规则如下：



单击“确定”后，在刷新关键字弹框页面中可以看到值成功继承。






### 📖 说明

刷新规则配置说明：

- 刷新规则中，新参数名称不能重复。
- 在脚本中，表单切换文本时，可以刷新关键字的内容，但是展示的文本不会随着关键字刷新。

**步骤6** 在刷新关键字弹框左侧，选择是否更新回收站用例，勾选“更新回收站用例”表示回收站用例的关键字同步更新。

- 测试用例名称前面带有  标识的，表示该测试用例已被删除到回收站中。
- 测试用例名称前面带有  标识的，表示该测试用例可正常使用。

**步骤7** 批量更新时将包含该关键字的用例及组合关键字进行数据更新。如果不需要更新某个用例或者组合关键字，单击该用例或者组合关键字名称后面的  将其删除。删除后如果做搜索操作，删除过的用例或者组合关键字也会出现在搜索结果列表中。

**步骤8** 单击“确定”，确认提示信息后刷新关键字。

---

### 须知

参数刷新后，该操作不可回退，请小心操作。

---

----结束

## 6.4.7 添加 CodeArts TestPlan 接口脚本的逻辑控制步骤

逻辑控制用于编排测试场景，包括等待时间、分组、判断、循环。

### 等待时间

如果执行某个测试步骤后需要间隔一段时间再继续执行后续测试步骤，可以使用等待时间。

等待时间设置方式如下：

**步骤1** 进入接口自动化用例编辑的“脚本”页签，单击“等待时间”。

**步骤2** 输入等待时间，支持范围在1~60的整数。



----结束

## 分组

**步骤1** 进入接口自动化用例编辑的“脚本”页签，单击“分组”。

**步骤2** 输入分组名称，并拖拽相关的测试步骤到分组。

- 分组中可添加URL请求、判断、等待、循环。
- 支持拖拽编排分组在测试用例中的顺序。
- 支持拖拽编排分组内部测试步骤的顺序。
- 支持整体禁用或者删除分组。



----结束

## 判断

如果需要根据前序测试步骤的结果决定后续需要执行的测试步骤，可使用判断。

判断设置方式如下：

**步骤1** 进入接口自动化用例编辑的“脚本”页签，单击“if判断”。



- 步骤2** 添加判断逻辑，输入if判断条件，在分支中添加后续测试步骤。  
判断分支中可添加URL请求、判断、等待、循环、测试关键字。



----结束

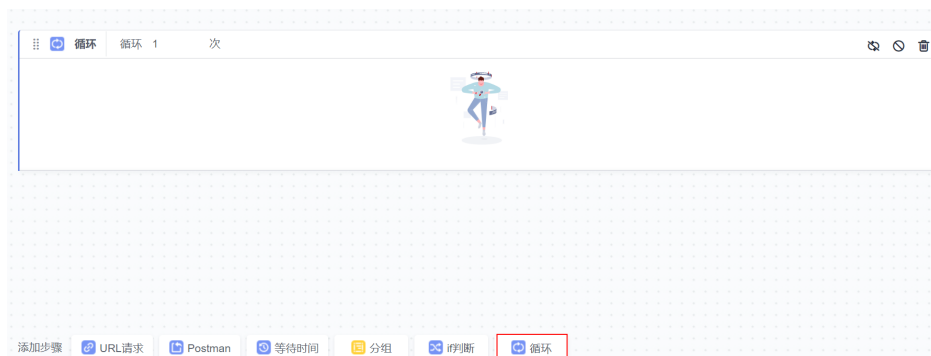
#### 📖 说明

if判断支持引用局部变量和传递参数，比较符的使用请参见[比较符说明](#)。

## 循环

- 步骤1** 进入接口自动化用例编辑的“脚本”页签，单击“循环”。
- 步骤2** 设置循环次数。

循环中可添加URL请求、判断、分组、等待、测试关键字。



----结束

## 6.4.8 设置 CodeArts TestPlan 接口脚本的测试用例参数

### 背景信息

合理的测试设计要求分离测试逻辑和测试数据，实现测试逻辑的最大化复用，增强测试用例的可维护性和投入产出比。例如不同测试环境的URL域名是一种独立于测试逻辑，和测试环境相关的测试数据。测试用例参数可以用来管理这些测试数据。

接口自动化中的测试用例参数分为三类：

- [局部参数](#)
- [内置参数](#)
- [响应提取参数](#)

## 局部参数

局部参数使用范围是当前测试用例，如：测试步骤的参数、检查点、变量等都可以引用局部参数。

局部参数引用形式为“\${参数名}”，如：参数名为“hostIp”，则可以使用“\${hostIp}”来引用该参数。

局部参数主要配置项如下：

表 6-1 参数主要配置项



配置项	是否必填	描述
名称	是	支持中英文、数字、点号、中文短横线、下划线的半角字符，并且长度范围是1~300。
类型	是	支持文本、随机字符串、随机整数、时间戳、格式化时间戳、生成UUID、Base64编码、MD5哈希值、密码或认证信息、SHA512编码。 详情介绍请参考 <a href="#">表6-2</a> 。
描述	否	对参数的简要描述，上限为3000字符。默认为当前参数类型的描述。 单击“文本框”后，用户可以直接输入文本描述。 单击  ，输入一个json，单击“Json转换”，可以给没有换行的Json添加换行和缩进，单击“回填”添加成功。
值	否	用户可以为不同类型的参数赋值，详情请参考 <a href="#">表6-2</a> 。
敏感变量	否	勾选为敏感数据后，测试计划对参数值加密存储，在测试结果日志中使用星号覆盖处理。敏感数据类型适用并不限于个人信息、鉴权信息，例如姓名、地址、用户名等。
动态变量	否	动态参数的值可以在用例执行过程中被赋值。动态参数初始值可为空，被赋值之后，此处显示的是最新值。 动态参数赋值方法：在用例测试步骤“响应提取”的“赋值给环境动态参数”中设置后，在测试执行时，响应提取的内容将被赋值给动态参数。

表 6-2 参数类型

名称	描述
文本	上限为10000字符，支持设置“敏感参数”、“动态参数”，默认为否。
随机字符串	随机生成指定长度字符串，随机字符串的变量值长度校验范围【1-9999】，不支持设置“敏感参数”及“动态参数”。
随机整数	随机生成指定范围整数，区间范围校验为【-999999999~999999999】，不支持设置“敏感参数”及“动态参数”。 例如：设置【-9999,9999】，会获取这个区间内的随机整数。
时间戳	无需输入，生成当前整数时间戳，不支持设置“敏感参数”及“动态参数”。
格式化时间戳	格式为yyyy-MM-dd HH:mm:ss或yyyy-MM-dd，不支持设置“敏感参数”及“动态参数”。 例如：1.yyyy-MM-dd HH:mm:ss: 33250825252000，预期值为3023-09-05 20:20:52。 2.yyyy-MM-dd: 33250825252000，预期值为3023-09-05。
生成UUID	无需输入，不支持设置“敏感参数”及“动态参数”。
Base64编码	使用Base64方法编码参数，上限为256字符，不支持设置“敏感参数”及“动态参数”。
MD5哈希值	使用指定参数生成MD5哈希值，上限为256字符，不支持设置“敏感参数”及“动态参数”。
密码或认证信息	上限为256字符，不支持设置“敏感参数”及“动态参数”。默认勾选“敏感参数”。
SHA512编码	上限为256字符，不支持设置“敏感参数”及“动态参数”。默认勾选“敏感参数”。
数组	数组的内容是json数组格式，上限为10000字符，不支持设置“敏感参数”及“动态参数”。

局部参数配置方法如下：

**步骤1** 进入接口自动化用例编辑的“脚本”页签，单击图标。

**步骤2** 单击左上方“新建变量”，输入参数名、类型、值。

全部参数设置完毕，单击“保存”。

----**结束**

## 内置参数

内置参数是将HTTP/HTTPS响应的对应部分参数化，在检查点、响应提取功能中的“来源”选项中选择内置参数。

测试计划服务中的内置参数如下表：

内置参数	参数说明	是否支持多级取值	用途	举例
响应体 (json)	表示接口返回的响应体。	是	<ul style="list-style-type: none"><li>检查点的属性字段</li><li>参数传递的属性字段</li></ul>	<ul style="list-style-type: none"><li>检查点：判断响应体中的id等于100。</li><li>设置方法：设置检查点来源为响应体（JSON），属性为id（前提条件响应体JSON串有id字段），设置比较符为等于（字符串、不区分大小写），设置目标值为100。</li></ul>
响应头	表示接口返回的响应头。	是	<ul style="list-style-type: none"><li>检查点的属性字段</li><li>参数传递的属性字段</li></ul>	<ul style="list-style-type: none"><li>检查点：判断响应头中的token等于abcd。</li><li>设置方法：设置来源为响应头，属性为token（前提条件响应头有token），设置比较符为等于（字符串、不区分大小写），设置目标值为abcd。</li></ul>
响应码	表示接口返回的响应码。	否	<ul style="list-style-type: none"><li>检查点的属性或值字</li><li>变量的属性字段</li></ul>	<ul style="list-style-type: none"><li>检查点：判断响应码等于200。</li><li>设置方法：设置来源为响应码，设置比较符为等于（数字），设置目标值为200。</li></ul>

### 说明

内置参数支持多级取值，例如

- 响应体为“`{\"result\":{\"appId\":\"12\"}}`”时，则取appId的格式为：来源选择响应体，属性为“`result.appId`”。如果result是数组格式。则属性为“`result[i].appId`”，其中，i为非负整数。

## 响应提取参数

响应提取参数是从接口的响应体中提取出来的参数，定义及使用方法请参考[设置 CodeArts TestPlan接口脚本的响应提取](#)。

## 6.4.9 设置 CodeArts TestPlan 接口脚本的环境参数

在自动化测试中，通常存在多个测试环境，每个测试环境的环境参数值会有不同，例如域名、账号等。这些参数常会在测试脚本中被使用到，如果将这些参数硬绑定到测试脚本中，脚本的冗余度很高，复用度很低。

为了解决上述问题，可以采用环境参数的方式统一管理和环境相关的参数，在测试脚本中参数化引用环境参数，执行时只需选择执行环境，即可使用对应的环境参数值完成测试。

### 环境参数使用范围


当前项目的所有测试用例，各个测试用例中的测试步骤的参数、检查点、变量、URL 等都可以引用环境参数。

### 环境参数引用形式

环境参数的引用形式为“`$${参数名}`”。例如：参数名为“hostname”，则可以使用“`$${hostname}`”来引用该参数。

根据需要可以对环境参数进行分组配置管理，例如：参数hostname在类生产环境中的值为“stage.example.com”，在生产环境中的值为“prod.example.com”，测试脚本使用“`$${hostname}`”引用该参数，执行测试时选择不同环境执行，实现一套接口自动化用例在所有环境复用。

### 环境参数设置方法

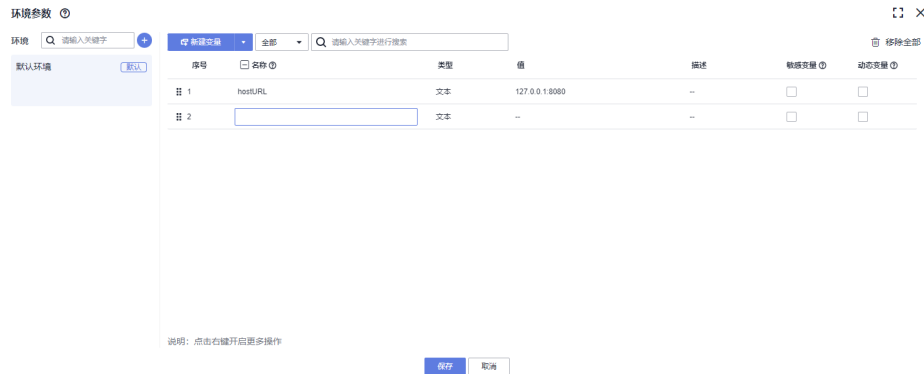
**步骤1** 进入接口自动化用例编辑的“脚本”页签，单击图标.




**步骤2** 单击“新建变量”，输入参数信息，单击“确定”。

配置项	说明
参数名	参数的名称。
类型	参数的类型。支持文本、随机字符串、随机整数、时间戳等多种类型。
值	参数的值。
描述	对参数的描述。
敏感参数	勾选为敏感数据后，对参数值加密存储，在测试结果日志中使用星号覆盖处理。敏感数据类型适用并不限于个人信息、鉴权信息，例如姓名、地址、用户名等。

配置项	说明
动态参数	<p>动态参数设置，动态参数的值可以在用例执行过程中被赋值。动态参数初始值可为空，被赋值之后，此处显示的是最新值。</p> <p>在用例测试步骤“响应提取”的“赋值给环境动态参数”列中设置后，在测试执行时，响应提取的内容将被赋值给动态参数。详细介绍请参见<a href="#">动态变量说明</a>。</p>



**步骤3** 页面中会有一个默认环境。

如果需要设置其它环境为默认环境，单击该环境卡片右上角图标 ，在下拉列表中选择“设为默认环境”。



----结束

### 环境参数值使用内置函数的设置说明

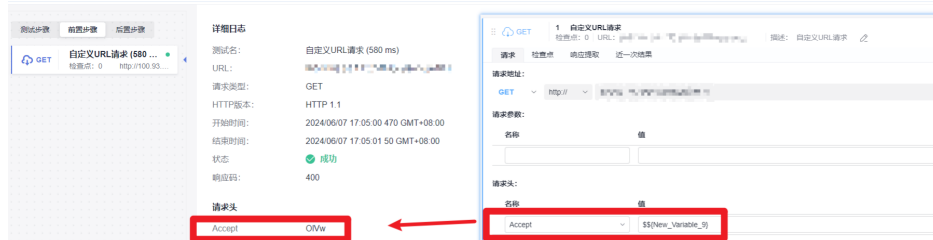
1. 环境参数不支持内置函数嵌套，如果有嵌套则原样输出。
2. 环境参数支持的内置函数如[表6-3](#)。

表 6-3 环境参数支持的内置函数表

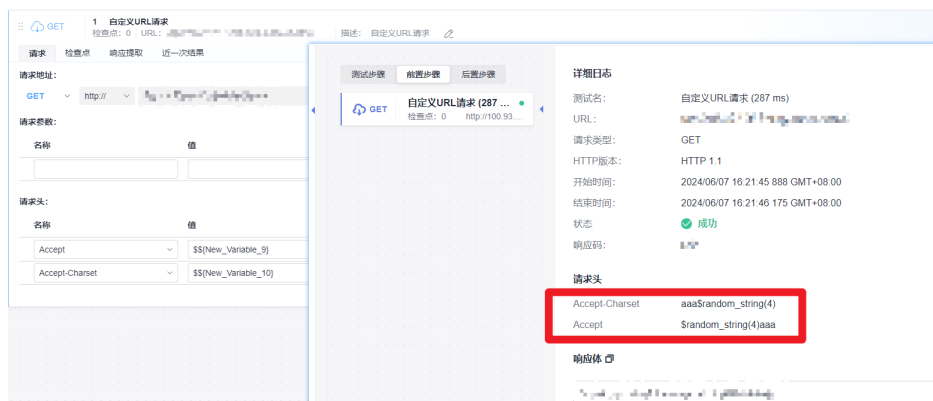
序号	函数	描述
1	\$random_string(intA)	生成指定长度随机字符串
2	\$random_int(intA, intB)	生成指定范围随机数
3	\$timestamp()	获取当前时间戳
4	\$timestamp_format(String A, String B)	时间戳转日期
5	\$uuid()	生成UUID
6	\$encode_base64(StringA)	生成Base64编码
7	\$md5(StringA)	生成MD5哈希值
8	\$sha512(StringA)	生成SHA512编码

3. 环境参数支持内置函数使用举例如下。

- 正常场景：引用\$random\_string(intA)，输出\$random\_string(intA)。



- 正常场景：函数前后拼接字符，按照字符串解析，原样输出。例如引用aa \$random\_string(10)或者\$random\_string(10)aa，则输出aa \$random\_string(10)或者\$random\_string(10)aa。

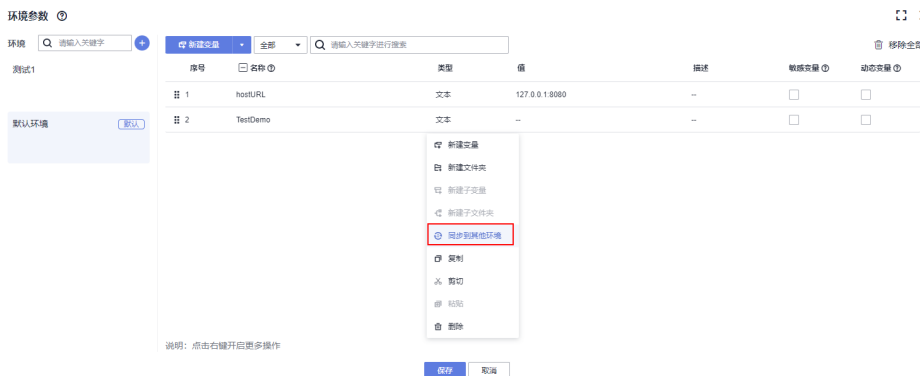


- 异常场景：传参所需一个参数，实际传两个参数，例如 \$random\_string(1,1)，执行时抛出异常。  
传参类型和实际函数所需参数类型不匹配，例如\$random\_string(fff)或 \$random\_string()，执行时抛出异常。



## 同步环境参数

鼠标右键单击需要同步的参数，选择“同步到其他环境”，可以将当前参数同步到全部的环境下。



## 动态变量说明

测试套件中的多个用例间可能存在上下文关系，后执行的用例依赖先执行用例中接口的返回结果。例如所有的接口都需要鉴权信息，且鉴权信息有时效性，如果在每个用例里都去获取鉴权信息，用例的测试步骤会显得冗余，不易维护。

使用动态全局变量即可避免上述问题，在第一个执行的用例中获取鉴权信息后赋值给动态全局变量，后面执行的用例直接使用动态全局变量即可，无需重复获取鉴权信息。

### 步骤1 设置动态变量。

1. 在“接口自动化测试用例列表”页面，单击页面右侧“更多”，选择“环境参数”。
2. 选中动态变量列的复选框，单击“保存”，将全局参数设置为动态参数。



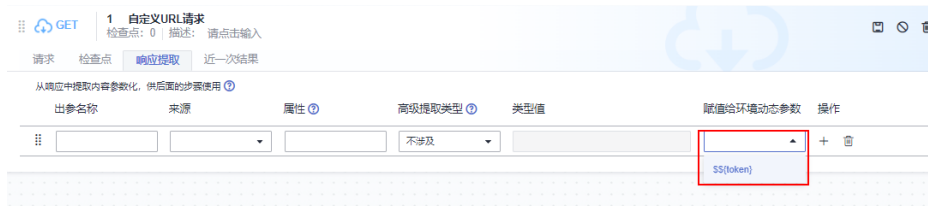


### 步骤2 赋值动态变量。

1. 在接口自动化用例中，选择URL请求的“响应提取”标签下，单击匹配列的“添加”按钮。

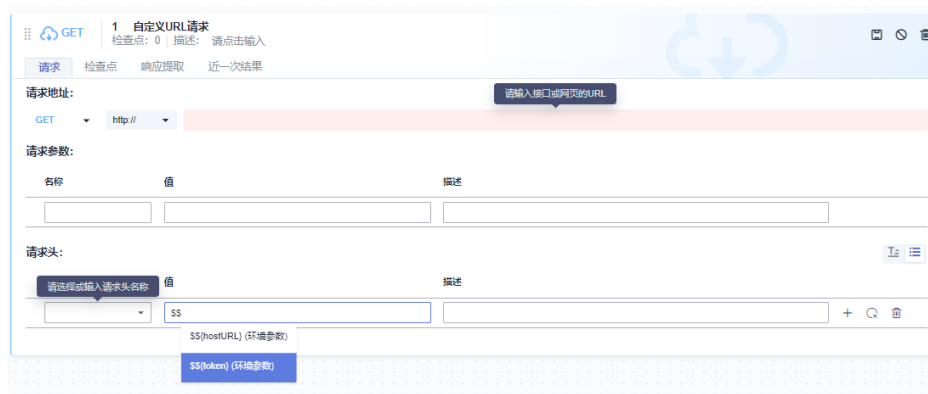


2. 在“赋值给环境动态参数”下拉列表中希望赋值的全局动态参数。



### 步骤3 使用动态变量。

- 在测试用例中使用动态变量。  
在需要使用全局动态变量的用例中，引用动态全局变量，引用方式请参见[环境参数引用形式](#)。



- 在测试套件中使用动态全局变量。  
将[步骤2](#)和[步骤3](#)中的用例按顺序添加到测试套件中，选择“串行执行”，即可让使用动态变量的用例执行时使用到动态全局变量被赋值的最新值。

 说明

并行执行时动态变量的赋值和使用顺序不能保证，因此不推荐在并行执行时使用动态全局变量。

----结束

## 6.4.10 导入 CodeArts TestPlan 接口自动化用例测试数据集

在设计接口自动化用例的脚本中，对于同一个测试逻辑需要读取其他数据，进行多轮参数数据组合，每一次对脚本进行手动配置将非常冗余。

CodeArts TestPlan支持数据驱动能力，支持导入Excel文件读取测试数据，复用到不同测试场景，高效生成并执行多轮接口自动化用例。



 说明

该特性当前“亚太-新加坡”、“拉美-墨西哥”区域支持。

### 前置条件

1. 接口测试用例的脚本中已配置可用的请求地址。
2. 需要将数据集Excel中的字段在接口自动化用例脚本中配置成局部参数，请参考[局部参数](#)。

### 操作步骤

- 步骤1** 单击导航栏“测试 > 测试用例”。
- 步骤2** 在“接口自动化”页签中单击目标用例名称，选择“脚本”页签。
- 步骤3** 单击 图标。
- 步骤4** 在弹框中，单击默认环境下拉栏，选择用例对应的目标环境。可参考[设置CodeArts TestPlan接口脚本的环境参数](#)。
- 步骤5** 单击弹框中“下载示例”，将数据集表格下载到本地。
- 步骤6** 在本地配置好“数据集描述”与“局部参数字段”，用户可根据需要增加多行数据，每一行将是测试用例执行的一轮。
  - 数据集描述：非必填，长度范围为1-128个字节。
  - 表头字段：每个表头字段需要在用例脚本中已创建局部参数，大小写需要和对应的局部参数保持一致，参数值的规范参照[表6-1](#)进行填写。
- 步骤7** 在弹框中单击 图标，选择本地编辑好的Excel文件，单击“上传”。

 说明

1. 最多上传100条数据集，最多上传50个参数字段。
  2. 一次只能上传一个Excel文件，不支持批量上传。
  3. 导入一次Excel后再次导入新的Excel，弹框中将会保留新导入Excel里的数据集数据，上一个导入的Excel数据将会被清除。
- 步骤8** 导入成功的参数数据将显示在列表中。单击“清空当前数据集”可以将导入的参数数据全部删除。

**步骤9** 导入成功后，单击“关闭”。

**步骤10** 在URL请求中，在参数值的值里输入“\$”，可以调用已配置的局部参数，执行测试用例时将读取该局部参数所属的参数值。导入数据集后执行接口自动化用例的操作请参考[接口自动化用例中导入数据集后多轮次执行](#)。



----结束

## 6.4.11 接口自动化用例内置函数说明

### 6.4.11.1 在 CodeArts TestPlan 中调用二元加法运算函数

#### 函数名

`$add(intA, intB)`

#### 功能说明

实现参数A与参数B的加法运算。其中，参数A、B支持以下类型：

- 数字
- 局部参数
- 二元运算

#### 使用场景

接口自动化用例中支持在以下场景中使用二元加法运算函数：

- 请求url路径
- 请求头
- 请求体
- 检查点属性
- if判断
- for循环中断条件

## 示例

- 请求url路径

如下图所示，请求url路径中参数“test”的值为二元加法运算函数，函数中的参数A、B均为“1000”。

请求参数：

名称	值
test	\$add(1000,1000)

- 请求头

如下图所示，请求头中参数“add”的值为二元加法运算函数，函数中的参数A为二元减法运算“\$subtract(1001,1000)”、参数B为“-1”。

请求头：

名称	值
Content-Type	application/json
add	\$add(\$subtract(1001,1000),-1)

- 检查点属性

如下图所示，检查点属性“result”的目标值为二元加法运算函数，函数中的参数A为局部参数“test”，参数B为“1”。局部参数的设置方式请参考[局部参数](#)。

来源	属性	高级提取类型	类型值	比较符	目标值
响应体(JSON)	result	不涉及		等于(数字)	\$add(\${test},1)

- if判断

如下图所示，if判断的目标值为二元加法运算函数，函数中的参数A为数字“1”，参数B为环境变量“status”。环境参数的设置方式请参考[设置CodeArts TestPlan接口脚本的环境参数](#)。

if	\${test}	小于(数字)	\$add(1,\${status})
----	----------	--------	---------------------

- for循环中断条件

如下图所示，for循环中断条件的目标值为二元加法运算函数，函数中的参数A、B均为“1000”。

循环	循环 10 次	中断条件	大于(数字)	\$add(1000,1000)
中断条件	\$\$status	大于(数字)	\$add(1000,1000)	

### 6.4.11.2 在 CodeArts TestPlan 中调用二元减法运算函数

#### 函数名

\$subtract(intA, intB)

## 功能说明

实现参数A与参数B的减法运算。其中，参数A、B支持以下类型：

- 数字
- 局部参数
- 二元运算

## 使用场景

接口自动化用例中支持在以下场景中使用二元减法运算函数：

- 请求url路径
- 请求头
- 请求体
- 检查点属性
- if判断
- for循环中断条件

## 示例

- 请求url路径

如下图所示，请求url路径中参数“test”的值为二元减法运算函数，函数中的参数A为“1001”、参数B为“1000”。

名称	值
test	\$subtract(1001,1000)

- 请求头

如下图所示，请求头中参数“subtract”的值为二元减法运算函数，函数中的参数A为二元乘法运算“\$multiply(100,100)”、参数B为“-1”。

名称	值
Content-Type	application/json
subtract	\$subtract(\$multiply(100,100),-1)

- 请求体

如下图所示，请求体中应用了二元减法运算函数，函数中的参数A二元除法运算“\$divide(1000,100)”，参数B为二元加法运算“\$add(1000,1000)”。

```

1  {
2      "add": "$add($multiply(100,100),$divide(1000,100))",
3      "subtract": "$subtract($divide(1000,100),$add(1000,1000))",
4      "multiply": "$multiply($add(1000,1000),$subtract(1001,1000))",
5      "divide": "$divide($subtract(1001,1000),$multiply(100,100))"
6  }
```

- 检查点属性

如下图所示，检查点属性“result”的目标值为二元减法运算函数，函数中的参数A为局部参数“test”，参数B为“1”。局部参数的设置方式请参考[局部参数](#)。

来源	属性	高级提取类型	类型值	比较符	目标值
响应体(JSON)	result	不涉及		等于(数字)	\$subtract(\${test},1)

- if判断

如下图所示，if判断的目标值为二元减法运算函数，函数中的参数A为数字“1001”，参数B为环境变量“status”。环境参数的设置方式请参考[设置 CodeArts TestPlan 接口脚本的环境参数](#)。

if	`\${test}`	小于(数字)			\$subtract(1001,\${status})
----	------------	--------	--	--	-----------------------------

- for循环中断条件

如下图所示，for循环中断条件的目标值为二元减法运算函数，函数中的参数A为“1001”、参数B为“1000”。

循环	循环 10 次				
中断条件	`\${status}`	大于(数字)			\$subtract(1001,1000)

### 6.4.11.3 在 CodeArts TestPlan 中调用二元乘法运算函数

#### 函数名

\$multiply(intA, intB)

#### 功能说明

实现参数A与参数B的乘法运算。其中，参数A、B支持以下类型：

- 数字
- 局部参数
- 二元运算

#### 使用场景

接口自动化用例中支持在以下场景中使用二元乘法运算函数：

- 请求url路径
- 请求头
- 请求体
- 检查点属性
- if判断
- for循环中断条件

#### 示例

- 请求url路径

如下图所示，请求url路径中参数“test”的值为二元乘法运算函数，函数中的参数A、B均为“100”。

名称	值
test	\$multiply(100,100)

- 请求头

如下图所示，请求头中参数“add”的值为二元乘法运算函数，函数中的参数A为二元除法运算“\$divide(1000,100)”、参数B为“-1”。

名称	值
Content-Type	application/json
multiply	\$multiply(\$divide(1000,100),-1)

- 请求体

如下图所示，请求体中应用了二元乘法运算函数，函数中的参数A为二元加法运算“\$add(1000,1000)”、参数B为二元减法运算“\$subtract(1001,1000)”。

```

1 {
2   "add": "$add($multiply(100,100),$divide(1000,100))",
3   "subtract": "$subtract($divide(1000,100),$add(1000,1000))",
4   "multiply": "$multiply($add(1000,1000),$subtract(1001,1000))",
5   "divide": "$divide($subtract(1001,1000),$multiply(100,100))"
6 }
    
```

- 检查点属性

如下图所示，检查点属性“result”的目标值为二元乘法运算函数，函数中的参数A为局部参数“test”，参数B为“1”。局部参数的设置方式请参考[局部参数](#)。

来源	属性	高级提取类型	类型值	比较符	目标值
响应体(JSON)	result	不涉及		等于(数字)	\$multiply(\${test},1)

- if判断

如下图所示，if判断的目标值为二元乘法运算函数，函数中的参数A为数字“100”，参数B为环境变量“status”。环境参数的设置方式请参考[设置 CodeArts TestPlan 接口脚本的环境参数](#)。

if	\$(test)	小于(数字)	\$multiply(100,\${status})
----	----------	--------	----------------------------

- for循环中断条件

如下图所示，for循环中断条件的目标值为二元乘法运算函数，函数中的参数A、B均为“100”。

循环	循环 10 次		
中断条件	\$\$status	大于(数字)	\$multiply(100,100)

### 6.4.11.4 在 CodeArts TestPlan 中调用二元除法运算函数

#### 函数名

\$divide(intA, intB, intC)

## 功能说明

实现参数A与参数B的除法运算，C为精度值。其中，参数A、B、C支持以下类型：

- 数字
- 局部参数
- 二元运算

-不带精度值除法运算，能除尽，则为除尽后的保留位数，不能除尽，默认保留6位小数，四舍五入规则。

-带精度的除法运算，精度值范围为大于1小于6包括边界值的整数。若能除尽，除尽后小数位数不超过精度值则按照原有位数显示，超过精度值则按精度值保留。除不尽则按给定精度值保留，四舍五入规则。

## 使用场景

接口自动化用例中支持在以下场景中使用二元除法运算函数：

- 请求url路径
- 请求头
- 请求体
- 检查点属性
- if判断
- for循环中断断条件

## 示例

- 请求url路径

如下图所示，请求url路径中参数“test”的值为二元除法运算函数，函数中的参数A为“1000”、参数B为“100”。

名称	值
test	<code>\$divide(1000,100)</code>

- 如下图所示，请求url路径中参数“test”的值为二元除法带精度值的运算函数，函数中的参数A为“1”、参数B为“3”，精度值为5。

名称	值
test	<code>\$divide(1,3,5)</code>

- 请求头

如下图所示，请求头中参数“divide”的值为二元除法运算函数，函数中的参数A为二元加法运算“`$add(1000,1000)`”、参数B为“-1”。

名称	值
Content-Type	application/json
divide	<code>\$divide(\$add(1000,1000),-1)</code>



如下图所示，请求头中参数“divide”的值为二元除法带精度值运算函数，函数中的参数A为“1”、参数B为“-3”、参数C精度值为全局环境参数 $\${scale}$ 。

- 请求体

如下图所示，请求体中应用了二元除法运算函数，函数中的参数A为二元减法运算“ $\$subtract(1001,1000)$ ”、参数B为二元乘法运算“ $\$multiply(100,100)$ ”。

```

1  {
2  |   "add": "$add($multiply(100,100),$divide(1000,100))",
3  |   "subtract": "$subtract($divide(1000,100),$add(1000,1000))",
4  |   "multiply": "$multiply($add(1000,1000),$subtract(1001,1000))",
5  |   "divide": "$divide($subtract(1001,1000),$multiply(100,100))"
6  }
    
```

如下图所示，请求体中应用了二元除法带精度值运算函数，函数中的参数A为“1”、参数B为“3”、参数C精度值为全局环境参数 $\${scale}$ 。

```

1  {
2  |   "divide": "$divide(1,3,\${scale})"
3  }
4
    
```

- 检查点属性

如下图所示，检查点属性“result”的目标值为二元除法运算函数，函数中的参数A为局部参数“test”，参数B为“1”。局部参数的设置方式请参考[局部参数](#)。

如下图所示，检查点属性“result”的目标值为二元除法带精度运算函数，函数中的参数A为局部参数“test”，参数B为“2”，参数C为“5”。局部参数的设置方式请参考[局部参数](#)。

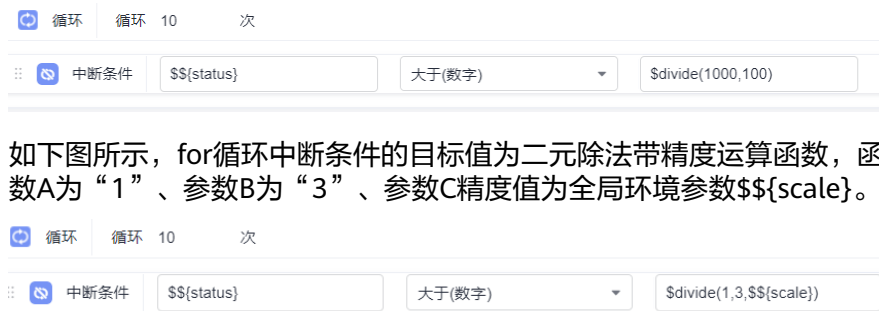
- if判断

如下图所示，if判断的目标值为二元除法运算函数，函数中的参数A为数字“1”，参数B为环境变量“status”。参数C精度值为局部参数“localScale”。局部参数的设置方式请参考[局部参数](#)。

如下图所示，if判断的目标值为二元除法带精度运算函数，函数中的参数A为数字“1”，参数B为“3”。环境参数的设置方式请参考[设置CodeArts TestPlan接口脚本的环境参数](#)。

- for循环中断条件

如下图所示，for循环中断条件的目标值为二元除法运算函数，函数中的参数A为“1000”、参数B为“100”。



如下图所示，for循环中断条件的目标值为二元除法带精度运算函数，函数中的参数A为“1”、参数B为“3”、参数C精度值为全局环境参数 $scale$ 。

### 6.4.11.5 在 CodeArts TestPlan 中调用获取当前时间戳函数

#### 函数名

\$timestamp()

#### 功能说明

获取从格林威治时间1970年01月01日00时00分00秒起到当前时间的总毫秒数。

#### 使用场景

接口自动化用例中支持在以下使用场景使用获取当前时间戳函数：

- 请求url路径
- 请求头
- 请求体
- 检查点属性
- if判断
- for循环中断条件

#### 示例

- 请求url路径  
如下图所示，请求url路径中参数“test”的值为生成当前时间戳函数。

名称	值
test	\$timestamp()

- 请求头  
如下图所示，请求头中参数“time”的值为生成当前时间戳函数。

名称	值
Content-Type	application/json
time	\$timestamp()

- 请求体  
如下图所示，请求体中应用了生成当前时间戳函数。

```

1  {
2  |   "time": "$timestamp()"
3  | }
4

```

- 检查点属性  
如下图所示，检查点属性“result”的目标值为生成当前时间戳函数。

来源	属性	高级提取类型	类型值	比较符	目标值
响应体(JSON)	result	不涉及		等于(数字)	\$timestamp()

- if判断  
如下图所示，if判断的目标值为生成当前时间戳函数。

if	`\${test}`	小于(数字)	\$timestamp()
----	------------	--------	---------------

- for循环中断条件  
如下图所示，for循环中断条件的目标值为生成当前时间戳函数。

循环	循环	次
中断条件	`\${status}`	大于(数字)

### 6.4.11.6 在 CodeArts TestPlan 中调用获取指定时间戳函数

#### 函数名

\$getTimeBeforeHour(doubleA)

#### 功能说明

获取当前时间之前A小时的时间戳。时间戳是指从格林威治时间1970年01月01日00时00分00秒起到指定时间的总毫秒数。

函数中的参数A支持以下类型：

- 数字
- 局部参数
- 其它内置函数

#### 使用场景

接口自动化用例中支持在以下使用场景使用获取指定时间戳函数：

- 请求url路径
- 请求头

- 请求体
- 检查点属性
- if判断
- for循环中断条件

## 示例

- 请求url路径

如下图所示，请求url路径中参数“test”的值为获取指定时间戳函数，函数中的参数A为数字“3”。

名称	值
test	\$getTimeBeforeHour(3)

- 请求头

如下图所示，请求头中参数“time”的值为获取指定时间戳函数，函数中的参数A为数字“3”。

名称	值
Content-Type	application/json
time	\$getTimeBeforeHour(3)

- 请求体

如下图所示，请求体中应用了获取指定时间戳函数，函数中的参数A为二元加法运算“\$add(2,2)”。

```

1  {
2  |   "time": "$getTimeBeforeHour($add(2,2))"
3  | }
4

```

- 检查点属性

如下图所示，检查点属性“result”的目标值为获取指定时间戳函数，函数中的参数A为局部参数“test”。局部参数的设置方式请参考[局部参数](#)。

来源	属性	高级提取器	数据类型	比较符	目标值
响应体(JSON)	result	不涉及		等于(数字)	\$getTimeBeforeHour(\$test)

- if判断

如下图所示，if判断的目标值为获取指定时间戳函数，函数中的参数A为环境变量“time”。环境参数的设置方式请参考[设置CodeArts TestPlan接口脚本的环境参数](#)。

if	if	小于(数字)	\$getTimeBeforeHour(\$\${state
----	----	--------	--------------------------------

- for循环中断条件

如下图所示，for循环中断条件的目标值为获取指定时间戳函数，函数中的参数A为数字“3”。



### 6.4.11.7 在 CodeArts TestPlan 中调用日期转时间戳函数

#### 函数名

\$dateFormat(String A)

#### 参数说明

String A: 日期时间值, 支持以下几种格式:

- yyyy-MM-dd HH:mm:ss或MM-dd-yyyy HH:mm:ss格式。
- yyyy MM dd HH:mm:ss或MM dd yyyy HH:mm:ss格式。
- yyyy.MM.dd HH:mm:ss或MM.dd.yyyy HH:mm:ss格式。
- yyyy/MM/dd HH:mm:ss或MM/dd/yyyy HH:mm:ss格式。

#### 功能说明

实现将字符串转化成对应的时间戳。时间戳是指从格林威治时间1970年01月01日00时00分00秒起到指定时间的总毫秒数。

函数中的参数A支持以下几种类型:

- 符合[参数说明](#)格式的日期时间值
- 局部参数
- 其它内置函数

#### 使用场景

接口自动化用例中支持在以下场景中使用日期转时间戳函数:

- 请求url路径
- 请求头
- 请求体
- 检查点属性
- if判断
- for循环中断条件

#### 示例

- 请求url路径  
如下图所示, 请求url路径中参数“test”的值为日期转时间戳函数, 函数中的参数A为环境参数“date”。环境参数的设置方式请参考[设置CodeArts TestPlan接口脚本的环境参数](#)。

名称	值
test	\$dateFormat(\$\$(date))

- 请求头

如下图所示，请求头中参数“date”的值为日期转时间戳函数，函数中的参数A为环境参数“date”。

名称	值
Content-Type	application/json
date	\$dateFormat(\$\$(date))

- 请求体

如下图所示，请求体中应用了日期转时间戳函数，函数中的参数A为“2020.09.11 11:00:00”。

```

1  {
2  |   "date": "$dateFormat(2020.09.11 11:00:00)"
3  | }
4

```

- 检查点属性

如下图所示，检查点属性“result”的目标值为日期转时间戳函数，函数中的参数A为环境参数“test”。局部参数的设置方式请参考[局部参数](#)。

来源	属性	高级提取类型	类型值	比较符	目标值
响应体(JSON)	result	不涉及		等于(数字)	\$dateFormat(\$\$(date))

- if判断

如下图所示，if判断的目标值为日期转时间戳函数，函数中的参数A为“2020-09-11 11:00:00”。

if	\$(test)	小于(数字)	\$dateFormat(2020-09-11 11:00:00)
----	----------	--------	-----------------------------------

- for循环中断条件

如下图所示，for循环中断条件的目标值为日期转时间戳函数，函数中的参数A为“2020-09-11 11:00:00”。

循环	循环 10 次		
中断条件	\$\$\$(status)	大于(数字)	\$dateFormat(2020-09-11 11:00:00)

### 6.4.11.8 在 CodeArts TestPlan 中调用时间戳转日期函数

#### 函数名

\$timestamp\_format(String A, String B)

## 参数说明

- String A: 要转化的时间戳，20位以内的纯数字串，也可使用内置获取当前时间戳函数\$timestamp()。
- String B: 日期时间值，由年、月、日、时、分、秒、毫秒组成。其中：
  - 年：使用字母“y”表示，共4位字符。
  - 月：使用字母“M”表示，共1~2位字符。
  - 日：使用字母“d”表示，共1~2位字符。
  - 时：使用字母“H”表示，共0~2位字符。
  - 分：使用字母“m”表示，共0~2位字符。
  - 秒：使用字母“s”表示，共0~2位字符。
  - 毫秒：使用字母“S”表示，共3位字符。
- 表 6-4 字母参数类型

字母	日期或时间元素	参数类型	示例
G	Era 标志符	文本	AD
y	年	年份	1996; 96
M	年中的月份	月份	July; Jul; 07
w	年中的周数	数字	27
W	月份中的周数	数字	2
D	年中的第几天	数字	189
d	月份中的第几天	数字	10
F	月份中的第几个星期	数字	2
E	星期中的某天	文本	Tuesday; Tue
a	Am/pm 标记	文本	PM
H	一天中的小时数 (0-23)	数字	0
k	一天中的小时数 (1-24)	数字	24
K	am(中午)/pm (下午) 中的小时数 (0-11)	数字	0
h	am(中午)/pm (下午) 中的小时数 (1-12)	数字	12
m	小时中的分钟数	数字	30
s	分钟中的秒数	数字	55

字母	日期或时间元素	参数类型	示例
S	毫秒数	数字	978
z	时区缩写名称	文本	PST;EST
Z	时区偏移量	文本	+800; -0530

### 📖 说明

1. 各字母与其代表的含义一一对应，请注意大小写格式。
2. H、m、s中若有一个为0位时，其它两个也必须为0位。

日期时间值中，连接符允许使用空格，“-”、“/”、“:”，也可以不使用连接符，且不能含转义连接符，比如“\n”。例如如下常用格式（包括但不限于）：

- yyyy-MM-dd HH:mm:ss
- yyyyMMddHHmmss
- yyyyMMddHHmmssSSS
- yyyy-M-d H:m:s
- MM-dd-yyyy HH:mm:ss
- MM/dd/yyyy HH/mm/ss
- MM/d/yyyy H/mm/ss
- MM/d/yyyy H/mm/ss SSS
- yyyyMMdd SSS
- yyyy-MM-dd HH:mm:ss SSS
- yyyy-MM-dd HH:mm:ss
- yyyy-MM-dd HH:mm
- yyyy-MM-dd HH
- yyyy-MM-dd
- yyyy-MM
- yyyy
- yy
- MM-dd HH
- MM-dd
- MM
- dd
- HH:mm:ss SSS
- HH:mm:ss
- HH:mm
- HH
- mm
- mm:ss
- ss



- SSS

## 功能说明

实现将时间戳转化成对应格式的日期。时间戳是指从格林威治时间1970年01月01日00时00分00秒起到指定时间的总毫秒数。

函数中的参数A支持以下几种类型：

- 符合[参数说明](#)格式的日期时间值
- 环境参数
- 局部参数
- 其它内置函数

函数中的参数B支持以下几种类型：

- 符合[参数说明](#)格式的日期时间值
- 环境参数
- 局部参数
- 其它内置函数

## 使用场景

接口自动化用例中支持在以下场景中使用时间戳转日期函数：

- 请求url路径
- 请求头
- 请求体
- 检查点属性
- if判断
- for循环中断条件

## 示例

- 请求url路径

如下图所示，请求url路径中参数“date”的值为时间戳转日期函数，函数中的参数A为内置获取当前时间戳函数“\$timestamp()”，获取当前时间戳内置函数的使用方式请参考[在CodeArts TestPlan中调用获取当前时间戳函数](#)，参数B为“yyyy-MM-dd HH:mm:ss”。

名称	值
date	\$timestamp_format(\$timestamp(),yyyy-MM-dd HH:mm:ss)

- 请求头

如下图所示，请求头中参数“date”的值为时间戳转日期函数，函数中的参数A为环境参数“date”，参数B为“yyyyMMddHHmmss”。环境参数的设置方式请参考[设置CodeArts TestPlan接口脚本的环境参数](#)。

名称	值
date	\$timestamp_format(\$\${date}, yyyyMMddHHmmss)

- 请求体

如下图所示，请求体中应用了时间戳转日期函数，函数中的参数A为“123456789”，参数B为“yyyyMMddHHmmssSSS”。

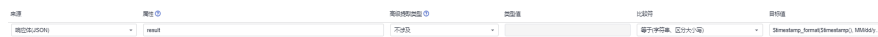
```

1  {
2  |   "date": "$timestamp_format(123456789, yyyyMMddHHmmssSSS)"
3  | }
4

```

- 检查点属性

如下图所示，检查点属性“result”的目标值为时间戳转日期函数，函数中的参数A为内置获取当前时间戳函数“\$timestamp()”，参数B为“MM/dd/yyyy HH/mm/ss”。



- if判断

如下图所示，if判断的目标值为时间戳转日期函数，函数中的参数A为内置获取当前时间戳函数“\$timestamp()”，参数B为“MM/d/yyyy H/mm/ss SSS”。



- for循环中断条件

如下图所示，for循环中断条件的目标值为时间戳转日期函数，函数中的参数A为内置获取当前时间戳函数“\$timestamp()”，参数B为“yyyyMMdd SSS”。



### 6.4.11.9 在 CodeArts TestPlan 中调用时间戳加减运算函数

#### 函数名

\$timeStampCalculation(longA, StringB)

#### 参数说明

- longA: 毫秒级时间戳。
- StringB: 时间差, integer范围内的数字加字母(天d/小时h/秒s), 例如1d代表对指定时间戳加一天, -1d代表对指定时间戳减一天。

#### 功能说明

实现long型参数A与字符串参数B的加减运算, 其中参数A、B支持以下类型:

- 符合参数说明格式的值
- 局部参数

- 其它内置函数

## 使用场景

接口自动化用例中支持在以下场景中使用时间戳加减运算：

- 请求url路径
- 请求头
- 请求体
- 检查点属性
- if判断
- for循环中断条件

## 示例

- 请求url路径

如下图所示，请求url路径中参数“test”的值为时间戳加减运算函数，函数中的参数A为“1607939485441”、参数B为“1d”。

名称	值
test	\$timeStampCalculation(1607939485441,1d)

- 请求头

如下图所示，请求头中参数“time”的值为时间戳加减运算函数，函数中的参数A为日期转时间戳函数“\$dateFormat(2020.09.11 11:00:00)”、参数B为“-86400s”。

名称	值
Content-Type	application/json
time	\$timeStampCalculation(\$dateFormat(2020.09.11 11:00:00),-86400s)

- 请求体

如下图所示，请求体中应用了时间戳加减运算函数，函数中的参数A为日期转时间戳函数“\$dateFormat(2020.09.11 11:00:00)”、参数B为“1d”。

```

1  {
2  |   "time": "$timeStampCalculation($dateFormat(2020.09.11 11:00:00),1d)"
3  | }
4

```

- 检查点属性

如下图所示，检查点属性“result”的目标值为时间戳加减运算函数，函数中的参数A为环境参数“time”、参数B为“-24h”。环境参数的设置方式请参考[设置CodeArts TestPlan接口脚本的环境参数](#)。

来源	属性	高级提取类型	类型	比较符	目标值
响应体(JSON)	result	不涉及		等于(数字)	tpCalculation(\$[time],-24h)

- if判断

如下图所示，if判断的目标值为时间戳加减运算函数，函数中的参数A为环境变量“status”、参数B为数字“1d”。环境参数的设置方式请参考[设置CodeArts TestPlan接口脚本的环境参数](#)。



- for循环中断条件

如下图所示，for循环中断条件的目标值为时间戳加减运算函数，函数中的参数A为“1607939485441”、参数B为“1d”。



### 6.4.11.10 在 CodeArts TestPlan 中调用生成 Base64 编码函数

#### 函数名

`$encode_base64(StringA)`

#### 参数说明

StringA：待编码的字符串。长度不大于256个字节，可包含特殊符号!\*()::@&=+\$//?#[~!@%<>\_]{}`^。

#### 功能说明

实现对字符串进行Base64编码。其中，参数A支持以下几种类型：

- 符合参数说明格式的值
- 环境参数
- 局部参数
- 其它内置函数

#### 使用场景

接口自动化用例中支持在以下场景中使用生成Base64编码函数：

- 请求url路径
- 请求头
- 请求体
- 检查点属性
- if判断
- for循环中断条件

#### 示例

- 请求url路径

如下图所示，请求url路径中参数“test”的值为生成Base64编码函数，函数中的参数A为字符串“abc123”。



- 请求头

如下图所示，请求头中参数“Accept-Encoding”的值为生成Base64编码函数，函数中的参数A为字符串“abc123”。

名称	值
Content-Type	application/json
Accept-Encoding	\$encode_base64(abc123)

- 请求体

如下图所示，请求体中应用了生成Base64编码函数，函数中的参数A为生成UUID函数“\$uuid()”。

```

1  {
2  |   "encoding": "$encode_base64($uuid())"
3  | }
4

```

- 检查点

如下图所示，检查点属性“result”的目标值为生成Base64编码函数，函数中的参数A为局部参数“test”。局部参数的设置方式请参考[局部参数](#)。

来源	属性	高级匹配模式	类型	比较符	目标值
响应体(JSON)	result	不涉及		包含(字符串)	\$encode_base64(\$test)

- if判断

如下图所示，if判断的目标值为生成Base64编码函数，函数中的参数A为环境变量“status”。环境参数的设置方式请参考[设置CodeArts TestPlan接口脚本的环境参数](#)。

if	`\${test}`	包含(字符串)	\$encode_base64(`\${status}`)
----	------------	---------	-------------------------------

- for循环中断条件

如下图所示，for循环中断条件的目标值为生成Base64编码函数，函数中的参数A为字符串“abc123”。

循环	循环 10 次		
中断条件	`\${status}`	不包含(字符串)	\$encode_base64(abc123)

### 6.4.11.11 在 CodeArts TestPlan 中调用生成 SHA512 编码函数

#### 函数名

\$sha512(StringA)

#### 参数说明

StringA: 待编码的字符串。长度不大于256个字节，可包含特殊符号!\*(());@&=+\$//?#[\].~%<>\_{}^。

#### 功能说明

实现对字符串进行SHA512编码。其中，参数A支持以下几种类型：

- 符合[参数说明](#)格式的值
- 环境参数
- 局部参数
- 其它内置函数

## 使用场景

接口自动化用例中支持在以下场景使用生成SHA512编码函数：

- 请求url路径
- 请求头
- 请求体
- 检查点属性
- if判断
- for循环中断条件

## 示例

- 请求url路径

如下图所示，请求url路径中参数“test”的值为生成SHA512编码函数，函数中的参数A为字符串“abc123”。

名称	值
test	\$sha512(abc123)

- 请求头

如下图所示，请求头中参数“Accept-Encoding”的值为生成SHA512编码函数，函数中的参数A为字符串“abc123”。

名称	值
Content-Type	application/json
Accept-Encoding	\$sha512(abc123)

- 请求体

如下图所示，请求体中应用了生成SHA512编码函数，函数中的参数A为生成UUID函数“\$uuid()”。

```

1  {
2  |   "encoding": "$sha512($uuid())"
3  | }
4
5

```

- 检查点属性

如下图所示，检查点属性“result”的目标值为生成SHA512编码函数，函数中的参数A为局部参数“test”。局部参数的设置方式请参考[局部参数](#)。

来源	属性	期望结果类型	类型	比较符	目标值
响应(JSON)	result	不涉及		包含(字符串)	\$sha512(test)

- if判断

如下图所示，if判断的目标值为生成SHA512编码函数，函数中的参数A为环境变量“status”。环境参数的设置方式请参考[设置CodeArts TestPlan接口脚本的环境参数](#)。



- for循环中断条件

如下图所示，for循环中断条件的目标值为生成SHA512编码函数，函数中的参数A为字符串“abc123”。



### 6.4.11.12 在 CodeArts TestPlan 中调用生成 SHA256 编码函数

#### 函数名

\$sha256(StringA)

#### 参数说明

StringA: 待编码的字符串。长度不大于256个字节，可包含特殊符号!'()::@&=+\$/?#[]-.~%<>\_{}^`。

#### 功能说明

实现对字符串进行SHA256编码。其中，参数A支持以下几种类型：

- 符合[参数说明](#)格式的值
- 环境参数
- 局部参数
- 其它内置函数

#### 使用场景

接口自动化用例中支持在以下场景使用生成SHA256编码函数：

- 请求url路径
- 请求头
- 请求体
- 检查点属性
- if判断
- for循环中断条件

#### 示例

- 请求url路径

如下图所示，请求url路径中参数“test”的值为生成SHA256编码函数，函数中的参数A为字符串“abc123”。

名称	值
test	\$sha256(abc123)

- 请求头

如下图所示，请求头中参数“Accept-Encoding”的值为生成SHA256编码函数，函数中的参数A为字符串“abc123”。

名称	值
Content-Type	application/json
Accept-Encoding	\$sha256(abc123)

- 请求体

如下图所示，请求体中应用了生成SHA256编码函数，函数中的参数A为生成UUID函数“\$uuid()”。

```

1  {
2  |   "encoding": "$sha256($uuid())"
3  | }
4

```

- 检查点属性

如下图所示，检查点属性“result”的目标值为生成SHA256编码函数，函数中的参数A为局部参数“test”。局部参数的设置方式请参考[局部参数](#)。

来源	属性	高级提取类型	类型值	比较符	目标值
响应体(JSON)	result	不涉及		包含(字符串)	\$sha256(\${test})

- if判断

如下图所示，if判断的目标值为生成SHA256编码函数，函数中的参数A为环境变量“status”。环境参数的设置方式请参考[设置CodeArts TestPlan接口脚本的环境参数](#)。

if	`\${test}`	包含(字符串)	\$sha256(status)
----	------------	---------	------------------

- for循环中断条件

如下图所示，for循环中断条件的目标值为生成SHA256编码函数，函数中的参数A为字符串“abc123”。

循环	循环 10 次		
中断条件	`\${status}`	不包含(字符串)	\$sha256(abc123)

### 6.4.11.13 在 CodeArts TestPlan 中调用生成 MD5 哈希值函数

#### 函数名

\$md5(StringA)



## 参数说明

StringA: 待编码的字符串。长度不大于256个字节，可包含特殊符号!\*();:@&=+\$//?#[\].~%<>\_[]{}^。

## 功能说明

实现将字符串转化成MD5哈希值。其中，参数A支持以下几种类型：

- 符合参数说明格式的值
- 环境参数
- 局部参数
- 其它内置函数

## 使用场景

接口自动化用例中支持在以下场景使用MD5哈希值函数：

- 请求url路径
- 请求头
- 请求体
- 检查点属性
- if判断
- for循环中断条件

## 示例

- 请求url路径

如下图所示，请求url路径中参数“test”的值为生成MD5哈希值函数，函数中的参数A为字符串“abc123”。

名称	值
test	\$md5(abc123)

- 请求头

如下图所示，请求头中参数“Accept-Encoding”的值为生成MD5哈希值函数，函数中的参数A为字符串“abc123”。

名称	值
Content-Type	application/json
Accept-Encoding	\$md5(abc123)

- 请求体

如下图所示，请求体中应用了生成MD5哈希值函数，函数中的参数A为生成UUID函数“\$uuid()”。

```

1  {
2      "encoding": "$md5($uuid())"
3  }
4  
```

- 检查点属性

如下图所示，检查点属性的“result”的目标值为生成MD5哈希值函数，函数中的参数A为局部参数“test”。局部参数的设置方式请参考[局部参数](#)。



- if判断

如下图所示，if判断的目标值为生成MD5哈希值函数，函数中的参数A为环境变量“status”。环境参数的设置方式请参考[设置CodeArts TestPlan接口脚本的环境参数](#)。



- for循环中断条件

如下图所示，for循环中断条件的目标值为生成MD5哈希值函数，函数中的参数A为字符串“abc123”。



#### 6.4.11.14 在 CodeArts TestPlan 中调用生成指定范围随机数函数

##### 函数名

\$random\_int(intA, intB)

##### 功能说明

实现在参数A与参数B之间的范围内生成随机数。测试计划服务支持生成10位以内的随机数，即最小值-9999999999、最大值为9999999999。

参数A、B以下几种类型：

- 数字
- 环境参数
- 局部参数
- 其它内置函数

##### 使用场景

接口自动化用例中支持在以下场景使用生成指定范围随机数函数：

- 请求url路径
- 请求头
- 请求体
- 检查点属性
- if判断
- for循环中断条件

## 示例

- 请求url路径

如下图所示，请求url路径中参数“test”的值为生成指定范围随机数函数，函数中的参数A为数字“1”、参数B为数字“100”。

名称	值
test	\$random_int(1,100)

- 请求头

如下图所示，请求头中参数“number”的值为生成指定范围随机数函数，函数中的参数A为数字“1”、参数B为数字“100”。

名称	值
Content-Type	application/json
number	\$random_int(1,100)

- 请求体

如下图所示，请求体中应用了生成指定范围随机数函数，函数中的参数A为二元加法运算“\$add(5,5)”、参数B为二元乘法运算“\$multiply(5,5)”。

```

1  {
2  |   "number": "$random_int($add(5,5), $multiply(5,5))"
3  | }
4

```

- 检查点属性

如下图所示，检查点属性“result”的目标值为生成指定范围随机数函数，函数中的参数A为数字“1”，参数B为局部参数“test”。局部参数的设置方式请参考[局部参数](#)。

来源	属性	高级搜索表达式	数据类型	比较符	目标值
响应体(JSON)	result	不等于	数字	等于(数字)	\$random_int(1,\$test)

- if判断

如下图所示，if判断的目标值为生成指定范围随机数函数，函数中的参数A数字“1”，参数B为环境变量“status”。环境参数的设置方式请参考[设置CodeArts TestPlan接口脚本的环境参数](#)。

if	\$(test)	等于(数字)	\$random_int(1,\$(status))
----	----------	--------	----------------------------

- for循环中断条件

如下图所示，for循环中断条件的目标值为生成指定范围随机数函数，函数中的参数A为数字“1”、参数B为数字“100”。

循环	循环 10 次
中断条件	\$(status) 不等于(数字) \$random_int(1,100)

### 6.4.11.15 在 CodeArts TestPlan 中调用生成指定长度随机字符串函数

#### 函数名

\$random\_string(intA)

## 功能说明

实现生成指定长度的随机字符串。其中，参数A支持以下几种类型：

- 数字
- 环境参数
- 局部参数
- 其它内置函数

## 使用场景

接口自动化用例中支持在以下场景使用生成指定长度随机字符串函数：

- 请求url路径
- 请求头
- 请求体
- 检查点属性
- if判断
- for循环中断条件

## 示例

- 请求url路径

如下图所示，请求url路径中参数“test”的值为生成指定长度随机字符串函数，函数中的参数A为数字“3”。

名称	值
test	\$random_string(3)

- 请求头

如下图所示，请求头中参数“string”的值为生成指定长度随机字符串函数，函数中的参数A为数字“3”。

名称	值
Content-Type	application/json
string	\$random_string(3)

- 请求体

如下图所示，请求体中应用了生成指定长度随机字符串函数，函数中的参数A为二元加法运算“\$add(2,2)”。

```

1  {
2    "string": "$random_string($add(2,2))"
3  }
4

```

- 检查点属性

如下图所示，检查点属性“result”的目标值为生成指定长度随机字符串函数，函数中的参数A为局部参数“test”。局部参数的设置方式请参考[局部参数](#)。

来源	属性	高级校验规则	类型	比较符	目标值
响应体(JSON)	result	不等于		数值(字符串)	\$random_string(5test)

- if判断

如下图所示，if判断的目标值为生成指定长度随机字符串函数，函数中的参数A为环境变量“status”。环境参数的设置方式请参考[设置CodeArts TestPlan接口脚本的环境参数](#)。



- for循环中断条件

如下图所示，for循环中断条件的目标值为生成指定长度随机字符串函数，函数中的参数A为数字“3”。



### 6.4.11.16 在 CodeArts TestPlan 中调用指定范围随机生成小数函数

#### 函数名

\$randomDecimal(double A, double B, int C)

#### 功能说明

实现在参数A与参数B之间的范围内生成随机C位小数。测试计划服务支持生成9位随机小数，即最小值-999999999.999999999、最大值为999999999.999999999。

参数A、B以下几种类型：

- 小数或整数
- 环境参数
- 局部参数
- 其它内置函数

参数C支持以下几种类型：

- 正整数
- 环境参数
- 局部参数
- 其它内置函数

若C为0时，函数会生成随机整数。

#### 使用场景

接口自动化用例中支持在以下使用场景使用指定范围随机生成小数函数：

- 请求url路径
- 请求头
- 请求体

- 检查点属性
- if判断
- for循环中断条件

## 示例

- 请求url路径

如下图所示，请求url路径中参数“test”的值为生成指定范围随机生成小数函数，函数中的参数A为数字“1”、参数B为数字“100”、参数C为数字“2”。

请求参数：

名称	值
test	\$randomDecimal(1, 100, 2)

- 请求头

如下图所示，请求头中参数“number”的值为生成指定范围随机小数函数，函数中的参数A为数字“1”、参数B为数字“100”，参数C为数字“2”。

请求头：

名称	值
Content-Type	application/json
number	\$randomDecimal(1, 100, 2)

- 请求体

如下图所示，请求体中应用了生成指定范围随机小数函数，函数中的参数A为二元加法运算“\$add(1,1)”、参数B为二元乘法运算“\$multiply(10,10)”、参数C为数字“2”。

```

1  {
2  |   "number": "$randomDecimal($add(1,0), $multiply(10,10), int 2)"
3  | }
4  
```

- 检查点属性

如下图所示，检查点属性“result”的目标值为生成指定范围随机小数函数，函数中的参数A为数字“1”，参数B为局部参数“test”。局部参数的设置方式请参考[局部参数](#)、参数C为数字“2”。

来源	属性	高级提取类型	类型值	比较符	目标值
响应体(JSON)	result	不涉及		等于(数字)	\$randomDecimal(1, \$(test), 2)

- if判断

如下图所示，if判断的目标值为生成指定范围随机小数函数，函数中的参数A数字“1”，参数B为环境变量“status”。环境参数的设置方式请参考[设置CodeArts TestPlan接口脚本的环境参数](#)，参数C为数字“2”。



- for循环中断条件

如下图所示，for循环中断条件的目标值为生成指定范围随机小数函数，函数中的参数A为数字“1”、参数B为数字“100”、参数C为数字“2”。



### 6.4.11.17 在 CodeArts TestPlan 中调用生成 UUID 函数

#### 函数名

\$uuid()

#### 功能说明

生成一串随机字符串。

#### 使用场景

接口自动化用例中支持在以下使用场景使用生成UUID函数：

- 请求url路径
- 请求头
- 请求体
- 检查点属性
- if判断
- for循环中断条件

#### 示例

- 请求url路径

如下图所示，请求url路径中参数“test”的值为生成UUID函数。



- 请求头

如下图所示，请求头中参数“time”的值为生成UUID函数。

名称	值
Content-Type	application/json
string	\$uuid()

- 请求体

如下图所示，请求体中应用了生成UUID函数。

```

1  {
2  |   "string": "$uuid()"
3  | }
4
5

```

- 检查点属性

如下图所示，检查点属性“result”的目标值为生成UUID函数。

来源	属性	高级提取类型	类型	比较符	目标值
响应体(JSON)	result	不涉及		包含(字符串)	\$uuid()

- if判断

如下图所示，if判断的目标值为生成UUID函数。

if	`\${test}`	包含(字符串)	\$uuid()
----	------------	---------	----------

- for循环中断条件

如下图所示，for循环中断条件的目标值为生成UUID函数。

循环	循环 10 次		
中断条件	`\${status}`	不包含(字符串)	\$uuid()

## 6.4.11.18 在 CodeArts TestPlan 中调用倒序索引提取数组函数

### 函数名

\$getReverseItem(StringA, intB)

### 参数说明

- StringA: 响应体或者响应头的数组/列表元素路径。
- intB: 倒序的数组下标，0代表倒数一组数据，2代表倒数第二组数据。

### 功能说明

根据倒序索引提取响应体或者响应头中的数组。

### 使用场景

接口自动化用例中支持在响应提取中使用内置函数倒序索引提取数组。



## 示例

如下图所示，响应体参数“item”的属性值为内置函数倒序索引提取数组，函数中的参数A为响应体属性“result”、参数B为“0”。

出参名称	来源	属性
item	响应体(JSON)	\$getReverseltem(result,0)

### 6.4.11.19 在 CodeArts TestPlan 中调用倒序索引提取数组元素的值函数

#### 函数名

`$getReverseltem(StringA, StringB, intC)`

#### 参数说明

- StringA: 响应体或者响应头的数组/列表元素路径。
- StringB: 数组对象中的属性名。
- intC: 倒序的数组下标，例如0代表倒数第一组数据，2代表倒数第二组数据。

#### 功能说明

根据倒序索引提取响应体或者响应头中的数组，获取指定元素的值。

#### 使用场景

接口自动化用例中支持在响应提取中使用内置函数倒序索引提取数组元素的值。

## 示例

如下图所示，响应参数“name”的属性值为内置函数倒序索引提取数组元素的值，函数中的参数A为响应体属性“result”、参数B为result中倒数第N+1数据中的参数名“name”、参数C为“0”。

出参名称	来源	属性
name	响应体(JSON)	\$getReverseltem(result,name,0)

### 6.4.11.20 在 CodeArts TestPlan 中调用大写转小写函数

#### 函数名

`$toLowerCase(String A)`

#### 参数说明

- String A: 需要转为小写的字符串。

参数A支持以下类型：

- 字符串
- 局部参数

## 功能说明

将指定输入的字符串中的字符全部转换成小写字符。

## 指定场景

接口自动化用例中支持在以下场景中使用大写转小写函数：

- 请求url路径
- 请求头
- 请求体
- 检查点属性
- if判断
- for循环中断条件

## 实例

- 请求url路径

如下图所示，请求url路径中参数“test”的值为大写转小写运算函数，函数中的参数A为“TEST”。

名称	值
test	\$toLower(TEST)

- 请求头

如下图所示，请求头中参数“lower”的值为大写转小写运算函数，函数中的参数A为“Test”。

名称	值
Content-Type	application/json
lower	\$toLower(Test)

- 请求体

如下图所示，请求体中应用了大写转小写运算函数，函数中的参数A为“Test”。

```

1  {
2  |     "id":1,
3  |     "lower":"$toLower(TEST)"
4  | }
5

```

- 检查点属性

如下图所示，检查点属性“result”的目标值为大写转小写运算函数，函数中的参数A为“Test”。

来源	属性	高级提取类型	类型	比较符	目标值
响应体(JSON)	result	不涉及		等于(数字)	\$toLower(Test)

- if判断

如下图所示，if判断的目标值为大写转小写运算函数，函数中的参数A为“AAAAA”。



- for循环中断条件

如下图所示，for循环中断条件的目标值为大写转小写运算函数，函数中的参数A为“OK”。



## 6.4.11.21 在 CodeArts TestPlan 中调用小写转大写函数

### 函数名

\$toUpper(String A)

### 参数说明

- String A: 需要转为大写的字符串。

参数A支持以下类型：

- 字符串
- 局部参数

### 功能说明

将指定输入的字符串中的字符全部转换成大写字符。

### 使用场景

接口自动化用例中支持在以下场景中使用小写转大写函数：

- 请求url路径
- 请求头
- 请求体
- 检查点属性
- if判断
- for循环中断条件

### 示例

- 请求url路径

如下图所示，请求url路径中参数“test”的值为小写转大写运算函数，函数中的参数A为“test”。



- 请求头

如下图所示，请求头中参数“upper”的值为小写转大写运算函数，函数中的参数A为“Test”。

名称	值
Content-Type	application/json
upper	\$toUpper(Test)

- 请求体

如下图所示，请求体中应用了小写转大写运算函数，函数中的参数A为“Test”。

```

请求体:
1 {
2   "id": 1,
3   "upper": "$toUpper(Test)"
4 }
    
```

- 检查点属性

如下图所示，检查点属性“result”的目标值为小写转大写运算函数，函数中的参数A为“Test”。

来源	属性	高级语法和类型	期望值	比较符	目标值
响应体(JSON)	result	不涉及		等于(字符串、区分大小写)	\$toUpper(Test)

- if判断

如下图所示，if判断的目标值为小写转大写运算函数，函数中的参数A为“aaaaa”。

if	\$toUpper(aaaaa)	等于(字符串、区分大小写)	AAAAA
----	------------------	---------------	-------

- for循环中断条件

如下图所示，for循环中断条件的目标值为小写转大写运算函数，函数中的参数A为“ok”。

循环	循环 10 次			
中断条件	<table border="1"> <tr> <td>`\${status}</td> <td>等于(字符串、区分大小写)</td> <td>\$toUpper(ok)</td> </tr> </table>	`\${status}	等于(字符串、区分大小写)	\$toUpper(ok)
`\${status}	等于(字符串、区分大小写)	\$toUpper(ok)		

## 6.4.11.22 在 CodeArts TestPlan 中调用字符串拼接函数

### 函数名

\$strConcat(String A, String B)

### 参数说明

- String A: 第一个参数为字符串1。
- String B: 第二个参数为字符串2。

参数A、B支持以下类型：

- 字符串
- 局部参数

## 功能说明

将字符串1和字符串2拼接成一个新的字符串。

## 使用场景

接口自动化用例中支持在以下场景中使用字符串拼接函数：

- 请求url路径
- 请求头
- 请求体
- 检查点属性
- if判断
- for循环中断条件

## 示例

- 请求url路径

如下图所示，请求url路径中参数“test”的值为字符串拼接运算函数，函数中的参数A为“abc”，参数B为“123”。

名称	值
test	<code>\$strConcat(abc, 123)</code>

- 请求头

如下图所示，请求头中参数“number”的值为字符串拼接运算函数，函数中的参数A为“00000”，参数B为环境参数“`$${number}`”。

名称	值
Content-Type	application/json
number	<code>\$strConcat(00000, \$\${number})</code>

- 请求体

如下图所示，请求体中应用了字符串拼接运算函数，函数中的参数A为环境参数“`$${info}`”，参数B为环境参数“`$${number}`”。

```
1  {
2      "id":1,
3      "address": "$strConcat($${info}, $${number})"
4  }
5
```

- 检查点属性

如下图所示，检查点属性“result”的目标值为字符串拼接运算函数，函数中的参数A为局部参数“`${str1}`”，参数B为局部参数“`${str2}`”。

来源	属性	高级提取类型	类型值	比较符	目标值
响应体(JSON)	result	不涉及		等于(字符串, 区分大小写)	<code>\$strConcat(\${str1}, \${str2})</code>

- if判断

如下图所示，if判断的目标值为字符串拼接运算函数，函数中的参数A为“abc”，参数B为“123”。



- for循环中断条件

如下图所示，for循环中断条件的目标值为字符串拼接运算函数，函数中的参数A为“0000”，参数B为“1111”。



### 6.4.11.23 在 CodeArts TestPlan 中调用字符串切割函数

#### 函数名

\$strCut(String A, int B, int C)

#### 参数说明

- String A: 切割原始字符串。
- int B: 切割字符串起始下标，从0开始。
- int C: 切割字符串末尾下标。

参数A、B、C支持以下类型：

- 字符串
- 局部参数

#### 功能说明

获取指定元素的值字符串切割，第一个参数为切割原始字符串，第二个参数为切割字符串起始下标，第三个参数为切割字符串末尾下标，返回新的字符串。

#### 使用场景

接口自动化用例中支持在以下场景中使用字符串切割函数：

- 请求url路径
- 请求头
- 请求体
- 检查点属性
- if判断
- for循环中断条件

#### 示例

- 请求url路径

如下图所示，请求url路径中参数“test”的值为字符串切割运算函数，函数中的参数A为环境参数“\${user}”，参数B为整数2，参数C为整数4。

名称	值
name	\$strCut(\${user}, 2, 4)

- 请求头

如下图所示，请求头中参数“name”的值为字符串切割运算函数，函数中的参数A为环境参数“\${user}”，参数B为整数2，参数C为整数4。

名称	值
Content-Type	application/json
name	\$strCut(\${user}, 2, 4)

- 请求体

如下图所示，请求体中应用了字符串切割运算函数，函数中的参数A为环境参数“\${user}”，参数B为整数2，参数C为整数4。

```

1  {
2      "id": "$toUpper(${a})",
3      "name": "$strCut(${user}, 2, 4)"
4  }
5

```

- 检查点属性

如下图所示，检查点属性“result”的目标值为字符串切割运算函数，函数中的参数A为环境参数“\${info}”，参数B为整数2，参数C为整数5。

来源	属性	高级提取类型	类型值	比较符	目标值
响应体(JSON)	result	不涉及		等于(数字)	\$strCut(\${info}, 2, 5)

- if判断

如下图所示，if判断的目标值为字符串切割运算函数，函数中的参数A为“abcdef”，参数B为整数2，参数C为整数4。

if	\$strCut(abcdef, 2, 4)	等于(字符串、区分大小...)	cd
----	------------------------	-----------------	----

- for循环中断条件

如下图所示，for循环中断条件的目标值为字符串切割运算函数，函数中的参数A为环境参数“\${test}”，参数B为整数2，参数C为整数4。

循环	循环 10 次
中断条件	\$\${info} 等于(字符串、区分大小...)
	\$strCut(\${test}, 2, 4)

## 6.4.11.24 在 CodeArts TestPlan 中调用获取字符串长度函数

### 函数名

\$strlen(String A)

## 参数说明

- String A: 需要获取长度的原始字符串。

参数A支持以下类型:

- 字符串
- 局部参数

## 功能说明

获取指定字符串的长度，参数为计算的原始字符串。

## 使用场景

接口自动化用例中支持在以下场景中使用获取字符串长度函数:

- 请求url路径
- 请求头
- 请求体
- 检查点属性
- if判断
- for循环中断条件

## 示例

- 请求url路径

如下图所示，请求url路径中参数“test”的值为获取字符串长度运算函数，函数中的参数为局部参数“\${test}”。

名称	值
test	\$strlen(\${test})

- 请求头

如下图所示，请求头中参数“test”的值为获取字符串长度运算函数，函数中的参数A为局部参数“\${test}”。

请求头:

名称	值
Content-Type	application/json
test	\$strlen(\${test})

- 请求体

如下图所示，请求体中应用了获取字符串长度运算函数，函数中的参数A为局部参数“\${test}”。



```

1  {
2  |   "String": "$strLen(${test})"
3  | }

```

- 检查点属性

如下图所示，检查点属性“result”的目标值为获取字符串长度运算函数，函数中的参数A为环境参数“\${info}”。

来源	属性	高级提取类型	类型值	比较符	目标值
响应体(JSON)	result	不涉及		等于(数字)	\$strLen(\${info})

- if判断

如下图所示，if判断的目标值为获取字符串长度运算函数，函数中的参数为局部参数“\${test}”。

if	\$strLen(\${test})	等于(数字)	10
----	--------------------	--------	----

- for循环中断条件

如下图所示，for循环中断条件的目标值为获取字符串长度运算函数，函数中的参数为局部参数“\${test}”。

循环	循环	次	
中断条件	\${info}	等于(数字)	\$strLen(\${test})

## 6.4.12 接口自动化用例系统关键字说明

### 6.4.12.1 系统关键字概述与一览表

接口自动化测试将常见的一些操作封装成关键字供用户使用，用于提升编写接口测试用例的效率，请参考表6-5。

表 6-5 系统关键字一览表

分类	关键字集
认证类	<ul style="list-style-type: none"> <li>● 在CodeArts TestPlan接口脚本中调用GetIAMToken关键字</li> </ul>

分类	关键字集
数据库操作类	<ul style="list-style-type: none"> <li>● 在CodeArts TestPlan接口脚本中调用MySQLQuery关键字</li> <li>● 在CodeArts TestPlan接口脚本中调用MySQLUpdate关键字</li> <li>● 在CodeArts TestPlan接口脚本中调用MySQLInsert关键字</li> <li>● 在CodeArts TestPlan接口脚本中调用MySQLDelete关键字</li> <li>● 在CodeArts TestPlan接口脚本中调用OpenGaussQuery关键字</li> <li>● 在CodeArts TestPlan接口脚本中调用OpenGaussUpdate关键字</li> <li>● 在CodeArts TestPlan接口脚本中调用OpenGaussInsert关键字</li> <li>● 在CodeArts TestPlan接口脚本中调用OpenGaussDelete关键字</li> <li>● 在CodeArts TestPlan接口脚本中调用PostgreSQLQuery关键字</li> <li>● 在CodeArts TestPlan接口脚本中调用PostgreSQLUpdate关键字</li> <li>● 在CodeArts TestPlan接口脚本中调用PostgreSQLInsert关键字</li> <li>● 在CodeArts TestPlan接口脚本中调用PostgreSQLDelete关键字</li> <li>● 在CodeArts TestPlan接口脚本中调用MongoDBQuery关键字</li> <li>● 在CodeArts TestPlan接口脚本中调用MongoDBInsert关键字</li> <li>● 在CodeArts TestPlan接口脚本中调用MongoDBUpdate关键字</li> <li>● 在CodeArts TestPlan接口脚本中调用MongoDBDelete关键字</li> </ul>

分类	关键字集
中间件类	<ul style="list-style-type: none"> <li>● 在CodeArts TestPlan接口脚本中调用RedisGet关键字</li> <li>● 在CodeArts TestPlan接口脚本中调用RedisSet关键字</li> <li>● 在CodeArts TestPlan接口脚本中调用OBSWrite关键字</li> <li>● 在CodeArts TestPlan接口脚本中调用OBSDelete关键字</li> <li>● 在CodeArts TestPlan接口脚本中调用OBSQuery关键字</li> <li>● 在CodeArts TestPlan接口脚本中调用KafkaProducer关键字</li> <li>● 在CodeArts TestPlan接口脚本中调用KafkaConsumer关键字</li> </ul>
协议类	<ul style="list-style-type: none"> <li>● 在CodeArts TestPlan接口脚本中调用TCP关键字</li> <li>● 在CodeArts TestPlan接口脚本中调用UDP关键字</li> <li>● 在CodeArts TestPlan接口脚本中调用WSConnect关键字</li> <li>● 在CodeArts TestPlan接口脚本中调用WSRequest关键字</li> <li>● 在CodeArts TestPlan接口脚本中调用WSWriteOnly关键字</li> <li>● 在CodeArts TestPlan接口脚本中调用WSReadOnly关键字</li> <li>● 在CodeArts TestPlan接口脚本中调用WSDisconnect关键字</li> <li>● 在CodeArts TestPlan接口脚本中调用DubboClient关键字</li> </ul>

### 6.4.12.2 在 CodeArts TestPlan 接口脚本中调用 GetIAMToken 关键字

#### 概述

该系统关键字可以用于通过用户名/密码的方式进行认证来获取IAM用户Token。Token是系统颁发给IAM用户的访问令牌，承载用户的身份、权限等信息。调用IAM以及其他云服务的接口时，可以使用该系统关键字获取的IAM用户Token。

## 参数详解

参数	是否必选	参数类型	默认值	描述
IAM Token URL	是	String	https://iam.myhuaweicloud.com/v3/auth/tokens	IAM的Endpoint（终端节点）。 该接口可以使用全局区域的Endpoint和其他区域的Endpoint调用。
Domain Name	是	String	<空>	IAM用户所属账号名称/租户名。 当使用华为账号登录时，租户(账号)名称与用户名相同。关于如何获取账号名称请参见： <a href="#">获取账号</a> 。
User Name	是	String	<空>	IAM用户名。关于如何获取用户名称请参见： <a href="#">获取IAM用户名</a> 。
Password	是	String	<空>	IAM用户登录密码（登录密码属于个人敏感信息，须在环境参数中定义为敏感参数）。
Region ID	否	String	<空>	区域ID（如：cn-north-1），关于如何获取区域ID请参见： <a href="#">获取区域ID</a> 。

## 默认检查点

名称	预期值
结果	成功

## 默认响应提取

名称	提取变量	含义
IAM_TOKEN	X-Subject-Token	用户TOKEN字符

## 响应

- 状态：成功

参数	参数类型	描述
Body	IAM接口返回体	IAM接口返回体

● 响应示例

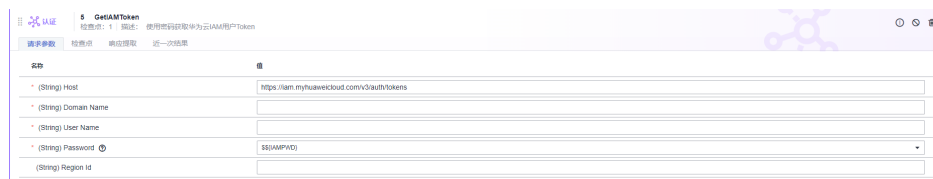
```
{
  "token": {
    "catalog": [],
    "expires_at": "2020-01-04T09:05:22.701000Z",
    "issued_at": "2020-01-03T09:05:22.701000Z",
    "methods": [
      "password"
    ],
    "project": {
      "domain": {
        "id": "d78cbac186b744899480f25bd022f...",
        "name": "IAMDomain"
      },
      "id": "aa2d97d7e62c4b7da3ffdfc11551f...",
      "name": "cn-north-1"
    },
    "roles": [
      {
        "id": "0",
        "name": "te_admin"
      },
      {
        "id": "0",
        "name": "op_gated_OBS_file_protocol"
      },
      {
        "id": "0",
        "name": "op_gated_Video_Campus"
      }
    ],
    "user": {
      "domain": {
        "id": "d78cbac186b744899480f25bd022f...",
        "name": "IAMDomain"
      },
      "id": "7116d09f88fa41908676fdd4b039e...",
      "name": "IAMUser",
      "password_expires_at": ""
    }
  },
  "X-Subject-Token": "MlIlatAYJKoZlhcNAQcCollapTCCGqECAQExDTALB..."
}
```

示例

● 获取IAM用户TOKEN

📖 说明

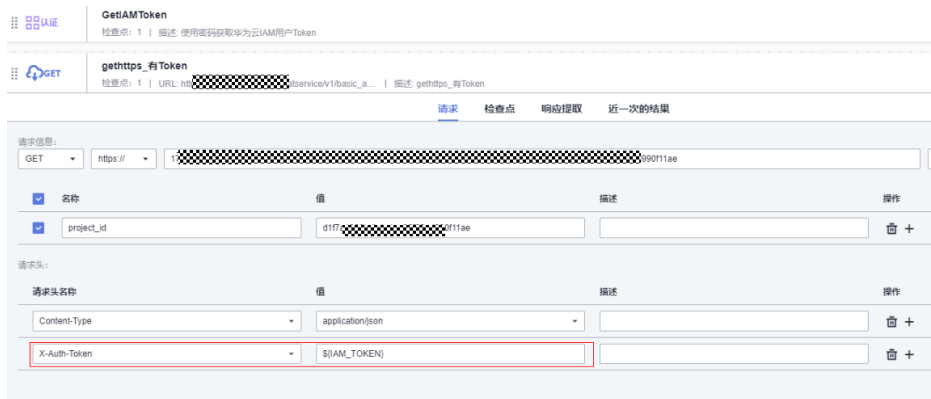
Password字段属于敏感参数，需要在变量里面配置且设置为敏感参数，在下拉框中选用。



● 在接口中引用TOKEN

📖 说明

IAM\_TOKEN为保留的默认响应提取，可以直接在请求里面引用。



6.4.12.3 在 CodeArts TestPlan 接口脚本中调用 MySQLQuery 关键字

MySQL SELECT操作使用，系统限制最大查询结果行数100条。

参数	是否必选	参数类型	描述
Ip	是	String	数据库ip 如果是华为云RDS实例，需要绑定EIP，并保证实例端口安全组策略开启放行，详细请参考华为云： <a href="#">通过公网连接MySQL实例。</a>
Port	是	Integer	数据库端口
DB Name	是	String	数据库实例名称
User Name	是	String	用户名
Password	是	String	密码（登录密码属于个人敏感信息，须在环境参数中定义为敏感参数）
Sql	是	String	sql查询语句

MySQLQuery 响应

状态：成功

参数	参数类型	描述
[数组元素]	Array of Row objects	sql查询结果列表返回结构体

## 参数列表

参数	参数类型	描述
field1	String	数据库字段1
field2	Integer	数据库字段2
...	String	数据库字段n..

## MySQLQuery 响应示例

```
[
  {
    "name": "张三",
    "id": "efdb403066474ab08836b9eeaaa23bca",
    "age": 18
  },
  {
    "name": "李四",
    "id": "g582b0d966611486f918bedb9c711b14",
    "age": 20
  }
]
```

## MySQLQuery 使用示例

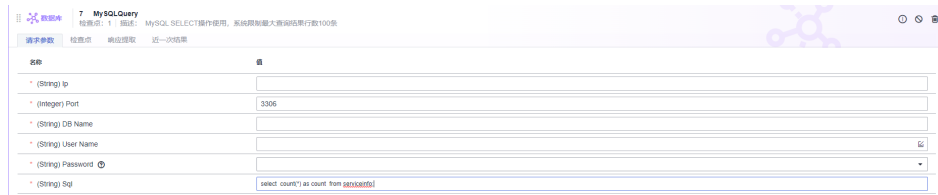
- 查询数据库表数据，系统关键字返回用户数据列表，用户可根据业务对返回进行判断。



用户可以在返回的List里面进行结果判断：



- 查询用户数据并进行数据统计，结果返回用户数据统计列表。



用户可以在响应提取里面进行结果判断：



### 6.4.12.4 在 CodeArts TestPlan 接口脚本中调用 MySQLUpdate 关键字

MySQL修改操作使用，适用于INSERT、UPDATE或DELETE等没有返回值的语句。

参数	是否必选	参数类型	描述
Ip	是	String	数据库ip 如果是华为云RDS实例，需要绑定EIP，并保证实例端口安全组策略开启放行，详细请参考华为云： <a href="#">通过公网连接MySQL实例。</a>
Port	是	Integer	数据库端口
DB Name	是	String	数据库实例名称
User Name	是	String	用户名
Password	是	String	密码（登录密码属于个人敏感信息，须在环境参数中定义为敏感参数）
Sql	是	String	UPDATE等没有返回值的sql语句

### MySQLUpdate 响应

状态：成功

参数	参数类型	描述
affected_row_count	Integer	sql语句影响行数

### MySQLUpdate 使用示例



对影响条数进行判断：





## MySQLUpdate 响应示例

```
{
  "affected_row_count": 3
}
```

### 6.4.12.5 在 CodeArts TestPlan 接口脚本中调用 MySQLInsert 关键字

适用于MySQLInsert 插入数据操作使用。

参数	是否必选	参数类型	描述
Ip	是	String	数据库ip: 如果是华为云RDS实例, 需要绑定EIP, 并保证实例端口安全组策略开启放行, 详情请参考华为云: <a href="#">通过公网连接MySQL实例</a> 。
Port	是	Integer	数据库端口
DB Name	是	String	数据库实例名称
User Name	是	String	用户名
Password	是	String	密码 (登录密码属于个人敏感信息, 须在环境参数中定义为敏感参数)
Sql	是	String	INSERT操作, 没有返回值的sql语句

## MySQLInsert 响应

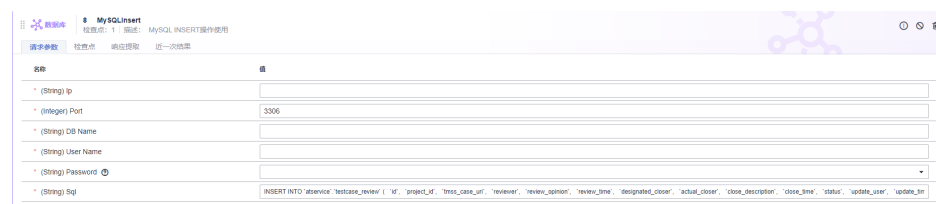
状态: 成功

参数	参数类型	描述
affected_row_count	Integer	sql语句影响行数

## MySQLInsert 响应示例

```
{
  "affected_row_count": 1
}
```

## MySQLInsert 使用示例



对影响条数进行判断：



### 6.4.12.6 在 CodeArts TestPlan 接口脚本中调用 MySQLDelete 关键字

适用于MySQLDelete删除操作使用。

参数	是否必选	参数类型	描述
Ip	是	String	数据库ip: 如果是华为云RDS实例, 需要绑定EIP, 并保证实例端口安全组策略开启放行, 详细请参考华为云: <a href="#">通过公网连接MySQL实例。</a>
Port	是	Integer	数据库端口
DB Name	是	String	数据库实例名称
User Name	是	String	用户名
Password	是	String	密码 (登录密码属于个人敏感信息, 须在环境参数中定义为敏感参数)
Sql	是	String	DELETE操作, 没有返回值的sql语句

### MySQLDelete 响应

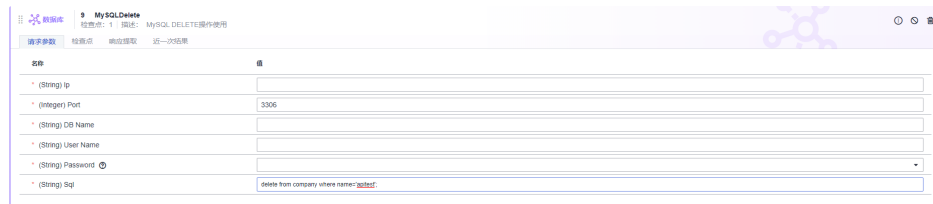
状态: 成功

参数	参数类型	描述
affected_row_count	Integer	sql语句影响行数

### MySQLDelete 响应示例

```
{
  "affected_row_count": 1
}
```

## MySQLDelete 使用示例



对影响条数进行判断:



### 6.4.12.7 在 CodeArts TestPlan 接口脚本中调用 OpenGaussQuery 关键字

适用于OpenGaussQuery SELECT操作使用，系统限制最大查询结果行数100条。

参数	是否必选	参数类型	描述
Ip	是	String	数据库ip: 如果是云数据库 GaussDB实例，需要绑定EIP，并保证实例端口安全组策略开启放行，详细请参考华为云： <a href="#">通过gsq连接实例</a> 。
Port	是	Integer	数据库端口
DB Name	是	String	数据库实例名称
User Name	是	String	用户名
Password	是	String	密码（登录密码属于个人敏感信息，须在环境参数中定义为敏感参数）
Sql	是	String	sql查询语句

## OpenGaussQuery 响应

状态: 成功

参数	参数类型	描述
[数组元素]	Array of Row objects	sql查询结果列表返回结构体

### 参数列表

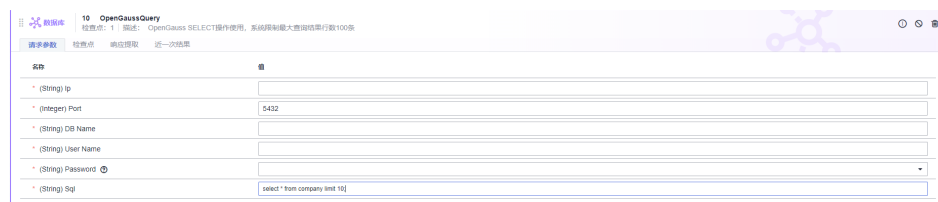
参数	参数类型	描述
field1	String	数据库字段1
field2	Integer	数据库字段2
...	String	数据库字段n

### OpenGaussQuery 响应示例

```
[
  {
    "name": "张三",
    "id": "efdb403066474ab08836b9eaaaa23bca",
    "age": 18
  },
  {
    "name": "李四",
    "id": "g582b0d966611486f918bedb9c711b14",
    "age": 20
  }
]
```

### OpenGaussQuery 使用示例

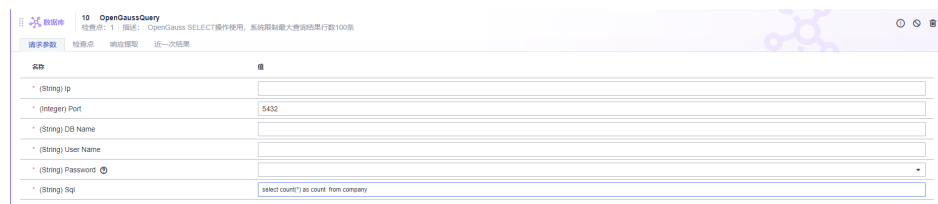
查询数据库表数据，系统关键字返回用户数据列表，用户可根据业务对返回进行判断。



用户可以在返回的List里面进行结果判断：



查询用户数据并进行数据统计，结果返回用户数据统计列表。



用户可以在响应提取里面进行结果判断：



### 6.4.12.8 在 CodeArts TestPlan 接口脚本中调用 OpenGaussUpdate 关键字

适用于OpenGaussUpdate 修改操作使用。

参数	是否必选	参数类型	描述
Ip	是	String	数据库ip: 如果是云数据库 GaussDB 实例, 需要绑定 EIP, 并保证实例端口安全组策略开启放行, 详细请参考华为云: <a href="#">通过gsq连接实例</a> 。
Port	是	Integer	数据库端口
DB Name	是	String	数据库实例名称
User Name	是	String	用户名
Password	是	String	密码 (登录密码属于个人敏感信息, 须在环境参数中定义为敏感参数)
Sql	是	String	UPDATE操作没有返回值的sql语句

### OpenGaussUpdate 响应

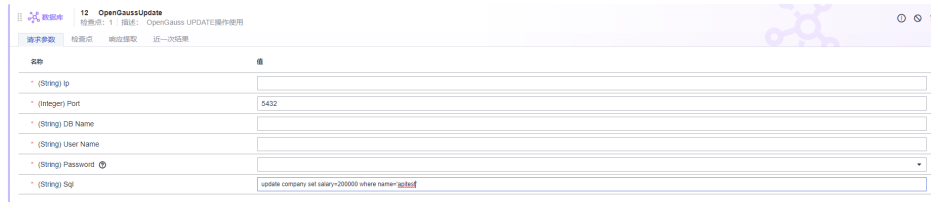
状态: 成功

参数	参数类型	描述
affected_row_count	Integer	sql语句影响行数

### OpenGaussUpdate 响应示例

```
{
  "affected_row_count": 1
}
```

## OpenGaussUpdate 使用示例



对影响条数进行判断:



### 6.4.12.9 在 CodeArts TestPlan 接口脚本中调用 OpenGaussInsert 关键字

适用于OpenGaussInsert 插入数据操作使用。

参数	是否必选	参数类型	描述
Ip	是	String	数据库ip: 如果是云数据库 GaussDB 实例, 需要绑定 EIP, 并保证实例端口安全组策略开启放行, 详细请参考华为云: <a href="#">通过gsqL连接实例</a> 。
Port	是	Integer	数据库端口
DB Name	是	String	数据库实例名称
User Name	是	String	用户名
Password	是	String	密码 (登录密码属于个人敏感信息, 须在环境参数中定义为敏感参数)
Sql	是	String	INSERT操作, 没有返回值的sql语句

## OpenGaussInsert 响应

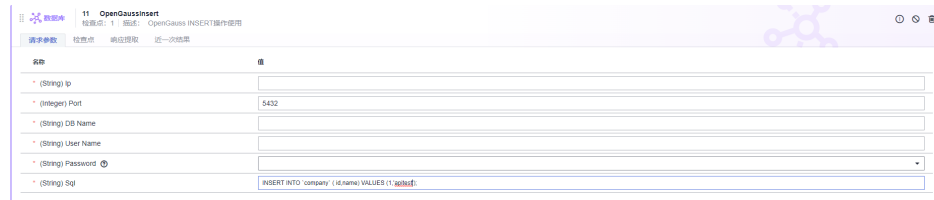
状态: 成功

参数	参数类型	描述
affected_row_count	Integer	sql语句影响行数

## OpenGaussInsert 响应示例

```
{
  "affected_row_count": 1
}
```

## OpenGaussInsert 使用示例



对影响条数进行判断：



### 6.4.12.10 在 CodeArts TestPlan 接口脚本中调用 OpenGaussDelete 关键字

适用于OpenGaussDelete删除操作使用。

参数	是否必选	参数类型	描述
ip	是	String	数据库ip: 如果是云数据库 GaussDB 实例, 需要绑定 EIP, 并保证实例端口安全组策略开启放行, 详细请参考华为云: <a href="#">通过gsqI连接实例</a> 。
Port	是	Integer	数据库端口
DB Name	是	String	数据库实例名称
User Name	是	String	用户名
Password	是	String	密码 (登录密码属于个人敏感信息, 须在环境参数中定义为敏感参数)
Sql	是	String	DELETE操作, 没有返回值的sql语句

## OpenGaussDelete 响应

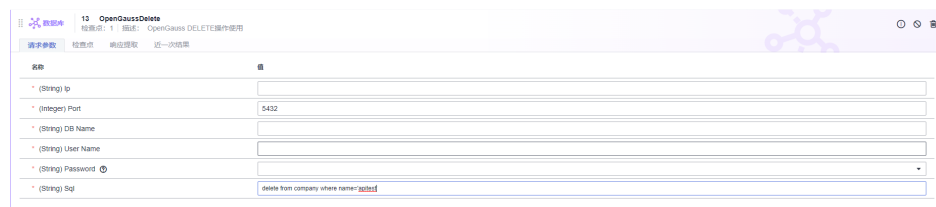
状态：成功

参数	参数类型	描述
affected_row_count	Integer	sql语句影响行数

## OpenGaussDelete 响应示例

```
{
  "affected_row_count": 1
}
```

## OpenGaussDelete 使用示例



对影响条数进行判断：



### 6.4.12.11 在 CodeArts TestPlan 接口脚本中调用 PostgreSQLQuery 关键字

适用于PostgreSQLQuery SELECT操作使用，系统限制最大查询结果行数100条。

参数	是否必选	参数类型	描述
ip	是	String	数据库ip：如果是华为云RDS实例，需要绑定EIP，并保证实例端口安全组策略开启放行，详细请参考华为云： <a href="#">连接RDSfor PostgreSQL实例的方式</a> 。
Port	是	String	数据库端口
DB Name	是	String	数据库实例名称
User Name	是	String	用户名



参数	是否必选	参数类型	描述
Password	是	String	密码（登录密码属于个人敏感信息，须在环境参数中定义为敏感参数）
Sql	是	String	sql查询语句

## PostgreSQLQuery 响应

状态：成功

参数	参数类型	描述
[数组元素]	Array of Row objects	sql查询结果列表返回结构体

## 参数列表

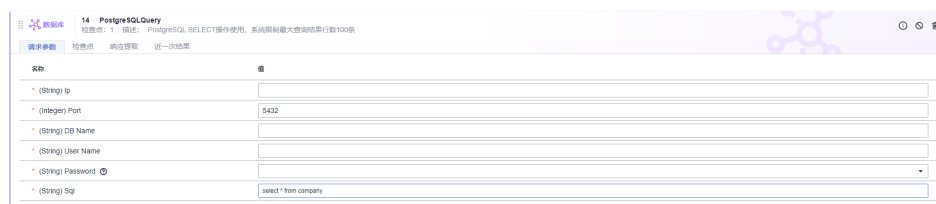
参数	参数类型	描述
field1	String	数据库字段1
field2	Integer	数据库字段2
...	String	数据库字段n

## PostgreSQLQuery 响应示例

```
[
  {
    "name": "张三",
    "id": "efdb403066474ab08836b9eeaaa23bca",
    "age": 18
  },
  {
    "name": "李四",
    "id": "g582b0d966611486f918bedb9c711b14",
    "age": 20
  }
]
```

## PostgreSQLQuery 使用示例

查询数据库表数据，系统关键字返回用户数据列表，用户可根据业务对返回进行判断。



用户可以在返回的List里面进行结果判断：



查询用户数据并进行数据统计，结果返回用户数据统计列表。



用户可以在响应提取里面进行结果判断：



### 6.4.12.12 在 CodeArts TestPlan 接口脚本中调用 PostgreSQLUpdate 关键字

适用于PostgreSQLUpdate 修改操作使用。

参数	是否必选	参数类型	描述
Ip	是	String	数据库ip: 如果是华为云RDS实例，需要绑定EIP，并保证实例端口安全组策略开启放行，详细请参考华为云： <a href="#">连接RDSfor PostgreSQL实例的方式</a> 。
Port	是	Integer	数据库端口
DB Name	是	String	数据库实例名称
User Name	是	String	用户名
Password	是	String	密码（登录密码属于个人敏感信息，须在环境参数中定义为敏感参数）
Sql	是	String	UPDATE操作没有返回值的sql语句

## PostgreSQLUpdate 响应

状态: 成功

参数	参数类型	描述
affected_row_count	Integer	sql语句影响行数

## PostgreSQLUpdate 响应示例

```
{
  "affected_row_count": 1
}
```

## PostgreSQLUpdate 使用示例



对影响条数进行判断:



### 6.4.12.13 在 CodeArts TestPlan 接口脚本中调用 PostgreSQLInsert 关键字

适用于PostgreSQLInsert 插入数据操作使用。

参数	是否必选	参数类型	描述
Ip	是	String	数据库ip: 如果是华为云RDS实例, 需要绑定EIP, 并保证实例端口安全组策略开启放行, 详细请参考华为云: <a href="#">连接RDSfor PostgreSQL实例的方式</a> 。
Port	是	Integer	数据库端口
DB Name	是	String	数据库实例名称
User Name	是	String	用户名

参数	是否必选	参数类型	描述
Password	是	String	密码（登录密码属于个人敏感信息，须在环境参数中定义为敏感参数）
Sql	是	String	INSERT操作，没有返回值的sql语句

## PostgreSQLInsert 响应

状态：成功

参数	参数类型	描述
affected_row_count	Integer	sql语句影响行数

## PostgreSQLInsert 响应示例

```
{
  "affected_row_count": 1
}
```

## PostgreSQLInsert 使用示例



对影响条数进行判断：



### 6.4.12.14 在 CodeArts TestPlan 接口脚本中调用 PostgreSQLDelete 关键字

适用于PostgreSQLDelete删除操作使用。

参数	是否必选	参数类型	描述
Ip	是	String	数据库ip: 如果是华为云RDS实例, 需要绑定EIP, 并保证实例端口安全组策略开启放行, 详细请参考华为云: <a href="#">连接RDSfor PostgreSQL实例的方式</a> 。
Port	是	Integer	数据库端口
DB Name	是	String	数据库实例名称
User Name	是	String	用户名
Password	是	String	密码 (登录密码属于个人敏感信息, 须在环境参数中定义为敏感参数)
Sql	是	String	DELETE操作, 没有返回值的sql语句

## PostgreSQLDelete 响应

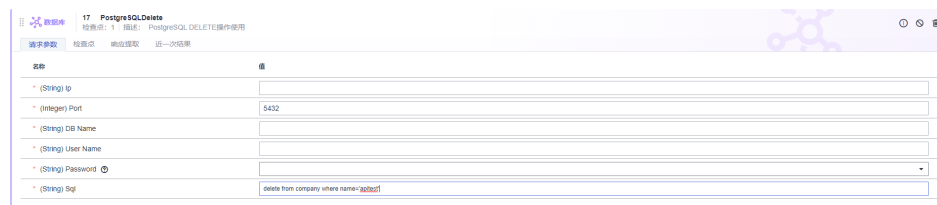
状态: 成功

参数	参数类型	描述
affected_row_count	Integer	sql语句影响行数

## PostgreSQLDelete 响应示例

```
{
  "affected_row_count": 1
}
```

## PostgreSQLDelete 使用示例



对影响条数进行判断:



### 6.4.12.15 在 CodeArts TestPlan 接口脚本中调用 MongoDBQuery 关键字

MongoDB查询操作使用，系统限制最大查询结果记录数100条。

参数	是否必选	参数类型	描述
Ip	是	String	数据库ip地址
Port	是	Integer	数据库端口
User Name	是	String	用户名
Password	是	String	密码（登录密码属于个人敏感信息，须在环境参数中定义为敏感参数）
DB Name	是	String	数据库实例名称
Collection	是	String	集合名称
Query	否	String	查询操作指定查询条件（Bson格式数据），默认为空查询集合下的所有数据，系统限制最大记录数为100条。
Limit	否	Integer	<ul style="list-style-type: none"> <li>查询结果限制条数，默认为空查询集合下的所有数据，系统限制最大记录数为100条。</li> <li>若输入的参数0-100之外的其他数字则取最大限制数100条。</li> </ul>

### MongoDBQuery 响应

状态：成功

参数	参数类型	描述
[数组元素]	Array of Row objects	查询结果列表返回结构体

### 参数列表

参数	参数类型	描述
field1	String	数据库字段1
field2	Integer	数据库字段2
...	String	数据库字段n..

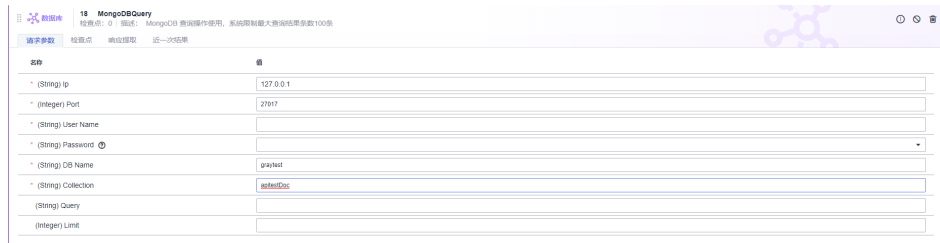
### MongoDBQuery 响应示例

```
[{
  "_id": {
```

```
    "$oid" : "62465c42907c00003d0076fe"  
  },  
  "title" : "MongoDB 教程",  
  "description" : "MongoDB 是一个 Nosql 数据库",  
  "by" : "test",  
  "url" : "",  
  "tags" : [ "mongodb", "database", "NoSQL" ],  
  "likes" : 100.0  
}, {  
  "_id" : {  
    "$oid" : "62465ce4907c00003d0076ff"  
  },  
  "title" : "PHP 教程",  
  "description" : "PHP 是一种创建动态交互性站点的强有力的服务器端脚本语言。",  
  "by" : "test",  
  "url" : "",  
  "tags" : [ "php" ],  
  "likes" : 200.0  
}, {  
  "_id" : {  
    "$oid" : "62465ce8907c00003d007700"  
  },  
  "title" : "Java 教程",  
  "description" : "Java 是由Sun Microsystems公司于1995年5月推出的高级程序设计语言。",  
  "by" : "test",  
  "url" : "",  
  "tags" : [ "java" ],  
  "likes" : 150.0  
}]
```

### MongoDBQuery 使用示例

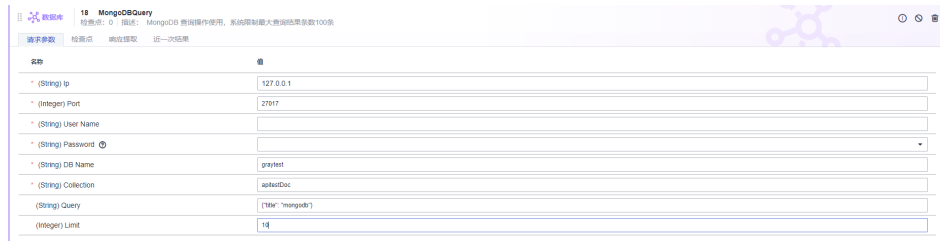
- 查询数据库集合下的文档数量，系统关键字返回查询文档数据列表，用户可根据业务对返回进行判断。



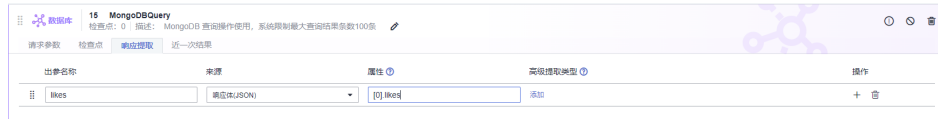
通过返回的List可以进行结果判断，例如：判断集合中的文档数量为10个。



- 查询用户数据并进行数据提取，结果返回查询出的数据列表。



通过响应提取可以进行结果提取或者判断。



### 6.4.12.16 在 CodeArts TestPlan 接口脚本中调用 MongoDBInsert 关键字

MongoDB插入操作使用，向集合中插入文档。

参数	是否必选	参数类型	描述
Ip	是	String	数据库ip地址
Port	是	Integer	数据库端口
User Name	是	String	用户名
Password	是	String	密码（登录密码属于个人敏感信息，须在环境参数中定义为敏感参数）
DB Name	是	String	数据库实例名称
Collection	是	String	集合名称
Bson	是	String	插入的数据（Bson格式数据）

### MongoDBInsert 响应

状态：成功

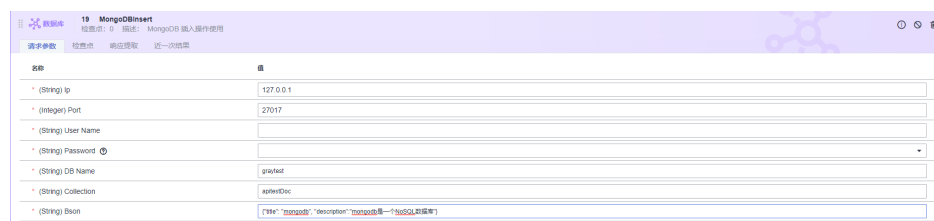
参数	参数类型	描述
affected_row_count	Integer	插入文档影响的行数

### MongoDBInsert 响应示例

```
{
  "affected_row_count": 1
}
```

### MongoDBInsert 使用示例

根据输入的参数数据将数据插入集合中，系统关键字返回插入的记录数。



通过测试检查点判断结果是否成功。





### 6.4.12.17 在 CodeArts TestPlan 接口脚本中调用 MongoDBUpdate 关键字

MongoDB更新文档操作使用，根据输入的参数更新集合中的文档。

参数	是否必选	参数类型	描述
Ip	是	String	数据库ip地址
Port	是	Integer	数据库端口
User Name	是	String	用户名
Password	是	String	密码（登录密码属于个人敏感信息，须在环境参数中定义为敏感参数）
DB Name	是	String	数据库实例名称
Collection	是	String	集合名称
Query	是	String	更新的查询条件（bson数据）
Update	是	String	需要更新的文档数据（bson数据）
Multi	否	Boolean	是否更新全部记录（默认是false,只更新找到的第一条记录，如果这个参数为true,就把按条件查出来多条记录全部更新）。

### MongoDBUpdate 响应

状态：成功

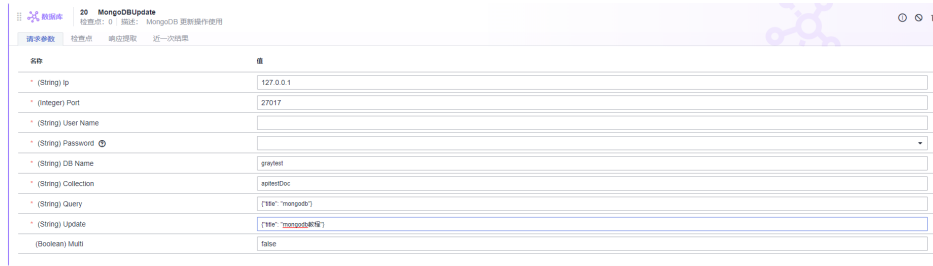
参数	参数类型	描述
affected_row_count	Integer	更新文档影响的行数

### MongoDBUpdate 响应示例

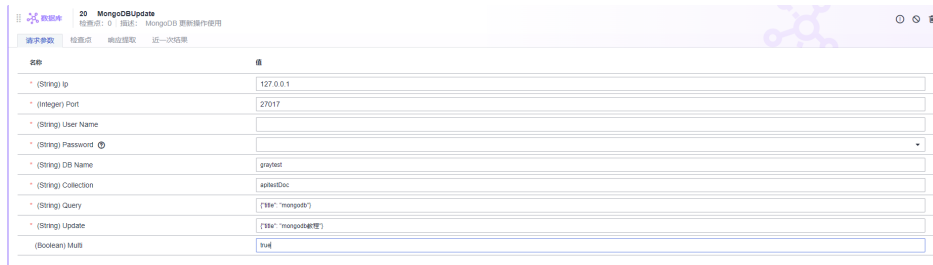
```
{
  "affected_row_count": 1
}
```

### MongoDBUpdate 使用示例

根据输入的查询数据和更新数据对集合中的文档进行更新，系统关键字返回更新的记录数。



根据multi参数设置为true来更新匹配到的所有文档。



判断结果是否成功，更新条数是否是预期值。



### 6.4.12.18 在 CodeArts TestPlan 接口脚本中调用 MongoDBDelete 关键字

MongoDB删除文档操作使用，根据输入的参数删除集合中的文档。

参数	是否必选	参数类型	描述
Ip	是	String	数据库ip地址
Port	是	Integer	数据库端口
User Name	是	String	用户名
Password	是	String	密码（登录密码属于个人敏感信息，须在环境参数中定义为敏感参数）
DB Name	是	String	数据库实例名称
Collection	是	String	集合名称
Query	是	String	删除操作指定的条件（Bson格式数据），默认为空查询集合下的所有数据进行删除
JustOne	否	Boolean	是否删除单条数据（如果设为 false，则删除所有匹配条件的文档，如果不设置该参数，或使用默认值 true，则只删除匹配的一个文档）。

## MongoDBDelete 响应

状态: 成功

参数	参数类型	描述
affected_row_count	Integer	删除文档影响的行数

## MongoDBDelete 响应示例

```
{
  "affected_row_count": 1
}
```

## MongoDBDelete 使用示例

根据输入的查询参数数据进行文档删除，系统关键字返回删除的记录数。

The screenshot shows the configuration for the MongoDBDelete test case. The 'JustOne' parameter is set to 'true'. Other parameters include IP (127.0.0.1), Port (27017), DB Name (gradtest), Collection (appleDoc), and Query ({"title": "mongodb教程"}).

根据justOne参数设置为false来删除匹配到的所有文档。

The screenshot shows the configuration for the MongoDBDelete test case. The 'JustOne' parameter is set to 'false'. Other parameters are the same as in the previous screenshot.

判断结果是否成功，更新条数是否是预期值。

The screenshot shows the test results for the MongoDBDelete test case. The result for 'affectedRowCount' is '等于(数字)' with a target value of '1'.

### 6.4.12.19 在 CodeArts TestPlan 接口脚本中调用 RedisGet 关键字

Redis字符串操作，获取指定key的值。

参数	是否必选	参数类型	描述
Ip	是	String	Redis数据库ip地址
Port	是	Integer	Redis数据库端口
Password	是	String	密码（登录密码属于个人敏感信息，须在环境参数中定义为敏感参数）
Key	是	String	指定的key名称

## RedisGet 响应

状态：成功

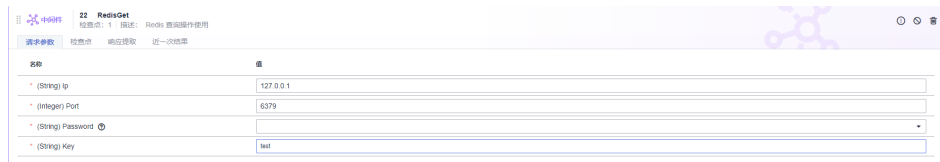
参数	参数类型	描述
key	String	指定key的value值

## RedisGet 响应示例

```
{
  "test" : "Redis"
}
```

## RedisGet 使用示例

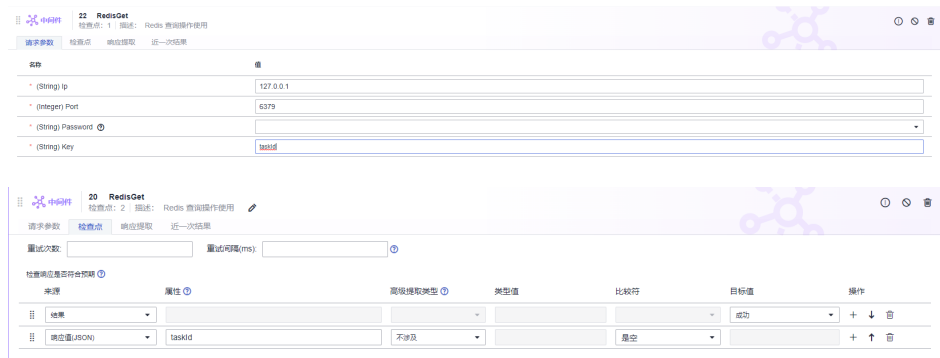
- 根据输入的key值，获取对应的value值。



判断结果是否成功，结果是否是预期值：



若key不存在，则返回的value值为空。可以在检查点判断。



### 6.4.12.20 在 CodeArts TestPlan 接口脚本中调用 RedisSet 关键字

Redis字符串操作，根据指定的key设置对应的value值，可设置过期时间。

参数	是否必选	参数类型	描述
Ip	是	String	Redis数据库ip地址
Port	是	Integer	Redis数据库端口
Password	是	String	密码（登录密码属于个人敏感信息，须在环境参数中定义为敏感参数）
Key	是	String	指定的key名称
Value	是	String	设置的value值
Expire	否	Long	过期时间，单位为秒。若使用默认值-1，则表示永不过期。使用Long类型的负整数则设置为永不过期。

### RedisSet 响应

状态：成功

参数	参数类型	描述
result	String	设置key对应的value值返回结果。若设置成功则返回OK。若设置失败则返回错误信息。

### RedisSet 响应示例

```
{
  "result": "OK"
}
```

### RedisSet 使用示例

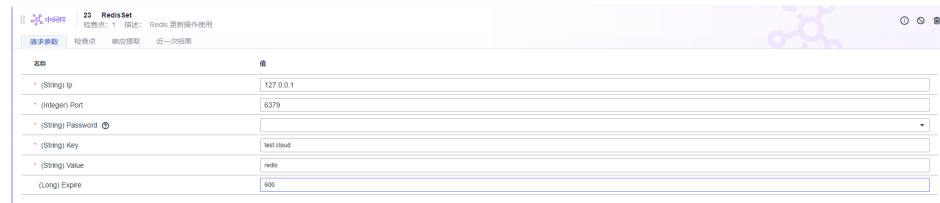
- 根据输入的key值，设置对应的value值，使用默认过期时间，永不过期。



判断是否设置成功：



- 根据输入的key值，设置对应的value值，使用过期时间，设置600秒后过期。



判断是否设置成功：



### 6.4.12.21 在 CodeArts TestPlan 接口脚本中调用 OBSWrite 关键字

OBS字符串操作，根据指定的key设置对应的value值。

参数	是否必选	参数类型	描述
Access Key ID	是	String	OBS 访问密钥AK（AK属于个人敏感信息，须在环境参数中定义为敏感参数）。
Secret Access ID	是	String	OBS 访问密钥SK（SK属于个人敏感信息，须在环境参数中定义为敏感参数）。
Rest Endpoint	是	String	OBS 终端节点（Endpoint）。
Bucket Name	是	String	OBS桶名称。
Key	是	String	OBS文件具体路径。
Value	是	String	OBS文件内容。

## OBSWrite 响应

状态：成功

参数	参数类型	描述
result	String	若设置成功则返回ok。
key	String	OBS文件具体路径。

状态：失败

参数	参数类型	描述
result	String	若设置成功则返回fail。
errorMessage	String	返回失败原因。

## OBSWrite 响应示例

```
{
  "result": "ok",
  "key": "/key"
}
```

## OBSWrite 使用示例

对“key1/key2”路径写入“测试数据”内容。



判断是否设置成功：



### 6.4.12.22 在 CodeArts TestPlan 接口脚本中调用 OBSDelete 关键字

OBS文件删除操作，根据指定的key删除对应文件。

参数	是否必选	参数类型	描述
Access Key ID	是	String	OBS 访问密钥AK（AK属于个人敏感信息，须在环境参数中定义为敏感参数）。
Secret Access ID	是	String	OBS 访问密钥SK（SK属于个人敏感信息，须在环境参数中定义为敏感参数）。
Rest Endpoint	是	String	OBS 终端节点（Endpoint）。
Bucket Name	是	String	OBS桶名称。
Key	是	String	OBS文件具体路径。

## OBSDelete 响应

状态：成功

参数	参数类型	描述
result	String	若设置成功则返回ok。
key	String	OBS文件具体路径。

状态：失败

参数	参数类型	描述
result	String	若设置成功则返回fail。
errorMessage	String	返回失败原因。

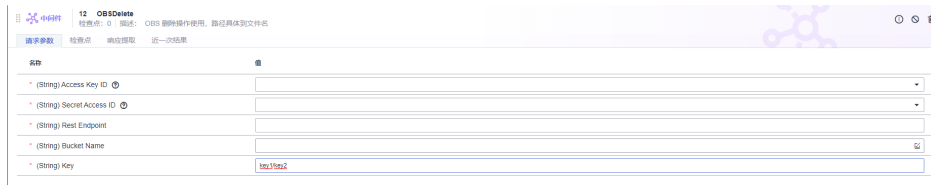
## OBSDelete 响应示例

```
{
  "result": "ok",
  "key": "/key"
}
```

## OBSDelete 使用示例

删除“key1/key2”路径数据。





判断是否设置成功：



### 6.4.12.23 在 CodeArts TestPlan 接口脚本中调用 OBSQuery 关键字

OBS内容查询操作操作，根据指定的key获取对应的文件内容。

参数	是否必选	参数类型	描述
Access Key ID	是	String	OBS 访问密钥AK（AK属于个人敏感信息，须在环境参数中定义为敏感参数）。
Secret Access ID	是	String	OBS 访问密钥SK（SK属于个人敏感信息，须在环境参数中定义为敏感参数）。
Rest Endpoint	是	String	OBS 终端节点（Endpoint）。
Bucket Name	是	String	OBS桶名称。
Key	是	String	OBS文件具体路径。

### OBSQuery 响应

状态：成功

直接返回文件内容，例如文件内容是“测试数据”，则展示“测试数据”。

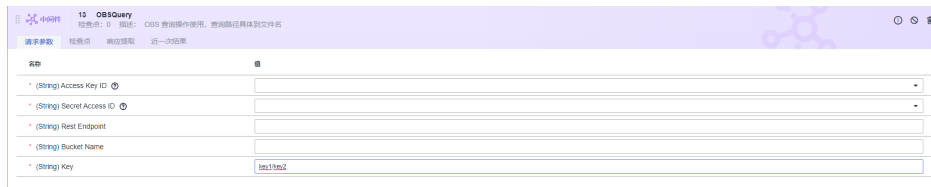
"测试数据"

状态：失败

参数	参数类型	描述
result	String	若设置成功则返回fail。
errorMessage	String	返回失败原因。

## OBSQuery 使用示例

根据“key1/key2”路径，查询数据。



判断是否获取成功：



### 6.4.12.24 在 CodeArts TestPlan 接口脚本中调用 KafkaProducer 关键字

#### KafkaProducer 简介

该系统关键字支持对Kafka的生产者进行测试。

参数	是否必选	参数类型	默认值	描述
Broker	是	String	127.0.0.1:9093	Kafka实例ip地址。
Topic	是	String	-	Kafka消息的主题。
Message	是	String	-	Kafka每一条发送的消息主体。
SASL Username	否	String	-	Kafka SASL 权限认证配置用户名。
SASL Password	否	String	-	Kafka SASL 权限认证配置密码。
Truststore	否	File	-	Kafka客户端证书。

参数	是否必选	参数类型	默认值	描述
Truststore Password	否	String	-	Kafka客户端证书密码。

## KafkaProducer 响应

状态：成功

参数	参数类型	描述
Body	String	KafkaProducer接口返回值。

## KafkaProducer 响应示例

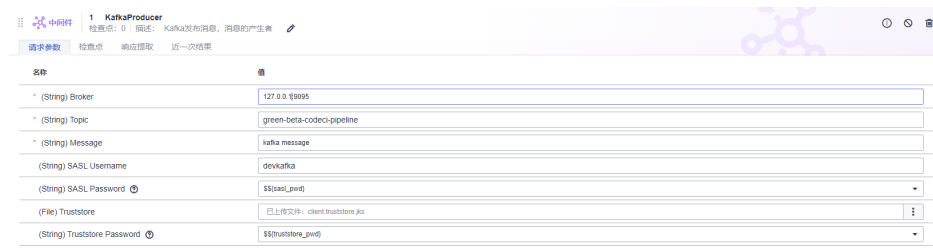
```
{
  "result" : "send message to kafka success.",
  "status" : "ok"
}
```

## 默认检查点

名称	预期值
结果	成功

## KafkaProducer 使用示例

填写参数，发送Kafka消息：



### 6.4.12.25 在 CodeArts TestPlan 接口脚本中调用 KafkaConsumer 关键字

## KafkaConsumer 简介

该系统关键字支持对Kafka的消费者进行测试。

参数	是否必选	参数类型	默认值	描述
Broker	是	String	127.0.0.1:9093	Kafka实例ip地址。
Topic	是	String	-	Kafka消息的主题。
Consumer Group	是	String	-	Kafka消息消费组。
SASL Username	否	String	-	Kafka SASL 权限认证配置用户名。
SASL Password	否	String	-	Kafka SASL 权限认证配置密码。
Truststore	否	File	-	Kafka客户端证书。
Truststore Password	否	String	-	Kafka客户端证书密码。

## KafkaConsumer 响应

状态：成功

参数	参数类型	描述
Body	String	KafkaConsumer接口返回值。

## KafkaConsumer 响应示例

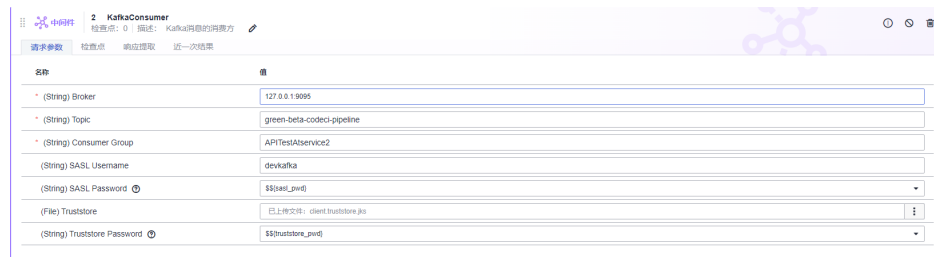
```
{
  "message": [{
    "offset": 102130,
    "value": "kafka message 1"
  }, {
    "offset": 102131,
    "value": "kafka message 2"
  }]
}
```

## 默认检查点

名称	预期值
结果	成功

## KafkaConsumer 使用示例

填写参数接受kafka消息：



### 6.4.12.26 在 CodeArts TestPlan 接口脚本中调用 TCP 关键字

#### TCP

该系统关键字支持对TCP基础协议进行测试，如果需要使用该系统关键字，需要用户IP开放公网访问并且对应端口的安全组策略开启放行。

参数	是否必选	参数类型	默认值	描述
Host	是	String	-	TCP服务地址
Port	是	Integer	-	TCP服务端口
Connect Timeout	是	Integer	-	TCP服务连接超时长 单位： MILLISECOND S
Read Timeout	是	Long	-	消息读取超时时间 单位： MILLISECOND S
Check End Type	是	Enum	CheckEndStr	消息结束标识： 数据长度 (CheckEndLen gth)， 结束字符 (CheckEndStr)

参数	是否必选	参数类型	默认值	描述
Body Type	是	Enum	CharSequence	消息体数据类型： 标准字符串 (CharSequence)， 16进制码流 (Hexadecimal CodeStream)

## TCP 响应

状态：成功

参数	参数类型	描述
Body	String	TCP接口返回值

## TCP 响应示例

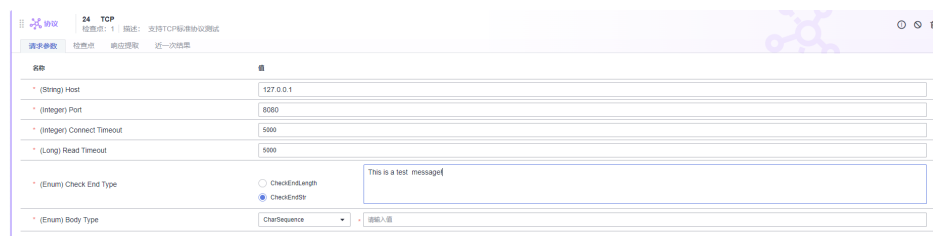
```
{
  "This is a test Message!"
}
```

## 默认检查点

名称	预期值
结果	成功

## TCP 使用示例

填写参数，访问TCP服务：



## 6.4.12.27 在 CodeArts TestPlan 接口脚本中调用 UDP 关键字

### UDP

该系统关键字支持对UDP基础协议进行测试，如果需要使用该系统关键字，需要用户IP开放公网访问并且对应端口的安全组策略开启放行。

参数	是否必选	参数类型	默认值	描述
Host	是	String	-	UDP服务地址
Port	是	Integer	-	UDP服务端口
Check End Type	是	Enum	CheckEndStr	消息结束标识： 数据长度 (CheckEndLength) 结束字符 (CheckEndStr)
Body Type	是	Enum	CharSequence	消息体数据类型： 标准字符串 (CharSequence) 16进制码流 (Hexadecimal CodeStream)

### UDP 响应

状态：成功

参数	参数类型	描述
Body	String	UDP接口返回值

### UDP 响应示例

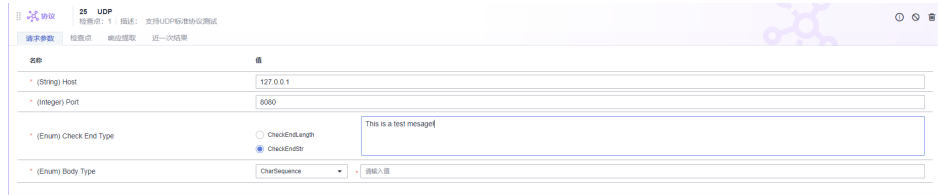
```
{
  "This is a test Message!"
}
```

### 默认检查点

名称	预期值
结果	成功

## UDP 使用示例

- 填写参数，访问UDP服务：



### 6.4.12.28 在 CodeArts TestPlan 接口脚本中调用 WSConnect 关键字

该关键字适用于Websocket客户端连接服务端操作。

参数	是否必选	参数类型	描述
Request Uri	是	String	websocket服务地址
Response Timeout	是	Long	响应超时时间
Header	是	String	请求头
Connect Timeout	是	Integer	连接超时时间

## WSConnect 响应

状态：成功

参数	参数类型	描述
Body	String	WSConnect接口返回值

## WSConnect 响应示例

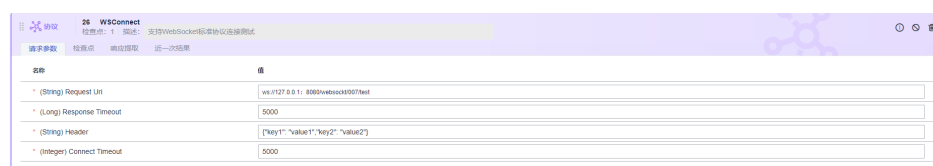
```
{
  "Connect to *** at port *** in time 5000 successfully!"
}
```

## 默认检查点

名称	预期值
结果	成功

## WSConnect 使用示例

填写参数，访问websocket服务：





### 6.4.12.29 在 CodeArts TestPlan 接口脚本中调用 WSRequest 关键字

该关键字适用于Websocket客户端请求服务端操作。

参数	是否必选	参数类型	描述
Request Uri	是	String	websocket服务地址
Response Timeout	是	Long	响应超时时间
Header	是	String	请求头
Request Type	是	String	请求数据类型
Request Body	是	String	请求数据
Response Type	是	String	响应数据类型

### WSRequest 响应

状态：成功

参数	参数类型	描述
Body	String	WSRequest接口返回值

### WSRequest 响应示例

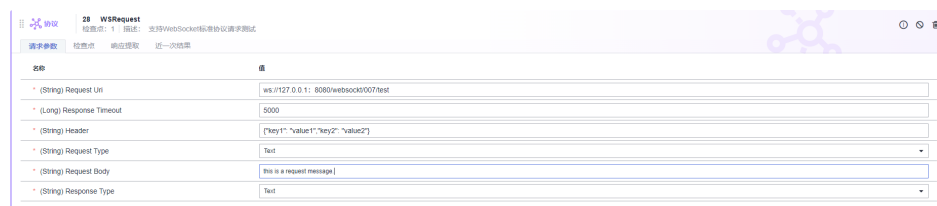
```
{
  "Write data to *** at port *** successfully! response: this is a response message."
}
```

### 默认检查点

名称	预期值
结果	成功

### WSRequest 使用示例

填写参数，访问websocket服务：



### 6.4.12.30 在 CodeArts TestPlan 接口脚本中调用 WsWriteOnly 关键字

该关键字适用于访问Websocket服务端进行只写操作。

参数	是否必选	参数类型	描述
Request Uri	是	String	websocket服务地址
Response Timeout	是	Long	响应超时时间
Header	是	String	请求头
Request Type	是	String	请求数据类型
Request Body	是	String	请求数据

### WsWriteOnly 响应

状态：成功

参数	参数类型	描述
Body	String	WsWriteOnly接口返回值

### WsWriteOnly 响应示例

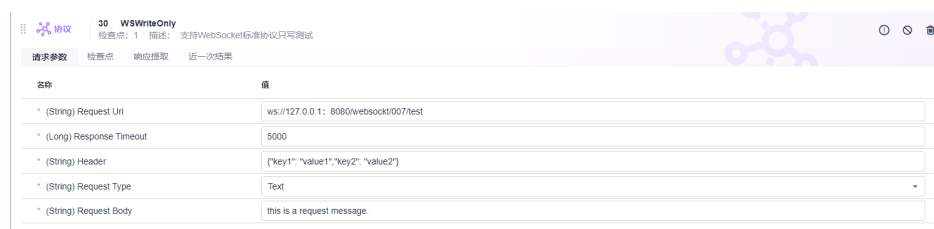
```
{
  "Write data to *** at port *** successfully! response: this is a response message."
}
```

### 默认检查点

名称	预期值
结果	成功

### WsWriteOnly 使用示例

填写参数，访问websocket服务：



### 6.4.12.31 在 CodeArts TestPlan 接口脚本中调用 WSReadOnly 关键字

该关键字适用于访问Websocket服务端进行只读操作。

参数	是否必选	参数类型	描述
Request Uri	是	String	websocket服务地址
Response Timeout	是	Long	响应超时时间
Header	是	String	请求头
Response Type	是	String	响应数据类型

### WSReadOnly 响应

状态：成功

参数	参数类型	描述
Body	String	WSReadOnly接口返回值

### WSReadOnly 响应示例

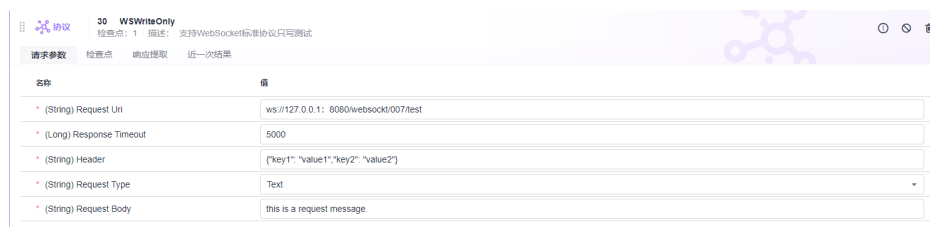
```
{
  "Read data from*** at port *** successfully! response: this is a response message."
}
```

### 默认检查点

名称	预期值
结果	成功

### WSReadOnly 使用示例

填写参数，访问websocket服务：



### 6.4.12.32 在 CodeArts TestPlan 接口脚本中调用 WSDisConnect 关键字

该关键字适用于访问Websocket服务端进行断连操作。

参数	是否必选	参数类型	描述
Request Uri	是	String	websocket服务地址
Response Timeout	是	Long	响应超时时间
Header	是	String	请求头
Status Code	是	String	响应数据类型
Message	是	String	断连信息

## WSDisConnect 响应

状态：成功

参数	参数类型	描述
Body	String	WSDisConnect接口返回值。

## WSDisConnect 响应示例

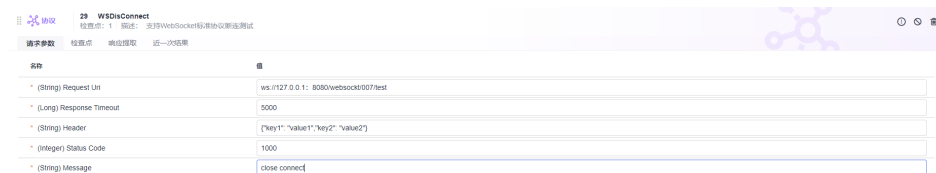
```
{
  "Disconnect to *** at port *** in time 5000 successfully!"
}
```

## 默认检查点

名称	预期值
结果	成功

## WSDisConnect 使用示例

填写参数，访问websocket服务：



### 6.4.12.33 在 CodeArts TestPlan 接口脚本中调用 DubboClient 关键字

#### Dubbo 简介

Apache Dubbo 是一款微服务开发框架，它提供了 RPC通信与微服务治理两大关键能力。使用 Dubbo 开发的微服务，将具备相互之间的远程发现与通信能力，同时利用

Dubbo 提供的丰富服务治理能力，可以实现诸如服务发现、负载均衡、流量调度等服务治理诉求。

## DubboClient 简介

该系统关键字支持对Dubbo协议进行测试，如果需要使用该系统关键字，需要用户将Dubbo服务开放公网访问并且将对应端口的安全组策略开启放行。

参数	是否必选	参数类型	默认值	描述
Dubbo Server IP Address	是	String	-	Dubbo服务地址
Dubbo Server Port	是	Integer	-	Dubbo服务端口
Dubbo operation instruction	是	String	LS	dubbo操作指令，支持：LS，INVOKE。
Dubbo Registration API	是	String	-	Dubbo注册的API名称，常见形式为全限定类名。
Dubbo Registration API	是	String	-	Dubbo注册API对应的方法签名。

## DubboClient 响应

- 状态：成功

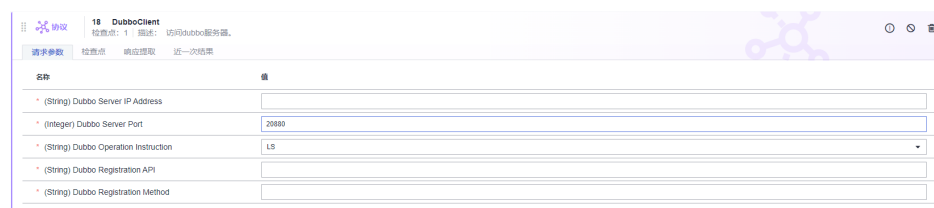
参数	参数类型	描述
Body	String	Dubbo服务端返回值

## 默认检查点

名称	预期值
结果	成功

## DubboClient 使用示例：获取服务器所有注册 API

参数填写示例：



### 响应示例：

测试步骤 | 前置步骤 | 后置步骤

协议 DubboClient (796 ms) 检查点: 1

协议 DubboClient (138 ms) 检查点: 1

协议 DubboClient (143 ms) 检查点: 1

#### 详情日志

测试名: DubboClient (796 ms)

开始时间: 2022/11/26 12:34:22 252 GMT+08:00

状态: ✔ 成功

结果: 成功

响应提取:

检查点: ● 结果是成功

#### 请求

Dubbo IP Address: [REDACTED]

Dubbo Port: 20880

Method: LS

Param: api

#### 响应值

```
[ "com.dubbo.demo.dubbo.api.DemoService" ]
```

## DubboClient 使用示例：获取服务器指定 Service 注册 API

### 参数填写示例：

18 DubboClient

检查点: 1 | 描述: 访问Dubbo服务器。

请求参数 | 检查点 | 响应提取 | 上一次结果

名称	值
* (String) Dubbo Server IP Address	
* (Integer) Dubbo Server Port	20880
* (String) Dubbo Operation Instruction	LS
* (String) Dubbo Registration API	
* (String) Dubbo Registration Method	

### 响应示例：

测试步骤 前置步骤 后置步骤

协议 DubboClient (796 ms) 检查点: 1

协议 DubboClient (138 ms) 检查点: 1

协议 DubboClient (143 ms) 检查点: 1

### 详情日志

测试名: DubboClient (138 ms)

开始时间: 2022/11/26 12:34:23 33 GMT+08:00

状态: ✔ 成功

结果: 成功

响应提取: ● 结果是成功

检查点: ● 结果是成功

### 请求

Dubbo IP Address: [REDACTED]

Dubbo Port: 20880

Method:

Param: LS

api: com.[REDACTED]demo.dubbo.api.DemoService

### 响应值

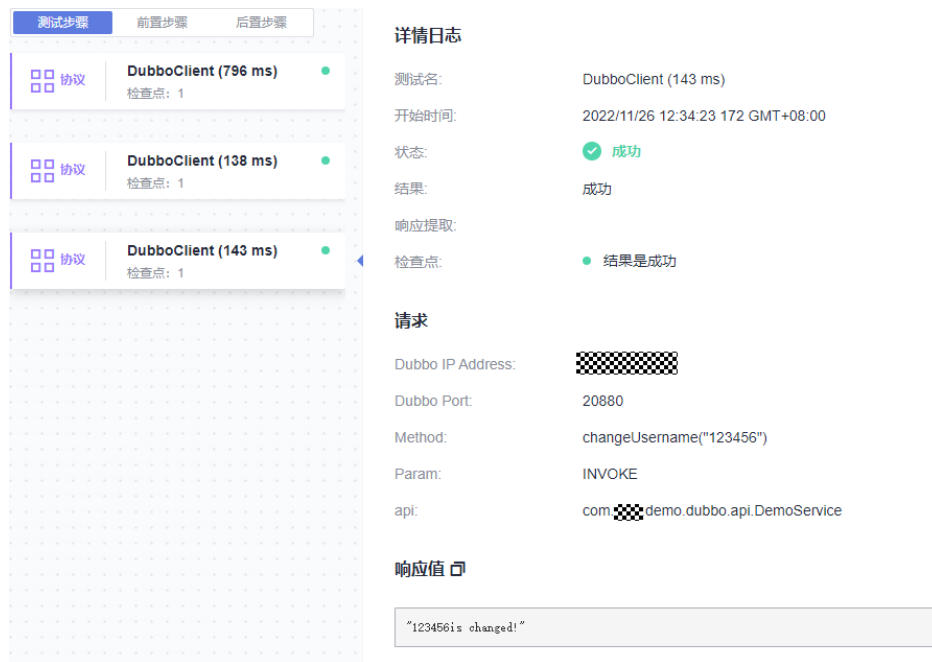
```
[ "changeUsername" ]
```

## DubboClient 使用示例：调用 Dubbo 指定 API

参数填写示例：

名称	值
(String) Dubbo Server IP Address	
(Integer) Dubbo Server Port	20880
(String) Dubbo Operation Instruction	INVOKED
(String) Dubbo Registration API	
(String) Dubbo Registration Method	


响应示例：



## 6.5 执行 CodeArts TestPlan 测试用例

测试用例创建完成后，可以执行用例记录结果。


### 普通执行测试用例

- 步骤1** 登录软件开发生产线首页，搜索目标项目并单击项目名称，进入项目。
- 步骤2** 单击导航栏“测试 > 测试用例”。
- 步骤3** 在对应的测试用例页签，例如手工测试用例单击“手工测试”页签，接口自动化测试用例单击“接口自动化”页签，找到待测试用例，单击“操作”列 , 进入执行页面。  
若需要同时执行多个用例，勾选需要执行的用户例，单击“批量设置结果”。
- 步骤4** 根据测试实际结果设置用例结果。
- 步骤5** 完成设置后单击“确定”，返回用例列表，可以在用例列表“结果”列查看执行结果。
- 步骤6** 单击用例名称，选择“执行历史”页签，可以查看测试用例的执行历史信息。

----结束

### 带参数执行手工测试用例

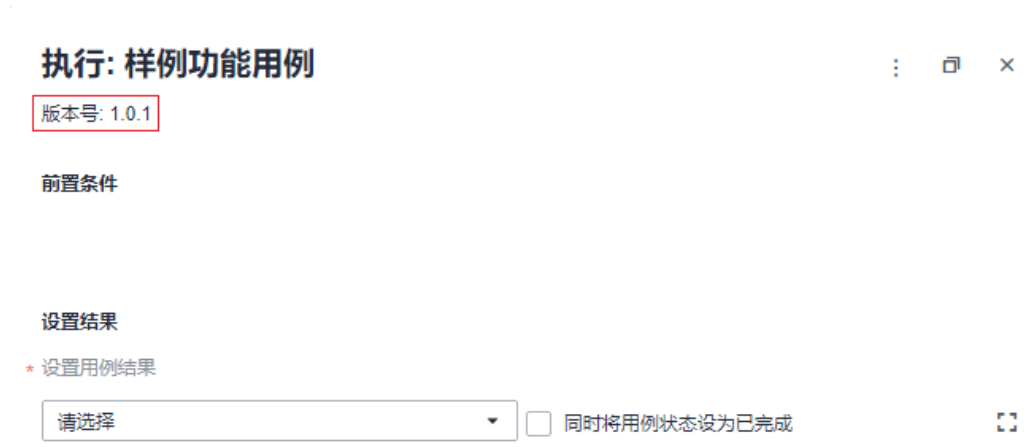
测试用例支持带参数执行。

- 步骤1** 在用例列表单击“操作”列 , 选择“带参数执行”。
- 步骤2** 在弹出框输入版本号，单击“执行”，进入执行页面。





**步骤3** 可以看到测试版本号，然后设置用例执行结果，填写实际结果信息，单击“保存”。




**步骤4** 单击对应的用例，选择“执行历史”页签，即可查询执行的结果信息。

----结束

## 接口自动化用例中导入数据集后多轮次执行

**步骤1** 完成[导入接口测试用例脚本的参数数据集](#)。


**步骤2** 单击导航栏“测试 > 测试用例”。

**步骤3** 单击“接口自动化”页签，找到待测试用例，单击“操作”列 ，开始自动执行。

**步骤4** 执行完成后，可以在用例列表“结果”列查看执行结果。

单击用例名称，选择“执行历史”页签，可以查看该测试用例的执行历史信息。

**步骤5** 用例执行结果卡片展示用例整体的执行结果，在卡片中可以查看执行失败的轮次和数据集总轮次。

**步骤6** 单击卡片中的  图标，可以查看各个轮次数据集的执行情况。

单击下拉栏，可以根据“成功”或“失败”的轮次进行筛选。

**步骤7** 单击某一个轮次，页面将显示该轮次的详细日志信息。

----结束

## 6.6 管理 CodeArts TestPlan 测试用例

测试计划服务支持从本地导入测试用例至用例库，也支持从用例库导出测试用例。可以批量添加测试用例、通过特性目录管理测试用例、测试用例关联需求、评论测试用例、过滤测试用例、自定义用例列表显示列，设置测试用例字段。

### 导入手工测试用例

**步骤1** 登录软件开发生产线首页，搜索目标项目并单击项目名称，进入项目。

**步骤2** 单击导航栏“测试 > 测试用例”。

**步骤3** 单击“手工测试”页签，单击页面右侧“导入”，在下拉菜单中单击“从文件导入”。

也可以单击“全量用例”页签，单击页面右侧“导入”，在下拉菜单中单击“从文件导入”，在弹窗中选择导入用例的执行类型为“手工测试和功能自动化”。

**步骤4** 设置当用例编号重复时，是否覆盖测试用例。

勾选“是”：若用例编号重复，新上传的用例将覆盖相同编号的用例。

勾选“否”：即使用例编号重复，所有用例都会上传至用例列表。

**步骤5** 在弹窗中单击“下载模板”。

按照模板中的格式要求填写用例信息，返回页面，上传编写好的用例文件，单击“确定”。

#### 说明

- 单次导入用例条数不超过5000条。
- 单次文件大小不超过5M。
- 测试计划服务当前支持Excel表格格式。如果数据不符合导入标准，页面提示下载错误报告，请修改数据后重新导入。

----结束

### 导入接口自动化用例

测试计划服务支持通过导入文件的方式生成测试用例，可导入以下类型的文件：

- Postman文件：支持Postman Collection v2.1标准，导入Postman Collection JSON文件。
- Swagger文件：支持Swagger 2.0和3.0标准，导入YAML格式文件。
- Excel文件：参照页面提供的模板编辑Excel格式文件。

#### 导入Postman文件/Swagger文件

每次只能导入一个测试用例。导入的测试用例只能生成测试步骤，不支持生成前置步骤与后置步骤。

**步骤1** 登录软件开发生产线首页，搜索目标项目并单击项目名称，进入项目。

**步骤2** 单击导航栏“测试 > 测试用例”。

**步骤3** 选择“接口自动化”页签，单击页面右侧“导入”，选择“从文件导入”，弹出“从文件导入用例”窗口。

**步骤4** 选择“上传Postman文件”或“上传Swagger文件”。

从本地拖拽文件到窗口，或单击“点击添加文件”并从本地选择文件。单击“下一步”。



**步骤5** 在列表中根据步骤的顺序勾选需要生成用例的条目，单击“保存”。



----结束

## 导入Excel文件

通过Excel文件导入测试用例时，单次导入用例条数不超过500条。

**步骤1** 登录软件开发生产线首页，搜索目标项目并单击项目名称，进入项目。

**步骤2** 单击导航栏“测试 > 测试用例”。

**步骤3** 选择“接口自动化”页签，单击页面右侧“导入”，在下拉菜单中选择“从文件导入”。

也可以单击“全量用例”页签，单击页面右侧“导入”，在下拉菜单中单击“从文件导入”，在弹窗中选择导入用例的执行类型为“接口自动化”。

**步骤4** 选择“上传Excel文件”，单击“下载模板”。



**步骤5** 在本地打开Excel模板，根据模板中标题的批注信息编辑测试用例信息，其中带有\*号列内容是必填内容。

模板中的字段信息如下：

字段	是否必填	说明
用例名称	是	字符长度为1-128个字符，可以包含中文、大小写字母、数字、特殊字符（-_ /*&`^~;; :: () ( ) {}=+,x¥……——’!@#%\$.![]<>?·! ”、 ‘ “ ” ? —。 , 《 》 【 】 ）的组合格式。
用例描述	否	字符长度不超过500个字符。
请求类型	是	只可输入GET、POST、PUT、DELETE四种请求类型。
请求头参数	否	格式为“key=value”。若存在多个参数，则参数之间使用&分割，即“key=value&key1=value1”。
请求地址	是	请求协议支持http和https。格式为“https://ip:port/pathParam?query=1”。
环境分组	否	对应环境参数分组。
IP变量名	否	在对应“环境分组”中生成该变量名，提取“请求地址”内容，生成相应全局变量内容。
请求体类型	否	支持填写raw、json、formdata，分别对应页面中文本、JSON请求体、表单参数三种请求体格式。若未填入，则默认为json。
请求体	否	若请求体类型为“formdata”，则请求体的格式为“key=value”。若存在多个参数，则参数之间使用&分割，即“key=value&key2=value2”。通过Excel导入用例时，“formdata”不支持文件格式请求体。
检查点匹配方式	否	支持精确匹配、模糊匹配。精确匹配代表“相等”，模糊匹配代表“包含”。

字段	是否必填	说明
期望检查点值	否	对应检查点“目标值”。

**步骤6** 保存编辑后的Excel文件，从本地拖拽文件到“从文件导入用例”窗口，或单击“点击添加文件”并从本地选择文件。单击“下一步”。

**步骤7** 查看导入结果：

- 导入成功，列表中将新增用例，新增的用例条数与Excel中编辑的行数相同。
- 导入失败，页面右上角会弹框提示失败。



在“从文件导入用例”窗口中可下载错误列表，参照列表中的错误原因修改Excel文件，重新导入。

----结束

## 导入自定义自动化用例

测试计划服务支持通过导入文件的方式生成测试用例，可导入以下类型的文件：

- Excel文件：参照页面提供的模板编辑Excel格式文件。

通过Excel文件导入测试用例时，单次导入用例条数不超过5000条。

**步骤1** 登录CodeArts首页，搜索目标项目并单击项目名称，进入项目。

**步骤2** 单击导航栏“测试 > 测试用例”。

**步骤3** 选择设置的自定义执行方式名称页签，单击页面右侧“导入”，在下拉菜单中选择“从文件导入”，单击“下载模板”。

**步骤4** 在本地打开Excel模板，根据模板中标题的批注信息编辑测试用例信息，其中带有\*号列内容是必填内容。

模板中的字段信息如下：

字段	说明
用例名称	用例的名称（必填项）。可以描述用例对应的测试场景或功能，字符长度为1-128个字符。
用例编号	用例的编号。字符长度为3-128个字符。

字段	说明
脚本路径	脚本文件在仓库中的相对路径。
处理者	指定完成对应测试用例的人员。
用例状态	用例的状态，可以选择：“新建”，“设计中”，“测试中”，“完成”。
用例结果	用例的结果，可以选择：“成功”、“失败”、“待核查”、“不可用”、“阻塞”或其他自定义结果。
用例类型	用例的类型，可以选择：“功能性测试”，“性能测试”，“兼容性测试”，“易用性测试”，“可靠性测试”，“安全性测试”，“可服务性测试”。
用例级别	请根据用例对应场景或功能的重要程度选择用例等级。 <ul style="list-style-type: none"><li>● 0级：最基本的功能验证，用例不宜过多，各模块尽量保证在10-20个，占比5%左右。</li><li>● 1级：基本功能验证，可用于继承特性的基本特性验证、迭代验收前的基本功能验证等，占比20%左右。</li><li>● 2级：重要特性验证，可用于测试版本（非回归版本）中功能测试，占比60%左右。</li><li>● 3级：一般功能/非重要功能验证，包括对基本/重要功能的异常测试，占比10%~15%左右。</li><li>● 4级：特殊输入、场景、阈值条件的用例，该级别用例不宜过多，占比0%~5%左右。</li></ul>
迭代	根据当前用例需要在哪个迭代进行测试而选择对应迭代。
模块	选择当前用例对应的模块，模块列表来源于项目设置。
需求编号	填写用例需要关联的需求编号。
需求名称	填写用例需要关联的需求名称。
描述	用户根据需要填写相关描述信息。
前置条件	描述执行当前用例所需要的必备前提条件。
归属目录	与特性目录相关联。
测试步骤	录入具体的步骤描述和对应的预期结果。
预期结果	对应检查点“目标值”。

**步骤5** 保存编辑后的Excel文件，从本地拖拽文件到“从文件导入用例”窗口，或单击“点击添加文件”并从本地选择文件。单击“下一步”。

**步骤6** 查看导入结果：

- 导入成功，列表中将新增用例，新增的用例条数与Excel中编辑的行数相同。
- 导入失败，页面右上角会弹框提示失败。

在“从文件导入用例”窗口中可下载错误列表，参照列表中的错误原因修改Excel文件，重新导入。

----结束

## 导出测试用例

**步骤1** 在测试用例对应页签，单击页面右侧“更多”，在下拉菜单中单击“导出”。

**步骤2** 在弹框中选择导出用例范围，可以选择“全部导出”。

选择部分导出(起始位置选择作为表格第一行的用例，结束位置默认为用例表最后一行的用例)，单击“确定”。



**步骤3** 在本地打开导出的excel表格，查看导出的用例内容。

----结束

## 批量添加测试用例

测试计划服务支持从用例库中向测试计划批量添加用例，包括手工测试用例和接口自动化用例。

**步骤1** 登录软件开发生产线首页，搜索目标项目并单击项目名称，进入项目。

**步骤2** 单击导航栏“测试 > 测试用例”。

**步骤3** 单击页面左上方用例库，在下拉列表中选择目标测试计划。

**步骤4** 在“手工测试”或者“接口自动化”，单击页面右侧“导入”，在下拉列表中选择“添加已有用例”。

**步骤5** 在弹框中勾选测试用例，单击“确定”完成测试用例的添加。

- 已存在于测试计划中的用例不能重复添加。
- 支持添加测试计划中需求相关的所有用例。

----结束

## 用例在线评审

用户可以对已创建的测试用例进行评审。

### 新建评审

**步骤1** 登录软件开发生产线首页，搜索目标项目并单击项目名称，进入项目。

**步骤2** 单击导航栏“测试 > 测试用例”。

**步骤3** 选择对应的测试类型页签，单击需要评审用例所在行右侧的图标 $\cdots$ ，单击“新建评审”。

**步骤4** 在弹框内输入配置如下信息后，单击“确定”。

配置项	是否必填	描述
名称	是	新建评审时，名称默认与用例名称保持一致。
用例修改时间	是	新建评审时，时间默认为当前系统日期。
自动评审闭环	是	用户可以勾选是否自动评审闭环。 <ul style="list-style-type: none"><li>选择“是”：表示该用例评审闭环状态会自动置为关闭状态，自动闭环。</li><li>选择“否”：表示该用例评审会走人工审批流程，需要手动闭环。</li></ul>
期望闭环时间	是	当用户将自动评审闭环勾选为否，可以选择期望闭环时间。
评审意见	是	输入不超过1000个字符的评审信息。
指定闭环人	是	当用户将自动评审闭环勾选为否，可以在下拉栏中选择指定闭环人。

### ---结束

### 批量评审

**步骤1** 登录软件开发生产线首页，搜索目标项目并单击项目名称，进入项目。

**步骤2** 单击导航栏“测试 > 测试用例”。

**步骤3** 在测试用例列表中勾选需要批量评审的用例。

**步骤4** 单击下方“批量评审”。

**步骤5** 在弹框中配置如下信息后，单击“确定”。

- 自动评审闭环：选择“是”：表示该用例评审闭环状态会自动置为关闭状态，自动闭环；选择“否”：表示该用例评审会走人工审批流程，需要手动闭环。
- 期望闭环时间：当用户将自动评审闭环勾选为否，可以选择期望闭环时间。
- 评审意见：输入不超过1000个字符的评审信息。



- 指定闭环人：当用户将自动评审闭环勾选为否，可以在下拉栏中选择指定闭环人。

----结束

### 查看评审记录

**步骤1** 登录软件开发生产线首页，搜索目标项目并单击项目名称，进入项目。

**步骤2** 单击导航栏“测试 > 测试用例”。




**步骤3** 单击“评审记录”，完成创建的评审记录将显示在页面中。

**步骤4** 单击评审记录页面中的搜索栏，可以添加筛选条件。

**步骤5** 输入筛选条件的关键字，可以查找对应的评审记录。

**步骤6** 对未闭环的评审记录，可以进行删除、编辑和闭环操作。

用例名称	用例编号	测试...	评审意见	评审人	评审时间	指定闭环人	实际闭环人	闭环状态	闭环...	期望闭环	操作项
登录学...	11706	web...	继续删...	devclo...	2024/05/13 1...	devcloud_...	--	未闭环	--	2024/0...	  

- 单击未闭环的评审记录操作项中的 ，依照提示信息单击“确认”，删除此评审记录。
- 单击未闭环的评审记录操作项中的 ，在“编辑评审”弹窗中编辑评审内容。
- 单击未闭环的评审记录操作项中的 ，在“闭环评审”弹窗中，选择“是否闭环”选项中的“是”，单击“确定”。

----结束

## 需求与测试用例

### 测试用例关联需求



#### 说明

测试用例仅可关联Scrum项目Epic/Feature/Story类型的工作项及看板项目默认的“需求”类型工作项。

测试计划服务支持测试用例与需求相关联，操作步骤如下：

**步骤1** 登录软件开发生产线首页，搜索目标项目并单击项目名称，进入项目。

**步骤2** 单击导航栏“测试 > 测试用例”。

**步骤3** 选择需要关联需求的测试用例，单击所在行右侧的图标 ，选择“关联需求”。或者在“全量用例”页面中，单击需要关联需求的用例所在操作列中的  图标，进行关联需求。

若需要将多个用例与同一个需求相关联，则可在列表中勾选需要关联需求的测试用例，单击页面下方“批量关联需求”。

**步骤4** 在弹窗中勾选需要关联的需求，可以选择当前计划下的需求或在所有需求中选择。单击“确定”完成关联。



#### ----结束

### 添加需求相关的测试用例

- **前置条件**

测试计划下的需求在用例库下已关联了测试用例。

添加需求相关的测试用例的步骤与**批量添加测试用例**相同，在弹出框中勾选“选择本测试计划中的需求相关的所有用例”即可。

### 按需求管理测试用例

在“测试用例”页面，单击页面左侧“需求目录”。

- 已关联的需求都默认归属于“需求目录”下。
- 单击“需求目录”下的某个需求，可以查看与该需求关联的所有用例。
- 单击需求名称右侧的 $\dots$ ，可以查看需求详情，也可以新建与所选需求关联的测试用例。

### 需求变更通知

当某个需求关联了测试用例且在需求管理服务中对该需求做出修改时，“测试用例”页面的对应需求名称会出现红点，提醒对此需求所关联用例做补充或修改。

## 缺陷与测试用例

测试用例执行失败时，通常会将该用例与缺陷相关联。可以新建缺陷，或者关联已存在的缺陷。

本节以手工测试用例为例介绍如何将测试用例与缺陷相关联。



### 📖 说明


测试用例仅可关联Scrum项目“Bug”类型的工作项及看板项目默认的“缺陷”类型工作项。

**步骤1** 登录软件开发生产线首页，搜索目标项目并单击项目名称，进入项目。

**步骤2** 单击导航栏“测试 > 测试用例”。

**步骤3** 选择需要关联缺陷的测试用例。在“测试用例列表”页面操作，可通过以下方式创建/关联缺陷。

- 单击操作列 ，关联当前项目下已有的缺陷。
- 单击操作列 ，选择“新建并关联缺陷”，并根据页面提示创建对应缺陷。
- 打开单个“测试用例”进行新建并关联缺陷。  
单击测试用例名称，页面中选择“缺陷列表”，单击“新建并关联缺陷”进行操作。

**步骤4** 缺陷创建或关联完成后，可以在测试用例详情页面的“缺陷列表”页签查看缺陷信息。单击  可以解除当前缺陷。

----结束

## 评论测试用例

测试计划服务支持对测试用例进行评论。

**步骤1** 登录软件开发生产线首页，搜索目标项目并单击项目名称，进入项目。

**步骤2** 单击导航栏“测试 > 测试用例”。

**步骤3** 选择需要评论的测试用例，单击用例名称，选择“详情”页签。

**步骤4** 在页面下方“评论”框中输入评论内容，单击“保存”。

保存成功的评论将显示在评论框下方。

----结束

## 过滤测试用例

测试计划服务支持自定义过滤测试用例，本节以“测试用例 > 手工测试”页面为例，介绍如何过滤测试用例。

### 使用默认过滤条件

**步骤1** 登录软件开发生产线首页，搜索目标项目并单击项目名称，进入项目。

**步骤2** 单击导航栏“测试 > 测试用例”。

**步骤3** 在“手工测试”页签，单击列表上方“所有用例”或“全部”2个下拉列表，在列表中选择需要的过滤条件。

- 所有用例：显示当前测试计划或用例库中所有的用例。
- 我的用例：显示“处理人”是当前登录用户的所有用例。
- 未关联测试套：显示在“测试执行”中，未关联任何测试套件的用例。

----结束

用户可以单击新建用例右键的两个下拉栏，根据“所有用例”或“我的用例”、“全部”或“未关联测试套”进行用例筛选。

### 设置高级过滤条件

当默认过滤条件不满足过滤需要时，可以添加自定义过滤条件。

**步骤1** 登录软件开发生产线首页，搜索目标项目并单击项目名称，进入项目。

**步骤2** 在导航栏单击“测试 > 测试用例”。

**步骤3** 单击用例列表上方的“高级过滤”，页面将显示常用的过滤条件。

**步骤4** 根据需要设置过滤条件，单击“过滤”，页面中将显示符合过滤条件的用例列表。

也可以单击“保存当前条件并过滤”，在弹框中输入过滤器名，单击“确定”。保存后的过滤器将会显示在“所有用例”下拉列表中。

**步骤5** （可选）若高级过滤条件仍不满足需求，可以单击“添加筛选条件”，根据需要在下拉列表中选择过滤项，单击该过滤项，完成添加。过滤项将显示在页面中，重复**步骤4**操作，完成过滤。高级过滤支持过滤自定义字段。


----结束

## 自定义用例列表显示列

测试计划支持自定义列展示测试用例表格，本节以测试用例下的手工测试用例为例介绍具体操作。

**步骤1** 登录软件开发生产线首页，搜索目标项目并单击项目名称，进入项目。

**步骤2** 单击导航栏“测试 > 测试用例”。

**步骤3** 在“手工测试”页面，单击表格最后一列 ，在弹出框中可以配置显示的列和隐藏的列，并可以拖拽排列显示顺序。用例列表支持配置自定义字段。


----结束

## 搜索用例

测试计划服务支持按用例名称、编号或描述进行搜索。

**步骤1** 完成创建测试用例。

**步骤2** 在用例列表上方的搜索框内输入用例名称、编号或描述的关键字。

**步骤3** 单击 。

**步骤4** 搜索到的用例将显示在用例列表中。

----结束

# 7 创建与执行 CodeArts TestPlan 测试套件

## 7.1 创建 CodeArts TestPlan 测试套件

测试套件用于将测试用例分配给指定的测试执行者。

### 前提条件

- 具有若干个已经创建的测试用例。

### 新建手工测试套件

**步骤1** 登录软件开发生产线首页，搜索目标项目并单击项目名称，进入项目。

**步骤2** 在导航栏单击“测试 > 测试执行”。默认显示“用例库”和“基线”。

**步骤3** 单击“用例库”旁的▼，根据需要选择测试计划。

**步骤4** 在“手工测试”页签，单击页面左上方“新建套件”，进入新建页面。

**步骤5** 在新建手工测试套件页面中，配置以下基本信息。

配置项	是否必填	描述
名称	是	套件的名称，名称长度为1-128个字符。
计划周期	否	选择测试套件开始时间和结束时间。
描述	否	对套件的简要描述，描述信息最大长度为500个字符。
标签	否	单击文本框，输入标签信息即可生成标签，最大可添加10个标签，用空格分隔。
编号	否	可以自定义用例编号，编号长度为3-128个字符；若不输入，则服务自动生成编号。
模块	否	可选择当前项目下已创建的模块，请参考 <a href="#">模块设置</a> 。
版本号	否	单击下拉栏，可输入长度不少于3个字符的版本号。

配置项	是否必填	描述
处理者	否	单击下拉栏，选择当前套件的处理器。

**步骤6** 单击“添加用例”或“立即添加”，勾选待测试的测试用例后单击“确定”，单击“保存”完成手工测试套件创建。

用户可以在“添加测试用例”弹框中的搜索框内输入用例名称、编号、描述等字段，也可以输入自定义字段，进行目标用例的搜索。

----结束

## 新建接口自动化套件

测试计划服务提供了接口测试套创建，用例串并行执行策略，测试用例分块加速能力，有效提高资源池利用率，减少任务阻塞情况。并且可提前拦截产品缺陷，更加快速地发现问題。

**步骤1** 登录软件开发生产线首页，搜索目标项目并单击项目名称，进入项目。


**步骤2** 在导航栏单击“测试 > 测试执行”。

**步骤3** 单击“用例库”旁的▼，根据需要选择测试计划。

**步骤4** 单击“接口自动化”页签，单击页面左上方“新建套件”，进入新建页面。

**步骤5** 在新建接口自动化测试套件页面中，配置以下基本信息。

配置项	是否必填	说明
测试套名称	是	套件的名称，名称长度为1-128个字符。
前置用例	否	准备测试阶段需要的前置条件数据，如果没有前置条件，可不添加前置用例。 单击“立即添加”，可以在用例列表中选择前置用例。
后置用例	否	释放或恢复准备阶段和测试阶段测试数据的，以还原测试环境，如果没有待恢复的数据，可不添加后置步骤。 单击“立即添加”，可以在用例列表中选择后置用例。
资源池	是	通过资源池，用户可以接入自己的执行资源，在执行任务时，可以选择接入的资源池中的代理机来执行任务，提高任务执行效率，不再依赖软件开发生产线公共执行资源。 单击“资源池管理”，可以查看用户已创建的资源池，可参考 <a href="#">新建资源池</a>
编号	否	用户可以输入长度为3-128个字符的编号。
标签	否	根据需要对当前任务设置标签，以空格分隔，每个任务最多可关联10个标签。
模块	否	选择当前测试套件对应的模块，模块列表来源于项目设置，请参考 <a href="#">模块设置</a> 。



配置项	是否必填	说明
处理者	否	指定完成对应测试任务的人员。
版本号	否	在下拉栏中，选择在“测试计划详情”、“用例详情”设置的版本号。
描述	否	输入对接口测试套件的简要描述，最大500个字符。
环境参数	否	环境参数可以用于整个项目，如测试步骤的参数、检查点、变量、URL等都可以引用环境参数。单击  图标，可新建环境参数。
用例最长执行时间	是	设置每个用例的最长执行时间，超过时间，用例则超时失败。任务继续执行，直到最后一个用例执行完毕，最大不超过10分钟。
测试套参数覆盖	否	可以在此对测试套参数进行高级配置，可以针对测试套件中的全局参数，进行参数覆盖，最多对3条全局参数进行覆盖。
定时类型	是	包括两种类型，默认仅执行一次： <ul style="list-style-type: none"> <li>仅执行一次：测试套只执行一次。</li> <li>周期性重复执行：设置一个执行频率，测试套按照这个频率周期重复执行。按天级设置。</li> </ul>
任务开始时间	是	包括两种模式，默认立即执行： <ul style="list-style-type: none"> <li>立即执行：运行后立即开始执行。</li> <li>指定开始时间：指定某个时刻后开始执行。</li> </ul>
执行顺序	是	包括两种模式，默认串行： <ul style="list-style-type: none"> <li>串行：接口测试套件中的用例将串行执行。</li> <li>并行：接口测试套件中的用例将并行执行。</li> </ul>

**步骤6** 完成基础信息配置后，单击“添加用例”或“立即添加”，勾选待执行的测试用例。

用户可以在“添加测试用例”弹框中的搜索框内输入用例名称、编号、描述等字段，也可以输入自定义字段，进行目标用例的搜索。

**步骤7** 根据需要完成执行设置，单击“保存”完成接口自动化套件创建。

**步骤8** 根据需要对已创建的接口测试套件进行以下操作：

- 修改测试套件：单击测试套件名称，可以重新配置该套件的基本信息。
- 删除测试套件：单击对应测试套件所在操作列中的  图标，单击“删除”可以删除该测试套件。在测试套件列表中，勾选多个测试套件，单击页面下方“删除”，可以批量删除测试套件。
- 拷贝测试套件：单击对应测试套件所在操作列中的  图标，单击“拷贝测试套件”，新套件名称将以“复制的套件名称\_copy\_当前时间戳”命名，用户也可以对新拷贝的套件重命名，输入相关描述，单击“确定”，完成拷贝测试套件。

----结束

## 7.2 执行 CodeArts TestPlan 测试套件

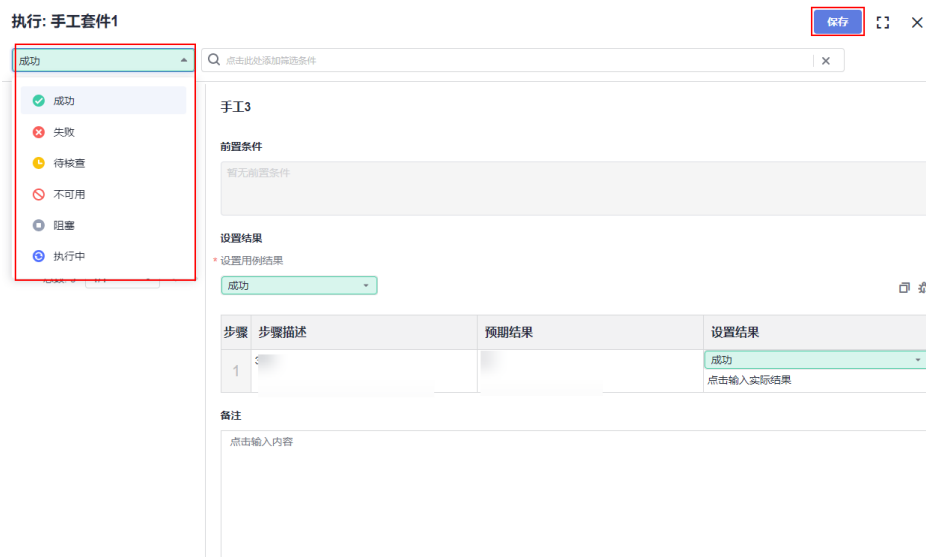
测试执行阶段中执行测试套件，检查被测系统是否符合测试套件预期结果，记录测试结果，发现产品问题及缺陷并帮助开发人员分析定位问题。

### 前提条件

具有已经创建的测试套件。


### 执行手工测试套件

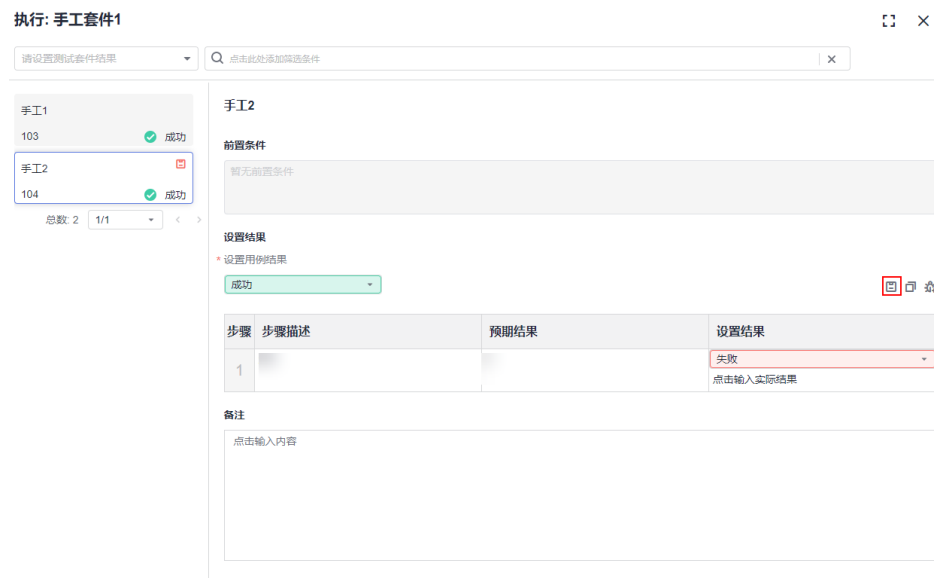
- 步骤1** 登录软件开发生产线首页，搜索目标项目并单击项目名称，进入项目。
- 步骤2** 在导航栏单击“测试 > 测试执行”。默认显示“用例库”。
- 步骤3** 单击“用例库”旁的▼，根据需要选择测试计划。
- 步骤4** 在“手工测试”页签，找到待执行测试套件，单击“操作”列中的▶。
- 步骤5** 进入执行手工测试套件页面，设置用例各步骤结果、描述及测试用例结果（用户可以切换套件内的用例并重复该步骤，直到手工测试套件内的用例全部执行完成）。
- 步骤6** 设置手工测试套件结果，单击右上角“保存”按钮。





## 说明

设置用例结果或切换用例会自动保存结果，若保存用例结果后修改步骤结果、描述或用例执行备注，可单击图标保存。



**步骤7** 执行完成后，可以在用例套件列表“执行历史”列查看执行结果。

----结束

## 带参数执行手工测试套件

手工测试套件支持带参数执行。

**步骤1** 在用例套件列表单击“操作”列，选择“带参数执行”。

**步骤2** 在弹出框输入版本号，单击“执行”，进入执行页面。



**步骤3** 在版本号下的下拉框中，根据实际执行结果，设置测试套件执行结果，填写实际结果信息，单击“保存”。



**步骤4** 单击对应的测试套件，选择“执行历史”页签，即可查询执行的结果信息。

----结束

## 批量执行手工测试套件

**步骤1** 在用例套件列表单击“操作”列 $\cdots$ ，选择“批量执行”。

**步骤2** 勾选需要执行的测试用例，单击页面下方“批量设置结果”。

**步骤3** 在弹框中可以设置用例状态以及用例结果。

**步骤4** 单击“确定”。

----结束

## 执行接口自动化套件

**步骤1** 登录软件开发生产线首页，搜索目标项目并单击项目名称，进入项目。

**步骤2** 在导航栏单击“测试 > 测试执行”。默认显示“用例库”。

**步骤3** 单击“用例库”旁的 $\nabla$ ，根据需要选择测试计划。

**步骤4** 找到待执行测试套件，单击“操作”列中的 $\triangleright$ ，开始自动执行，执行完成后，可通过“执行历史”列查看执行结果。

单击操作列中的 $\odot$ 图标可以停止执行接口自动化套件。

**步骤5** 在测试套件列表，单击“操作”列 $\odot$ ，进入“执行历史”页签，可以查看测试套件历次执行的用例执行详情信息。

----结束

# 8 查看与评估 CodeArts TestPlan 测试质量

## 8.1 查看 CodeArts TestPlan 测试质量看板

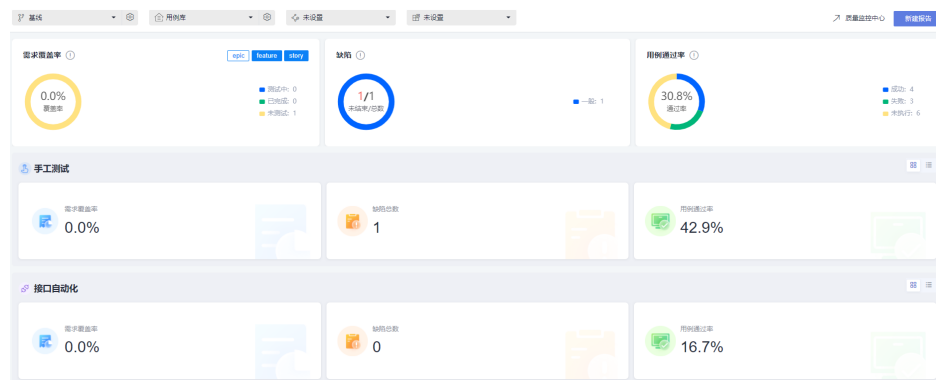
CodeArts TestPlan提供需求覆盖率、需求通过率、用例执行率、遗留缺陷指数等10多个质量度量指标，支持功能、性能、可靠性等维度的质量评估，用户可以通过看板清晰掌握整体测试计划的进度和风险。



### 项目级仪表盘

项目级仪表盘展示了用例库和测试计划对应的需求覆盖率、缺陷、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。

#### 用例库的质量报告

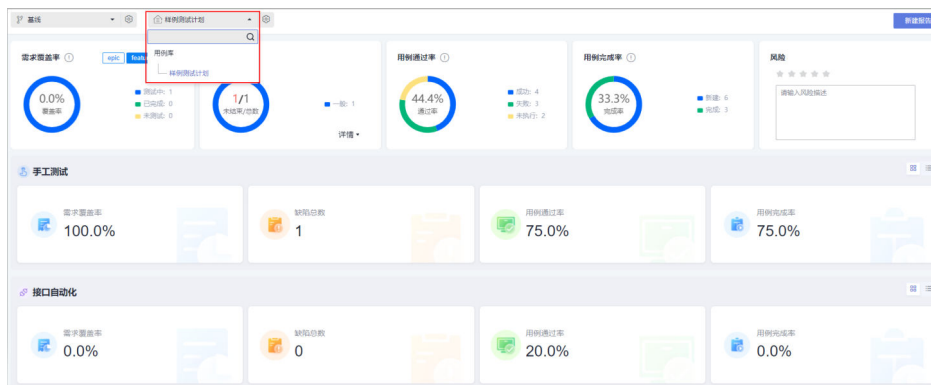
登录软件开发生产线首页，搜索并进入目标项目。在导航栏单击“测试 > 测试质量看板”，默认进入用例库“质量报告”页面。



报告项	说明
需求覆盖率	<p>需求覆盖率反映功能点测试覆盖情况，统计选中的迭代和模块下所有需求的测试覆盖率。</p> <ul style="list-style-type: none"> <li>● <b>未测试</b>：需求没有关联测试用例，或者关联的每个测试用例状态均未完成。</li> <li>● <b>测试中</b>：需求关联的测试用例的状态有一部分为未完成。</li> <li>● <b>已完成</b>：需求关联的测试用例的状态均为完成。</li> <li>● <b>需求覆盖率=已完成需求数 / 需求总数。</b></li> </ul> <p><b>说明</b> 在Scrum项目类型中，默认仅统计Feature和Story类型的需求，如需要统计Epic类型的需求，请选中Epic。</p>
缺陷	<p>统计选中的迭代和模块下未结束的缺陷数和缺陷总数，并按缺陷重要程度分组统计。</p> <p><b>说明</b> 未解决的缺陷数包含以下情况：</p> <ul style="list-style-type: none"> <li>● Scrum项目下为状态属于“开始态”和“进行态”的缺陷数。</li> </ul>
用例通过率	<p>用例通过率和缺陷综合反映产品质量，统计选中的迭代和模块下所有用例的通过率，并按执行结果分组统计，未执行用例计入未执行分组。</p> <p><b>用例通过率=执行结果字段为成功的用例数 / 用例总数。</b></p>
手工测试	<p>统计选中的迭代和模块下，手工用例关联的需求的覆盖率和缺陷总数，手工用例的通过率和完成率。</p>
接口自动化	<p>统计选中的迭代和模块下，接口自动化用例关联的需求的覆盖率和缺陷总数，接口用例的通过率和完成率。</p>
缺陷列表	<p>显示选中的迭代和模块下用例关联的缺陷列表，单击手工测试或接口自动化后面的  即可查看，可通过缺陷名称和编号进行过滤，单击缺陷名称可跳转到缺陷详情页面。</p>
用例列表	<p>显示选中的迭代和模块下的用例列表，单击手工测试或接口自动化后面的  ，即可查看，可通过用例名称和编号进行过滤，单击用例名称可跳转到用例详情页面。</p>

### 测试计划的质量报告

在“测试质量看板”页面，单击左上角用例库旁边的下拉箭头，选择一个测试计划，进入测试计划质量报告页面。



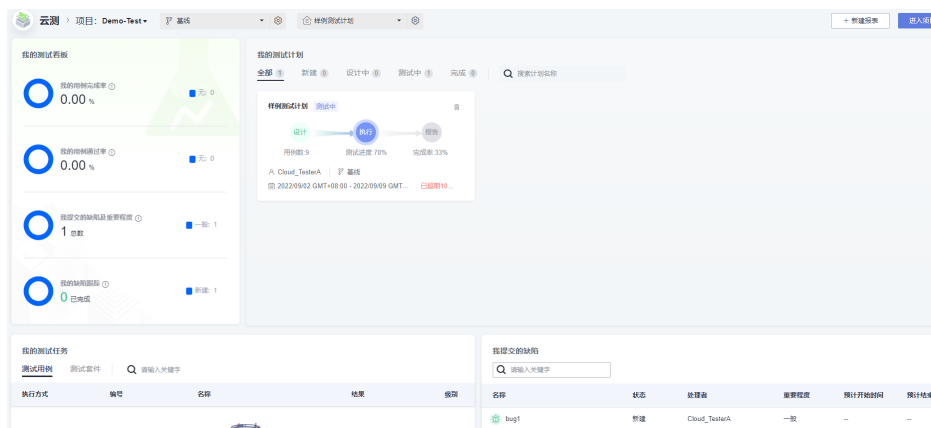
测试计划的质量报告与用例库的质量报告相比，不按迭代和模块过滤，增加用例完成率的统计报告和测试计划风险描述，其它报告项同用例库的质量报告。

报告项	说明
用例完成率	完成率反映测试活动的进度，统计选中的计划下所有用例的完成率，并按用例的状态分组统计。 用例完成率=用例处于完成状态的个数/用例总数
风险	测试计划的风险。可以评定测试计划的风险等级并添加风险描述。
手工测试	手工测试用例的需求覆盖率、缺陷总数、用例通过率、用例完成率。
接口自动化	接口自动化用例的需求覆盖率、缺陷总数、用例通过率、用例完成率。

## 个人级仪表盘

登录软件开发生产线首页，在导航栏单击“服务 > 测试计划”，进入个人仪表盘界面，展示处理者为当前登录用户的统计数据。

- 按项目中的用例库和测试计划展示我的用例完成率、我的用例通过率、我提交的缺陷及重要程度、我的缺陷跟踪的统计信息。
- 按项目中的用例库和测试计划展示我的测试计划、我的测试任务、我提交的缺陷列表信息。



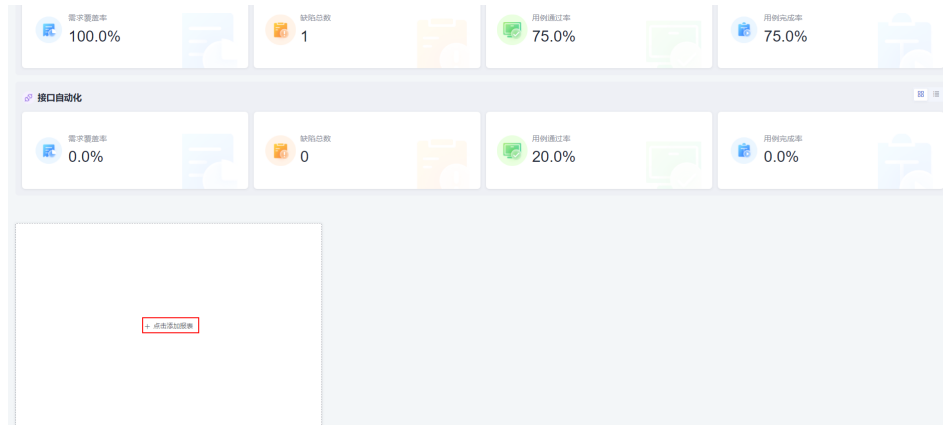
## 自定义测试报表

测试计划服务可以通过以下两个入口创建自定义测试报表：

- 通过首页创建
  - a. 登录软件开发生产线首页，在导航栏单击“服务 > 测试计划”，进入首页。
  - b. 单击右上角“新建报表”，或在左下角当前项目自定义测试报表中“点击添加报表”。
  - c. 弹出“选择报表类型”页面，可以根据需要选择要创建的报表类型。



- 通过质量看板页面创建
  - a. 登录软件开发生产线首页，搜索目标项目并单击项目名称，进入项目。
  - b. 在导航栏单击“测试 > 测试质量看板”，进入用例库“测试质量看板”页面。
  - c. 选择用例库或某一个测试计划，单击页面左下方“点击添加报表”，可以根据需要选择要创建的报表类型。



### 选择报表类型

- 自定义报表
- 测试用例
- 测试套件

#### 自定义报表

自定义报表  
可创建自定义报表

#### 测试用例

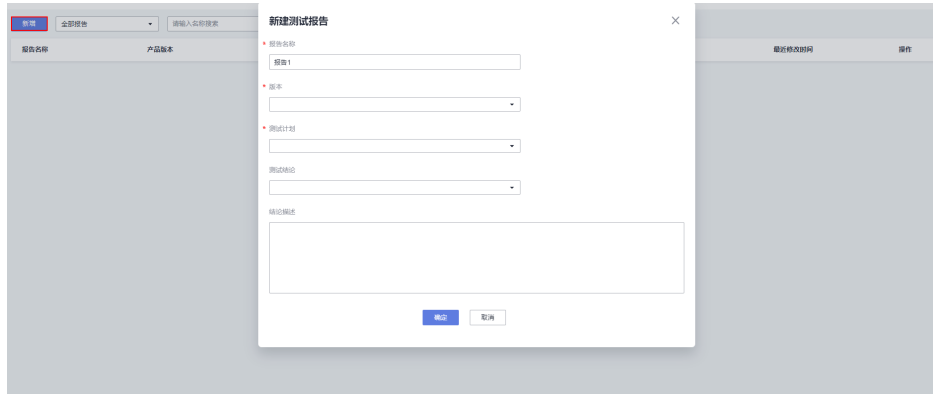
- 用例按模块分布  
按模块查看用例的分布
- 用例按处理者分布  
按处理者查看用例的分布
- 用例按级别分布  
按级别查看用例的分布
- 用例按结果分布  
按结果查看用例的分布
- 用例按状态分布  
按状态查看用例的分布

## 8.2 评估 CodeArts TestPlan 测试质量

**步骤1** 登录软件开发生产线首页，搜索并进入目标项目。

**步骤2** 在导航栏单击“测试 > 测试质量评估”，进入“测试质量评估”页面。

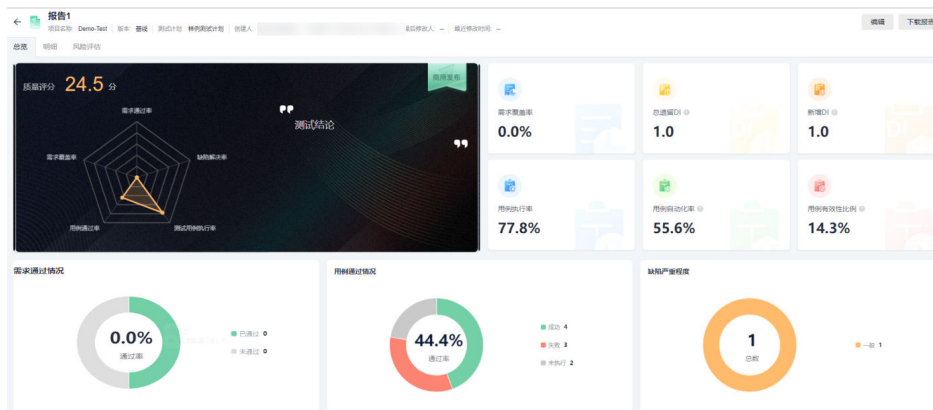
**步骤3** 单击页面左上角“新建”，输入报告名称，选择版本、测试计划、测试结论，输入结论描述，单击“确定”完成报告的创建。



----结束

## 测试报告总览

单击报告名称，默认进入测试报告的总览页面。总览页面展示了所选测试计划的相关测试数据情况。



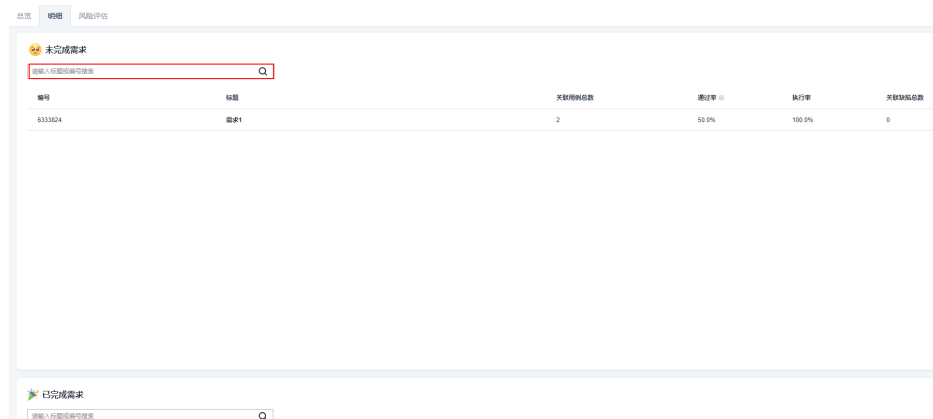
报告项	说明
需求覆盖率	需求覆盖率反映功能点测试覆盖情况，统计所选计划关联的所有需求 的测试覆盖率。需求覆盖率 = 已完成数/需求总数。
总遗留DI	统计此版本所有的遗留缺陷所算出的DI值。 <b>说明</b> 所有缺陷按照严重等级进行计算。每个缺陷DI值对应类型如下：提示：0.1 一 般：1 严重（重要）：3 致命（关键）：10
新增DI	统计此测试计划所关联的遗留缺陷所算出的DI值。
用例执行率	统计用例的执行情况。用例执行率 = 计划内执行有执行结果的用例数 量/计划内所有用例数量。
用例自动化率	非手工测试用例占有所有用例数的比例，用例自动化率=（测试计划所 有的用例总数-测试计划内手工测试用例数）/测试计划所有的用例总 数。

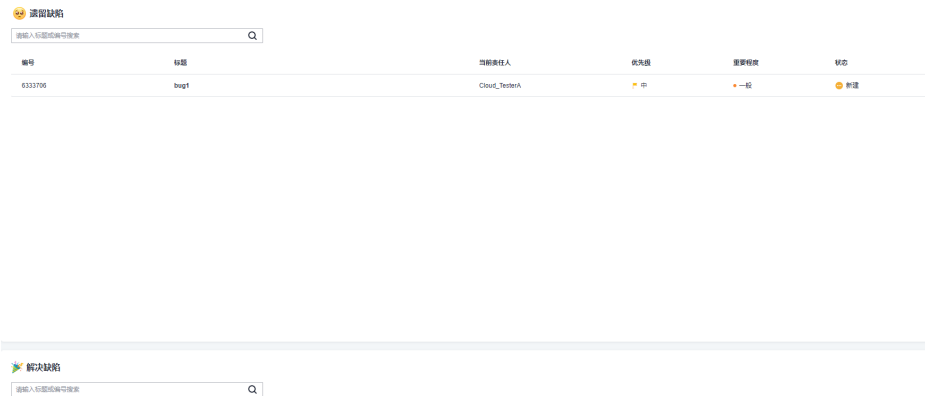


报告项	说明
用例有效性比例	发现缺陷的测试执行用例占有所有测试执行用例的百分比，用例有效性比例=发现缺陷的测试执行用例数量/测试执行用例数量。
需求通过情况	需求所关联的测试用例结果都为成功时，当前需求为已通过。需求通过率=已通过需求数/需求总数。
用例通过情况	用例通过率反映产品质量，统计选中的计划下所有用例的通过率，并按用例的结果分组统计，未执行的用例计入“未执行”分组。用例通过率=成功数/总数
缺陷严重程度	显示选中的计划下关联的缺陷数量。
自动化率	通过饼图以及详细的用例数量进行展示。 非手工测试用例占有所有用例数的比例，用例自动化率=(测试计划所有的用例总数-测试计划内手工测试用例数)/测试计划所有的用例总数。
各测试类型的用例通过率	显示不同测试类型的用例通过率。
缺陷按照模块分布情况	按照模块显示对应的缺陷数量。

## 测试报告明细

用户在“明细”界面中，可以查看测试计划下需求、缺陷的遗留和完成情况。  
在搜索栏内输入标题或编号，可以查找对应的需求或缺陷。



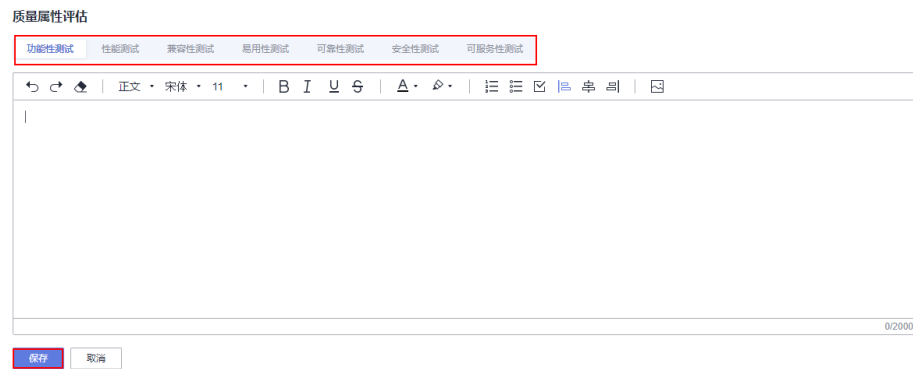


## 风险评估

用户可以在风险评估界面中，配置“质量属性评估”和“风险分析”信息，也可以自定义模块输入相关报告信息。

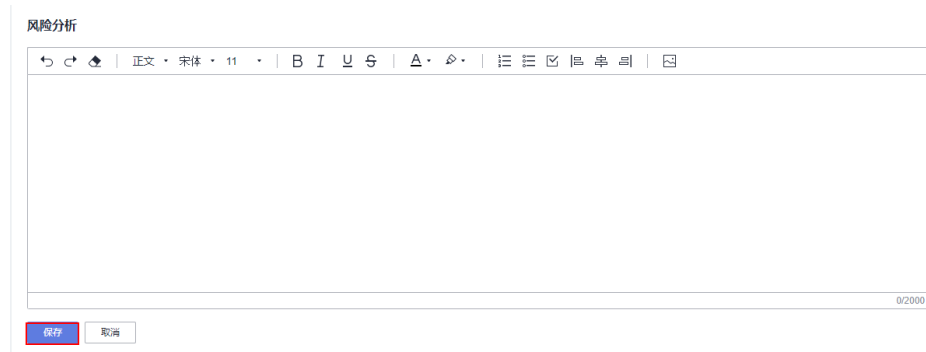
- 质量属性评估

在顶部选择测试类型，单击文本框区域，输入评估信息，单击“保存”完成创建。



- 风险分析

用户可以根据测试进展评估风险，单击文本框区域，输入风险分析的信息，单击“保存”完成创建。





- 自定义信息

单击“新建自定义信息”，新增一条信息模块。

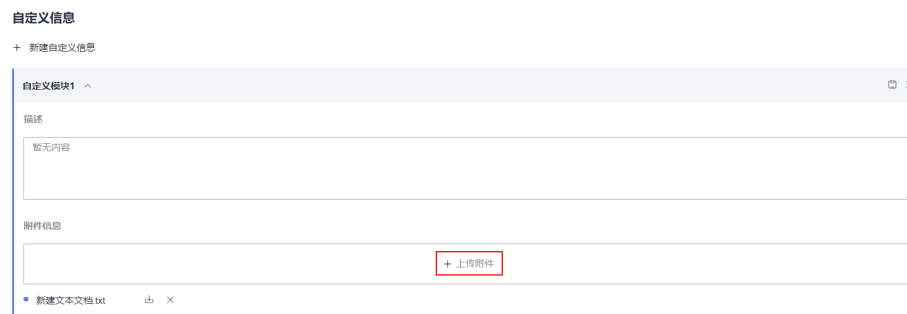
## 自定义信息

+ 新建自定义信息

单击模块名称可以重新定义模块标题，输入相关描述，单击  完成保存。单击  可以将新建的模块删除。



单击“上传附件”，可以将本地文件上传至自定义模块中。

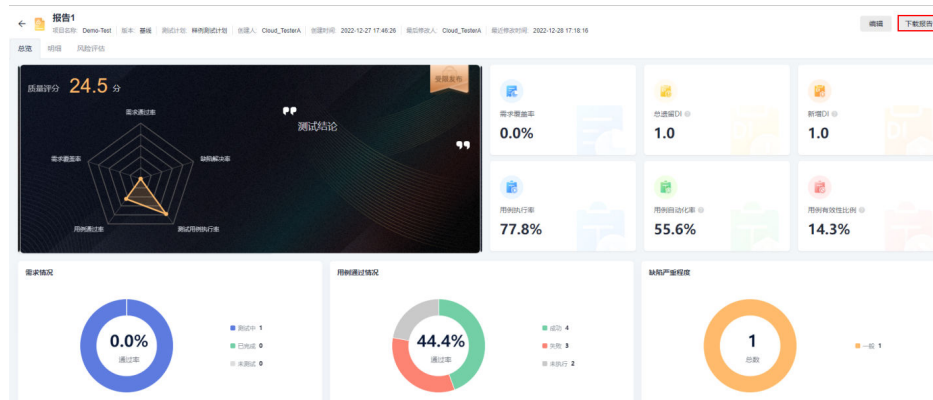


### 说明

- 1.上传单个文件的大小不能超过10M。
- 2.最多只能上传15个附件。

## 下载报告

单击页面右上方“下载报告”，可以将创建好的测试评估报告以PDF格式下载到本地。



# 9 配置 CodeArts TestPlan 全局配置项

## 9.1 配置 CodeArts TestPlan 发送消息通知

在测试计划服务中，可以对各项操作设置是否发送通知。

**步骤1** 登录软件开发生产线首页，搜索目标项目并单击项目名称，进入项目。

**步骤2** 在导航栏单击“设置 > 测试设置”。

**步骤3** 在“消息通知”页签，可以对系统事件配置是否发送服务动态和发送邮件。

表 9-1 通知发送策略

操作类型	操作者本人是否接收通知	通知策略
分配用例处理者	否	用例处理者
更新用例	否	项目经理、测试经理、用例创建者、用例处理者
删除用例	否	项目经理、测试经理、(如果不存在项目经理、测试经理，则给项目创建者发送)，用例处理者
删除目录	否	项目经理、测试经理、(如果不存在项目经理、测试经理，则给项目创建者发送)
新建套件	否	测试套件处理者
更新套件	否	测试套件创建者，测试套件处理者
删除套件	否	项目经理、测试经理、(如果不存在项目经理、测试经理，则给项目创建者发送)，测试套件处理者
新建质量报表	否	项目经理、测试经理、(如果不存在项目经理、测试经理，则给项目创建者发送)

操作类型	操作者本人是否接收通知	通知策略
更新质量报表	否	项目经理、测试经理、(如果不存在项目经理、测试经理, 则给项目创建者发送), 自定义报表创建者
删除质量报表	否	项目经理、测试经理、(如果不存在项目经理、测试经理, 则给项目创建者发送), 自定义报表创建者
编辑风险信息	否	项目经理、测试经理、(如果不存在项目经理、测试经理, 则给项目创建者发送), 测试计划创建者, 测试计划处理者
新建计划	否	项目经理、测试经理、(如果不存在项目经理、测试经理, 则给项目创建者发送), 测试计划处理者
更改计划	否	项目经理、测试经理、(如果不存在项目经理、测试经理, 则给项目创建者发送), 测试计划创建者, 测试计划处理者
删除计划	否	项目经理、测试经理、(如果不存在项目经理、测试经理, 则给项目创建者发送), 测试计划创建者, 测试计划处理者
执行手工用例	否	用例创建者, 用例处理者
计划完成后发送质量报告	是	测试计划设置页该项设置的自定义通知者 ( 如果未设置, 则给项目经理、测试经理发送 )
评论通知	否	用例详情评论被@的用户
接口测试用例执行完成	是	接口自动化用例执行者
接口测试套件执行完成	是	接口自动化测试套件执行者

----结束

## 9.2 配置 CodeArts TestPlan 功能开关

**步骤1** 登录软件开发生产线首页, 搜索目标项目并单击项目名称, 进入项目。

**步骤2** 在导航栏单击“设置 > 测试设置”。

**步骤3** 单击“功能开关”页签, 根据需要打开或关闭以下功能的开关。

- 需求更新提醒: 需求有更新时, 在测试用例页面的需求目录中对应的需求名称右侧出现红点, 单击红点可查看需求更新历史, 具体可参考[测试用例关联需求](#)。

- 导入Swagger文档时使用接口定义的以下字段作为脚本模板名称：脚本模板名称默认使用operationId，可切换选择summary。
- 测试设计分步骤生成用例：通过思维导图的方式生成用例，一个步骤对应一个预期结果，具体可参考[按照节点步骤生成预期结果](#)。

----结束

## 9.3 管理 CodeArts TestPlan 项目成员

登录并访问过测试计划服务页面的项目成员，将会显示在测试计划服务用户列表中。项目创建者可以根据需要管理用户列表。

**步骤1** 登录软件开发生产线首页，搜索目标项目并单击项目名称，进入项目。

**步骤2** 在导航栏单击“设置 > 测试设置”。

**步骤3** 在“用户管理”页签展示使用过测试计划服务的用户，项目创建者可以根据需要删除用户管理下的用户。

----结束

## 9.4 设置 CodeArts TestPlan 测试用例字段



如果预置的用例字段不满足实际使用需求，用户可以按需自定义添加其它类型字段。

**步骤1** 登录软件开发生产线首页，搜索目标项目并单击项目名称，进入项目。

**步骤2** 在导航栏单击“设置 > 测试设置”。

**步骤3** 选择“测试用例设置”页签，页面将显示测试用例下的字段信息。

**步骤4** 用户可以根据需要完成以下字段配置：

- 单击操作列，可以修改部分字段的名称，添加或删除选项、打开或关闭“是否显示”或“是否必填”。
- 用户可以在设置页面中打开或关闭“是否显示”或“是否必填”。
- 用户可以单击对应字段默认值的下拉栏，选择用例创建后该字段显示的默认值。
- 单击操作列可以删除对应的字段。

----结束

### 自定义字段配置



用户可以根据需要设置自定义字段，设置成功的字段将显示在测试用例的详情中。

**步骤1** 在测试用例设置中，单击“新增字段”，最大添加25个自定义字段。

**步骤2** 输入“字段名称”、“描述”。选择“字段类型”，打开或关闭“是否显示”以及“是否必填”，单击“确定”。

- 单行文本：只能输入一行字段，最大长度不超过100个字符。
- 多行文本：可以输入多行字段，最大长度2000个字符。
- 单选列表：用户可以在字段中单选设置的信息，单击“增加选项”完成配置，最大添加10个选择。

- 多选列表：用户可以在字段中多选设置的信息，单击“增加选项”完成配置，最大添加10个选择。
- 日期时间：用户可以设置日期和时间。
- 日期：用户可以设置日期。
- 整数：用户可以在字段中输入整数，可以输入-9999999999到9999999999范围内的值。
- 小数：用户可以在字段中输入小数，最大支持小数点后两位。
- 单选用户：在下拉栏中选择用户。
- 多选用户：在下拉栏中可以勾选多个用户。

**步骤3** 新建成功的字段显示在列表中，单击可以重新编辑对应的字段，单击可以删除对应的字段。

----结束

## 9.5 设置 CodeArts TestPlan 测试套件的状态和结果

如果预置的测试套件状态和结果不满足实际使用需求，用户可以按需自定义添加其它状态和结果。

**步骤1** 登录软件开发生产线首页，搜索目标项目并单击项目名称，进入项目。

**步骤2** 在导航栏单击“设置 > 测试设置”。

**步骤3** 在“测试套件设置”页签，单击“添加状态”或“添加结果”，根据需要配置新的套件状态与结果。

----结束

## 9.6 设置 CodeArts TestPlan 测试报告背景图和 Logo

**步骤1** 登录CodeArts首页，搜索目标项目并单击项目名称，进入项目。

**步骤2** 在导航栏单击“设置 > 测试设置”。

**步骤3** 单击“封面背景图”、“内容背景图”以及“设置Logo”右侧的“替换图片”，在本地选择需要替换的图片。

- 封面背景图：推荐上传JPG或PNG格式，推荐尺寸794px\*1123px。
- 内容背景图：推荐上传JPG或PNG格式，推荐尺寸794px\*1123px。
- 设置Logo：推荐上传JPG或PNG格式，推荐限制宽度≤90px、高度≤40px。

**步骤4** 上传完成后，单击“保存设置”。

**步骤5** 用户可以[下载报告](#)查看重新配置的风格样式。

----结束



## 9.7 设置 CodeArts TestPlan 接口自动化用例请求超时时间、资源池和 DNS 映射

用户配置DNS映射后，当接口自动化用例中的URL请求路径为域名时，会自动映射请求对应的IP。

用户可以在“测试计划设置”中设置请求超时时间、调试资源池、设置DNS映射。

**步骤1** 登录CodeArts首页，搜索目标项目并单击项目名称，进入项目。

**步骤2** 在导航栏单击“设置 > 测试设置”。

**步骤3** 单击“其他”页签，在资源池中选择接口用例调试所需的目标资源池（DEFAULT是默认的公共资源池）。

**步骤4** 根据需要设置“请求超时时间”，默认显示“10000”ms。

**步骤5** 根据需要设置“DNS映射”。

- 域名：用于标识一个网站的唯一标识符。
- IPS：用于标识一个设备或主机的位置，由4段0~255的数字组成，中间用"."连接。

----结束