

代码托管 操作指南 产品文档





【版权声明】

©2013-2023 腾讯云版权所有

本文档(含所有文字、数据、图片等内容)完整的著作权归腾讯云计算(北京)有限责任公司单独所有,未经腾讯云事先明确书面许可,任何主体不得以任何形式 复制、修改、使用、抄袭、传播本文档全部或部分内容。前述行为构成对腾讯云著作权的侵犯,腾讯云将依法采取措施追究法律责任。

【商标声明】

# 🔗 腾讯云

及其它腾讯云服务相关的商标均为腾讯云计算(北京)有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标,依法由权利人所有。未经腾讯云及有关 权利人书面许可,任何主体不得以任何方式对前述商标进行使用、复制、修改、传播、抄录等行为,否则将构成对腾讯云及有关权利人商标权的侵犯,腾讯云将依 法采取措施追究法律责任。

【服务声明】

本文档意在向您介绍腾讯云全部或部分产品、服务的当时的相关概况,部分产品、服务的内容可能不时有所调整。 您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定,除非双方另有约定,否则,腾讯云对本文档内容不做任何明示或默示的承 诺或保证。

【联系我们】

我们致力于为您提供个性化的售前购买咨询服务,及相应的技术售后服务,任何问题请联系 4009100100。



文档目录

操作指南
开通服务
主账号
子用户
权限说明
Git 仓库
创建仓库
导入或关联外部仓库
浏览仓库文件
修改远程仓库地址
管理仓库
恢复已删除仓库
管理代码仓库卡片
通过本地命令行管理仓库
模板仓库
提交日期格式
代码对比
SVN 仓库使用
创建 SVN 仓库
访问 SVN 仓库
管理 SVN 目录权限
SSH 协议
配置 SSH 公钥
密钥指纹鉴权
通过 SSH 协议推拉代码
分支管理
创建分支
设置默认分支
设置保护分支
设置隐藏分支
设置只读分支
代码所有者
合并请求与代码评审
合并请求设置
发起合并请求
评审合并请求
版本与标签
代码版本
代码标签
代码仓库安全
配置权限方案
检查仓库安全风险
调整访问限制
使用 GPG 签名 commit 记录
提交者及提交注释验证
锁定文件
限制推送文件大小与仓库容量
文件推送限制
仓库规范
Git 通识
常用命令



LFS 大文件支持 Go get 支持

操作指南 开通服务



# 主账号

最近更新时间: 2023-01-05 18:12:51

## 操作场景

本文为您介绍腾讯云主账号如何开通并使用 CODING DevOps 服务,您可以按照以下几种情况有选择性操作。

## 主账号未开通服务

此场景适用于未曾在腾讯云中使用 CODING DevOps 服务的用户,请访问 CODING DevOps 控制台 并按照提示完成策略权限授权后,开通 CODING 服 务。

←	角色	<b>管理</b>
服	<b>發授权</b>	
Q	意赋予 C	ODING DevOps 权限后,将创建服务预设角色并授予 CODING DevOps 相关权限
角	色名称	CODING_QCSRole
角	色类型	服务角色
角	色描述	当前角色为 CODING DevOps 服务角色,该角色将在已关联策略的权限范围内访问您的其他云服务资源。
授	ē权策略	预设策略 QcloudAccessForCODINGRole①、预设策略 QcloudAccessForCODINGRoleInThroughTCR①、预设策略 QcloudAccessForCODINGRoleInAccessTKE①、预设策略 QcloudSLSFullAccess①、预设策略 QcloudSSLFullAccess①、预设策略 QcloudSSLFullAccess①、预设策略 QcloudAccessForCODINGRoleInCD①、预设策略 QcloudAccessForCODINGRoleInCD①、预设策略 QcloudSSLFullAccess①、预设策略
	同意授	

#### 🕛 说明

腾讯云主账号服务开通后,您还需要 CODING 账号才能正常使用服务。

#### 未注册过 CODING 账号

1. 若您未注册过 CODING 账号,请单击没有 CODING 账号,去开通。

概览	
	开通 CODING DevOps 服务 CODING DevOps 涵盖了软件开发从构想到交付的一切所需,使研发团队在云端高效协同, 实践敏捷开发与 DevOps,提升软件研发效能。了解详情 已有 CODING 账号,去关联 没有 CODING 账号,去开通

2. 按照提示填写邮箱、验证码、密码进入下一步,输入团队名称和域名后完成团队创建。



获取验证码			
Ø			
夺号。			
NG 服务协议》			
-	获取验证码	获取验证码 ダ <sup>好号</sup> 。 NG服务协议》	获取验证码

## 关联已有 CODING 账号

若您有印象曾使用过 CODING 服务,但并非通过腾讯云控制台入口注册,那么可以选择将腾讯云账号与已有 CODING 账号相关联。请单击控制台中的**已有** CODING 账号,去关联。





输入已注册的 CODING 账号完成关联操作,关联完成后将继承原有 CODING 团队数据。

← 关联 CODING 账号	
验证需要关联的 CODING 账号	没有 CODING 账号?去开通
请输入邮箱	
请输入邮箱地址	
验证码	获取验证码
下一步	

## 主账号已开通服务

单击 控制台 中的立即使用或团队域名,跳转至 CODING 团队页后,开始使用 CODING 服务。



# 子用户

最近更新时间: 2021-11-24 10:04:09

## 操作场景

子用户是由主账号创建的实体,有确定的身份 ID 和身份凭证,能够登录并独立设置控制台,且具有 API 访问策略。若您希望通过腾讯云子用户使用 CODING DevOps 服务(以下简称服务),下文将按照多种场景进行指引。

- 主账号已开通服务,需帮助子用户开通服务。
- 主账号未开通服务,需使用子用户协助主账号开通服务。
- 子用户已开通服务,但部分功能受限。

### 主账号已开通服务

- 1. 主账号登录 CODING DevOps 控制台。
- 2. 单击快速创建子用户。

概览		
8	前往 CODING DevOps 团队域名 立即使用   立即选购	
快速创建子用户		
点击以下按钮快速创建拥有 COI 快速创建子用户	DING DevOps 权限的子用户。	

 在快速新建用户页面填写子用户相关信息,访问方式请同时勾选编程访问及腾讯云控制台访问,密码设置可选自动生成或自定义密码,单击创建用户后即可创 建出拥有 CODING DevOps 策略的子用户。



• 为保障	目户登录使用用户名,不支持中文,用户 章子账号的账户安全,未完善手机信息的	<sup>12</sup> 名一经确定将无法更改。 在创建用户后,您可以重 的子账号在登录时将被要求绑定和验证手机	酒看并下载密钥等相关信息		
2置用户信息•	用户名 ★	备注	手机	邮箱	
	test		中国大陆(+86) *		
	新增用户 (单次最多创建101	个用户)			
5问方式•	<ul> <li>✓ 编程访问 启用SecretId和SecretKey,支持// ● 請讯云控制台访问 启用密码,允许用户登录到腾讯Z</li> </ul>	腾讯云API、SDK和其他开发工具访问 云控制台			
空制台密码•	<ul> <li>自动生成密码</li> <li>自定义密码</li> </ul>				
需要重置密码	✔ 用户必须在下次登录时重置密码				
可接收消息类型	<ul> <li>✓ 財务消息 ①</li> <li>✓ 产品消息 ③</li> </ul>				

## ▲ 注意

新建子用户时,请务必勾选访问方式的编程访问和腾讯云控制台访问,避免子用户无法登录 CODING DevOps 控制台的情况。

#### 4. 创建完成后,使用子用户登录控制台,单击加入团队。

概览		
您的主账号已开通 CODING 服务,您可加入团团队域名 https:// C	L.	
CODING DevOps 简介	援 〉 🖾 测试 〉 🞑 交付	
<b>构想</b> 从您的产品需求构想开始,即可在这里发起并进行全过程 管理,分解需求、设计产品、直至需求开发完成落地。在 这里,团队成员还可通过团队 Wiki、文件共享等工具进	文件网盘     全部文件     Q = 世景全部文件     + 新建 ●     2 上信       全部文件     文件名 ○     创建者 ○     最后       ● 最近文件     可用发三部产品设计     陈江     5 分付       ▲ 星标文件     ●     研发三部产品设计     陈江     5 分付       ▲ 星标文件     ●     体验报告     陈江     5 分付       ● 回收站     ●     ○     CODING Enterprise Logo.png V2 ★     陈泽药…       ●     快捷访问     ●     ○     送代与事务规划通知和动态.txt     彭可     昨天	20.32



#### 按照提示填写账号信息后加入团队。

加入主账号团队	
<b>填写账号信息</b> 加入主账号团队前,请填写以下信息。	
请输入邮箱	
验证码	获取验证码
请设置成员名	
便于团队内成员快速识别,设置后如需任	修改,请联系团队管理员。
下一步	

### 主账号未开通服务

#### 使用主账号直接开通服务

请联系主账号拥有者,参见 <mark>主账号</mark>,了解如何使用主账号直接开通服务。

#### 使用子用户帮助主账号开通服务

- 1. 子用户需前往 访问管理控制台,确认拥有 AdministratorAccess 权限策略,然后前往 CODING DevOps 控制台 > 概览页面。
- 2. 单击**开通服务**,进入主账号开通服务流程。

#### 概览

<b>开通 CODING</b> CODING DevOps 涵語 高效协同,实践敏捷牙 开通服务	<b>DevOps 服务</b> <sup>盖了软件开发从构 F发与 DevOps,排</sup>	<b>5</b> 想到交付的一切所需,使研发团队在 是升软件研发效能。 <mark>了解详情</mark>	王云述
CODING DevOps 简介 [文] 构想 〉 [公] 计划 〉 [五]	◎ 开发 〉	运 测试 > 🛕	交付
	文件网盘	<b>全部文件</b> □ Q = 搜索全部文件 + 1 □ 文件名 ≎	新建 →
构想	<ul> <li>● 最近文件</li> <li>★ 星标文件</li> <li>&lt; 分享中的</li> </ul>	研发三部产品设计	<ul><li>陈江 5 分钟前</li><li>陈江 5 分钟前</li></ul>
从您的产品需求构想开始,即可在这里发起并进行全过程 管理,分解需求、设计产品、直至需求开发完成落地。在 这里、团队成员还可通过团队 Wiki、文件共享等工具进	<ul><li>□ 回收站</li><li>◆ 快捷访问 ~</li></ul>	<ul> <li>CODING Enterprise Logo.png <sup>V2</sup> ★</li> <li>送代与事务规划通知和动态.txt</li> </ul>	陈泽芮 今天 20:32 彭可 昨天 20:32



#### 输入主账号邮箱与验证码完成开通。

← 开通 CODING 服务			
开通主账号 CODING 服务			
您的主账号暂未开通 CODING DevOp	s 服务,您可先帮主账号开通再加入团	团队。同时主账号将成为团队所有者	
请输入主账号邮箱			
验证码	获取验证码		
下一步			

○ 若主账号邮箱不存在,需进入主账号创建流程,输入密码与团队名称后开通账号。

	@ .com		9	
请输入密	母		ø	
支持 6~32	位的字符组合。			
团队名称				
支持3~32位	的中英文组合			
https://	团队域名	.coding	et	
支持3~32位		开头。注册后不可修改。		
.持3~32位	的字母数字组合,需以字母 云我已阅读并同章《COF	开头。注册后不可修改。 NING 服务协议》		
<b>干通即表</b> ;	N我已阅读开问总《COL			
开通即表; 确定				
开通即表 确定				

○ 若主账号邮箱已关联多个 CODING 团队,选择需要的团队即可。



	开通 CODING 服务
	选择 CODING 团队
	您输入的邮箱已有 CODING 账号,请选择要关联的团队。
	请选择 CODING 团队 ▼
	· 腾云扣钉 · · · · · · · · · · · · · · · · · · ·
	Washington
	腾讯云 (1) (1) (1) (1) (1) (1) (1) (1) (1) (1)
	Balbala
	zh zhott
	十 新建团队
<del>(</del>	开通 CODING 服务
	开通成功
	成日成功为王城与开通 CODING 服务。总可加入王城与包队开成为包队官连风。

子用户已开通服务



子用户能否继续创建新的子用户,取决于 访问管理控制台 中所选择的策略。

<ul> <li>         ·</li></ul>	前账号下全部资源的全部访问权限, 前账号下除去访问管理(CAM)、 iss 即可 前账号下全部资源的只读访问权限,	请单选 Admir 费用中心以外的 请单选 Read(	nistratorAccess 即可 D全部资源访问权限,请单选 OnlyAccess 即可	
新建自定义策略			支持搜索策略名称/描述	Q
策略名	描述	引用次数	策略类型 ▼	
AdministratorAccess	该策略允许您管理账户内所	2	预设策略	
ReadOnlyAccess	该策略允许您只读访问账户	0	预设策略	
QCloudResourceFullAccess	该策略允许您管理账户内所	0	预设策略	
QCloudFinanceFullAccess	该策略允许您管理账户内财	0	预设策略	
QcloudAAFullAccess	活动防刷(AA)全读写访	0	预设策略	
QcloudABFullAccess	代理记账(AB)全读写访	0	预设策略	
持按住 shift 键进行多选 自助管理 API 密钥 🕣 🦳 自助	管理 MFA 设备 (j) 确定	取消		

当子用户具备 AdministratorAccess 策略时,控制台将出现快速创建子用户,意味着该子用户能够创建新的子用户。





若子用户仅具备 QcloudCODINGFULLAccess ,控制台中将不再出现快速创建子用户,意味着该子用户仅具备服务的访问策略,无法继续新建子用户。

概览	
8	<b>前往 CODING DevOps</b> 团队域名 https://tt1123.coding.net ☑ 立即使用
CODING DevOps 简介	



# 权限说明

最近更新时间: 2021-08-25 15:26:36

在使用腾讯云 CODING Devops 产品的过程中,为了能够调取相关云资源,会遇到需要进行服务授权的场景。在使用该产品的过程中主要涉及 CODING\_QCSRole 角色,本文档会展示此角色所包含的预设策略详情。

## CODING\_QCSRole 角色

开通 CODING Devops 服务后,腾讯云会授予您的账户 CODING\_QCSRole 角色的权限。该角色默认关联多个预设策略,对应策略会出现在该角色的已授 权策略列表中。 CODING\_QCSRole 角色默认关联的预设策略包含如下:

- QcloudAccessForCODINGRole : CODING 服务对云资源的访问权限。
- QcloudAccessForCODINGRoleInAccessTKE : CODING 服务所需要的容器服务(TKE)权限。
- QcloudAccessForCODINGRoleInThroughTCR : CODING 服务所需要的容器镜像(TCR)权限。

#### 预设策略 QcloudAccessForCODINGRole

#### 授权场景

当您已注册并登录腾讯云账号后,首次登录 CODING 控制台 时,需前往"访问管理"页面对当前账号赋予 CODING DevOps、访问管理(CAM)等权限。

#### 授权步骤

1. 首次登录 CODING 控制台 时会弹出服务授权窗口。

2. 单击前往访问管理,进入角色管理页面。

3. 单击同意授权,完成身份验证后即可成功授权。

#### 权限内容

• 访问管理相关

权限名称	权限说明
cam:ListPolicies	查询策略列表

#### • 容器镜像服务相关

权限名称	权限说明
tcr:DescribeInstances	查询实例信息
tcr:CreateInstanceToken	创建实例访问凭证
tcr:DescribeNamespaces	查询命名空间信息
tcr:DescribeRepositories	查询镜像仓库信息
tcr:DescribeRepositoryOwnerPersonal	查询个人版所有仓库

#### 预设策略 QcloudAccessForCODINGRoleInAccessTKE

#### 授权场景

开通 CODING Devops 服务并完成 CODING\_QCSRole 角色授权后,该策略将会与 CODING\_QCSRole 角色相关联,完成操作后即可获得腾讯云容器服务 (TKE) 云资源的相关权限。

#### 授权步骤

该策略与预设策略 QcloudAccessForCODINGRole 同时授权,无需额外操作。

#### 权限内容

权限名称

权限说明



ccs:DescribeCluster	查询集群列表
ccs:CreateClusterEndpoint	创建集群访问端口
ccs:CreateClusterEndpointVip	创建托管集群外网访问端口
ccs:ModifyClusterEndpointSP	修改托管集群外网端口的安全策略
ccs:DescribeClusterEndpointVipStatus	查询托管集群开启外网端口流程状态

## 预设策略 QcloudAccessForCODINGRoleInThroughTCR

#### 授权场景

开通 CODING Devops 服务并完成 CODING\_QCSRole 角色授权后,该策略将会与 CODING\_QCSRole 角色相关联,完成操作后即可获得腾讯云容器镜像 服务(TCR)云资源的相关权限。

#### 授权步骤

该策略与预设策略 QcloudAccessForCODINGRole 同时授权,无需额外操作。

#### 权限内容

#### 容器镜像服务相关:

权限名称	权限说明
tcr:CreateWebhookTrigger	创建触发器
tcr:ModifyWebhookTrigger	更新触发器
tcr:DeleteWebhookTrigger	删除触发器
tcr:DescribeWebhookTriggerLog	获取触发器日志
tcr:PushRepository	推送镜像
tcr:PushRepositoryPersonal	个人版推送镜像



# Git 仓库 创建仓库

最近更新时间: 2023-02-07 14:51:27

本文为您详细介绍如何创建 Git 代码仓库。

## 进入项目

- 1. 登录 CODING 控制台,单击团队域名进入 CODING 使用页面。
- 2. 单击团队首页左侧的项目,进入项目列表页,选择目标项目。
- 3. 选择左侧菜单代码仓库,进入代码仓库首页。

### 手动创建仓库

1. 登录团队后,单击首页左侧的代码仓库中的创建代码仓库进行创建。



2. 代码仓库无法独立存在,需从属于某个项目。两者的对应关系为一个项目对应多个代码仓库。



		•	<ul> <li>← 创建代码仓库 普通创建 模板创建</li> </ul>	
e	工作台	*	所属项目 *	
	项目		所属项目	
	项目集		Q 搜索	
功能	模块		P         newone         下划线(_)、中划线(-)和点(.)的组合         0/1	00
	代码仓库		● 经典项目类型	
	知识管理	Beta	● 度量测试	
()	仪表盘		test0001     g	
	工作负载			- //
	<b>工</b> 作 久報	Beta	12 项目协同测试	
	「「人」の主	Deta	123	
	日你		添加 .gitignore 文件	
Ũ	应用中心		◎ 添加分支模型(仓库创建后将根据所选模型创建分支)	
9	基础设施		是否开源	
8	全部事项		● 私有仓库(仅对仓库成员可见,仓库成员可访问仓库)	
L	全部制品		○ 公开仓库(公开后,任何人都可以访问代码仓库,请谨慎考虑!)	
۶.	Cloud Studio		仓库规范	
守田	商日			
	编辑菜单			
٠	通知		完成创建 取消	
•	团队设置中心			

3. 仓库类型选择 Git,输入符合条件的仓库名称。

4. 创建代码仓库时还可以选择是否启用 仓库规范,用以快速制订团队开发工作流。



← 创建代码仓库 = =	全部仓库规范方案 4	
所属项目 * 所属项目 仓库类型 * 仓库名称 *	<b>单分支规范 系统</b> 支持 master 分支	分支类型
◆ GIT 仓库 ~         仓库名称只支持           仓库描述	<b>功能分支规范 系统</b> 支持 master、feature/* 分支	分支类型  分支名匹配规则  描述
请输入仓库描述	Gitflow 分支规范 系统 支持 master、develop、feature/*、 release/*、hotfux/* 分支	暂无数据 ○ か许创建规定分支类型以外的分支
<ul> <li>✓ 生成 README 文件</li> <li>□ 添加 .gitignore 文件</li> </ul>	GitLab 分支规范 系统 支持 master、feature/*、*-stabl e、environment/* 分支	合并方向 规范仓库分支间的合并方向,只允许创建列表中规定方向的合并请求,列表为空则不会对仓库中的合并请求方向做限制。
<ul> <li>添加分支機型(仓库创建后将根据</li> <li>是否开源</li> <li>私有仓库(仅对仓库成员可见,仓</li> </ul>		<b>游分支</b> 图示 暂无添加任何合并方向限制,允许创建任意分支间的合并请求
<ul> <li>公开仓库(公开后,任何人都可以</li> <li>仓库规范</li> <li>+</li> </ul>		
完成创建取消		确 认 取 消

模板创建仓库



CODING 提供数个预置开发框架的代码仓库模块,助您快速体验代码仓库时如何与持续集成或构建产物进行关联。

← 创建代码仓库 普通	创建 模板创建	
所属项目 *		
所属项目	-	
仓库名称 *		
仓库名称只支持字母、数字、下划线(	)、中划线(–)和点(.)的组合	0/100
仓库描述		
请输入仓库描述		
选择模板 *         预置模板         自定义模板           Spring-demo         基于简单的 Java 网页应用。	<b>ruby-on-rails-demo</b> 图LLS 基于简单的 Ruby on Rails 网 页应用。	<b>ruby-sinatra-demo</b> 基于简单的 Ruby Sinatra 网页 应用。
express ngde 基于简单的 Node.js 网页应 用。	<b>android-demo</b> 基于简单的 Android APP 应 用。	Flask flask-demo Flask 基于简单的 Python Flask 网页 应用。
<b>是否开源</b> <ul> <li>私有仓库(仅对仓库成员可见,仓库</li> <li>公开仓库(公开后,任何人都可以认</li> <li>完成创建</li> <li>取消</li> </ul>	成员可访问仓库) 访问代码仓库,请谨慎考虑!)	

## 导入外部仓库

您可以将开源 Git 仓库或其他平台的仓库快速迁移至 CODING DevOps 平台,详情请参见 导入或关联外部仓库 。

## 初始化 Git 仓库



创建仓库后可以通过四种方式进行初始化。

	代码仓库还未初始化	
当您使用客户	P端克隆代码仓库时,终端提示的用户名是您在 CODING 个人设置 里填写的「手机」	或「邮箱」。
HTTPS 🔻	https://e.coding.net/	จ
HTTPS ▼ 导入外部仓库	https://e.coding.net/	Q
HTTPS ▼ 导入外部仓库 快速初始化仓库	https://e.coding.net/    /lans-project/init.git	ច
HTTPS ▼ 导入外部仓库 快速初始化仓库 使用命令行创翼	https://e.coding.net/    /lans-project/init.git 库 建仓库	បា

### 导入外部仓库

输入 Git 仓库地址后进行导入,完成初始化。

导入外部仓库		~
仓库克隆地址	请输入仓库克隆地址,如 https://github.com/Coding/WebIDE.git	开始导入

#### 快速初始化仓库

此方法将通过生成 README.md 文件完成初始化。

快速初始化仓库			$\sim$
快速初始化仓库会为您的仓库新建一个	README 文件。	同时您也可以选择为仓库增加一个 .gitignore 文件。	
None -			
✓ 启用README.md文件初始化仓库			
快速初始化仓库			

使用命令行创建仓库

 $\sim$ 



此方法本质上是在本地仓库中生成一个 README.md 文件后上传至远端仓库完成初始化。

#### 使用命令行创建仓库

如果您还没有任何代码, 可以通过命令行工具创建一个全新的 Git 仓库并初始化到本项目仓库中。

```
git clone https://e.coding.net, nit.git
cd init
echo "# init" >> README.md
git add README.md
git commit -m "first commit"
git push -u origin master
```

### 使用命令行推送本地仓库

您也可以直接将本地已初始化的仓库上传至远端仓库中。

使用命令行推送已存在的仓库	$\sim$
如果您已有一个 Git 仓库, 可以通过命令行工具将其直接推送到本仓库中。	
git remote <b>add</b> origin http://e.staging-corp.coding.ic ting/restore. git push -u origin master	git co



# 导入或关联外部仓库

最近更新时间: 2023-02-07 14:51:32

## 进入项目

- 1. 登录 CODING 控制台,单击团队域名进入 CODING 使用页面。
- 2. 单击团队首页左侧的项目,进入项目列表页,选择目标项目。
- 3. 选择左侧菜单**代码仓库**,进入代码仓库首页。

## 导入仓库

CODING 代码仓库不仅提供了一键导入外部仓库功能,还支持定时同步外部开源仓库。 1. 前往任意项目,单击左侧栏的**代码仓库**,单击页面右上角的**点击导入**。

← 创建代码仓库 普通创建 模	板创建	在其他网站已有仓库? 点击导入
仓库类型 * 仓库名称 *		
◆ GIT 仓库 🝷 / 仓库名称只支持字母、	数字、下划线(_)、中划线(-	-)和点(.)的组合 0/100
仓库描述		
请输入仓库描述		6
快速初始化仓库		
生成 README 文件		
添加 .gitignore 文件 请选择 .gitignore 文件	*	
<ul> <li>是否开源</li> <li>私有仓库(仅对仓库成员可见,仓库成员可访问仓</li> <li>公开仓库(公开后,任何人都可以访问代码仓库,</li> </ul>	3年) 请谨慎考虑! )	
<b>代码扫描</b> 开启代码扫描可以发现代码中的安全漏洞、功能缺 行代码评审。查 <b>看详情 忆</b>	陷等代码问题,结果将展示	在合并请求详情中,辅助您进
<ul> <li>自动匹配语言</li> <li>通过模板创建</li> <li>复用E</li> <li>初次扫描到有效代码,根据语言属性自动匹配推荐;</li> </ul>	<b>己有方案</b> 规则集。	
完成创建 取消		

2. 按照提示选择仓库来源类型。



← 导入外部仓库					
仓库导入方式					
	IDI將变持 ODI SitHub				
仓库导入记录					
原仓库路径	导入方式	导入平台	导入时间	状态	操作
adolflan/warehouse-2	G Gitee OAuth	gitee.com	2022–03–09 15:51:39	成功	查看新仓库
codingopenapi/api/openapi	🖉 URL	e.coding.net/codingopenapi/api/op	2022-03-09 15:00:29	成功	查看新仓库

3. 您可以选择通过 URL 导入私有或开源仓库。若源仓库类型为私有,那么需按照页面提示填写用户名与密码。

← 导入外部仓库 / URL 导入	× 这是一个私有仓库
Git 仓库 URL *	① 需要填写您登录 时的用户名或邮箱、密码。为 了安全考虑,我们不会记住它们。
http:// /zwk-test1230/0907.git	用户名或邮箱
仓库名称 * 0907	请输入用户名或邮箱
是否开源	密码
● 私有仓库(仅对仓库成员可见,仓库成员可访问仓库)	请输入密码
○ 公开仓库(公开后,任何人都可以访问代码仓库,请谨慎考虑!)	确认 取消
<b>代码扫描</b> 开启代码扫描可以发现代码中的安全漏洞、功能缺陷等代码问题,结果将展示在 行代码评审。查看详情 I	合并请求详情中,辅助您进
完成创建取消	

## 同步仓库



同步功能仅对开源仓库开放。这意味着与源仓库保持一致,将覆盖导入 CODING 仓库后做出的变更。您可以在仓库设置中修改同步频率或关闭自动同步功能。

🔶 giteego 👻	浏览 提交 分支 合并请求 版本 对比 设置
仓库设置	同步信息
基本设置	自动同步触发时间
部署公钥	19:00 - 一 开启自动同步
文件锁定	保存
分支设置	最近 30 天同步记录
访问设置	同步外部仓库地址: https://gitee.corgit 。
推送设置	
合并请求	
版本发布	
代码标签	
保护目录	
仓库安全	
同步信息	

#### 在已导入仓库的首页中还可以通过单击导入按钮手动强制同步代码仓库。

				强制同步远端仓库 🔍 🧳 🍃
← openapi - 浏览	提交 分支 合并请求 版本	标签 对比	设置	○ + 创建代码仓库 -
n openapi	8 dev → 合 查找文件 →	输入以查找文件		<b></b>
> module	<b>文件</b> 历史 68			\$1. 创建合并请求 ····
MI README.md	参数补全			最后提交 decc39ff58 🗊 于 7 天前 按名称排序▼ ☴↓
api-Jenkinsfile	imodule	-	fix: API 参数补全	7 天前
api-content.json	🗋 .gitignore		add issue change log	10 个月前
apilison	MI README.md		fix: 更改测试环境链接	3 个月前
() <u>()</u>	🗋 api-Jenkinsfile	OPENAPI		10 个月前
	api-content.json		docs: 补充参数	5 个月前
	🗁 api.json		the straight data terms	3 个月前
	[Pr generator.py		docs: api 文档更新	10 个月前
	README.md			
	OPEN API 更新指	引		

## 导入镜像仓库

若出于安全原因考虑,您也可以选择导入外部私有(非开源)的 Git 镜像仓库,以下是操作步骤。

#### 步骤1: 将私有仓库拉取至本地

git clone --mirror 私有仓库地址







拉取至本地的镜像仓库文件夹通常带有 .git 后缀。

#### 步骤2: 获取目标 CODING 代码仓库地址

在 Web 端进入 CODING 项目,获取目标仓库的地址。

#### 步骤3: 推送至目标 CODING 仓库

在终端中输入命令进入私有仓库。



使用推送命令将私有仓库推送至目标 CODING 仓库。

git push --mirror 目标仓库地址



若出现 refs/pull 报错,可以使用以下命令避免报错:

ð

Ð

Ð



## git push URL "+refs/heads/\*:refs/heads/\*" "+refs/tags/\*:refs/tags/\*"

#### 推送成功后,可以看到的私有仓库已上传至目标仓库中。

← gitlab → 浏览	提交 分支 合并请求 版本	对比 设置		+ 创建代码仓库 🖌 🕶
✿ gitlab	<b></b>			
<ul> <li>iOSTemplate</li> <li>iOSTemplate</li> </ul>	<b>文件</b> 历史 1			33 创建合并请求 …
> 📄 iOSTemplateT	🚯	tion		最后提交 a804df1405 🗐 于 3 年前
> 📄 iOSTemplateU	iOSTemplate	GitLab	Initial template creation	3 年前
.gitignore	iOSTemplate.xcodeproj	GitLab	Initial template creation	3 年前
MI README.md	iOSTemplateTests	GitLab	Initial template creation	3 年前
	iOSTemplateUITests	GitLab	Initial template creation	3 年前
	.gitignore	GitLab	Initial template creation	3 年前
	MJ README.md	GitLab	Initial template creation	3 年前
	README.md			
	This is a template for getting st	arted with iOS deve	lopment using GitLab and fastlane.	

## 关联代码仓库

关联仓库功能本质上是将访问外部仓库的凭据"暂存"至 CODING,当您使用持续集成或持续部署时,能够直接调用第三方仓库作为代码源,而省去了频繁迁移 的繁琐流程。

ŵ	项目概览		代码仓库 关联仓库						+ 关联代码仓库
	项目协同		仓库来源全部→ 认证方式全部→	已开启项目模块 全部 ▼	关联人 全部 ▼	搜索仓库 Q			
>	代码仓库		代码仓库	仓库来源	认证方式	已开启项目模块	关联人	关联时间	操作
٢	代码扫描 beta	>	Add activity documents	GitHub	OAuth	全部模块	▶ 主账号	2021-06-11 11:38	
00	持续集成	>	110世辺序 #10				-		
Ŷ	持续部署	>	-   (响件, 共						
	制品管理	>							
Ā	测试管理	>							
.8	文档管理	>							

支持的关联仓库类型有 GitHub 、GitLab 、私有 GitLab、Gitee、工蜂、通用 Git 仓库与其他项目中的 CODING 仓库。前五种仓库类型支持 OAuth 认证



方式,通用 Git 仓库支持账号密码认证,关联后的仓库代码不会存储至 CODING 代码仓库。



### 关联 GitLab 私有仓库

如需关联私有 GitLab 仓库,需要在 GitLab 创建应用然后由团队管理员绑定私有 GitLab 服务。具体操作请参见 绑定 GitLab 私有云服务 。

#### 关联 GitLab SaaS 仓库

如需关联 GitLab SaaS 版本上的仓库,在**关联代码仓库**页面选择 **GitLab** 代码源,然后单击**前往认证**,在跳转的 GitLab 验证页面单击 Authorize 完成授 权。授权成功后,选择需要关联的代码仓库即可完成操作。

← 关联代码仓库						
仓库来源						
<b>~</b>	0			G	<b></b>	
CODING	GitHub	GitLab	私有 GitLab	Gitee	工蜂	通用 git 仓库
认证方式		授权人				
OAuth	-	您尚未认	证 GitLab OAuth	,前往认证		
代码仓库 *						
请选择	•					
确认关联	取消					



### 关联 GitHub 仓库

在**关联代码仓库**页面,选择 GitHub 代码源,然后单击前往认证使用 OAuth 认证即可跳转至 GitHub 进行授权认证。若提示失败,有可能是因为您未在 GitHub 中填写用户名。请前往 Settings > Profile > Name 进行填写。

#### 关联其他项目的代码仓库

若希望关联其他 CODING 项目中本人名下的代码仓库,需前往**关联代码仓库**页面,仓库来源选择 CODING,勾选目标仓库后进行关联。关联后可用作持续集成 或持续部署中的代码源。

C	C anas		← 关联代码仓库	
Fo	项目概览			
:	项目协同		🗢   🖓   🦊   🐓   🤤 👳   🍕	
>	代码仓库		CODING GitHub GitLab 私有 GitLab Gitee 工蜂 通用 gi	it 仓库
$\sim$	持续集成		认证方式	
<b>A</b>	持续部署		个人令牌 -	
C	代码扫描	Beta	代码仓库 *	
	应用管理	Beta		
	制品管理		确认关联取消	
-	测试管理			
•	文档管理			
设置				
٩	项目设置			

若希望将仓库直接导入至项目中进行修改,请前往目标仓库的**访问设置**并将其设置为开源状态,然后参见 <mark>导入开源仓库</mark> 导入目标代码。

#### 修改关联仓库

本章节以 GitHub 为例:

登录 GitHub 后点击右上角头像 Settings > Applications > Authorized OAuth Apps 取消关联。



Your personal account	Go to your personal profi
Account settings	Applications
Profile	Installed GitHub Apps Authorized GitHub Apps Authorized OAuth Apps
Account	
Appearance	You have granted 4 applications access to your account. Sort - Revoke all
Accessibility	
Account security	Last used within the last week · Owned by Coding     Report abuse
Billing & plans	Revoke
Security log	iconfont     Last used within the last 6 months · Owned by iconfont-cn
Security & analysis	
Sponsorship log	Linode
Emails	Last used within the last 3 months - Owned by linode
Notifications	····
Scheduled reminders	Last used within the last 12 months - Owned by easychen
SSH and GPG keys	
Repositories	(i) Read more about connecting with third-party applications at GitHub Help.

切换为其他的 GitHub 账号重新进行 OAuth 认证完成 关联 GitHub 仓库。



# 浏览仓库文件

最近更新时间: 2023-08-02 18:00:52

本文为您详细介绍如何浏览仓库文件。

## 进入项目

1. 登录 CODING 控制台,单击团队域名进入 CODING 使用页面。

2. 单击团队首页左侧的**项目**,进入项目列表页,选择目标项目。

3. 选择左侧菜单代码仓库,进入代码仓库首页。

单击任一仓库名称进入该仓库的详情页面,您可以在此处查看该代码仓库下的所有文件目录,或预览单个代码文件下的所有内容、提交历史、各提交内容间的代码 对比等信息。

## 仓库文件列表

进入仓库页面后,将自动显示仓库默认分支的文件目录结构。您可以新增文件或文件夹,或通过单击 ··· 对指定文件执行锁定、设为只读、Raw(查看未经渲染的 原始文本 )、复制路径、下载或删除等操作。

ji test 与切换仓库		
		新版仓库使用引导 🖸
<pre></pre>	۹	€ 云原生开发 …
make commit code     小松鼠 提交于 1 年前	-~ d	锁定 设为只读
		Raw 复制路径
[,2initpy 1	☑ ◆ 编辑代码	下载

#### () 说明:

您可以通过分支名称旁的面包屑导航快速定位目录结构中的文件。

#### 仓库提交历史

单击仓库首页的提交数,默认进入 master 分支的提交历史页面。

jit	est 与切换仓库		
	本	✿ 设置	新版仓库使用引导 2
💡 master 👻 🐉 分支 1 🛛 🖉	标签		Q + 🗘 😋 云原生开发 克隆/下载 •
make commit code CODING-Support 提交于1年前	ΰ		◆df86d4a 页91 个提交
🚞 api	小松鼠	make commit code	1 年前
ases 🔁	小松鼠	make commit code	1 年前
🗋 pytest.ini	小松鼠	make commit code	1 年前
ML readme.md	小松鼠	make commit code	1 年前
T requirements.txt	小松鼠	make commit code	1年前

您可以切换为其他分支,查看相应的提交历史记录,也可以通过网络视图模式查看来自其他分支的合并情况。



ji lic1	与切换仓库 😮 公开	
小代码      記 合并请求     谷版本	☞ 云原生构建 ● 设置	新版仓库使用引导 🖸
😵 master 👒		
		网络视图 >
<b>P</b> master	◆ 上传文件 ● 小松鼠 提交于1年前	列表视图           ● 已验证         4c4ea24           网络视图
	Initial Commit     O 小松鼠 提交于1年前	● 已验证 79f7c60 少

## 文件提交详情

在仓库指定分支的提交历史页面,提交记录按照日期倒序排列。单击任一提交记录的名称或右侧的 SHA ID 可以跳转至该提交记录的详情页面。

jic1 与切换仓库 ❷ 公开			
◆代码		新版仓库使用	引导区
\$ <sup>2</sup> master ▼ 提交人 全部 ▼ 提交日期 开始日期 ▼ 至 结束日期 ▼	Q. 搜索	列表视图	- = <b>↓</b>
上传文件     ① 小松鼠 提交于1年前	● 已验证	4c4ea24	
Initial Commit ○ 小松鼠 提交于1年前	⊘ 已验证	79f7c60 🧊	

提交记录详情页面会清晰地列出本次提交中所有更改过的文件。

jiy c1 ≒ 切换仓库 ❷ 公开	
	新版仓库使用引导 🛽
上传文件 查 CODING-Support 提交于 1 年前 → 4c4ea24 Diff Patch 1 parent 79f7c60	浏览代码
2 个文件被修改 (0 行增加 / 0 行减少)	※ 全部展开 ≪ 全部収起 ∲ 切扱为左右模式 ◎ 高级透现
▶ 0 ■■■■■ CODING-CD产品说明.pdf	
> 0 ====== 上线鹅讯云测试报告.docx	
(字): (评论) (评论)	

代码的行级对比支持使用**普通模式、左右模式与高级选项**,以便更快发现代码间的异同,还支持对每行代码发起评论。

#### 普通模式

普通模式下,代码文件中会显示该提交改动过的所有文件。展开任一文档,可查看具体的提交改动。每个文件内会显示具体的行级增删修改信息。其中,绿色代表 增加,红色表示删减。

1 个文	件被修	8改(1 行増加 / 1 行减少)				≫ 全部展开	☆ 全部收起	∳ 切换为左右模式	✿ 高级选项
¥ 2		readme.md							
		testing							
		1 This is a test.							
	2	2 testing S							
	-T								
	_								
li	ΖE								
		评论							
				© \$\$	0				
		评论							

## 左右模式



单击右上角**切换为左右模式**,代码行级对比由普通模式切换为左右模式。用户可清晰地对比代码改动前后的内容。改动哪一行、修改了什么内容均一目了然。 您还可以选择浏览改动前或后的文件或 Diff 文件。

1 个文件被修改(1 行增加 / 1 行减少)		≫ 全部展开	≪ 全部收起	∳ 切换为普通模式	✿ 高级选项
✓ 2 ■■■■■ readme.md					
1 testing	1 This is a test.				新文件
2 testing⊙	2 testing ⊘				Diff
· · · · · · · · · · · · · · · · · · ·					
◎ \$\$\$ \$\$\$\$ \$\$\$\$ \$\$\$\$\$ \$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$					
评论					

#### 高级选项

高级选项支持折行显示、显示制表符模式。您可以按需打开或关闭对应选项。

#### 查看仓库容量

() 说明:

仅项目管理员具备修改仓库设置的权限。

单击**设置**,在基本设置中可以看到目前代码仓库用量剩余,在此页面中还可以修改仓库图标。

jiyunkeji	i/jack-den	no/public1	늨 切换仓库 🥝 公开						
৵代码	<b>?</b> ]合并请求	❸版本	☞ 云原生构建	✿ 设置					
仓库设置		基本设置							
基本设置		仓库名称							仓库图标
权限方案		HTTPS 👻	https://		demo/	public1	7 / 100	J	
部署公钥		<ol> <li>修改仓库</li> <li>git remot</li> </ol>	名称将导致仓库的访问 L e set–url origin https:/	JRL 改变,在L	此之前的 URI	_ 地址将失效。Git ic1.c	仓库地址修改方法: ait		Ρ
文件锁定		3							
分支设置		仓库描述							更改图标
仓库规范		请输入仓库描述							
访问设置								1,	
推送设置		设为自定义模     白定义模板合体	<b>板仓库</b>	的日录结构和文体	性生成新的合度	横板合废当前团队同	TE		
合并请求		保存		99 H 7 ST 979 X 1	11 - E. 169 Mar 19 19 19 19	, 来成 6月 二时四阶号	11110		
版本发布	_	PRIJ			_				
代码标签		仓库容量管理 •							
仓库安全	L	已使用 476.00 KB /	/ 总容量 5.00 GB 修改仓	库容量上限					



# 修改远程仓库地址

最近更新时间: 2023-08-02 18:00:52

本文为您详细介绍如何修改远程仓库地址。

## 进入项目

- 1. 登录 CODING 控制台,单击团队域名进入 CODING 使用页面。
- 2. 单击团队首页左侧的项目,进入项目列表页,选择目标项目。
- 3. 选择左侧菜单代码仓库,进入代码仓库首页。

### 操作说明

1. 进入一个项目之后,在**代码仓库**模块,单击任一仓库进入其详情页面。在**设置**页签即可对当前代码仓库进行设置。

<ol> <li>说明:</li> <li>只有项目管理</li> </ol>	里员才能进入仓库的设置页面。	
j –	est 与切换仓库	
৵代码	合并请求	
仓库设置	基本设置	
基本设置	仓库名称	仓库图标
权限方案	HTTPS - Api-test 8 / 100	
部署公钥	修改仓库名称将导致仓库的访问 URL 改变,在此之前的 URL 地址将失效。Git 仓库地址修改方法: git remote set-url origin https://e.coding.net//api-test.git	A
公支设置	仓库描述	更改图标
仓库规范	请输入仓库描述	
访问设置		
准送设置		
合并请求	日走又懷做它库可让用广创建它库时以相同的日求结构和文件生成新的仓库,模板仓库当前团队可用。	
版本发布	はたけ	

2. 您可以在**基本设置**页面修改您的仓库名称与图标。修改名称后会导致仓库的访问 URL 改变,在此之前的 URL 地址将失效。修改名称后需在您的本地仓库与 新的地址相匹配。



除此之外,您还可以在该页面添加仓库描述、查看代码仓库用量情况。如有必要,您还可以归档、重置或删除代码仓库。



# 管理仓库

最近更新时间: 2023-08-02 18:00:52

本文为您详细介绍如何在仓库中执行部分操作。

## 进入项目

- 1. 登录 CODING 控制台,单击团队域名进入 CODING 使用页面。
- 2. 单击团队首页左侧的**项目**,进入项目列表页,选择目标项目。
- 3. 选择左侧菜单**代码仓库**,进入代码仓库首页。

## 清理仓库

远程仓库在使用的过程中会产生缓存文件,您可以单击**仓库清理**减少缓存文件对仓库容量的占用。

ji	st ≒切换仓库
৵代码 \$16	合并请求 診版本 ③云原生构建 ❹设置
仓库设置	仓库描述 更改图标
基本设置	请输入仓库描述
权限方案	
部署公钥	设为自定义模板仓库 自定义模板仓库可让用户创建仓库时以相同的目录结构和文件生成新的仓库,模板仓库当前团队可用。
文件锁定	保存
分支设置	
仓库规范	仓库容量管理 ●
访问设置	已使用 40.00 KB / 总容量 5.00 GB 修改仓库容量上限
推送设置	仓库清理
合并请求	仓库清理使用 git ge 饥化代龄仓库存储空间,压缩存储库对源,减少存储库磁盘占用。

## 归档仓库

如需归档仓库,在代码仓库列表单击 … 并按照提示确认即可。

我的星标 已归档 全部仓库	未分组 更多 ▼		
api-test		cd-multi-env-demo	
未填写仓库描述	移动到分组	≥库描述	
1 0 更新于 1 年前	仓库设置	更新于 2 年前	
	归档仓库		
P public1 公开 未填写仓库描述	里 <u>且</u> 0 件 删除仓库		
╏ 0 更新于 1 年前			

归档代码仓库之后,将会自动阻隔 Git 或 Web 端的代码仓库访问请求,用户无法继续访问和操作该仓库。已归档的仓库只能在**已归档**分类内查看,若希望恢复仓 库的正常访问,需重新解除归档。



我的星标 已归档 全部仓库 未分组 更多		
api-test		
未填写仓库描述	解除归档	
\$ ⑦ 更新于1年前		

## 删除与重置

如需重置或删除仓库,在代码仓库列表单击 … 并按照提示确认即可。

删除后的代码仓库将移入回收站保留 30 天,项目管理员可在到期前进入回收站恢复。逾期将会彻底删除当前代码仓库内的所有代码,包括代码分支、合并请求、 代码版本,数据无法恢复。



单击**重置仓库**后,将重置仓库内的所有代码,包括代码分支、合并请求、代码版本。重置后数据无法恢复,并且仓库也会被重置为空仓库。


# 恢复已删除仓库

最近更新时间: 2023-08-02 18:00:53

本文为您详细介绍如何恢复已删除仓库。

# 进入项目

- 1. 登录 CODING 控制台,单击团队域名进入 CODING 使用页面。
- 2. 单击团队首页左侧的**项目**,进入项目列表页,选择目标项目。
- 3. 选择左侧菜单**代码仓库**,进入代码仓库首页。

# 操作说明

对于已删除的代码仓库,项目管理员可在 30 天内进入回收站恢复。

1. 在**代码仓库**页面,单击右上角回收站图标。

标 已归档 <b>全部仓库</b>	未分组 更多 →			搜索仓库	
<b>api-test</b> 未填写仓库描述	•••	C cd-multi-env-demo 未填写仓库描述	 C cd-multi-env-deploy-d 未填写仓库描述		jack-docker-push-demo - 未填写仓库描述
\$ 0 更新于1年前		\$ 0 更新于 2 年前			\$ <sup>*</sup> 】0 更新于 2 年前
<b>public1</b> 公开 未填写仓库描述	***				
1 <sup>●</sup> 1 0 更新于1年前					

2. 进入回收站后,选择要恢复的代码仓库,单击恢复即可恢复该仓库。

← 回收站 ⑦				
仓库名称	删除操作人	删除时间	剩余时间	操作
♦ test		2021-10-28 15:02:02	30 天	恢复



# 管理代码仓库卡片

最近更新时间: 2023-08-02 18:00:53

本文为您详细介绍如何管理代码仓库卡片。

# 进入项目

- 1. 登录 CODING 控制台,单击团队域名进入 CODING 使用页面。
- 2. 单击团队首页左侧的**项目**,进入项目列表页,选择目标项目。
- 3. 选择左侧菜单代码仓库,进入代码仓库首页。
- 在 CODING DevOps 平台上,每个代码仓库默认以卡片的形式展示。您可以在个人工作台或各个项目中快速查看代码仓库。

# 个人工作台

个人工作台将展示所有与"我"有关的代码仓库,包括由"我"创建、"我所加入"的代码仓库。单击设为星标后代码仓库将置顶以供快速查看。

代码仓库 最近访问 全部仓库 开源仓库					创建代码仓库
星标仓库					
A android 未填写仓库描述	E express ★ 未填写仓库描述	openapi 未填写仓库描述	*		
🔓 蓋的測试 🕄 3	☐ 莖的測试 \$\$ 2	🗁 蓋的測试 🏾 🕄 0			
所属项目 ▼ 公开性 ▼ Q 搜索					
仓库名称	所属项目	合并请求	更新时间	操作	٥
A android	■人間の	<b>11</b> 3	2022-08-23 16:50:38	*	
J java-spring-example	CODING电商平台mini1-不要用来测试	<b>%]</b> 2	2022-09-08 18:11:14	*	
A auto1	CODING电商平台mini1-不要用来测试	<b>31</b> O	2022-06-21 18:24:06	*	
J java-spring-example2	CODING电商平台mini1-不要用来测试	87 I	2022-02-17 10:10:37	*	
O openapi	<b>三</b> 代版(1)	11 0	2022-03-04 12:18:28	*	

#### 单击右侧齿轮按钮新增更多查看项。

代码仓库 最近访问 全部仓库	开源仓库						创建代码仓库 ···
星标仓库							
N new-one ★ 未填写仓库描述	B blog 未填写仓库描述	*	D demo-tool 未填写仓库描述	★ <b>k8sDemo</b> ★ 未填写仓库描述	D docs 未填写仓库描述	*	
□ 飞鸟集 \$1.2	合个人博客 \$1.0		音 CODING 洞见 11 0	一 演示項目 \$1 0	□飞鸟集 11 0		
所属項目 ▼ 公开性 ▼ Q 授索							
仓库名称	所属项目	合并请求	仓库容量	最后一次提交	更新时间	创建时间	操作 🗘
D demo-tool	CODING 洞见	<b>3</b> ] O	84 KiB / 1 GiB	更新文件 README.md	2021-04-01	2021-04-21	*
D Demo	Gartner DEMO	<b>3</b> ] O	12 KiB / 1 GiB	Initial Commit	2022-03-24	2022-03-24	*
B blog	个人博客	<b>31 o</b>	58.6 MiB / 1 GiB	fix:链接修复	2020-11-19	2020-11-15	*
B blog_uQSjwQ	个人博客	<b>31 o</b>	940 KiB / 1 GiB	Initial commit	2020-11-16	2020-11-16	*
A awesome-blogs	我导入的项目oll8	11 o	284 KiB / 1 GiB	Merge pull request #2 from delta4d/master	2017-08-02	2020-10-12	*

通过左侧的拖拽选项为各项属性进行排序。支持快速展示下列属性:

- 所属项目
- 合并请求
- 仓库容量
- 创建时间
- 最后一次提交
- 更新时间



								8	表格显示设置	
代码仓库 最近访问 全部仓库	开源仓库								表头设置	恢复默认
星标仓库									仓库名称	
N new-one ★ 未填写仓库描述	B blog 未填写仓库描述	*	D demo-tool 未填写仓库描述	*	K k8sDemo 未填写仓库描述	*	D docs 未填写仓库描述		<b>操作</b> 以上属性不可排序	
☐ 飞鸟集 \$1 2	合个人博客 \$3.0				🗃 演示项目 🕄 0		☐ 飞鸟集 \$3.0		所属项目	
新属项目 ▼ 公开性 ▼ Q 搜索									:: 百开頃水 ::: 仓库容量 ::: 创建时间	
仓库名称	所属项目	合并请求	仓库容量	创建时间	0)	最后一次提交		更新	□最后一次提交	
D demo-tool	CODING 洞见	31 O	( 84 KiB / 1 GiB	2021-	04–21	更新文件 README.n	nd	201	更新时间	
Demo	Gartner DEMO	81 O	( 12 KiB / 1 GiB	2022-	-03–24	Initial Commit		201		
B blog	个人博客	81 O	58.6 MIB / 1 GIB	2020-	-11–15	fix:链接修复		202		
B blog_uQSjwQ	个人博客	<u> </u>	940 KIB / 1 GIB	2020-	-11–16	Initial commit		202		
A awesome-blogs	我导入的项目oll8	<u> </u>	( 284 KiB / 1 GiB	2020-	-10-12	Merge pull request a	#2 from delta4d/master	201		

# 项目内

进入任一项目后对仓库卡片进行操作。

### 分组与排序

在代码仓库页面,您可以创建仓库分组,然后通过批量操作将多个仓库同时添加到同一分组。通过管理分组,您可以调整仓库分组的展示顺序。

<b>� 代码仓库</b> � 关联仓库			窗 + 创建代码仓	2库 🔹
战的星标 已归档 全部仓库 未分组	更多▼	搜	索仓库 Q 默认排序 ▼	∷ 88
A api-test	公共分组 ⑦     ①       创建分组     -env-demo       管理分组	<b>C</b> cd ···· 未填写仓库描述	J jack- 未填写仓库描述	***
\$ <sup>℃</sup> 】0 更新于1年前	批量整理仓库 i于 2 年前	\$ 0 更新于 2 年前	\$℃ 0 更新于 2 年前	
P public1 公开	- P public2 公开 未填写仓库描述			
\$℃ 0 更新于1年前	\$ <sup>4</sup> 】0 更新于1年前			

## 可通过单击 \*\*\* 将其添加至任一分组。



### 公共分组

仓库分组默认为个人视图,即每个人都可以配置互相独立的分组视图。当项目管理员根据团队情况,需要项目成员统一并复用相同分组视图时,可以在**代码仓库**页 面的**更多**下拉列表中开启公共分组。



我的星标 已归档 全部仓库 未分组	更多 ▼
A aaas 更新于 2 天前	公共分组 ⑦ ()) 创建分组 管理分组
<b>C CC</b> 更新于 6 个月前	批量整理仓库
<b>G</b> git-3 更新于 9 个月前	

开启公共分组时,所有项目成员的分组视图都将切换至公共分组视图。此时,仅项目管理员可管理公共分组。
 开启公共分组对话框中默认勾选同步现有分组至公共分组,如不勾选选项则需项目管理员创建公共分组。

开启公共分组	×
开启后,项目成员将 <b>全部切换至</b> 公共分组,请谨慎操作!	
✔ 同步现有分组至公共分组 ⑦	
确 定 取 消	

• 未开启公共分组时,所有项目成员的分组视图为自己的个人分组视图。此时,可以自行修改、仅对自己可见。



Ð

Ð

Ð

Ð

# 通过本地命令行管理仓库

最近更新时间: 2023-02-07 14:52:15

本文为您详细介绍如何通过本地命令行管理仓库。

# 进入项目

- 1. 登录 CODING 控制台,单击团队域名进入 CODING 使用页面。
- 2. 单击团队首页左侧的项目,进入项目列表页,选择目标项目。
- 3. 选择左侧菜单代码仓库,进入代码仓库首页。

## 从远程仓库获取数据

您可以使用 git clone 命令克隆远程仓库至本地,并自动与之关联。

git clone [remote-name]

### 推送数据到远程仓库

使用 git push [remote-name] [branch-name] 可以将本地仓库中的数据推送到远程仓库,例如: git push learn-git master 会将本地的仓库数据推送到 远程仓库的 "master"分支。

### 重命名远程仓库

使用 git remote rename [old-name] [new-name] 命令修改某个远程仓库在本地的简称,例如想把 origin 改成 learn-git,可以运行:

git remote rename origin learn-git

重命名远程仓库之后,若在使用 Git 命令时需要指定远程仓库的名字,请使用更新后的命名作为推送命令。

git push learn-git [branch-name]:mr/master/[branch-name]

### 解除远程仓库关联

例如要解除和远程仓库 origin 的关联,运行:

git remote rm origin

▲ 注意

此命令是解除了本地仓库和远程仓库的关联,并不是删除了远程仓库的数据。

如需了解更多常用的 Git 命令,请参见 Git 常用命令速查表 。



# 模板仓库

最近更新时间: 2023-08-02 18:00:53

本文为您详细介绍如何使用模板仓库功能。

# 进入项目

1. 登录 CODING 控制台,单击团队域名进入 CODING 使用页面。

2. 单击团队首页左侧的**项目**,进入项目列表页,选择目标项目。

3. 选择左侧菜单**代码仓库**,进入代码仓库首页。

模板仓库本质上是快照保存一个仓库中的所有文件与分支为母本。项目管理员可以将需频繁调用的**脚手架**代码仓库直接设为模板仓库,让其他开发成员在创建代码 仓库时能够快速复用已有的代码沉淀。

# 创建模板仓库

您需要先创建一个 代码仓库,再将其作为模板仓库。由**项目管理员**前往仓库中的设置页面,勾选设为自定义模板仓库。

jiy	34 ≒ 切换仓库
৵代码 沿合	并请求 1 ❷ 版本 3 云原生构建 ♀ 设置
仓库设置	基本设置
基本设置	仓库名称
权限方案	HTTPS - https://e 3/ 1234 4/100
部署公钥	<ul> <li>         修改仓库名称将导致仓库的访问 URL 改变,在此之前的 URL 地址将失效。Git 仓库地址修改方法:         git remote set-url origin https://         4.git         4</li></ul>
文件锁定	合废描述 更改图标
分支设置	
仓库规范	地理として、
访问设置	
推送设置	设为自定义模板仓库 自定义模板仓库可让用户创建仓库时以相同的目录结构和文件生成新的仓库,模板仓库当前团队可用。
合并请求	保存
版本发布	

创建完成后,模板仓库会出现标记。模板仓库不仅限于本项目使用,还能够被团队中的其他项目所引用。因此团队可以建立一个存储公关资源的项目,团队成员无 需加入此项目便能够调用此模板仓库。



# 基于模板创建仓库



模板仓库设置完成后,在新建仓库时,单击上方的**模板创建**,选择自定义模板下的仓库完成创建。您还可以按照需求选择是否导入仓库中的所有分支。

← 创建代码仓库 普通创建 模板创建	
仓库名称 *	
A-new-one	9/100
仓库描述	
请输入仓库描述	
	1.
选择模板 * 预置模板 自定义模板	
S lans-project/spring 未填写仓库描述	
✓ 导入模板仓库所有分支 未勾选则导入模板仓库的默认分支。	
是否开源	
● 私有仓库(仅对仓库成员可见,仓库成员可访问仓库)	
○ 公开仓库(公开后,任何人都可以访问代码仓库,请谨慎考虑!)	
完成创建 取消	



# 提交日期格式

最近更新时间: 2023-08-02 18:00:53

本文为您详细介绍如何修改仓库中的提交日期格式。

# 进入项目

- 1. 登录 CODING 控制台,单击团队域名进入 CODING 使用页面。
- 2. 单击团队首页左侧的**项目**,进入项目列表页,选择目标项目。
- 3. 选择左侧菜单代码仓库,进入代码仓库首页。

每当我们希望追溯代码历史提交信息时,通常会按照大致的项目时间进行回溯。在代码仓库的提交记录中不仅可以查看任意提交记录的作者,还能够查看详细的提 交日期。

ji4 与切换仓库			
◆代码  \$1 合并请求 1  ◎版本  ③ 云原生构建  ○ 设置		新版仓库使用引	导区
😢 master 👒			
提交人 全部 × 提交日期 开始日期 × 至 结束日期 ×	Q 搜索	列表视图 👻	∕ ≡↓
updated readme 🛕 @ Chasy 提交于 13 分钟前	❷ 已验证	a715d69	
→ s's 🛕	♥ 已验证	2c3debe	
		cfcafc0	
→ 版本提交		58544cd	
◆ Accept Merge Request #17 : (feat/application-init-at-1666172855 -> master) ····· ◆ Chasy 提交于 9 个月前		9790574	

您可以调整日期查看格式。单击左下角的个人账户设置 > 仓库设置进行调整。更改完成后单击保存即可生效。



<ul> <li>€</li> <li>账户f</li> <li>▲</li> </ul>	◆ 个人账户设置 信息 个人账户	<ul> <li>仓库设置</li> <li>在这里可以进行一些仓库相关的个人偏好设置。</li> <li>仓库日期格式</li> <li>系统默认日期格式 根据最佳体验针对性的展示绝对日期格式/相对日期格式。</li> <li>自定义日期格式</li> </ul>	
金人		长日期格式(YYYY-MM-DD HH:mm:ss) 👻	
	仓库设置 2		
ď	SSH 公钥	保存	
di.	GPG 公钥		
20	访问令牌		
٢	两步验证		
6	通知设置		
ତ	绑定设置		
	开放生态		
		团队负责人	
		服务订购	
		邀请成员	
		<b>切换语言</b> 简体中文	
		工单中心	
		更新日志	
	M8 .	退出	



# 代码对比

最近更新时间: 2023-08-02 18:00:53

本文为您详细介绍如何使用代码对比功能。

## 操作步骤

- 1. 登录 CODING 控制台,单击团队域名进入 CODING 使用页面。
- 2. 单击团队首页左侧的**项目**,进入项目列表页,选择目标项目。
- 3. 选择左侧菜单**代码仓库**,进入代码仓库首页。
- 4. 进入指定代码仓库,可单击**分支、标签**或修订版本(哈希值)进入对应页面。

testli/演示/demo <u> 少代码</u> 和合并请求 1	◎版本 🗘 云原生构建	2 🗘 设置		新版仓库使用引导 [2
demo  demo  src  Dockerfile  Jenkinsfile	ジ master → ジ 分支 Initial commit ▲ reviewer-1 提交于 6	5 2标签 2		Q + ♀ ⓒ 云原生开发 克隆/下截 ▼ ◆ c30a6e4 ④1个提交
ML README.md mvnw mvnw.cmd pom.xml	<ul> <li>src</li> <li>Dockerfile</li> <li>Jenkinsfile</li> <li>MI README.md</li> <li>mvrww</li> <li>mvrww.cmd</li> <li>pom.xmi</li> </ul>	reviewer-1 reviewer-1 reviewer-1 reviewer-1 reviewer-1 reviewer-1	Initial commit Initial commit Initial commit Initial commit Initial commit Initial commit	6 天前 6 天前 6 天前 6 天前 6 天前 6 天前 6 天前
	README.md			

- 5. 在对应页面中单击 · · · 并选择对比。
  - 以分支列表为例,选择指定分支,单击 ・・・并选择对比。

【码 13 合并请求 ◎ 版本 53 云原生构建 0 设置	
<b>苦跃分支</b> 我的分支 保护分支 只读分支 所有分支 过时分支	湿索 ♀ + 慈雄分支
master 默认分支 ② ③ Chasy 更新于1小时前 ◇ b4afd5f ①	0 0 创建合并请求
dev @ Chasy 更新于 3 天前 -> baScc38	2 2 设为只读
2个, 共2个	编辑备注 < Cherry-pic
	revert

○ 以标签列表为例,选择指定标签,单击 ・・・ 并选择对比。



jìT		
共 4 条标签	搜索	Q + 创建标签
<ul> <li>✓ v4.0 ····</li> <li>Chasy 创建于 3 天前 ◆ 0ccc9f3 □</li> </ul>	创建版本描述	下载
<ul> <li>Ø 100 mm</li> <li>Chasy 创建于 3 天前 ↔ 0cec9f3</li> </ul>	版本描述	別除
<ul> <li>✓ V2.0 ····</li> <li>Chasy 创建于 3 天前 ↔ d2d652c</li> </ul>	版本描述	
Ø 10 ···· Chasy 创建于 3 天前 ◆ d2d652c	版本描述	
1-4个, 共4个		< 1 > 15条/页 >

○ 如果进入了指定的修订版本(哈希值),在右上角单击 ・・・并选择对比。

浏览代码 ··
创建标签
※全部成計 《全部收起 於切換力普通模式 対比

6. 在代码对比页面,可选择对应分支、标签或修订版本进行对比。

			/Т	jiy
		1	診版本 ♀云原生构建	<b> </b>
+ 创建合并请求	引祖先的差异 😨 🛛 存在代码冲突,你仍然可以创建合并请求 🧬	✔ 相对共同祖先的差	▼ 目标分支: master ▼	源分支: dev 👻 🔶
				「分支、标签、修订版本号。 <b>Q</b>
				<b>分支</b> 标签 修订版本 29 dov
✿ 高级选项	≫ 全部展开 《 全部收起 ↓ 0 切换为左右模式		臧少)	2° master
				> 4 README.md

## 功能说明

- 为保证网页响应速度,若超过 20 个文件存在差异将无法使用全部展开功能。
- 代码对比功能本质上是 git diff 命令的在线图形化表达。若勾选相对共同祖先的差异选项,那么将展示源分支与共同祖先间提交的差异。



不勾选相对共同祖先的差异选项将直接展示源分支与目标分支间的对比。







# SVN 仓库使用 创建 SVN 仓库

最近更新时间: 2023-08-02 18:00:53

本文为您详细介绍如何创建 SVN 仓库。

# 进入项目

- 1. 登录 CODING 控制台,单击团队域名进入 CODING 使用页面。
- 2. 单击团队首页左侧的项目,进入项目列表页,选择目标项目。
- 3. 选择左侧菜单代码仓库,进入代码仓库首页。

### 操作步骤

- 1. 进入一个项目之后,单击左侧导航栏代码仓库进入代码仓库管理页面。
- 2. 单击页面右上角创建代码仓库,选择仓库类型为 SVN 仓库。

c	C anau	•	← 创建代码仓库 / 普通创建 模板创建 在其他网站已有仓库? 点击导入
	项目概览		所属项目 *
22	项目协同		<ul> <li>盛的测试</li> </ul>
< <b>/</b> >	代码仓库		仓库类型 * 仓库名称 *
$\sim$	持续集成		≥ SVN 仓库 ▼         仓库名称只支持字母、数字、下划线(_)、中划线(-)和点(.)的组合         0/100
Ą	持续部署		◆ GIT 仓库
C	代码扫描	Beta	NVN 仓库
•	应用管理	Beta	
L	制品管理		初始化合库
. I	测试管理		<ul> <li>创建 SVN 仓库推荐布局 (tags, branches, trunk) SVN 仓库使用帮助</li> </ul>
.3	文档管理		今古 合 日 建 日 7 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
设置			
₽	项目设置		

3. 选择创建 SVN 仓库推荐布局时,将会自动创建 tags、branches、trunk 三个目录。这是多数 SVN 仓库的推荐目录布局。

4. 仓库初始化完成之后,即可在代码界面看到 SVN 仓库的内容。

chi	st					
<b>小代码 </b> の 提交	♀设置					新版仓库使用引导 🖸
svn_test branches	head 🔻 🗀 /					
<ul> <li>brancisco</li> <li>trunk</li> <li>tags</li> </ul>	svn_test 文件					检出▼
	名称	提交人	更新时间	提交		
	branches		2023-08	r1	Initial project layout	
	trunk		2023-08	r1	Initial project layout	
	tags	100	2023-08	r1	Initial project layout	

5. 单击右上角检出可以看到这个仓库的 SVN 地址。



	cł	st					
E svn_test      head マ 「 /      head マ 「 /      head マ 「 /      trunk     svn_test      文件     HTTPS	<b>∽ 代码 つ</b> 提交	✿设置				新版仓库使用引导(	Z
名称         提交人         更新时间         提交         当您使用客户端克隆代码时, 终端提示的用户名是您在 CODING 个人设置 里填写的邮箱           branches         2023-08         rl         https://svn.coding.net;         isst         it           trunk         2023-08         rl         initial project rayout         initial project layout	<ul> <li>Svn_test</li> <li>branches</li> <li>trunk</li> <li>tags</li> </ul>	head   □ / svn_test 文件 合称 branches trunk trunk tags	提交人	更新时间 2023-08 2023-08 2023-08	提交 rl rl	松出 HTTPS 当您使用客户端克隆代码时,终端提示的用户名是您在 CODING 个人设置 里填写的邮箱 https://svn.coding.net, pst Initial project rayout Initial project layout	-

▲ 注意:

目前只支持在创建项目中开启 SVN 仓库,不支持在 Git 仓库中新建 SVN 仓库。



Ð

Ð

# 访问 SVN 仓库

最近更新时间: 2023-02-07 14:53:44

SVN 仓库服务目前支持大多数主流 SVN 客户端。推荐使用各客户端的最新稳定版本。

# 进入项目

- 1. 登录 CODING 控制台,单击团队域名进入 CODING 使用页面。
- 2. 单击团队首页左侧的**项目**,进入项目列表页,选择目标项目。
- 3. 选择左侧菜单**代码仓库**,进入代码仓库首页。

# Mac 环境

在 Mac 环境,可使用 Homebrew 安装 SVN 客户端。

1. 运行下面命令安装 Homebrew:

/usr/bin/ruby -e "\$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"

2. Homebrew 安装好之后,在终端输入以下命令完成 SVN 安装:

brew install subversion

3. 使用命令 svn --version 验证 SVN 是否已正确安装:

svn, version 1.9.7 (r1800392) compiled Feb 28 2018, 15:54:50 on x86_64-apple-darwin17.3.0 Copyright (C) 2017 The Apache Software Foundation. This software consists of contributions made by many people; see the NOTICE file for more information. Subversion is open source software, see http://subversion.apache.org/ The following repository access (RA) modules are available:	ð
<ul> <li>* ra_svn : Module for accessing a repository using the svn network protocol.</li> <li>- with Cyrus SASL authentication</li> <li>- handles 'svn' scheme</li> <li>* ra_local : Module for accessing a repository on local disk.</li> <li>- handles 'file' scheme</li> <li>* ra_serf : Module for accessing a repository via WebDAV protocol using serf.</li> <li>- using serf 1.3.9 (compiled with 1.3.9)</li> <li>- handles 'http' scheme</li> <li>- handles 'https' scheme</li> <li>The following authentication credential caches are available:</li> <li>* Plaintext cache in /Users/Liwenqiu/.subversion</li> <li>* Mac OS X Keychain</li> </ul>	

4. 使用命令 svn checkout svn://subversion.e.coding.net/example/example-project (请将地址替换为您的 SVN 仓库地址)来检出 SVN 仓库:



5. 接下来可以使用 add、commit 命令往仓库中新添加内容:



🛑 😐 🕒		: ~/Documents/Coding/svn-project/trunk (zsh)	
~/Documents/Coding/svn-project/trunk	🖌 svn	status	?
? README.md			
? src			
~/Documents/Coding/svn-project/trunk	🖌 svn	add <u>README.md</u> <u>src</u>	?
A README.md			
A src			
A src/main.c			
~/Documents/Coding/svn-project/trunk	🖌 svn	commit -m "First commit"	+
Adding README.md			
Adding src			
Adding src/main.c			
Transmitting file datadone			
Committing transaction			
Committed revision 2.			
~/Documents/Coding/svn-project/trunk	$\checkmark$		

6. 除了使用 SVN 协议之外,还可以使用 svn+ssh 协议来访问仓库,如下图所示:



# **Cornerstone 工具**

您可以通过 Cornerstone 来使用 SVN 仓库。

1. 打开 Cornerstone 后,单击 Add Repostory 来添加 SVN 仓库(请将地址替换为您的 SVN 仓库地址)引用:



$\frown$			
Cloud Service	File Repository	HTTP Server	SVN Server
	SERVER LOCATIC	N	
	Enter the location of the repository in the	the server. The title source list.	is used to identify
Tunnel:	None		
Server:	subversion.coding	g.net	٢
Port:			
Path:	/svn-proj	ect	
	svn:// @subv	version.coding.net/	/svn-project
Title:	svn-project (subv	version.coding.net	:)
	ACCOUNT INFOR	MATION	
	Specify the account y fields blank when usi agent (such as SSH y	you use to access th ng a tunnel that requ vith private key auth	e server. Leave the uires the use of an entication).
Name:			
Password:	•••••		
	Save name and	password in my l	keychain
Paste URL F	Reset	Car	Add

#### 然后可以看到仓库的内容:

		Cornerstone		
		$\Theta$ $\phi$ $\delta$ $\Theta$	Ð G	
Check Out New Folder	Tag Branch	Revert Update Commit Merge	Import Export	View Options Panels
svn-project (subversion.coding.net)	< > svn-project			
	● ✓ Latest in Repository (HEAD)	■ ALL MOST RECENT TODAY THE	IS WEEK THIS MONTH LOCKED	Q Name, Author or Revision
	Name	▲ Date Modified Size K	Kind Revision 🗣 🖬 Author	
	branches	• Today 21:17 F	Folder 1	
	tags	• Today 21:17 F	Folder 1	
	🕨 🚞 trunk	• Today 21:21 F	Folder 2 liwenqiu	

2. 把仓库 checkout 出来,并且编辑文件之后,就可以 commit 进仓库,如下图:



Check Out New Folder	Tag Branch	Cornerstone	⊡ ⊡→ Import Export	Image: Second
svn-project (subversion.coding.net) subversion.coding.net/ /svn-project	<pre>&lt; Cancel LOG MESSAGE amend main.cl</pre>		Commit	Show Recent ≡
	CHANGES Name C a main.c	∧ ►	ing/svn-project/src	1/1
+~	OPTIONS Before continuing: Run script: After committing: Run script: Supported script Unlock files	None   None   Second Perl, Python, AppleScript, Unix she after commit	Il scripts and Automator actions.	Cancel Commit Changes

# Windows 环境

在 Windows 平台, 推荐使用 TortoiseSVN。

1. 下载 安装完成之后,在任意文件目录单机鼠标右键。



选择 checkout 把 SVN 仓库 checkout 到本地 (请将地址替换为您的 SVN 仓库地址)。



Repository		
URL of repository:	dia and	
svn://subversion.e.c	oding.net/ /svn-repo	· · · · · · · · · · · · · · · · · · ·
Checkout <u>d</u> irectory:		
C:\Workspace\svn-repo		
Multiple, independent	working copies	
Checkout D <u>e</u> pth		
Fully recursive		~
Omit e <u>x</u> ternals		<u>C</u> hoose items
Revision		
<u>H</u> EAD revision		
○ <u>R</u> evision		Show log

2. 第一次 checkout 需要输入用户名和密码。勾选 Save authentication 保存认证信息之后就不需要每次都输入密码。

Station 44	>
<svn: subversion.e<="" td=""><td>e.coding.net: &gt; /svn-repo</td></svn:>	e.coding.net: > /svn-repo
Request	s a username and a password
Username:	and the second sec
Password:	•••••
Password.	
✓ Save authentication	1
	OK Cancel

3. 进入检出的文件夹,新建 README.md 文件。



messages							
ple-click on fil	le for diff):						
ple-click on fil	le for diff): ioned Versione	d Added Delet	ed Modified I	Fi <b>les</b> Directori	ies		
ole-click on fi • <b>Non-versi</b> Extension	le for diff): ioned Versione Status	ed Added Delet Property status	ed Modified I Lock	Fi <b>les</b> Directori	ies		
ble-click on fi <b>Non-versi</b> Extension .md	le for diff): ioned Versione Status non-versioned	ed Added Delet Property status	ed Modified I Lock	Fi <b>les</b> Directori	ies	J	
	messages	messages	messages	messages	messages	messages	messages

#### 在空白处右键鼠标,选择 SVN commit... 将新建的文件保存进版本库:

Commit Finished!			-		×
Action	Path	Mime type			
Command	Commit to svn://subversion.e.coding.net/ /sv_repo				
Adding	C:\Workspace\svn-repo\README.md				
Sending content	C:\Workspace\svn-repo\README.md				
Committing transaction Completed	At revision: 2				
Added:1	<u>M</u> erge		<u>0</u> K	Cance	el

Linux 环境



在 Linux 下可以直接用系统的包管理工具安装 SVN。

# 在 Fedora 上用 yum 安装

\$ sudo yum install subversion

# 在 Ubuntu 或 Debian 上用 apt-get 安装

\$ sudo apt-get install subversion

安装成功之后,就可以用 svn checkout / commit 来访问 SVN 仓库。

### 🕛 说明

使用方法与在 Mac 平台使用命令行没有太大区别。

#### Ubuntu 下使用 SVN 命令行出现协商认证机制错误

在 Ubuntu 下使用 SVN 命令行客户端可能出现以下错误:

svn: E210007: Cannot negotiate authentication mechanism

这是由于 SVN 的认证过程使用到了 SASL 库来完成,所以需要运行以下命令安装依赖库来使用 SASL 认证:

\$ sudo apt-get install cyrus-sasl2-dbg

ð

ð

ð

ð



# 管理 SVN 目录权限

最近更新时间: 2023-02-07 14:53:48

本文为您详细介绍如何管理 SVN 仓库权限。

# 进入项目

- 1. 登录 CODING 控制台,单击团队域名进入 CODING 使用页面。
- 2. 单击团队首页左侧的**项目**,进入项目列表页,选择目标项目。

3. 选择左侧菜单代码仓库,进入代码仓库首页。

- SVN 仓库现支持权限控制,管理员能够为单独的用户设置指定目录的权限。管理员可以为仓库及子目录单独设置以下 3 种权限:
- 只读:只能查看设置的目录,不能写入,允许检出。
- 读写:可对设置的目录进行查看和写入,允许检出。
- 无权限:不能查看也不能写入,禁止检出。

### 设置权限

因每个用户默认对仓库都有读写权限,若需对 SVN 仓库中的某一目录进行权限控制,单击该目录的更多操作按钮,选择**权限**。

🔶 svn-go 👻	<b>浏览</b> 提交	设置						新建代码仓库
🗅 svn-go			head 💌 🎦 /					
> 🖿 branches	□> 已有权限							
> 🖿 trunk			svn-go					检出 🗸
> 🖿 tags			- 					
			名称	提交人	更新时间	提交		
			branches	<b>HE</b>	2022-09	r1	Initial project layout	
			🖿 trunk	MB	2022-09	r1	Initial project layout	
			🖿 tags	718	2022-09	r1	Initial project layout	

#### 在弹出的权限设置页面,您可以为该目录添加单独的用户与相应权限。

4	svn-	-go ≖	<b>浏览</b>	昆交 设置										新建代码1	仓库
	/branc	ches									+ 用户权限	+ 用户组权限	×	1	检出 ~
	用户/	/用户组				权限			操作						
	项	目管理员		•		读写 只读	无权限		保存	取消					
		筛选用户组	Q	]											
	项目	目管理员											- 1		
le,	开发	发			🖿 tags		阿蓝	2022–09 r1		Initial project	layout				
	测记	试													
	项目	目经理													
	产品														
	计元名	維													

设置完成后,已配置了权限控制的目录将以不同颜色来区分。若没有对目录配置过权限控制,目录默认显示为黑色,用户拥有对其的**读写**权限。

- 读写:黑色(默认)
- 只读: 黄色
- 无权限: 灰色



← svn-go → 3	<b>り览</b> 提交 设置				新建代码仓库	
<ul> <li>svn-go</li> <li>branches</li> <li>trunk</li> </ul>	head 💌 🗀 /				检出、	~
> Cags	<b>文件</b> 	提交人	更新时间	提交		
	branches		2022-09	r1	Initial project layout	
	🖿 trunk		2022-09	r1	Initial project layout	
	🖿 tags		2022-09	r1	Initial project layout	

### 权限覆盖说明

在某些场景下,可能会存在父目录与子级目录均设置了权限,且权限不一致的情况(例如,父目录权限为只读而子目录权限为读写)。 SVN 仓库中父目录与子目录的权限覆盖规则如下:

- 若父目录设置了权限,子目录未设置权限,则子目录继承父目录权限。
- 若父目录与子目录均设置了权限,以子目录的权限为准。例如:
  - 父目录权限为**读写**,子目录权限为**只读**,则子目录实际为**只读**权限。
  - 父目录权限为**只读**,子目录权限为**读写**,则子目录实际为**读写**权限。
  - 父目录权限为**读写**或只读,子目录无权限,则子目录实际为无权限。



Ð

ð

# SSH 协议 配置 SSH 公钥

最近更新时间: 2023-02-10 11:06:55

本文为您详细介绍如何配置 SSH 公钥。

## 进入项目

- 1. 登录 CODING 控制台,单击团队域名进入 CODING 使用页面。
- 2. 单击团队首页左侧的项目,进入项目列表页,选择目标项目。
- 3. 选择左侧菜单代码仓库,进入代码仓库首页。

### 功能介绍

SSH 的全称为 Secure Shell 即安全外壳协议,是一种加密的网络传输协议。它能够在公开的网络环境中提供安全的数据传输环境,通常用于登录远程主机与推 拉代码。

同样一个 SSH 公钥文件,如果添加至某一个代码仓库,则称为**部署公钥**,配置后默认拥有该项目的只读权限,支持新增读写权限;如果添加至团队设置中心,则 称为**团队部署公钥**,仅拥有只读权限;如果添加至个人账户,称为**账户 SSH 公钥**,配置后拥有账户内所有代码仓库的读写权限。同一个 SSH 公钥无法既作为部 署公钥,又作为个人账户 SSH 公钥。

#### 🕛 说明

若 SSH 公钥未用作账户 SSH 公钥,但添加为部署公钥时依然提示错误,那么有可能是此公钥已被其他人用作账户 SSH 公钥。若您的名下有多个团队,也可能是在其他团队中的个人账户设置中添加了此 SSH 公钥。

### 生成公钥

本文使用 ssh-keygen 工具生成 SSH 公钥,执行命令:

```
ssh-keygen -m PEM -t ed25519 -C "your.email@example.com" // 创建新的 SSH 私钥与公钥密钥对,输入您的邮箱作为标签
Enter file in which to save the key (/Users/you/.ssh/id_rsa): [Press enter] // 推荐使用默认地址
Enter passphrase (empty for no passphrase): // 此处直接回车即可;若设置密码,则每次使用 SSH 方式推送代码时都会要求输入密码
```

#### () 说明

- 若您需要使用多个 SSH 密钥对,在 Enter file in which to save the key 步骤时,输入一个新的文件名称就可以避免覆盖已有的密钥对。有关 SSH 更多信息可参见 维基百科。
- 若您的系统不支持 Ed25519 算法,请使用命令 ssh-keygen -m PEM -t rsa -b 4096 -C "your.email@example.com" 。

#### 成功之后显示如下信息:

- Your identification has been saved in /Users/you/.ssh/id\_ed25519.
- # Your public key has been saved in /Users/you/.ssh/id\_ed25519.pub.
- # The key fingerprint is:
- # 01:0f:f4:3b:ca:85:d6:17:a1:7d:f0:68:9d:f0:a2:db your.email@example.com

#### 添加公钥

您可以按需添加公钥至单个仓库或个人账户,相同的 SSH 公钥无法重复添加。

#### 添加至部署公钥

1. 打开上文中生成的密钥对的地址(通常为 ~/.ssh/ 文件夹)找到后缀为 pub 的公钥文件,使用 cat 命令输出所有内容并复制。

XEu2fu2f2l8	mv0iSs6Tt	N99CL 1565
	XEu2fu2f2l8	XEu2fu2f2l8mv0iSs6Tt

2. 前往代码仓库的**设置 > 部署公钥**页面,单击添加部署公钥,粘贴复制的公钥全文,末尾邮箱将自动填充为公钥名称。



-	a manual second	<b>已添加的公</b> 在添加过 SSH	<b>钥 ⑦</b> 公钥后 您可以使用 SSF	+ 协议访问 推送代码 占此查看 SSH 公租使田办法	列出公钥 新增公钥
÷	个人账户设置				
账户信	言息				
	个人账户				
	邮箱设置				
个人说	2置				
	仓库设置				
ď	SSH 公钥				
di.	GPG 公钥				
20	访问令牌			暂无公钥,您可以点击右上角按钮 新 <sup>始</sup> 公组	
0	两步验证			が12日 ム 77	
6	通知设置				
ତ	绑定设置				
	开放生态				
		CODING 的	SSH 公钥指纹		
		• SHA256:jo	k3FH7q5LJ6qvE7	;0Dn+Vg/Ik (RSA)	
		阿蓝	团队负责人 2:86:bb:8	5:db:87:22:c4:4f:b1	
		个人账户设置			
		服务订购			
		邀请成员			
		切换语言	简体中文		
		工単中心			
i.	HE -	退出			
•	<b>说明:</b> 部署公钥默认拥有该项目	的只读权限。如复	果需要获取推送权限,	请勾选部署公钥设置里的 <b>授予推送权限</b> 。	
	<u>ــــــــــــــــــــــــــــــــــــ</u>				
成后	,在本地运行百次连接时	1的公钥认证命令:	ssh -I git@e.cod	ing.net 。	



## 添加至团队部署公钥

团队部署公钥仅具备代码拉取权限,主要用于便捷地拉取不同项目下的代码仓库。团队管理员录入某个 SSH 公钥作为团队部署公钥后,可以将相对应的私钥分享 给团队成员。无需开源代码仓库、无需担心个人私钥泄露,仅需通过私钥便能够拉取代码仓库。

在持续交付的场景下,提供私钥便能够让相应的构建机或主机组检出代码。无需繁杂的二次关联与验证,让构建过程更加便捷。

团队所有者或管理员前往团队设置中心,单击**功能设置**中的团队部署公钥后录入 SSH 公钥。



#### 录入后添加至相应的目标仓库。

腾讯云



添加完成后,通过私钥便能够使用 SSH 协议拉取代码仓库。

#### 添加至个人账户 SSH 公钥

1. 打开上文中生成的密钥对的地址(默认地址通常为 ~/.ssh/ )找到后缀为 pub 的公钥文件,使用 cat 命令输出所有内容并复制。

<pre>~/.ssh cat id_ed25519.pub</pre>	SIGINT(2) 🗸 🚺 10381 🤇 1	1:27:12
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5A/	GUGXEu2fu2f2l8mv0iSs6TtN9	9CL 1565
83328@dd_com		

2. 登录 CODING,单击右上角个人头像进入个人账户设置 > SSH 公钥页面,然后单击新建公钥。



⊘ 飞鸟集 ▼		ā 🗘 🛤
个人账户设置	已添加的公钥 ⑦ 可以使用 SSH 协议访问,推送代码。点此查看 SSH 公钥使用办法	主账号团队所有者 个人账户设置
<ul> <li>个人账户</li> <li>曲/注入業</li> </ul>		团队管理 邀请成员
一 邮相设直		切换语言 中文 > 帮助中心
□ 模板设置		更新日志•
♂ SSH公钥 2		地田 ビ
🖬 访问令牌		
☑ 两步验证	暂无公钥,您可以点击右上角按钮	
□ 提醒设置	新增公钥	
∂ 绑定设置		
⑦ 开放生态		

#### 3. 根据提示粘贴已复制的公钥内容,按需填写公钥名称。

4. 完成后,在本地运行首次连接时的公钥认证命令: ssh -T git@e.coding.net

e e adolf@P_ADOLFLAN-MB0:~	ን የ የ የ የ የ የ የ የ የ የ የ የ የ የ የ የ የ የ የ
<pre>&gt; ssh -T git@e.coding.net</pre>	SIGINT(2) ← < 11127 < 10:46:42
CODING 提示: Hello 主账号, You've connected to coding.net via SSH.	This is a Personal Key.
主账号,你好,你已经通过 SSH 协议认证 coding.net 服务,这是一个个丿	<b>、</b> 公钥.
公钥指纹: a1:b9:65	
	✓ 11127 10:46:44



# 密钥指纹鉴权

最近更新时间: 2023-02-07 14:54:05

本文为您详细介绍如何使用密钥指纹进行代码仓库鉴权,用以确认所连接的远程仓库是否为真正的 CODING 仓库。

## 进入项目

- 1. 登录 CODING 控制台,单击团队域名进入 CODING 使用页面。
- 2. 单击团队首页左侧的**项目**,进入项目列表页,选择目标项目。
- 3. 选择左侧菜单代码仓库,进入代码仓库首页。
- 4. 若左侧未显示代码仓库,需项目管理员前往**项目设置 > 项目与成员 > 功能开关**打开功能开关。

# 功能介绍

代码安全是永不过时的议题,为了保证您所连接的远端仓库是真正的 CODING 代码仓库,现提供 SSH 密钥指纹用于鉴权。您只需要在本地运行命令后,验证返 回的结果就可以知晓远端代码仓库的真实性。

# 验证 SHA256 算法指纹

查看本地 .ssh/know\_hosts 文件中关于 e.coding.net 的 SHA256 算法的指纹,如果返回值为

jok3FH7q5LJ6qvE7iPNehBgXRw51ErE77S0Dn+Vg/lk ,则证明您连接到了正确的 CODING 服务器。可在终端中运行命令查看结果。

<pre>&gt;&gt; ssh-keygen -lf ~/.ssh/known_hosts 256 SHA256:Bdo9PWvc9YJra+FK28v7oxW0dghA/DI3ZLT3BhDz/nQ [ ]:12400 (ECDSA) 256 SHA256:V8qgXUfieT6Q//G/miMxK+8Dx05gS/2NaNpPYAU629s [ ]:28954 (ECDSA) 256 SHA256:ox9ko2YsRDgwp4C9im0Tha9FWxAXhfe7H7yLIXhcT5A [ ]:11900 (ECDSA) 256 SHA256:u2Xk2ekDfmrav2FAlTPPnGX9seyiZEk0vsFiGyp/EKo ]:28524 (ECDSA) 256 SHA256:za8qm0BYDLKBCx+hG4gT4/0iq06ZR/w00JhMoqJIWtA ECDSA) 2048 SHA256:jok3FH7q5LJ6qvE7iPNehBgXRw51ErE77S0Dn+Vg/Ik e.coding.net, (RSA)</pre>

# 验证 MD5 算法指纹

查看本地 .ssh/know\_hosts 文件中关于 e.coding.net 的 MD5 算法的指纹,如果返回值是 98:ab:2b:30:60:00:82:86:bb:85:db:87:22:c4:4f:b1 ,则 证明您连接到了正确的 CODING 服务器。可在终端中运行命令查看结果。

ssh-keygen -E md5 -lf ~/.ssh/known_hosts É
<pre>ssh-keygen -E md5 -lf ~/.ssh/known_hosts 256 MD5:cd:aa:1f:f5:f7:4d:44:a9:37:93:7d:22:94:6a:bb:b9 [ ]:12400 (ECDSA) 256 MD5:72:c3:89:c4:3e:d3:9d:6d:3e:d8:bc:af:bf:91:61:ce [ ]:28954 (ECDSA) 256 MD5:81:e4:07:c6:ec:38:5c:1a:da:03:d9:fb:34:83:9f:5c [ :11900 (ECDSA) 256 MD5:b1:ea:db:6b:04:36:d7:b5:03:57:28:7b:85:33:f4:7e [ ]:28524 (ECDSA) 256 MD5:20:8f:38:63:70:15:70:cc:a7:81:79:ff:ea:e5:11:98 (ECDSA) 2648 MD5:98:ab:2b:30:60:00:82:86:bb:85:db:87:22:c4:4f:b1 e.coding.net. (BSA)</pre>



# 通过 SSH 协议推拉代码

最近更新时间: 2023-02-07 14:54:11

本文为您详细介绍如何通过 SSH 协议进行代码推拉。

# 进入项目

- 1. 登录 CODING 控制台,单击团队域名进入 CODING 使用页面。
- 2. 单击团队首页左侧的**项目**,进入项目列表页,选择目标项目。
- 3. 选择左侧菜单**代码仓库**,进入代码仓库首页。
- 若左侧未显示代码仓库,需项目管理员前往项目设置>项目与成员>功能开关打开功能开关。

# 操作步骤

CODING 支持使用 SSH 协议推拉代码。

- 1. 参见 配置公钥 生成公钥并添加至代码仓库或个人账户。
- 2. 在代码仓库的浏览页面复制其 SSH 地址。

← Coding - 浏览	提交 分支 合并请求 版本 对比 设置	+ 创建代码仓库
1 Coding	P master →	※完隆
ML README.md	<b>文件</b> 历史 ②	<b>克隆仓库</b> 克隆代码仓库时,终端提示的用户名是您在 CODING 个人设置 里填写的「手机」或「邮箱」
	陈星 提交       ML README.md     陈星	HTTPS  https://e.coding.net/ /repo/Coding.git
	README.md	SSH

3. 在本地运行 git clone + 仓库地址 命令行即可完成拉取。





# 分支管理 创建分支

最近更新时间:2023-02-07 14:54:21

本文为您详细介绍如何使用在代码仓库中创建分支。

## 进入项目

1. 登录 CODING 控制台,单击团队域名进入 CODING 使用页面。

2. 单击团队首页左侧的项目,进入项目列表页,选择目标项目。

3. 选择左侧菜单**代码仓库**,单击任一代码仓库进入详情页,然后再单击**分支**进入分支管理页面。

分支是 Git 中的常用功能。常用的开发模式是保留一个主干,各项开发或修补工作在其他分支进行,完成后再合并入主干。因为直接在主干开发是一件危险系数较 高的活动,分支功能可以视为一道安全的阀门,将各项开发工作分隔开来,它能够保证主要版本的稳定性不被破坏,不同的人可以专注于不同的开发任务。

#### () 说明

下文中简称 CODING 代码仓库中的分支为远程分支、本地 Git 代码仓库的分支为本地分支。在本地终端运行相关命令也可以快速创建分支,请参见 Git 常用命令速查表。

### 查看分支

进入代码仓库的详情页面之后,单击**分支**页签即可查看目前远端仓库中的所有分支。您可以在此处进行新建分支、启用保护分支等操作,分支列表页会显示当前相 比于默认分支提前或落后多少提交。

← spring-go → 浏览 提交 分支 合并请求 版本 标签 对比 设置		+ 创建代码分支
<b>活跃分支</b> 保护分支 只读分支 所有分支 过时分支 搜索分支 Q	计算行公式多了1个提交	批量删除 分支设置
ジ master-patch-1 更新于 2 小時前	1 0	d557ab0 <b>①</b> 创建合并请求 设为只读 …
2 <sup>9</sup> master 献以分支 更新于2个月前	0   0	<b>b51ad0d 司</b> 创建合并请求 设为只读 …
1-2个, 共2个		

# 新建分支

单击右上角的**创建代码分支**,在弹窗中按照提示输入相关配置信息,默认以 master 分支作为创建来源。新旧分支内容一致,相当于在原分支的基础上衍生出新 分支。



## 添加备注



### 当分支名称不能完整描述分支用途时,可以添加一个简单的描述信息用以说明分支用途。

← spring-go -	浏览 提交 分支 合并请求 版本 标签 对比 设置		+ 创建代码分支
<b>活跃分支</b> 保护分支	<b>只读分支 所有分支 过时分支</b> 搜索分支 Q		批量删除 分支设置
影 master-patch-1 更新于 2 小时前		1 0	d557ab0 🗊 创建合并请求 设为只读 …
?? master 默认分支 更新于 2 个月前		٥ ٥	
1-2个, 共2个			

# 分支规范

CODING 内置了四种分支规范方案,详细内容请参见 仓库规范。



# 设置默认分支

最近更新时间: 2023-02-10 11:07:15

本文为您详细介绍如何在代码仓库中设置默认分支。

## 进入项目

- 1. 登录 CODING 控制台,单击团队域名进入 CODING 使用页面。
- 2. 单击团队首页左侧的项目,进入项目列表页,选择目标项目。
- 3. 选择左侧菜单**代码仓库 > 分支**,进入分支管理页面。

## 设置分支

默认分支被视为仓库中的基本分支,除非您指定其他分支,否则将基于默认分支自动生成所有拉取请求和代码提交。默认分支可在代码仓库的**设置 > 分支设置**页面 中可修改。

<ul> <li>android -</li> </ul>	浏览 提交 分支 合并请求 版本 标签 对比 设置		
仓库设置	分支设置		
基本设置	默认分支		
权限方案	\$ <sup>₽</sup> master		
部署公钥	默认分支被视为仓库中的基本分支,除非您指定其他分支,否则将基于默认分支自动生成所有拉取请求和代码提交。		
文件锁定	保护分支规则		+ 添加分支规则
分支设置	开启保护分支后,创建合并请求并邀请其他成员评审代码,其他成员「允许合并」后可自行合并分支。		
仓库规范	保护分支规则保护分支	御童	操作
访问设置		暂无数据	
推送设置	隐藏分支		+ 添加隐藏分支
合并请求	说明:隐藏分支访问权限优先级为 用户 > 用户组 > 所有用户,当某个成员属于多个用户组时,分支访问权限以所在的;	未排除的用户组为依据。	
版本发布			
代码标签			
仓库安全			
推送权限		暂无隐藏分支	

#### 当用户所在的权限组具备保护分支权限时,将具备调整默认分支的权限。

<b>项目与成员</b> <sup>项目</sup> 基本设置	◎ 项目经理 系 具备项目协问、测试管理、代码仓库、持续集成、文件管理的访问与操作权限。 配置权限					
菜单管理	分类	功能权限				٠
通知设置	项目协同	<ul> <li>访问项目协同</li> <li>删除事项</li> </ul>	❷ 编辑迭代	🕑 删除迭代	🕑 编辑事项	
成员配置		☑ 访问测试管理	✓ 编辑测试计划	☑ 归档测试计划	✓ 删除测试计划	
成员	测试管理	☑ 编辑用例	─ 删除用例	✓ 编辑报告	✓ 删除报告	
权限组		☑ 执行测试	<ॆ>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>	< 删除自动化用例		
个人偏好	代码扫描	☑ 访问代码扫描	扫描任务管理	扫描问题管理	扫描方案管理	٠
每日工作邮件提醒		☑ 访问代码	创建仓库	仓库设置	☑ 下载代码	
	代码仓库	✓ 拉取代码	✓ 本地推送代码	👽 在线编辑仓库文件	🕑 创建分支和编辑备注	
其他		✓ 删除普通分支	🕑 删除保护分支	🕑 保护分支规则	✓ 创建标签	٠
分类标签		✓ 删除普通标签	☑ 删除保护标签	🛃 创建版本	✓ 编辑版本	
		✓ 删除版本	☑ 创建合并请求	部署公钥	🛃 解锁锁定文件	
		☑ 访问持续集成	✓ 持续集成管理	✓ 手动触发/停止构建	☑ 重置缓存	
	持续集成	✓ 删除构建记录	✓ 创建构建计划	✓ 修改构建计划	💟 复制构建计划	
		✓ 删除构建计划	人工确认			
	持续部署	☑ 访问持续部署	持续部署管理	删除部署记录		۰
	应用管理	☑ 访问应用管理	☑ 应用发布	☑ 应用编辑	数据库变更管理	•
	制品度	访问制品库	制品库设置	● 删除制品仓库	□ 制品扫描	
	市100/牛	拉取制品	删除制品	推送制品		



# 设置保护分支

最近更新时间: 2023-02-07 14:54:30

本文为您详细介绍如何在代码仓库中设置保护分支。

# 进入项目

1. 登录 CODING 控制台,单击团队域名进入 CODING 使用页面。

2. 单击团队首页左侧的**项目**,进入项目列表页,选择目标项目。

3. 选择左侧菜单**代码仓库 > 分支**,进入分支管理页面。

保护分支是 CODING 针对 Git 中有关代码权限开发的一个特色功能,可以将选中的分支保护起来,防止未经报备、允许的更改。

开启后,保护分支在分支列表中将以绿色盾牌为标志。成员修改保护分支时需新建一个分支并在其中进行修改,创建合并请求后邀请其他成员评审代码,评审完成 并允许合并后才能执行合并操作。

<ul> <li>android -</li> </ul>	浏览 提交 分支 合并请求 版本 标签 对比 设置	
仓库设置	分支设置	
基本设置	默认分支	
权限方案	₽ <sup>9</sup> master ~	
部署公钥	默认分支被视为仓库中的基本分支,除非您指定其他分支,否则将基于默认分支自动生成所有拉取请求和代码提交。	
文件锁定	保护分支规则	+ 添加分支规则
分支设置	开启保护分支后,创建合并请求并邀请其他成员评审代码,其他成员「允许合并」后可自行合并分支。	
仓库规范	保护分支规则 保护分支数量	操作
访问设置	智无数据	
推送设置	隐藏分支	+ 添加隐藏分支
合并请求	说明:隐藏分支访问权限优先级为 用户 > 用户组 > 所有用户,当某个成员属于多个用户组时,分支访问权限以所在的未排除的用户组为依据。	
版本发布		
代码标签		
仓库安全		
推送权限	暂无隐藏分支	

设置保护分支规则



← android -	浏览 提交 分支 合并请求 版本 标签 对比 设置
仓库设置	添加保护分支规则
基本设置	分支规则名称
部署公钥	master 添加
文件锁定	匹配到1个保护分支
分支设置	master
仓库规范	
访问设置 推送设置	<b>禁止强制推送</b> 开启禁止强制推送后,将不允许使用 git push –f 强制修改分支历史。
合并请求	开启状态检查
版本发布	开启状态检查后,所有合并到此分支的合并请求必须在全部 CI 任务都通过之后才能合并。 查看设置 CI 触发规则的帮助文档 C
代码标签	自动添加分支管理员为评审者
仓库安全 <sub>推送权限</sub>	开启自动添加后,所有合并到此分支的合并请求,都会自动将分支管理员设置为评审者。当分支管理员人数 大于授权数量时,会从所有分支管理员中随机选取足够的人数。
	开启代码所有者评审
	开启代码所有者评审后。所有合并到此分支的合并请求,如果有修改代码所有者的文件,则必须经过相应的 代码所有者评审之后才允许合并。
	<b>合并请求允许合并授权数量</b> 请选择 ~
	设置合并请求必须有多少位分支管理员授权之后才允许合并到目标分支。

在代码仓库的**设置 > 分支设置**中的分支规则名称中输入需要被保护的分支,符合命名规则的分支都会被视为保护分支。

- 禁止强制推送:默认打开。即使有 git push 的权限,也不允许通过 git push -f 的方式强制修改分支的提交历史。对于多人合作的分支,强烈建议打开此 选项。它确保了只能通过增加新的提交来改变分支内容,而不是修改历史提交的方式来提交变更。
- 开启状态检查:通过在 CI 中设置规范性检查条件或设置代码扫描方案,运行 CI 成功后才被允许合并,更多详情请参见 触发规则 。
- 合并前必须处理完所有的评审讨论:开启此选项后,若评审者的代码评审意见为需要改进时,那么本次合并请求将不允许放行。直到代码作者针对此评审意见 重新发起合并请求,并且评审者再无其他异议后,此合并请求才会被允许通过。



← Angular-demo → 浏览 提交 分支 合并请求 版本 标签 对比 设置	
ズITLX 評事不通过 裕分支 master-patch-3 ① 合并到分支 master ②	
概览 ? 提交记录 ③ 文件改动 ②	+4 ######
> 全部展开 ≪ 全部収起 枠 切換为左右領式 ◆ 高級選項	0 / 2 文件查看 ⑦ 完成评审 🔻
> 4 BERE README.md	完成代码评审
> 0 ■■■■■ 街道 heic	请输入评论内容
	© \$\$ @ O
	评论 提交评审评论
	允许合并 提交评审反馈,并允许合并
	<ul> <li>需要改进 提交评重反馈,并要求修改完善后才允许合并</li> </ul>
	完成评审 取消

- 自动添加分支管理员为评审者:该功能开启之后,针对所有合并到此保护分支的合并请求,都会自动将分支管理员设置为评审者。当分支管理员的数量大于设置的合并请求允许合并授权数量时,会随机选择对应授权数量的分支管理员作为评审者。例如,当配置了3位分支管理员而授权数量为2时,系统会从3位分支管理员中随机选取2位作为评审者。
- **开启代码所有者评审:** 该功能开启之后,针对合并到该保护分支的合并请求,如果存在对代码所有者的文件的修改,则必须经过代码所有者的评审之后才允许 合并。一般而言不同的代码路径决定了应用中的某项功能,这些功能的维护者天然不希望代码被其他成员无意间改动,详情请参见代码所有者。
- 合并请求允许合并授权数量:用于设置合并请求必须经过多少位分支管理员的授权之后才允许合并到目标分支。如果该保护分支没有设置分支管理员,需经过 1 位普通成员授权之后才允许合并。

# 指定分支管理员

分支管理员为可选项。添加管理员后,所有的合并请求需得到管理员的允许才能被允许合并。管理员默认受到保护分支的条件限制,需创建合并请求才可修改分 支。勾选**允许直接推送**后,管理员将不受保护限制,可以直接修改保护分支内容。

<b>分支管理员</b> → 添加分支管理员 分支管理员为可选项。添加分支管理 制、需创建合并请求修改分支、若将	员后,所有合并请求需分支管理员「允许合并」 其设置为「允许直接 Push」则可以直接推送代债	。分支管理员也受保护分支限 <sup>四修改分支,</sup>
管理员	权限	操作
<b>●</b> 阿乔	允许直接推送	移除



若成员没有权限(即保护分支的非分支设置员 )push 至该分支,当其尝试 push 至该分支的时候,会得到如下错误提示:

IN-MB0:/Volumes/CODING/coding-help-generator	ເສ1
/Volumes/CODING/coding-help-generator Paper 1 @2 git push Enumerating objects: 5, done. Counting objects: 100% (5/5), done. Delta compression using up to 12 threads Compressing objects: 100% (3/3), done. Writing objects: 100% (3/3), 378 bytes   378.00 KiB/s, done. Total 3 (delta 2), reused 0 (delta 0) remote: [err=31] You have no permission to update protected branch (refs/heads/spi). remote: 你没有权限推送到保护分支 (refs/heads/api) , 查看更多信息: https://coding. .net/help/doc/git/git-branch.html#保护分支 remote: error: hook declined to update refs/heads/api To https://e.coding.net/codingcorp/coding-help-generator.git ! [remote rejected] api -> api (hook declined) error: failed to push some refs to 'https://e.coding.net/codingcorp/coding_help- generator.git: /Volumes/CODING/coding-help-generator / api 11 @2	a


## 设置隐藏分支

最近更新时间: 2023-02-07 14:54:35

本文为您详细介绍如何设置隐藏分支。

## 进入项目

- 1. 登录 CODING 控制台,单击团队域名进入 CODING 使用页面。
- 2. 单击团队首页左侧的项目,进入项目列表页,选择目标项目。
- 3. 选择左侧菜单**代码仓库 > 分支**,进入分支管理页面。

如需控制单个分支的访问权限,您可以将默认分支以外的任一分支设为隐藏分支,只有经授权的用户或用户组才能访问该分支,保证代码的安全性。

#### 操作步骤

1. 在代码仓库的详情页面,单击**设置 > 分支设置**进入分支设置页面。

← test - 浏	览 提交 分支 合并请求 版本	对比 <b>设置</b>						
仓库设置	分支设置							
基本设置	默认分支							
部署公钥	₿ <sup>9</sup> master							
文件锁定	默认分支被视为仓库中的基本分支,除非您	3指定其他分支,否则将基于默认分支自动	主成所有拉取请求和代码提交。					
分支设置	保护分支规则			+ 添加分支规则				
访问设置	开启保护分支后,创建合并请求并邀请其他成员评审代码,其他成员「允许合并」后可自行合并分支。							
推送设置	保护分支规则	操作						
合并请求	暂无数据							
版本发布	隐藏分支			+ 添加隐藏分支				
代码标签	说明:隐藏分支访问权限优先级为 用户 >	说明: 隐藏分支访问权限优先级为 用户 > 用户组 > 所有用户,当某个成员属于多个用户组时,分支访问权限以所在的未排除的用户组为依据。						
研发规范 beta	隐藏分支	用户组/成员	访问权限	操作				
			暂无数据					

2. 单击添加隐藏分支,选择或输入分支后单击保存。添加成功的隐藏分支会显示在隐藏分支列表。

#### 3. 通过新增用户组或新增成员添加允许或拒绝分支访问的用户。

	加快限以附任时木排陈时用尸组为依据。		
dev/001		+ 新增用户组/成员	<b>會</b> 删除
😬 所有人 (3)	拒绝访问	+ 新增用户组	>
9 项目管理员 (2)	6 允许访问	+ 新增成员	>

🕛 说明

隐藏分支访问权限优先级为用户 > 用户组 > 所有用户。当某个成员归属多个用户组时,若某个用户组具备访问权限,那么其能够访问隐藏分支。 例如,用户 A 同时归属于用户组 1 (允许访问 dev/001 分支)和用户组 2 (拒绝访问 dev/001 分支),在该种情况下用户 A 可以访问 dev/001 分支。



## 设置只读分支

最近更新时间: 2023-02-07 14:54:47

本文为您详细介绍如何在代码仓库中设置只读分支。

### 进入项目

- 1. 登录 CODING 控制台,单击团队域名进入 CODING 使用页面。
- 2. 单击团队首页左侧的项目,进入项目列表页,选择目标项目。
- 3. 选择左侧菜单**代码仓库 > 分支**,进入分支管理页面。

### 设置分支

说明
 只读分支为旗舰版功能。

#### 设置只读分支后,该分支将无法被团队中的任何成员写入或提交合并请求,仅允许拉取。

#### 进入任一代码仓库后,在**分支**栏中设置只读分支。

← android → 浏览 提交 分支 合并请求 版本 标签 对比 设置		+ 创建代码分支
<b>活跃分支</b> 保护分支 只读分支 所有分支 过时分支 分支类型 提索分支 Q		批量删除 分支设置
不允许创建或推送非规范分支,在页面上创建或者本地推送规范分支以外的分支,均会报错,已存在仓库中的非规范分支会有「不规范	5」的标签。	不再显示
診 master NL分支 主干分支 更新子 2022-08-23 16:50:38 ✓	0 0	3902994 🗊 创建合并请求 设为只读 …
診 master-patch-1 不規范 更新子 2022-08-23 16:50:01 ✓	5 1	f76adθc 🗊 创建合并请求 设为只读 …
1-2个, 共2个		

#### 开启只读模式后,将在分支页出现只读标识。

← android → 浏览 提交 分支 合并请求 版本 标签 对比 设置		+ 创建代码分支
活跃分支 保护分支 只读分支 所有分支 过时分支 分支类型 ▼ 提案分支 Q		批量删除 分支设置
不允许创建或推送非规范分支,在页面上创建或者本地推送规范分支以外的分支,均会报错,已存在仓库中的非规范分支会有	「不规范」的标签。	不再显示
※ feature/orbit 功能分支 更新子 2022-08-23 16 50.38 3	0 0	<b>3902994 (引</b> 创建合并请求 取消只读 …
② master 取込分支 主干分支 更新子 2022-08-23 16 50.38  ③	0 0	<b>3902994 ③</b> 创建合并请求 设为只读 …
診 master-patch-1 不現語 更新子 2022-08-23 16:50:01 ✓	5 1	<b>f76ad0c (</b> ] 创建合并请求 设为只读 …
※ new-br 不規范 更新于 2022-04-22 16.16:31 ✓	1 7	<b>2dbde83 词</b> 创建合并请求 取消只读 …
	1 7	<b>3f09863 </b>
》 new 不规范 更新子 2022-03-14 13:59:54	1 7	0e0a0b8 ସ
1-6个, 共6个		



## 代码所有者

最近更新时间: 2023-02-07 14:54:41

本文为您详细介绍如何使用代码仓库中的代码所有者机制。

## 进入项目

- 1. 登录 CODING 控制台,单击团队域名进入 CODING 使用页面。
- 2. 单击团队首页左侧的项目,进入项目列表页,选择目标项目。
- 3. 选择左侧菜单**代码仓库 > 分支**,进入分支管理页面。

代码所有者

代码所有者机制需配合 保护分支 功能一起使用。在代码仓库中放置声明文件 CODEOWNERS 后,就可以声明此仓库内代码文件的所有者,通常为项目负责 人。

当合并请求的目标分支为保护分支,且在请求中的改动文件涉及到声明文件中设定的路径或文件,合并请求详情中将列出对应的所有者与他们的评审状态。



① 说明 亦支持使用持续集成插件自动添加评审者,请参见 自动添加合并请求评审者。



#### 在**保护分支**设置中勾选开启代码所有者评审后,管辖范围的代码变动需通过代码所有者的评审后方能合并。

← Angular-demo -	浏览 提交 分支 合并请求 版本 标签 对比 设置									
仓库设置										
基本设置	禁止强制推送  T  F  F  F  F  F  F  F  F  F  F  F  F  F									
部署公钥										
文件锁定	开启状态检查     〇       旋定     开启状态检查后,所有合并到此分支的合并请求必须在全部 CI 任务都通过之后才能合并。									
分支设置	查看设置 CI 触发规则的帮助文档 亿									
访问设置	合并前必须处理完所有评审讨论									
推送设置	即使已经获取足够的合并请求允许,如果合并请求仍有评审讨论未被标记为已处理,合并请求仍不能合并。									
合并请求	自动添加分支管理员为评审者									
版本发布	开启自动添加后,所有合并到此分支的合并请求,都会自动将分支管理员设置为评审者。当分支管理员人数 大于授权数量时,会从所有分支管理员中随机选取足够的人数。									
代码标签										
仓库安全	<b>开启代码所有者评审</b>									
推送日志	代码所有者评审之后才允许合并。									
推送权限	<b>合并请求允许合并授权数量</b> 1 ▼ 设置合并请求必须有多少位分支管理员授权之后才允许合并到目标分支。									

#### 声明文件地址

代码所有者的声明文件 CODEOWNERS 默认从以下位置逐层检索,文件名必须是大写格式,找到一个后就会停止搜寻。

- 根目录
- docs/ 目录

#### 声明文件格式参考

声明文件是一个普通的文本文件,空行与 # 开始的行会被忽略,每行的格式为:

pattern email email email ...

```
pattern 指定了一个文件路径模式。email 为所有者邮箱,可以填写多个所有者,以空格分隔。
示例文件:
```

```
# 声明所有后缀名是 js 的文件
*.js yourname@coding.net
# 声明仓库根目录下 build/logs/ 目录内的文件(包括子目录)
/build/logs/ yourname@coding.net
# 声明所有 docs/ 文件夹内的文件(不包括子目录)
```

```
docs/* yourname@coding.net
```

```
# 声明所有根目录下 docs/ 文件夹内的文件(包括子目录)
/docs/ yourname@coding.net
```

ð



## 合并请求与代码评审 合并请求设置

最近更新时间: 2023-02-07 14:54:55

本文为您详细介绍如何在代码仓库中调整合并请求设置。

## 进入项目

1. 登录 CODING 控制台,单击团队域名进入 CODING 使用页面。

2. 单击团队首页左侧的项目,进入项目列表页,选择目标项目。

3. 单击左侧菜单栏的代码仓库 > 合并请求 , 进入功能页面。

#### () 说明

合并请求的全称为 Merge Request,下文将简称为 MR。

项目管理员可以在代码仓库的**设置 > 合并请求**页面设置合并请求的基础设置、调整默认合并方式、修改 MR 默认目标分支与设置合并提交信息模板(Accept Merge Request)。

← unit3 - 浏览	现 提交 分支 合并请求 版本 标签 对比 <b>设置</b>
仓库设置	合并请求
基本设置	基础设置
部署公钥	是否默认删除源分支
文件锁定	开启后,合并请求时默认删除源分支,但不强制,如不需要删除源分支,合并者取消勾选即可。
分支设置	是否默认以 Fast-Forward 模式合并
访问设置	开启后,合并请求时默认以 Fast–Forward 模式合并,但不强制,如不需要以 Fast–Forward 模式,合并者 取消勾选即可。
推送设置	
合并请求	合并请求源分支有新提交时目动取消合并授权
版本发布	
代码标签	开启状态检查后,所有合并到此分支的合并请求必须在全部 CI 任务都通过之后才能合并。
仓库安全	
推送权限	合并前必须获得所有评审者的允许合并 所有评审者是指合并请求评审者列表里的所有用户,如果该评审者是强制评审者,则无法从评审者列表中移 除。
	合并方式选择 ⑦ ● 默认直接合并 ○ 默认 Squash 合并 ○ 只能 Squash 合并

### 基础设置

是否默认删除源分支



删除源分支有助于保持仓库整洁,打开此开关后,MR 请求将自动勾选图中的开关。如不需要删除源分支,合并者取消勾选此按钮即可。

合并此请求
Accept Merge Request #12: (new-br -> master)
Merge Request: A new MR
Created By Accepted E URL: https://
<ul><li>✓ 删除源分支</li><li>✓ Fast-Forward 模式合并</li><li>合并分支</li><li>取消</li></ul>

### 是否默认以 Fast-Forward 模式合并

此功能主要用于维护主干分支的整洁。这种模式的合并过程将不会在目标分支中留下曾与源分支进行合并的历史信息。源分支的提交信息将直接与目标分支融合, 而不会产生一个两者曾经发生过合并行为的提交记录,相当于使用 git merge 命令时添加 -ff 参数 。

#### 状态检查

开启此功能后,所有的 MR 将在源分支自动触发持续集成任务,通过后才能允许合并至目标分支。



#### 合并前必须获得所有评审者的允许合并



此功能主要用于确保所有的 MR 经评审者全体同意后才能够被允许合并,无论目标分支是否为保护分支。

<ul> <li>← unit3 → 浏览 提交 分支 合并请求 版本 标签 对比 设置</li> </ul>	
第 #30 fx: Modify README.md 評單送行中, 将分支 [master-patch-1] ③ 合井到分支 [master] ③	
概2 6 提交记录 1 文件改动 1	+3
> 描述 │ ∠ 编辑 没有描述	发起人 ✓ coding
✓ 关联资源 ◎   + 添加资源 点击右上角*+*关联项目资源(迭代、任务、合并请求等)及添加外部链接	评审者 ● 5 ▼   +
合并请求源分支与目标分支无冲突     自动合并可以工作	标签 未选择   (+) 关注者
● 目标分支未开启代码扫描 合并请求未触发扫描任务,无法发现合入代码中潜藏的代码缺陷、安全漏洞,来帮助您评估合入代码质量 前往启用 ☑	未选择 (+)
<ul> <li></li></ul>	
合并分支 关闭 👎 0 👎 0	

## 合并方式

当源分支有多个提交的时候,提供三种合并模式:

- •默认直接合并:会产生一个合并提交。
- 默认 Squash 合并: 会把源分支的多个提交合并成一个提交,用户可以取消这个行为。
- 只能 Squash 合并: 强制把源分支的多个提交合并成一个提交,用户不能取消。

#### 默认目标分支

指定了默认分支之后,在创建合并请求时会自动设置该分支为目标分支。建议使用主干分支为合并请求的默认目标分支。

#### 合并提交消息模板

所有的合并请求都会在目标分支中留下一个提交消息,例如 Git 中的默认消息为 Accept Merge Request #xxx ,您可以在此处修改这条消息模板,更加清晰的记录合并结果。



#### 合并提交消息模版

#### 如果合并时产生一个合并提交,提交消息将默认使用此模版

Accept Merge Request \${id}: (\${source\_branch} -> \$(target\_branch)) Merge Request: \${title}

Created By: \${creator} Reviewed By: \${reviewers} Approved By: \${approvers} Accepted By: \${merger} URL: \${url}

0 / 400

0 / 400

#### 压缩提交消息模版

#### 如果合并时产生一个压缩提交,提交消息将默认使用此模版

Accept Merge Request \${id} : (\${source\_branch} -> \${target\_branch}) Merge Request: \${title} Created By: \${creator}

Reviewed By: \${reviewers} Approved By: \${approvers} Accepted By: \${merger} URL: \${url}

保存



## 发起合并请求

最近更新时间: 2023-02-07 14:55:06

本文为您详细介绍如何发起合并请求。

## 进入项目

1. 登录 CODING 控制台,单击团队域名进入 CODING 使用页面。

2. 单击团队首页左侧的项目,进入项目列表页,选择目标项目。

3. 选择左侧菜单**代码仓库 > 分支**,进入分支管理页面。

在采用多分支开发工作流时,建议开发组长将主分支设置为 保护分支,开发者创建临时开发分支,完成后向主分支发起合并请求。通过持续集成和代码评审后, 开发者将开发分支合并至主分支。

## 创建合并请求

支持通过命令行或在 CODING DevOps 平台手动创建合并请求。

#### 通过命令行创建

git push origin local-branch:mr/target-branch/local-branch

#### 手动创建

1. 在代码仓库详情页面,进入**合并请求**页签,然后单击右上角**创建**。

<ul> <li>← Coding → 浏览 提交 分支 合并请求 版本 对比 设置</li> </ul>	+ 创建合并请求
<b>开启中的 ▼ 开启中的</b> 已关闭的 全部 目标分支 全部 ▼ 发起者 全部 ▼ 标签 全部 ▼ 评审者 全部 ▼	标题或描述 ♀ 技更新时间 ▼ ➡
<b>11 test</b> #6 dev/001 → master ○ 0 条评论	<b>可合并</b> 陈星 创建 于 9 分钟前
±1^	

2. 指定合并请求的源分支与目标分支。当源分支与目标分支对比后没有发现冲突,系统提示源分支可合并到目标分支。您可以通过**文件改动**页签查看文件差异。

# 说明 通过 设置合并请求 可指定合并请求的默认目标分支。

Ð



建合并请求 选择合并请求的源分支和目标分支,并宣看提交记录和文件改动,合并之后目标分支将会包括源分支的改动内容 第分支: dev/001 ▼ 合并到 目标分支: master ▼ 交換 ✓ 可合并	
2 提交记录 1 文件改动 1	
<b>井请求标题 ★</b> 请输入合并请求标题	译审者 未选择   (1
述	标签
编辑 預覧 HBI ♂ " 小 図 ■ 三 三 ☑ @ 井 ✔ ③ 模	未指定 (1
请在此输入合并请求描述内容(支持 Markdown)	<b>关注者</b> 未选择   (·

3. 填写合并请求的标题、描述,关联资源。

① 说明
当源分支的提交不领先于目标分支时,不可创建合并请求。

## 解决不可自动合并的请求

.

当源分支与目标分支对比后发现存在冲突,系统提示源分支和目标分支不可自动合并。您可以在**对比**页签查看文件改动记录,解决冲突后才能继续合并分支。

<ul> <li>node–express–exar</li> </ul>	nple -	浏览	提交 分支	合并请求	版本 对日	比 设置	
创建合并请求							
请选择合并请求的源分支和目标	示分支,并打	查看提交记录	录和文件改动,合邦	f之后目标分支%	将会包括源分支	的改动内容	
源分支: dev/001	- f	合并到 目	标分支: master	Ŧ	交换	× 不可自动合并	
i) dev/001 相对于 maste	r 不可自动 <sup>;</sup>	合并。					

例如:当 branch-01 合并入 master 分支时提示有冲突时,可以先在本地切换至 master 分支并运行命令:

git merge branch-01	ð
找到冲突文件,此时冲突文件会标识冲突内容并询问保留何种内容。选择需保留的内容,保存后重新提交 commit,完成后切换至 branch-01 分支并输	入命

송:

git merge master

再将修改后的代码推送至远端仓库即可。

ð



#### 发起评审

在发起一次分支合并请求时,建议让相关人员参与到代码审视中,以确保准合并代码的正确性。

① 说明	
如果合并请求的目标分支为 保护分支 ,默认会自动添加分支管理员为评审者。如需更改设置,参见 保护分支规则 。	

<b>创建合并请求</b> 请选择合并请求的源分支和目标分支,并重看提交记录和文件改动,合并之后目标分支将会包括源分支的改动内容 源分支: dev/001 ▼ 合并到 目标分支: master ▼ 交換 ✓ 可合并 概览 提交记录 1 文件改动 1	
合并请求标题 ★ 请输入合并请求标题 描述 编辑 预览 H B I ② ** 少 四 単 注 Ⅲ ☑ ⑨ # ✓ ⑨ 模板	<b>评审者</b> 未选择   ◆ ↓ 送筛选 Q :× 清除已选择项
请在此输入合并请求描述内容(支持 Markdown)	● 阿乔 未进择↓ (▲)

评审者在完成 合并请求评审 之后,将会在该合并请求详情页面显示评审结果。

### 检查合并状态

除了上述常见的人工代码审查外,结合 CODING 持续集成,我们还提供了自动化代码评审工具集成的解决方案。基于预定义的规则先行对代码进行扫描,当代码 质量有问题时会拒绝合并代码。只有通过了自动化工具检查的代码才允许合并,显著提高代码审查效率。

#### 开启状态检查

仅 保护分支 支持开启状态检查。勾选之后,要求状态检查(CI 任务)全部运行通过后才允许合并。

← coding-help-generator ▼ 浏览 提交 分支	合并请求 版本 对比 设置	+ 创建代码分支
活跃分支 保护分支 所有分支 过时分支 搜索分支	Q 搜索条件:无	分支设置
<ul> <li>✓ master 默认分支</li> <li>更新于 3 小时前 ✓</li> </ul>	0 0	<b>75b2b60 司</b> 创建合并请求 …
<ul> <li>☑ api …</li> <li>更新于 8 天前 ✓</li> </ul>	32 91	<b>18f7ad0 </b>
1-2个, 共2个		



#### 在持续集成中的触发规则内需要选中**创建合并请求时触发构建**,这样才能在创建合并后立即触发构建任务。

<b>⑥</b> 项目概览		← 人工确认 ☑ │ 基础信息 流程配置 触发规则 变量与缓存 通知提醒
☑ 项目协同		
<♪ 代码仓库		CODING 持续集成支持通过多种方式来触发构建计划,查看完整帮助文档 [2
∞ 持续集成	~	代码源触发 🖌 代码更新时自动执行
构建		选择需要触发持续集成的事件
↓ 持续部署	>	● 推送到 master ▼ 时触发构建
毌 制品库		推送新标签时触发构建
囚 测试管理	>	○ 推送到分支时触发构建
立 任务(旧版)	>	○ 符合分支或标签规则时构建 ⑦
文档管理	>	合并请求
		合并请求触发会构建源分支与目标分支合并后的结果, 能够尽可能早地发现集成中的错误,查看完整帮助文档 <sup>[2]</sup>
		✓ 创建合并请求时触发构建
		✔ 合并合并请求时触发构建
		✓ 源分支变更时触发构建
		✓ 目标分支变更时触发构建
		✓ 自动取消相同合并请求 ②

#### 查看状态检查结果

完成上述设置后,在正确触发构建任务的情况下,可以看到合并检查的状态。如果没有显示如图界面,可能是没有在 Cl 构建任务中选择**创建合并请求时触发构 建**。



← coding-help-generator → 浏览 提交 分支 合并请求 版本 对比 设置	
<ul> <li>第802</li> <li>docs: 更新文档</li> <li>执行状态检查 将分支 mr/master/issue/765 副 合并到分支 master 副</li> </ul>	
<b>概览</b> 5 提交记录 7 文件改动 15	
<ul> <li>✓ 描述</li> <li>Ø 编辑</li> <li>docs: 更新文档</li> </ul>	发起人
<ul> <li>✓ 关联资源 0 ▲ ★添加资源</li> <li>点击右上角"+"关联项目资源(迭代、任务、合并请求等)及添加外部链接</li> </ul>	▼   + 标签  + 添加
○ 合并请求分支与目标分支无冲突 自动合并可以工作	<b>关注者</b> 未选择 │ <b>+</b>
合并状态检查:进行中         〇           1个合并检查进行中         〇	
<ul> <li></li></ul>	
(码扫描未自动触发 代码扫描可以发现合入代码中潜藏的代码缺陷、安全漏洞,帮助您评估合入代码质量。 前往启用 区	
<ul> <li>         保护分支: 等待其他成员授权     </li> <li>         目标分支开启了保护分支且有分支管理员:分支管理员可以合并任何 MR,非管理员发起的 MR 需要至少有 0 位分支管理员授权后才可自行合并     </li> </ul>	

您可以单击右上角的刷新按钮随时获取最新状态,成功后会在下方出现分支已经合并的提示,失败则会拒绝合并。状态检查有四种状态:

- 进行中: 您可以耐心地等待构建完成。
- 成功:此时合并请求可以正常合并。
- 失败:构建过程发生错误,合并检查不通过,您可以修改代码、推送触发新的构建任务直到构建成功完成。
- 异常:构建过程发生了异常,可以尝试进行手动触发。



如果存在多个状态检查的情况下,只有所有的状态检查都成功通过后才会允许分支合并。您也可以在代码浏览、提交历史、分支列表中查看状态检查过程。

← coding-help-generator ◄ 浏览	提交 分支	合并请求 版本	对比 设置		+ 创建代	6.666 ▼
✿ coding-help-generator	😵 master 👻	♠ 查找文件 ✔	输入以查找文件			瀏克隆
> scaffolds	<b>六</b> 件 正中 (1005				0	¥:=+
> scripts	<b>又件</b> 历史 1360				Cloud Studio 11 回建合	开頃水 …
> isource	升级				✔ 最后提交 75b2b60115	<b>司</b> 于 3 小时前
> hemes/coding-help	scaffolds	<b>状态检查:成功</b> 1 个状态检查成功		0		2 年前
🔄 .fnlint.json	scripts		Th	244年	pdate-time2 -> master)	9 个月前
🗋 .git-pre-commit		✓ dev — 构建成	(/ـــ	注情		2 小时前
🗋 .gitignore	Source			71928310374 00.24 -> 00.01		0,1,0,1,0
🗋 .lintmdrc	themes/coo	ling-help	1.64	Accept Merge Request #769 :	(mr/master/issue/765	20 天前
🗋 Jenkinsfile	im .fnlint.json		**	ci: lint filename slug		1 年前
MI OPENAPI-README.md	🗋 .git-pre-co	mmit	**	ci: lint filename slug		1 年前
MI README.md	🗋 .gitignore		1.8*	Accept Merge Request #266 :	(commit-standary -> m	9 个月前
L _config.yml	🗋 .lintmdrc		**	docs: lint markdown		1 年前
🕒 api–Jenkinsfile	🗋 Jenkinsfile		**	refactor: #546 use changeRequ	Jest	4 个月前
🔤 api.json				docs: ani 再新流程修改		2 个日前
[]₃ generate–overview.js	WW OFENAFI-F	EADME.Ind		0003. apr 史洞加住修成		2 1 / 1 / 1
🞰 package-lock.json	M↓ README.m	d	1.64	docs: readme 文件更新		6 个月前
En package.json	🗋 _config.yml		**	fix: #261 hexo chinese anchor l	JRL	9 个月前
🖞 yarn.lock	🗋 api-Jenkins	file	1.87	Accept Merge Request #420 :	(mr/master/issue/419	6 个月前

## 确认合并

• 合并的目标分支为保护分支

若合并请求的发起者为分支管理员,那么可以自行合并。若发起者为普通成员,那么需要通过分支管理员的评审才能完成合并。

• 合并的目标分支为非保护分支

发起者无需经过评审与授权,可以自行发起并完成分支合并。

```
() 说明
```

如需了解如何修改默认分支与设置保护分支,请参见 设置默认分支 或 设置保护分支。

#### 删除源分支

合并分支时,勾选删除源分支,可在合并分支时删除源分支。

合并此请求	
Accept Merge Request #79: (new-feature -> master)	
Merge Request: demo Created By: @主账号	
Accepted By: @主账号 URL: https://straybirds.coding.net/	
✓ 删除源分支 Fast-Forward 模式合并	

### Fast-Forward 模式合并

常规的分支合并时会默认产生一个合并提交记录。若勾选了 **Fast-Forward 模式合并**,远端仓库会在合并时判断是否符合 Fast-Forward 规则。若符合则此 合并不会产生新的合并提交记录;若不勾选此模式,则会在合并时保留过往开发记录并产生一个新的合并记录。此选项相当于使用 git merge 时添加 -ff 参



数。

🕛 说明

通过设置合并请求可实现合并请求默认删除源分支及默认以 Fast-Forward 模式合并。



## 评审合并请求

最近更新时间: 2023-07-26 17:18:21

本文为您详细介绍如何在代码仓库中评审合并请求。

### 进入项目

- 1. 登录 CODING 控制台,单击团队域名进入 CODING 使用页面。
- 2. 单击团队首页左侧的项目,进入项目列表页,选择目标项目。
- 3. 选择左侧菜单代码仓库 > 合并请求,进入合并请求功能页面。

#### 发起评审

在发起一次分支合并请求时,建议让相关人员参与到代码审视中,以确保准合并代码的正确性。如果合并请求的目标分支为保护分支,默认会自动添加分支管理员 为评审者。如需更改设置,请参见 保护分支规则 。

创建合并请求	
请选择合并请求的源分支和目标分支,并查看提交记录和文件改动,合并之后目标分支将会包括源分支的改动内容	
源分支: dev/001 ▼ 合并到 目标分支: master ▼ 交換 ✓ 可合并	
概览 提交记录 1 文件改动 1	
合并请求标题*	评审者
请输入合并请求标题	未选择   +
描述	快速筛选 Q
编辑 预览 HBI & "小 四 里 三 三 回 @ 井 ノ ⑦ 模板	·∞ 清除已选择项
请在此输入合并请求描述内容(支持 Markdown)	主选择 (二)

评审者在完成合并请求评审之后,将会在该合并请求详情页面显示评审结果。

#### 评审内容

评审者收到邀请评审通知后,对代码进行评审的内容一般会聚焦于以下两点:

- 该合并请求的标题、描述及关联资源是否对代码改动作出充分说明,评审者可通过评论和发起者进行沟通确认。
- 针对提交记录对应的文件改动,进行代码行级评审(评论)。

### 开始评审

在**合并请求**的**文件改动**页签,代码评审者可以针对代码文件进行逐行评论以完成评审。鼠标悬停在代码文件中的某一行,单击界面中的蓝色 + 号填写评论。

#### 直接评论

#### 单击评论则直接发布该评论。

★ template-1 → 浏览 提交 分支 合并请求 版本 标签 对比 设置	
1) #33 feat:update port number 可合并 将分支 master-patch-1 (1) 合并到分支 master (1)	
概览 ④ 提交记录 ② <b>文件改动</b> ①	+1 -1
⇒ 全部収起 村 切換为普通模式 ◆ 高級追項	0/1文件显者 ⑦
V 2 E READWE.md	日 日 立 香 一 一 日 立 香 一 一 一 日 立 香 一 一 一 一 一 一 一 一 一 一 一 一 一 一 一 一 一 一
	36 37 4. 亜信 Toncat 38
39 5. 11 http://127.0.0.1:8080/ .	39 5. a at http://127.0.0.1:80/ .
	对+39 行添加评论
	編輯 預え <b>D</b> H B <i>I &amp; P + 4</i> K2 四 注 Ξ ビ Ø # ✓ ② 请检查。
	评论 取消



#### 代码建议

代码建议功能可以让评审者直接上手修改合并请求中的代码,告别与提交者之间反复沟通代码应该如何修改。

← template-1 → 浏览 提交 分支 合并请求 版本 标签 对比 设置		
33 #33 feat:update port number 町倉井 将分支 master-patch-1 ① 合并到分支 master ③		
概览 ② 提交记录 ① <b>文件改动</b> ①		+1 -1
◇ 全部展开 灸 全部收起 村 切换为普通模式 ♀ 高级选项	0 / 1	文件查看 ⑦
✓ 2 ■■■■ README.md           36           37         4. 重目 Tomcat           38           39         5. 打开 http://127.0.0.1:8080/.	36 37 4. 重音 Toscat 38 39 5. 打开 http://127.0.0.1:編3/. 对 +39 行酒加评论 编辑 预宽	<u>- 已京者</u> …

#### 合并请求发起者单击应用建议后,将直接在源分支上提交新代码覆盖旧有内容。

← template-1 → 浏览 提交 分支 合并请求 版本 标签 对比 设置	
<b>31</b> #33	
feat:update port number	
可合并 将分支 master-patch-1 ③ 合并到分支 master ③	
概览 ③ 提交记录 1 文件改动 1	+1 -1
> 全部展开 <>> 全部收起  ♦ 切块为普通模式	0 / 1 文件發者 ⑦
✓ 2 ■■■■■ README.md	✓ 显示评论
<u>.1.</u>	
3b 37 4 曲户 Tampat	3b 37 4 車中 Tampat
38	38
39 5. 打开 http://127.0.0.1:8080/ .	39 5. 打开 http://127.0.0.1:443/ .
	✓ coding 几秒前
	修改建议
	5 1TH http://127.0.0.1.442/
	+ 5. 打开 http://127.0.0.11:80/ .
	应用建议。
	Ø 0 <u>2</u>

代码建议发布成功之后,评论内容和评审状态会同步至合并请求概览页面的活动日志。





~ 活动	日志 民看日志 民看诗论	
+	coding 创建了合并请求	2022-08-09 16:21:19
ß	coding 更新了标题	2022-08-09 16:21:37
Q	coding 对过时的文件改动发起了讨论	2022-08-09 16:36:05
	> README.md	
¢	coding 推送了新的提交,更新了合并请求	2022-08-09 16:41:58
Q	coding 对文件改动发起了讨论	2022-08-09 16:59:27
	✓ README.md	
	36 37 37 <b>4. 重启 Toncat</b> 38 38	
	39       5. 打开 http://127.0.0.1:8080/.         39       5. 重 抽 http://127.0.0.1:80/.	
	✓ coding 2 分钟前 请检查。	
	请输入评论内容(译 + Enter 直接评论)	

#### 会议评审

评审合并请求时支持一键发起腾讯会议,快速和与会者对齐合并内容能够促进生产与提高沟通效率。

	新版仓库使用引导 🖸
\$1 #4 合并分支修改 test1 可合并 将分支 test1-patch-1 副 合并到分支 test1 副	
<b>概览 1</b> 提交记录 1 文件改动 1	+2 -1
<ul> <li>✓ 描述 / </li> <li>② 编辑     没有描述     </li> </ul>	发起人
✓ 关联资源 0 │ + 添加资源 点击右上角"+"关联项目资源(迭代、任务、合并请求等)及添加外部链接	<b>评审者</b> 未选择│ <b>+</b> 标签
< 合并请求源分支与目标分支无冲突 自动合并可以工作	未选择   + 关注者
● 目标分支未开启代码扫描 合并请求未触发扫描任务,无法发现合入代码中潜藏的代码缺陷、安全漏洞,来帮助您评估合入代码质量 前往启用	未选择 ◆
✔ 非保护分支 目标分支为非保护分支,可由发起者自行合并 MR	<ul><li></li></ul>
合并分支 关闭 关闭	

授权访问及邀请与会人入会的方法可以参考 快速发起腾讯会议。

### 评审进度

当需要评审多个代码文件时,可在评审完一个文件之后单击**已查看**进行跟踪与标记,文件查看进度条会随之自动更新。

 说明: 文件已查看并不影响评审状态,仅用于跟踪文件查阅进度,且只针对当前用户生效。



← test-0 → 浏览 提交 分支 合并请求 版本 对比 设置		
11 #2 Merge feature/add-chars into master 可合并 将分支 feature/add-chars ③ 合并到分支 master ③		
概览 7 提交记录 3 <b>文件改动</b> 3		+84
※ 全部展开 《 全部收起 ∲ 切换为左右模式 ✿ 高级选项	2 / 3 文件查看	⑦ 完成评审 1 ▼
> 36 mmmm find-common-chars.js		◎ 1 🔽 已查看 …
> 24 mmm fixed-pointers.js		✓ 已查看 …
> 24 semes symmetric-tree.js		已查看 …

## 完成评审

评审过程中相关代码建议仅自己可见。待所有代码文件均已被评审之后,单击右上角**完成评审**发布评审结果供其他成员参考。

① 说明: 完成评审上的数字代表当前页面所有代码评论的数量。
-----------------------------------

- 评论: 必须填写评论。评论内容为填写内容。
- **允许合并:**可不输入评论,直接发布评审结果为:**允许合并**。
- 需要改进: 可不输入评论,直接发布评审结果为:需要改进。

	3 / 3 文件查看	0	完成评审 2 🔻
完成代码评审 2个评审中			
请输入评论内容			
			◎ > @ ⊙
<ul> <li>评论</li> <li>提交评审评论</li> </ul>			
允许合并 提交评审反馈,并允许合并			
<b>需要改进</b> 提交评审反馈,并要求修改完善后才允许合并			
完成评审 取消			



#### 完成评审之后,评论内容或评审结果会同步显示在合并请求概览页面的活动日志。

¢	<b>test-0 ▼</b> 浏览 提交 分支 合并请求 版本 对比 设置	
<b>11</b> M	erge feature/add-chars into master	
可合并	· 将分支 feature/add-chars <b>司</b> 合并到分支 master <b>司</b>	
	1         发表了评论           请继续检查。         请继续检查。	
	> find-common-chars.js	
	✓ fixed-pointers.js	
	<pre>1 /** 2 * @param {TreeNode} root 3 * @return {boolean}</pre>	
	9 分钟前 OK。	Đ
	• D2	
	标记为已处理	
	B 完成了代码评审,评审结果为:允许合并 几秒前	

若合并请求的目标分支为保护分支,将会显示合并请求的评审状态(只要存在**需要改进**的评审结果,评审状态即为不通过)。

← test-0 → 浏览 提交 分支 合并请求 版本 对比 设置	
<b>11</b> #2	
Merge feature/add-chars into master	
<b>可合并</b> 将分支 feature/add-chars <b>司</b> 合并到分支 master <b>司</b>	
◇ 关联资源 ◎ │ + 添加资源	
点击右上角"+"关联项目资源(迭代、任务、合并请求等)及添加外部链接	
✔ 合并请求源分支与目标分支无冲突 自动合并可以工作	
目标分支未开启代码扫描 合并请求未触发扫描任务,无法发现合入代码中潜藏的代码缺陷、安全漏洞,来帮助您评估合入代码质量	前往启用 🖸
评审状态:不通过           1 个评审结果:需要改进,1 个评审结果:允许合并	
• (三)       评审结果:需要改进         • (回)       评审结果:允许合并	
保护分支: 等待其他成员授权           目标分支开启了保护分支且有分支管理员:分支管理员可以合并任何 MR,非管理员发起的 MR 需要至少有 1 位分支管理员授权后才可自行合并           ■         已允许合并	
撤销允许合并 合并分支 关闭 关闭	👍 0 👎 0

### 检查合并状态

除了上述常见的人工代码审查外,结合 CODING 持续集成,我们还提供了自动化代码评审工具集成的解决方案。基于预定义的规则先行对代码进行扫描,当代码 质量有问题时会拒绝合并代码。只有通过了自动化工具检查的代码才允许合并,显著提高代码审查效率。

#### 开启状态检查

仅 保护分支 支持开启状态检查。勾选之后,要求状态检查(CI 任务)全部运行通过后才允许合并。在持续集成中的触发规则内需要选中**创建合并请求时触发构** 建,这样才能在创建合并后立即触发构建任务。



	← Android-test ② 基础信息 流程配置 触发规则 变量与缓存 通知提醒						
<ul> <li>← 持续集成</li> <li>构建计划</li> </ul>	CODING 持续集成支持通过多种方式来触发构建计划,查看完整帮助文档 IZ 代码源触发 V 代码更新时自动执行 选择需要触发持续集成的事件						
构建节点	推送到 master       时触发构建         推送新标签时触发构建       推送到分支时触发构建         推送到分支时触发构建       (heads/content of the state of						
	<ul> <li>✓ 创建合并请求时触发构建</li> <li>✓ 合并合并请求时触发构建</li> <li>✓ 源分支变更时触发构建</li> <li>✓ 目标分支变更时触发构建</li> <li>✓ 自动取消相同合并请求 ⑦</li> </ul>						
	定时触发     分支     执行时间     操作						
	API 触发       触发地址       https://zzz7.coding.net/api/cci       ①       生成 curl 命令触发示例         需使用具有持续集成 API 触发权限的项目令牌触发						
	手动触发 指定立即构建的默认构建目标 master 🔻						

#### 查看状态检查结果

完成上述设置后,在正确触发构建任务的情况下,可以看到合并检查的状态。如果没有显示如图界面,可能是没有在 Cl 构建任务中选择**创建合并请求时触发构 建**。



<b>1</b> #802					
docs: 更新文档					
执行状态检查 将分支 mr/master/issue/765 司 合并到分支 master 司					
概览 5 提交记录 7 文件改动 15					
◇ 描述 │ ◇ 编辑	发起人				
docs: 更新文档	<b>6</b>				
	评审者				
✓ 关联资源 0 + 添加资源	+ ا *				
点击右上角"+"关联项目资源(迭代、任务、合并请求等)及添加外部链接	标签				
	+ 添加				
✓ 台开请求分支与目标分支无冲突 自动合并可以工作	关注者				
	禾选择│(+)				
合并状态检查:进行中					
代码扫描未自动触发					
代码扫描可以发现合入代码中潜藏的代码缺陷、安全漏洞,帮助您评估合入代码质量。    前往启用 🖸					
保护分支:等待其他成员授权 目标分支 市局了保护分支目有分支管理局:分支管理局可以合并任何 MR 非管理局发起的 MR 需要至少有 0 位分支管理局授权后才可自行合并					

您可以单击右上角的刷新按钮随时获取最新状态,成功后会在下方出现分支已经合并的提示,失败则会拒绝合并。状态检查有四种状态:

- 进行中: 您可以耐心地等待构建完成。
- 成功:此时合并请求可以正常合并。
- 失败:构建过程发生错误,合并检查不通过,您可以修改代码、推送触发新的构建任务直到构建成功完成。
- 异常:构建过程发生了异常,可以尝试进行手动触发。



如果存在多个状态检查的情况下,只有所有的状态检查都成功通过后才会允许分支合并。您也可以在代码浏览、提交历史、分支列表中查看状态检查过程。

← coding-help-generator ▼       浏览 提交 分支 合并请求 版本 对比 设置       + 创建代码仓库 ▼					码仓库 🚽	
✿ coding-help-generator \$P master ▼ ▲ 首找文件 ∨ 输入以查找文件 器克						I 克隆
> 🖿 scaffolds	文件 压电 1260			E	Cloud Studio 91 All建合	计连步 …
> 🖿 scripts						<b>叶</b> ·明·尔
> 🖿 source	6 计组				■ 🔽 最后提交 75b2b60115 🕻	<b>]</b> 于3小时前
> 📔 themes/coding-help	scaffolds	<b>状态检查:成功</b> 1 个状态检查成功		C		2 年前
inlint.json	scripts	/ doy — 构建成	Th	(光)時	pdate-time2 -> master)	9 个月前
🕒 .git-pre-commit	source		-/]	升级时间从 05.24 -> 06.01		3 小时前
🗋 .gitignore						
🗋 .lintmdrc	themes/cod	ling-help	184	Accept Merge Request #769 :	(mr/master/issue/765	20 天前
] Jenkinsfile	ifnlint.json.		6.6	ci: lint filename slug		1年前
MI OPENAPI-README.md	🗋 .git-pre-cor	nmit	6.0	ci: lint filename slug		1年前
MI README.md	🗋 .gitignore		2.8*	Accept Merge Request #266 :	(commit-standary -> m	9 个月前
Config.yml	lintmdrc		8.8	docs: lint markdown		1年前
🗋 api-Jenkinsfile	🗋 Jenkinsfile		8.8	refactor: #546 use changeRequ	est	4 个月前
ison api.json				doogo ogi 再新运程修改		0 公日前
[]₅ generate–overview.js	MI OPENAPI-R	EADME.IIId		uous. api 更新加在珍女		21月前
ban package-lock.json	M↓ README.mo	Ł	184	docs: readme 文件更新		6 个月前
be package.json	Config.yml		-	fix: #261 hexo chinese anchor L	IRL	9个月前
l yarn.lock	api-Jenkins	file	2.87	Accept Merge Request #420 :	(mr/master/issue/419	6 个月前

在新建合并请求或合并请求合并前,开发者可更新标题和描述,添加其他项目成员为评审者,并关联项目内资源(如任务、文件、合并请求、Wiki)。您还可以通 过 CI 插件 自动添加评审者 。

#### () 说明:

如果合并请求的目标分支为保护分支,且为保护分支配置了管理员并开启**自动添加分支管理员为评审者**,对应授权数量的分支管理员会自动添加为评审 者。



## 版本与标签 代码版本

最近更新时间: 2023-02-10 11:07:33

本文为您详细介绍如何使用代码仓库中的代码版本功能。

## 进入项目

1. 登录 CODING 控制台,单击团队域名进入 CODING 使用页面。

2. 单击团队首页左侧的项目,进入项目列表页,选择目标项目。

3. 单击左侧菜单栏的代码仓库 > 版本,进入版本功能页面。

代码版本可以理解为代码仓库某一时刻的快照。在代码仓库管理列表,单击指定代码仓库进入其详情页面之后,再单击**版本**进入管理发布页。

← android → 浏览 提交 分支 合并请求 版本 标签 对比 设置	+ 创建代码版本
<b>全部 →</b> 状态 全部 → 负责人 全部 → 标签 全部 → 创建日期 开始日期 → 至 结束日期 →	标题或描述 ♀ 技更新时间 ~ =↓
	<b>预发布</b> 创建于几秒前
共1个	

版本发布列表按创建时间倒序列出项目已发布的代码版本,并展示了代码版本对应的标签名、提交的代码版本等信息。

#### 新建代码版本

1. 在版本管理页面,单击右上角的创建代码版本。

说明
 您可以在代码仓库设置页面 设置版本发布默认分支。

2. 输入标签版本、版本发布标题、版本描述等,支持上传 100 MB 以下所有格式的文件以及关联项目内的资源(例如任务、文件、Wiki、合并请求)。其中, 标签名只能输出新标签名,需要给新标签版本选择创建来源(分支、标签或修订版本)。

← android ▼ 浏览 提交 分支 合并请求 版本 标签 对比 设置	
创建代码版本发布	摘要
标签版本 * 创建来源 *	版本格式: 主版本号.次版本号.修订号,版本号通增规则如下: 1. 主版本号: 当你做了不兼容的 API 修改;
v1.3 @ §9 master ~	<ol> <li>次版本号:当你做了向下兼容的功能性新增;</li> <li>修订号:当你做了向下兼容的问题修正。</li> <li>先行版太是及版本编译使自可以加到"主版太是"次版本是 修订是"的</li> </ol>
版本发布标题	后面,作为延伸。
这是一个即将发布的版本	
描述	
编辑 预览 H B I ℓ " ψ 図 Ⅲ Ⅲ ☑ @ # ✔ ⑦	
条件满足后将尽快发布。	
▲ 点击或将文件拖拽至此上传!	
支持所有文件格式,单个文件不超过100M。	

 上图中右侧摘要内容支持自定义修改,可作为团队公告板告知如何填写合乎规范的版本号。团队负责人或管理员可以前往团队设置中心 > 功能设置 > 仓库设置 中自定义摘要中的内容。



Q 搜索设置项	代码仓库 / 仓库设置 仓库设置 统一设置您的团队中所有仓库的部分设置	
<ul> <li>              代码仓库      </li> <li>             プになる</li> <li>             プロ人が新聞していていていていていていていていていていていていていていていていていていてい</li></ul>	#送设置         一 检查 Git 提交的提交者 (Committer) 和提交作者 (Author) 必须是已验证的邮箱。         一 单次提交的文件总大小不能超过 0       MB, Git LFS 文件除外。         保存         固定义版本创建页面右侧摘要内容         版本格式:         请遵循相应的版本规范。	
	保存           超建更确要           质签版本格式           6 / 400	

4. 在填写以上信息后,您可以标记为该版本为预发布状态,然后创建代码版本发布。代码版本发布后,显示情况参考下图。



## 编辑代码版本



在版本发布列表单击任一代码版本进入其详情页面之后,版本发布创建者或项目管理员单击**编辑**版本描述,即可编辑除标签名以外的其他所有信息。

🔶 a	ndroid -	汶	览 提ぶ	δ 分支	合并请求	版本	标签	对比	设置
	<b>这布的版本</b>	布于 1 分	钟前	🗣 v1.3 -œ-	8b21ae08				
<b>描述</b> 条件满》		<b>一</b>							
<b>引用(</b> 点击"+ <sup>*</sup>	0 "关联项目资》	原(迭代	、任务、合	并请求等)	及添加外部链	妾			
下载 [ 『 源	代码 (zip)	: 源	代码 (tar.gz	z)					

## 删除代码版本

在代码版本详情页面,版本发布创建者或项目管理员单击**删除版本**按钮,即可删除该版本。

<ol> <li>说明</li> <li>在删除版本时,可同时将对应的版本标签删除。只有删除了对应的版本标签,才能使用或创建同名的标签。</li> </ol>	
₩ #18 预发布内容	删除版本
前双右  - 💽 发布于 8 分钟前   💊 release01 - 🗢 8541b4ff	
描述 ク 編組	
没有描述	
	+ 添加资源
点击"+"关联项目资源(迭代、任务、合并请求等)及添加外部链接	<ul> <li></li></ul>
下载	+ 上传文件
豫代码 (zip)     豫代码 (tar.gz)	

## 设置版本发布默认分支



#### 前往代码仓库的**设置 > 版本发布**页面,项目管理员可以指定版本发布时所选择的默认分支。

← test - 浏览	提交	分支	合并请求	版本	对比	设置
仓库设置	版本发	<b>え</b> 布				
基本设置	版本发	布默认分支	支			
部署公钥	ş		Ŧ	保存	7	
文件锁定	新建版	本发布时,	目标分支会點	认选择出	:分支	
分支设置						
访问设置						
推送设置						
合并请求						
版本发布						
代码标签						
研发规范 beta						



## 代码标签

最近更新时间: 2023-02-10 11:04:41

本文为您详细介绍如何管理代码仓库中的代码标签。

## 进入项目

- 1. 登录 CODING 控制台,单击团队域名进入 CODING 使用页面。
- 2. 单击团队首页左侧的**项目**,进入项目列表页,选择目标项目。
- 3. 选择左侧菜单代码仓库 > 分支,进入分支管理页面。

在代码仓库管理列表,单击指定代码仓库进入其详情页面之后,再单击**标签**进入代码标签管理列表。

← android → 浏览 提交 分支 合并请求 版本 标签 对比 设置	+ 创建标签
◇ v1.3 … ◇ 8b21ae0 副 Lar.gz   版本描述   (○ 15 分钟朝)	编辑版本描述
◇ v1.2 ◇ 8b21ae0 司 管 zip 管 tar.gz ≧ 版本描述 ① 20 分钟前	编辑版本描述
◇ v1.0 ◇ 8b21ae0 司 管 zip 管 tar.gz ③ 1 个月前	创建版本描述
1-3个, 共3个	

标签列表显示了该仓库所有标签,按照创建顺序倒序排列。标签列表显示了标签名、标签说明、标签对应版本,并提供 zip 和 tar.gz 下载入口和删除标签入口。 单击标签名或者版本号可进入对应的代码版本详情页。

#### 新建标签

在标签管理列表,单击右上角**创建标签**,输入标签名并选择待创建标签对应的代码版本(分支、标签、修订版本号)即可创建新标签。页面右侧展示的**摘要**可作为 公告板,告知团队成员如何书写统一格式的代码标签。

创建标签			摘要
			版本格式: 主版本号.次版本号.修订号, 版本号递增规则如下:
标金服本 *		J)建米ぷ ▲	1. 主版本号: 当你做了不兼容的 API 修改;
请输入标签版本	0	°g master →	2. XXXX4号·当协取了同下兼容的初期注制相, 3. 修订号: 当你做了向下兼容的问题修正。
描述			先行版本号及版本编译信息可以加到"王版本号·坎版本号·修订号"的后面,作为 延伸。
添加额外说明(不支持 Markdo	wn)		

团队负责人或管理员可以前往团队设置中心 > 功能设置 > 仓库设置中自定义摘要中的内容。



		代码仓库 / 仓库设置	
		仓库设置	
Q 搜	搜索设置项	统一设置您的团队中所有仓库的部分设置	
🗲 f	代码仓库	<b>推送设置</b> 开启后将会强制覆盖团队中所有仓库的相同设置点	
ť	全库设置	● 位置 Git 提交的提交者 (Committer) 和提交作者 (Author) 必须是已短证的邮相。	
Ø	团队部署公钥		
ť	全库规范	休仔	
		<b>版本创建页摘要</b> 自定义版本创建页面右侧摘要内容	
		版本格式: 请遵循相应的版本规范。	
			17 / 400
		保存 标签创建页摘要 自定义标签创建页面右侧摘要内容	
		标签版本格式	
			6 / 400
		保存	

若希望代码标签严格遵守团队规范,那么可以在代码仓库设置页面 设置保护标签,以限制成员创建或修改特定格式的代码标签。

<ul> <li>android -</li> </ul>	浏览 提交 分支 合并请求 版本 标签 对比 设置	
仓库设置	代码标签	
基本设置		
权限方案	木勾述时,任何人都尤法删除桥金,已括尤加进过 git push -1 米強制删除标金。	
部署公钥	保护标签规则	+ 添加标签规则
文件锁定	保护标签用于控制指定的标签模式只能由授权的标签管理员创建或更新。	
分支设置	保护标签规则	操作
仓库规范	V1.5	编辑删除
访问设置	prod-*	编辑删除
推送设置		
合并请求		
版本发布		
代码标签		
仓库安全		
推送权限		

## 删除标签



#### 在**标签**页面,标签创建者或管理员只能删除未与任何版本发布对应的代码标签。

<ul> <li>android -</li> </ul>	3	刘览 提注	交 分支	合并请求	版本	标签	对比	设置
✓ v1.3 ···· -0- 8b21ae0	🖺 zip	🖺 tar.gz	🖹 版本描述	③ 17 分钟前				
✓ v1.2 ···· -> 8b21ae0 □	🖺 zip	🖺 tar.gz	🖻 版本描述	③ 22 分钟前				
✓ v1.0 ···· -> 8b21ae0 □	🖺 zip	🖺 tar.gz	① 1 个月前	<del>ت</del>				
1-3个, 共3个								

若任一标签与版本发布关联,只能在**版本发布**删除对应版本的同时将标签删除。若不希望标签被删除,项目管理员可以前往代码仓库的<mark>设置 > 代码标签</mark>页面,可以 勾选是否允许删除标签。若不允许删除标签,那么所有项目成员将无法通过命令行删除标签,同时网页上的标签不提供删除功能。

🗲 test - 浏览	近 提交 分支 合并请求 版本 对比 <b>设置</b>	
仓库设置	代码标签	
基本设置	✓ 是否允许删除或者强制推送 Git 标签 关闭后 则所有项目成员均不可删除或考证讨案户端强制推送修改 Git 标答 同时页面代码版本模块下的标签不提供删除功能	
部署公钥		
文件锁定	保护标签规则	+ 添加标签规则
分支设置	保护标签用于控制指定的标签模式只能由授权的标签管理员创建或更新。	
访问设置	保护标签规则	操作
推送设置	暂无数据	
会社语少		
57月7		
版平反中		
代码标签		
研发规范 beta		

## 设置保护标签

保护标签主要用于规范特定的成员进行创建、更新或删除标签等操作。开启保护标签后需设置标签管理员,仅管理员被允许在此标签下创建匹配标签规则的标签。 当设置了 \*-release 为保护分支规则之后,非管理员用户通过 Git 推送标签 xxx-release 的时候有如下提示:

gie push eug drigin XXXX recease
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
remote: [err=32] You have no permission to update protected tag (refs/tags/xxxx-release).
remote: 你没有权限推送到保护标签 (refs/tags/xxxx-release), 查看更多信息: https://coding.net/help/doc/git/git-branch.html#保护分支
remote: <b>error</b> : hook declined to update refs/tags/xxxx-release
To https://e.coding.net/vcs-test/coding-demo/coding-demo.git
! [remote rejected] xxxx-release -> xxxx-release (hook declined)
error: failed to push some refs to 'https://e.coding.net/vcs-test/coding-demo/coding-demo.git'

在 Web 端创建标签或新建版本时也同样会失败。

← android ◄ 浏览	〕 提交	分支 🐔	合并请求	版本	标签	对比	设置
创建标签							
标签版本 *		创建立调 *					
你还似乎		的连木林					
prod-1	0	影 master	•				

#### 使用场景举例

某团队使用标签作为触发 CI 构建的条件,即在生产分支中,通过推送 v1.0-release 字样的标签作为发布命令。设置保护标签后,仅标签管理员能够创建此类



型标签并完成发版动作,保持了代码仓库内各版本序列的整洁与规范。

## 查看版本信息

代码标签可作为某一项代码版本,单击**版本描述**可以查看该版本发布详情;单击**编辑版本描述**即可进行编辑。

<ul> <li>android -</li> </ul>		浏览 提	交 分支	合并请求	版本	标签	对比	设置
v1.3 …								
-o-8b21ae0 🧻	🖺 zip	🖺 tar.gz	🖥 版本描述	① 17 分钟前				
v1.2 …								
- <b>o</b> -8b21ae0 🧻	🖺 zip	🖺 tar.gz	🖹 版本描述	① 22 分钟前				
v1.0 …								
8b21ae0 🏢	🖺 zip	🖺 tar.gz	① 1 个月前					
1-3个, 共3个								

#### 若一个标签并没有对应任何版本发布,可单击**创建版本描述**快速为该标签创建一个发布版本。

← android → 浏览 提交 分支 合并请求 版本 标签 对比 设置	+ 创建标签
<ul> <li>◇ v1.3</li> <li>◇ 8b21ae0 (] E zip</li></ul>	编辑版本描述
<ul> <li>◇ v1.2</li> <li>◇ 8b21ae0 司 臣 zip 臣 tar.gz ≧ 版本描述 () 22 分钟前</li> </ul>	编辑版本描述
✓ v1.0 → 8b21ae0 副 暨 zip 暨 tar.gz ③ 1 个月前 量	创建版本描述
1-3个, 共3个	



## 代码仓库安全 配置权限方案

最近更新时间: 2023-02-07 14:56:11

本文为您详细介绍如何配置仓库权限方案。

## 进入项目

1. 登录 CODING 控制台,单击团队域名进入 CODING 使用页面。

- 2. 单击团队首页左侧的项目,进入项目列表页,选择目标项目。
- 3. 单击左侧菜单栏的代码仓库 > 设置,进入仓库安全页。

#### 配置权限

除了项目权限方案中的全局代码仓库权限,CODING 也支持在单个代码仓库内单独配置权限方案,且不会覆盖已存在的项目权限方案。只要用户的项目权限组或 仓库权限组拥有对应的权限,即可在仓库内指定对应操作。

• 项目权限方案: (配置路径:团队设置中心 > 全局设置 > 组织和成员 > 项目权限方案)



• 仓库内权限方案: (配置路径: 仓库设置 > 权限方案)





## 检查仓库安全风险

最近更新时间: 2023-02-07 14:56:33

本文为您详细介绍如何检查代码仓库中的安全风险。

## 进入项目

- 1. 登录 CODING 控制台,单击团队域名进入 CODING 使用页面。
- 2. 单击团队首页左侧的**项目**,进入项目列表页,选择目标项目。
- 3. 单击左侧菜单栏的**代码仓库 > 设置**,进入仓库安全页。

```
仓库安全
```

项目管理员可以在代码仓库的设置 > 仓库安全页面查看该仓库存在的安全风险。

← hello ◄ 浏览	提交 分支 合并请求 版本 对比 设置					
仓库设置	仓库安全					
基本设置	<ol> <li>为了提升仓库的安全等级,建议您修复以下风险。</li> </ol>					
部署公钥						
文件锁定	★开启 Git 提交的提交者和提交作者检查 →  → → → → → → → → → → → → → → → → → →					
分支设置	确以旋义有构拢义作有的即相定 CODING 验证过,能一足性皮病少态总推达。					
访问设置	未上传 GPG 公钥					
推送设置	L传 GPG 公钥后,可使用私钥对提交签名,以确保提交的可信度。 去上传					
合并请求						
版本发布	保护分支 master 存在以下风险					
代码标签	▲ 未设置分支管理员 去设置					
仓库安全	△ 木廾后代码所有者评単					

目前,系统支持对代码仓库进行以下检查:

- 是否开启了 Git 提交者和提交作者检查。
- 是否上传了 GPG 公钥。
- 是否设置了保护分支;若已设置保护分支,是否设置了分支管理员和开启代码所有者评审。

#### 🕛 说明

为了提高仓库的安全等级,建议您参见提交者及提交注释验证、使用 GPG 签名 commit 记录、保护分支 配置对应功能。



## 调整访问限制

最近更新时间: 2023-02-07 14:56:37

本文为您详细介绍如何调整代码仓库的访问限制。

## 进入项目

- 1. 登录 CODING 控制台,单击团队域名进入 CODING 使用页面。
- 2. 单击团队首页左侧的项目,进入项目列表页,选择目标项目。
- 3. 单击左侧菜单栏的**代码仓库 > 设置** ,进入仓库安全页。

### 访问设置

项目管理员进入代码仓库设置中的访问设置中能够调整以下限制:

- 仅允许 HTTPS 协议推拉代码仓库。
- 仅允许 SSH 协议推拉代码仓库。
- 公开源代码。



出于安全考虑,项目管理员可以限制代码仓库仅使用 SSH 协议或 HTTPS 协议进行代码推拉。相比 HTTPS 协议,SSH 协议会更加安全,详情请参见 配置 SSH 公钥 。

公开源代码需提前进行实名认证。单击右上角齿轮按钮进入团队设置中心,在团队设置中心的实名认证中进行实名。



		搜索	4 🔹 🗸
← unit3 - 浏	」览  提交   分支   合并请求   版本   标签   对比 <b>设置</b>		
仓库设置	访问设置		
基本设置	✓ 允许使用 HTTPS 协议访问代码仓库		
部署公钥	✓ 允许使用 SSH 协议访问代码仓库		
文件锁定	🗹 公开源代码(公开后,任何人都可以访问代码仓库,请慎重考虑! ) 🗊		
分支设置	保存		
访问设置			
推送设置			
合并请求			
版本发布			
代码标签			
仓库安全			
推送权限			

#### 实名后勾选**公开源代码**按钮完成代码仓库开源。

💽 团队设置中心				
Q 搜索设置项	☑ 团队信息 团队基本信息设置	<ul> <li>         ・</li></ul>	∧ ● 服务订购 服务订购与订购信息管理	∧ ♥ 安全性 ∧ 図A 安全相关功能设置
	基本信息	成员管理	概览	访问审计
<b>F</b> 功能设置	实名认证	<b>未认证</b> 用户组	1119	会话管理
🚼 生态能力	注销团队	团队权限方案	资源用量	登录设置
		项目权限方案	账单管理	水印设置 Pro
			优惠券管理	日志
	論第三方应用 第三方服务期定	^		
	第三方应用			



Ð

Ð

Ð

Ð

Ð

## 使用 GPG 签名 commit 记录

最近更新时间: 2023-05-30 17:26:54

本文为您详细介绍如何使用 GPG 签名 commit 记录。

## 进入项目

1. 登录 CODING 控制台,单击团队域名进入 CODING 使用页面。

2. 单击团队首页左侧的**项目**,进入项目列表页,选择目标项目。

CODING 支持使用 GPG 对 Git commit 进行签名验证。对于验证通过的 commit 提交记录,将会打上**已验证**标签,确保代码提交者是可靠来源,增强代码安 全性。

使用 GPG 签名 Git commit 时需要执行以下步骤:

步骤1: 生成 GPG 密钥对

步骤2:添加 GPG 公钥至个人账户设置

步骤3: 与本地 Git 仓库关联

步骤4: 签名 Git commit

步骤5: <u>验证签名</u>

### 操作步骤

#### 步骤1: 生成 GPG 密钥对

1. 下载 并安装 GPG。如果使用 macOS,可直接使用 brew 包管理工具运行以下命令:

#### brew install gpg

2. 运行以下命令生成 GPG 密钥对 ( 公钥或私钥 ):

gpg --full-gen-key

#### 🕛 说明

在某些场景下,例如使用了 Windows Gpg4win 或其他 macOS 版本,使用 gpg --gen-key 命令生成密钥对。

该命令为交互式命令,需要根据提示选择算法类型,指定密钥的有效期,输入您的真实姓名和电子邮件,设置密码等。

- 密钥类型:选择使用的密钥类型,或按 Enter 键选择默认的 RSA 和 RSA。
- 椭圆曲线: 按 Enter 键选择默认的椭圆曲线 Curve 25519 。
- 有效期限:按需指定密钥有效期,或按 Enter 键选择默认的永不过期。
- 电子邮件地址: 需为 CODING 账户内配置的邮箱地址。
- 3. 运行以下命令列出已创建的 GPG 密钥 (命令中的邮箱地址需填写 步骤 3 中指定的邮件地址):

#### gpg --list-secret-keys --keyid-format LONG "your\_email"

4. 复制以 sec 开头的 GPG 密钥 ID。以下示例中,复制 4AEA00A342C24CA3 :

```
sec ed25519/4AEA00A342C24CA3 2021-09-14 [SC]
6DE3507E82DEB6E8828FAAC34AEA00A342C24BD4
uid [绝对]your_name "your_email"
ssb cv25519/812B586FD245B560 2021-09-14 [E]
```

5. 利用复制的 ID 导出该 ID 的公钥(以上述 ID 为例):

gpg --armor --export 4AEA00A342C24CA3

6. 生成公钥之后,可将其 添加至您的 CODING 账户 。


÷

## 步骤2:添加公钥至个人账户设置

- 1. 登录 CODING 之后,单击页面左下角中的**个人账户设置**选项。
- 2. 在左侧导航栏选择GPG 公钥,进入公钥管理页面。
- 3. 单击新增公钥,将导出的 GPG 公钥粘贴至内容框,完成确认。

	GPG 公钥         新增公钥
← 个人账户设置	社旅加 GPG 公钥后, 您可以对提文添加 GPG 並名。黑面直看如阿主成 GPG 盔钥
账户信息	へのためでは、「「」」では、「」」では、「」」では、「」」では、「」」では、「」」では、「」」では、「」」では、「」」では、「」」では、「」」では、「」」では、「」」では、「」」では、「」」では、「」」では、「」」では、「
💄 个人账户	◇·知内交 *
■ 邮箱设置	
个人设置	请粘贴以BEGIN PGP PUBLIC KEY BLOCK 开头的公钥,例如: BEGIN PGP PUBLIC KEY BLOCK
♂ SSH 公钥	UNMN4eL220LyxtK6RKXEMIZ7+CRTA4Gx11efS0o1oL+oGie7oIjYro+UxN/BV9o9hhAvwAeHW
🛃 GPG 公钥	pmC9d7NnjaJ3SbfHtG7uu9j6QkNkw3fWjry0W3M3z6Z6KV8UXs6jg/qC4T8HiGqLsa8
20 访问令牌	=atpj 暫无公钥、您可以END PGP PUBLIC KEY BLOCK
💙 两步验证	
📑 通知设置	确认 取消
<ul><li>∂ 绑定设置</li></ul>	
<b>3</b> 开放生态	



()	<b>说明</b> 若邮箱地址显示未验证状态,	意味着该邮箱没有在 CODING 账户中配置。请在 <b>个人账户设置 &gt; 邮箱设置</b> 中添加该邮箱。	

<b>GPG 公钥</b> 在添加 GPG 公钥	月后,您可以对提交添加 GPG 签名。	新增公钥
R	<ul> <li>邮箱地址</li> <li>密钥 ID</li> <li>子密钥</li> <li>添加于 2021-09-14</li> </ul>	删除

#### 步骤3: 与本地 Git 仓库关联

1. 运行以下命令列出您已创建的 GPG 密钥 ( 命令中的邮箱地址需填写生成密钥时指定的邮件地址 ):

 gpg --list-secret-keys --keyid-format LONG "your\_email"
 □

 2. 复制 sec 开头的 GPG 密钥 ID。以下示例中,复制 4AEA00A342C24CA3 :
 sec ed25519/4AEA00A342C24CA3 2021-09-14 [SC]

 Sec ed25519/4AEA00A342C24CA3 2021-09-14 [SC]
 □

 6DE3507E82DEB6E8828FAAC34AEA00A342C24BD4
 □

uid [绝对 ] your\_name "your\_email"

ssb cv25519/812B586FD245B560 2021-09-14 [E]



ð

3. 在本地 Git 仓库中配置该密钥,对 commit 提交进行签名:

git config --global user.signingkey 4AEA00A342C24CA3

4. 至此,您已经成功将创建的 GPG 密钥与本地 Git 仓库进行关联。在本地修改完代码后书写 Git commit message 时进行签名,以此验明提交者的真实 性。

## 步骤4: 签名 Git commit

运行 Git commit 命令时需要用到 -S 参数。

1. 在本地完成代码编辑需要提交更改时,将 -S 参数添加到 git commit 命令中:

git commit -S -m "your_commit_message"	Ð
① 说明 如果不希望每次都要输入 -S 标志,您可以使用以下命令行设置 Git 自动为 commit 签名:	
git configglobal commit.gpgsign true	٥

2. 如提示输入密码,则提供生成 GPG 密钥时设置的密码。

#### 步骤5:验证签名

将签了名的提交推送至 CODING 代码仓库后,您可以在代码仓库的提交 tab 页查看提交验证是否签名成功。

← Angular-demo → 浏览 提交 分支 合并请求 版本 标签 对比 设置	
\$P master ▼ 提交日期 开始日期 ▼ 至 结束日期 ▼	搜索提交内容 Q <b>列表 ▼ =↓</b>
新的东西 st 提交于 2022-06-10 10:25:44	⑦ 已验证 b574191 ↓
測试测试 ● 31 提交于 2022-06-10 10:15:39	<ul> <li>♥ 当前提交已通过验证,且提交签名密</li> <li>钥的邮箱属于提交者。</li> <li>♥</li> </ul>
Updated build and version.  II 提交于 2020-12-23 00:33:14	鳥 GPG 公朝 拥有者: ◆ CODING 官方 密钥D: AD9( 4
Update README.md I 提交于 2020-12-06 04:57:44	xullyyjac xully
Updated build and version.	2b28034 <b>ູ</b> ີຟ ↔
Updated demo for Angular 11.	7176332 🧃 🚸

提交验证状态的说明如下:

验证状态	说明
已验证	使用 GPG 私钥签名,CODING 账户中有对应公钥,且公钥邮箱已验证
未验证	使用 GPG 私钥签名,但 CODING 账户中无对应公钥或公钥邮箱未验证(若出现未验证的邮箱,请前往 <b>个人账户设置 &gt; 邮箱设</b> 置中添加该邮箱。)
无验证状态标签	没有使用 GPG 私钥签名

# 删除 GPG 公钥



#### 如果您的 GPG 公钥有泄露风险或已不再使用 GPG 签名,可在个人账户设置 > GPG 公钥中删除该公钥。

个人账户设置	<b>GPG 公钥</b> 在添加 GPG 公钥后,您可以对提交添加 GPG 签名。	新增公钥
② 个人账户	邮箱地址 图 图 图 图 图 图 图 图 图 图 图 图 图 图 图 图 图 图 图	
▶ 邮箱设置	密钥 ID 子密钥	删除
个人设置	添加于 2021-09-14	
三 模板设置		
✔ SSH 公钥		
ℜ GPG 公钥 beta		
🗵 访问令牌		
♥ 两步验证		
□ 提醒设置		
∂ 绑定设置		
⑦ 开放生态		

#### 公钥删除之后:

• 已验证的提交变成未验证状态。

• 仍使用 GPG 私钥签名的提交(即使用 git commit -S -m )将变成未验证状态。

• 无签名的提交(即使用 git commit -m )将不被验证,无验证状态标签。

① 项目概览		← node-express-example → 浏览 提交 分支 合并请求 版本 对比 设置	
🗵 项目协同		\$* master ▼ 提交人 全部 ▼ 提交日期 开始日期 ▼ 至 结束日期 ▼	列表 → =↓
<♪ 代码仓库		Corrected mistakes	
⑦ 代码扫描 beta	>	◎ 报交于 3.小时前 ×	2dc0094 🗊 ↔
∞ 持续集成	>	Corrected mistakes	
▲ 持续部署	>	4. 提交于 3 小时前 ×	未验证 953a011 司 ↔
🗔 制品管理	>	Updated contents	
J) 测试管理	>	提交于 3 小时前 ×	未验证 c7bbe4c 🗊 🛷
文档管理	>	Updated contents	
		4 提交于 3 小时前 ×	
		Updated contents	未赊证 ab3812b 回 🚸
			NORTH CONCERNMENT

#### 🕛 说明

若已配置 Git 自动签名,可运行 git config --global commit.gpgsign false 命令取消自动签名。否则 GPG 公钥删除后,推送至远端仓库的提交依 然显示**未验证**状态。

# GPG 签名提交报错处理

如果参考上文完成所有操作之后,在使用 git commit -S -m 签名提交时出现以下报错,可以参见 解决 GPG 签名失败的问题修改相关配置。

error: gpg failed to sign the data fatal: failed to write commit object

ð



# 提交者及提交注释验证

最近更新时间: 2023-02-07 14:56:48

本文为您详细介绍如何在代码仓库中开启提交者及提交注释验证。

# 进入项目

1. 登录 CODING 控制台,单击团队域名进入 CODING 使用页面。

2. 单击团队首页左侧的项目,进入项目列表页,选择目标项目。

### 提交认证

项目管理员可以针对特定代码仓库,在设置 > 推送设置页面开启 Git 提交者(Committer)和 提交作者(Author)的验证。开启后将要求 committer 与 author 为同一邮箱,并且需前往个人账户设置中的邮箱设置进行添加与验证操作。开启此功能后可能会影响到 cherry-pick 功能。

<ul> <li>testing -</li> </ul>	浏览	提交	分支	合并请求	版本	标签	对比	设置
仓库设置	推	送设置						
基本设置		检查 Gi	t 提交的打	是交者 (Comm	nitter) 和排	是交作者	(Author)	必须是已验证的邮箱。
部署公钥		禁止强制	別推送 (Fo 欠提交的)	orce Push) 🕜 文件总大小限制	], Git LF	-S 文件除	外	
文件锁定		开启 Gi	t 提交信』	息的格式校验。				
分支设置								
仓库规范								
访问设置								
推送设置								
合并请求								
版本发布								
代码标签								
仓库安全								
推送权限								

团队负责人或管理员还可以强制团队中的所有仓库开启该验证。单击左下角的**团队设置中心**,在**仓库设置**中进行开启。

💽 团队设置中心					
Q 搜索设置项	☑ 项目协同 团从级项目协同配置	^	⑦ 代码扫描 ^ 図从代码扫描工具管理	∞ 持续集成 ^ 团从级持续构建配置	
✿ 全局设置	配置方案		工具规则	构建节点池	云账号
<b>示</b> 功能设置	事项类型		方案模版	构建计划模版	堡垒机
📲 生态能力	事项属性			构建插件	主机组
	事项状态				
	公开资源 管理公开代码仓库	^	√→ 代码仓库 ^ 図从級代码仓库配置	□ 制品管理 ^ 図从感制品管理功能	☆菜単管理 ^ 団队级菜半配置
	公开资源		仓库设置	制品晋级	菜单管理
			团队部署公钥		
			仓库规范		

开启后,若提交者与提交作者邮箱未经认证,所推送的代码将会被拦截。



<ul> <li>         参山科技         Q 搜索设置项     </li> </ul>	代码仓库 / 仓库设置 仓库设置 统一设置您的团队中所有仓库的部分设置
< 代码仓库	推送设置 开启后将会强制覆盖团队中所有仓库的相同设置点 ✓ 检查 Git 提交的提交者 (Committer) 和提交作者 (Author) 必须是已验证的邮箱。
仓库设置	● 单次提交的文件总大小不能超过 0 MB, Git LFS 文件除外。
团队部署公钥 仓库规范	保存 版本创建页摘要 自定义版本创建页面右侧摘要内容 版本格式: 请遵循相应的版本规范。

#### 此处的已验证邮箱指的是已在**个人账户设置 > 邮箱设置**中新增的邮箱。

💽 叁山科技 🔹	个人邮箱		
← 个人账户设置	主邮箱用于萤录 CODING,及代码托管 HTTPS 克隆或制品库的萤录	3用尸名。以下邮箱均可用于代码托管提交者身份识别。	×
<ul> <li>账户信息</li> <li>个人路户</li> </ul>	gala****@gmail.com - 主部商	新增邮箱	
● 邮箱设置		由印刷	
个人设置		BA-172.222	11 Rold ALCER
		944 HE (PD)	3大山(北)近山正位9
♂ SSH 公钥		确认	
of GPG 公钥			
🧝 访问令牌		取消	
两步验证			
通知设置			
<b>詳語</b> 开放生态			

#### 提交注释规范

项目管理员还可以设置 Git 提交注释(commit-message)的规则,不符合该规则的提交将被拦截。CODING 代码仓库具备 自动关联事项 能力,在 commit message 中附上事项号便能够与事项相关联,对齐本次开发工作的背景。

此时可以将提交注释规则设置为 ^fix #[0-9]+ ,规定每次提交时必须要在 commit message 中附上关联项目事项号。系统中还内置了优质开源社区的



#### commit message 参考,让每次代码提交记录更加清晰。

← testing -	浏览 提交 分支 合并请求 版本 标签 对比 设置	
仓库设置	推送设置	
基本设置	检查 Git 提交的提交者 (Committer) 和提交作者 (Author) 必须是已验证的邮箱。	
部署公钥	票正强制推迭(Force Push) ♥ 开启单次提交的文件总大小限制, Git LFS 文件除外	
文件锁定	✓ 开启 Git 提交信息的格式校验。	
分支设置	请输入有效的正则表达式,如: ^fix #[A–Z]+–[0–9]+	
仓库规范	0 / 2	256
访问设置	常用提交信息格式推荐 >	
推送设置	Angular 社区提交规范 ^(feat fix docs style refactor test chore ci)((.+))?: .{1,100} 使	用
合并请求	jQuery 社区提交规范 ^(Fixes Closes Ref)? .{1,100} 使	用
版本发布	Atom 社区提交规范 ^(:.*:)? .{1,72} 使	用
代码标签	JSHint 社区提交规范	用
仓库安全	ESLint 社区提交规范 ^(Fix Update New Breaking Docs Build Upgrade Chor)?: .{1,72} 使	用
推送权限	保存	



# 锁定文件

最近更新时间: 2023-02-07 14:56:53

本文为您详细介绍如何锁定代码仓库中的文件。

# 进入项目

1. 登录 CODING 控制台,单击团队域名进入 CODING 使用页面。

2. 单击团队首页左侧的**项目**,进入项目列表页,选择目标项目。

针对代码仓库中的**默认分支(通常为 master 分支)**,CODING 支持对该分支下的文件或路径进行锁定。锁定后的文件仅限锁定者修改(编辑、删除),若选 择锁定路径,则该路径下的所有文件同样会被锁定并仅限锁定者修改。

#### 权限说明

项目管理员可进入项目,单击左下角的**项目设置**,前往**项目与成员 > 用户组**设置代码仓库权限。设置完成后将团队成员添加至相关用户组中即可完成项目内的代码 仓库权限控制。

O	😧 蓋的測试 🚽	项目与成员	測试管理	😒 编辑用例	🕑 删除用例	☑ 编辑报告	副餘报告	
4	项目设置	項目		💟 执行测试	✓ 编辑自动化用例	✓ 删除自动化用例		
		基本设置	代码扫描	☑ 访问代码扫描	✓ 扫描任务管理	☑ 扫描问题管理	✓ 扫描方案管理	
61	项目与成员	菜单管理		🔽 访问代码	✓ 创建仓库	✓ 仓库设置	✓ 下载代码	
20	项目协同	通知设置		🔽 拉取代码	☑ 本地推送代码	☑ 在线编辑仓库文件	☑ 创建分支和编辑备注	
- ette	项目公告		代码仓库	🔽 删除普通分支	☑ 删除保护分支	☑ 保护分支规则	▽ 创建标签	
	开发者选项	成员配置		☑ 删除普通标签	✓ 删除保护标签	✓ 创建版本	✓ 解锁锁定文件	
		成员		☑ 删除版本	✓ 创建合并请求	✓ 部署公钥	☑ 解锁锁定文件	
		权限组		☑ 访问持续集成	✓ 持续集成管理	✓ 手动触发/停止构建	☑ 重置缓存	
		个人偏好	持续集成	☑ 删除构建记录	✓ 创建构建计划	✓ 修改构建计划	✓ 复制构建计划	
		每日工作邮件提醒		☑ 删除构建计划	✓ 人工确认			
			持续部署	☑ 访问持续部署	✓ 持续部署管理	✓ 删除部署记录		
		其他	应用管理	😒 访问应用管理	💟 应用发布	💟 应用编辑	💟 数据库变更管理	
		万央标並	And D rate	💟 访问制品库	🔽 制品库设置	✓ 删除制品仓库	♥ 制品扫描	
			利血件	✓ 拉取制品	✓ 删除制品	✓ 推送制品		
			Wiki	☑ 访问 Wiki	✓ 编辑 Wiki	☑ 删除 Wiki	✓ 分享 Wiki	
			文件	☑ 访问文件	☑ 编辑文件	☑ 删除文件	☑ 分享文件	
			API 文档	访问 API 文档				

● 具备访问代码仓库权限的项目成员可以在代码的浏览页面锁定相关文件或路径。锁定后,如果希望取消锁定,可以在代码浏览页面进行解锁。

• 普通用户只能解锁自己锁定的文件或路径;拥有**解锁锁定文件**权限的成员可以解锁他人锁定的文件或路径。

# 锁定文件

进入任一项目的代码仓库的浏览页,选择拟锁定文件,单击右上角更多操作按钮,选择锁定操作。

� coding-help-generator ◄ 浏览 提交	分支 合并请求 版本 对比 设置	+ 新建代码仓库
✿ coding-help-generator	第 master ▼ 合 当找文件 ~ README.md	瀏克隆
> 📄 scaffolds	<b>文件</b> 预览 历史 代码对比 按行查看	⊘ 编辑
> scripts		
> isource	docs: readme 文件更新 最后提交 5cd1e16	🖸 重命名
> intermes/coding-help	MIREADME.md 3.62 KB	☆ 锁定
5 .fnlint.json	1 # CODING 帮助中心 2 2 / / 世刊和社社 / / / / / / / / / / / / / / / / / /	
🕒 .git-pre-commit	3 [![特u述(な)]( <u>nttps://codingcorp.coding.net/badg</u> 4 5 [![Cammitizen_friend]u]/http://ima.chiolds.in	回 删除
🗋 .gitattributes	6 7 参与开放 APT 文档协作请直接阅读[《开放 APT 更新控记》](https://codingco	ator/d/coding_hel
🗋 .gitignore	8 9 本合库是 CODING 帮助中心的文档,采用 Hexo 框架, Markdown 格式, 通过持续集成强制检查下列规范:	
🗋 .lintmdrc	10 11 - Git commit: [commitlint](https://www.inter.in	
🗋 Jenkinsfile	12 - Markdown 英文: [remark]( <u>https://remark-cli</u> )       13 - Markdown 中文: [lint=md](https://int=md)	
MI OPENAPI-README.md	14 15 参与本项目之前 建议先学习如緬運程: [DevOos 代码质量定税: 代码物节与 Git Flow](http	learning/live-2
MI README.md	15 ダライヤベロとNY, たたメルテラススがKWATE: (DCCOPS 100000年天間・10000次20-3 011-1000)( <u>ITTP</u> 16 17 ## 目录结构	tearning/ tive-2



锁定后文件会出现小锁标志,除了锁定者以外的成员将无法编辑或删除此文件。

� coding-help-generator   浏览 提交	分支 合并请求 版本 对比 设置	+ 新建代码仓库
✿ coding-help-generator	💡 master 👻 🔒 查找文件 → README.md	I □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □
> 🧧 scaffolds	<b>文件</b> 预览 历史 代码对比 按行查看	⊿ 编辑 …
> 📄 scripts	_	
> isource	docs: readme 文件更新 最后提交 5cd1e18	☑ 重命名
> intermes/coding-help	MI README.md   3.62 KB   🔂	合 解锁
5 .fnlint.json	1 # CODING 帮助中心 2 2 // 世世神法之/ http://	□ 下载
🕒 .git-pre-commit	3 [![例建获念]( <u>https://coaing.orp.coaing.net</u> 4 5 [![Commitizen_friend]v](https://img.shie <sup>2</sup> 3 ]](http://img.shie <sup>2</sup> 5 ]][Commitizen_friend]v](https://img.shie <sup>2</sup>	回 删除
🗋 .gitattributes	6 7 参与开放 API 文档协作请直接阅读[《开放 API 更新指引》](https://coding	ator/d/coding-hel
🗋 .gitignore	8 9 本仓库是 CODING 帮助中心的文档、采用 Hexo 框架、Markdown 格式,通过持续集成强制检查下列规范:	
lintmdrc	10 11 - Git commit: [commit][ht 103710112350221]	
🗋 Jenkinsfile	12 – Markdown 英文: [remark]( <u>http://remark-cli</u> ) 13 – Markdown mx: [litti //remark_di	
MI OPENAPI-README.md		anning (live 2
MI README.md	13 参司卒4时之前,建以元学习代列味程: [Devups 17.始成重头战: 17.的成犯与 Git Flow]( <u>nttps:</u> 16 17 ## 目录结构	arning/live-2

# 锁定文件夹

选择仓库的任一文件夹,单击右上角更多操作按钮,选择锁定操作,即可锁定该路径下所有文件。

� coding-help-generator ▼ <b>浏览</b> 提交	分支 合并请求 版本	对比 设置			+ 新建代码仓库
↑ coding-help-generator	😵 master 👻 🏠	查找文件 ~ scaffolds			ℤ 克隆
✓ ■ scaffolds	<b>文件</b> 历史				+ 新建
M↓ draft.md					
M↓ page.md	init			最后提交 c5e1	🗹 重命名
M↓ post md					合 锁定
	<b>\$</b>				世 上传
> scripts	M↓ draft.md	yuanzhe	init		⊥ 下载
> isource	ML page md	vuenzho	init		❺ 删除
> 📒 themes/coding-help	™≢ page.md	yuanzne	IIIIt		
🔤 .fnlint.json	M↓ post.md	yuanzhe	init		1 年前
🕒 .git-pre-commit					
🗋 .gitattributes					
🗋 .gitignore					

# 锁定效果

锁定后的文件只有锁定者才能进行编辑或删除操作。锁定后的路径只有锁定者才能在路径下新建、编辑或删除文件。若其他用户推送的代码中包含对锁定文件的操 作记录,则会提示推送失败。

Counting objects: 3, done.	
Compressing objects: 100% (3/3), done.	
Writing objects: 100% (3/3), 290 bytes   290.00 KiB/s, done.	
Total 3 (delta 2), reused 0 (delta 0)	
remote: Permission denied. Your edited files are locked.	
remote: 没 有 权 限 操 作 。 您 修 改 的 文 件 已 被 锁 定 。	
<b>remote</b> : 被锁定的文件:	
remote: pom.xml	
remote: error: hook declined to update refs/heads/master	
To https://e.coding.n ng-demo.git	
! [remote rejected] master -> master (hook declined)	
error: failed to push some refs to 'https://e.co g-demo.gi	t'
ubuntuaVM-0-14-ubuntu:~/temp/sprina-demo\$	

在合并请求中,若目标分支为默认分支且待写入的目录中包含被锁定的文件或路径,那么该合并请求只能由被锁定的文件或路径的锁定者进行合并。其他成员单击



#### 合并会提示失败。

	⊗ 有文件被锁定	普·有奖体验。 搜索		
♦ spring-demo ▼ 浏览 提交 分支 合并请求 版本	对比 设置			
調试合并请求         可合并       将分支         dev       司         合并到分支       master				
概览 1 提交记录 1 文件改动 1				
<ul> <li>✓ 描述 // 参 编辑</li> <li>没有描述</li> </ul>				
✓ 关联资源 0 + 添加资源 点击右上角"+"关联项目资源(迭代、任务、合并请求等)及添加外	耶链接			
✔ 合并请求分支与目标分支无冲突 自动合并可以工作				
89 非保护分支 日运公支为非促护公支 可由发起来自行会并 MD				

# 解锁被锁定文件/文件夹

项目普通成员可直接在代码浏览页面解锁自己锁定的文件或文件夹;拥有**解锁锁定文件**权限的成员还可以解锁他人锁定的文件或文件夹。

← test - 浏览	提交 分支 合并请求 版本 对比 设置 + 1	刘建代码仓库
♠ test	V master ▼ 合 道找文件 ✓ README.md	<b>※克隆</b>
MI README.md	<b>文件</b> 预宽 历史 代码对比 按行查看	∠ 编辑 …
		🕑 重命名
	MI README.md   6 Bytes   🛱	合 解锁
	1 #test	🗋 Raw
	2	土 下载
		<b>茴</b> 删除

项目管理员可以在仓库的设置 > 文件锁定页面中查看被锁定的文件或文件夹。单击删除即可解锁文件或文件夹。

🚸 spring-demo 👻	浏览 提交 分支 合并请求 版本 对比 设置			
仓库设置	<b>已锁定文件</b> 目前锁定文件只适用于默认分支。			
基本设置		No. 1 a de		10.4
部署公钥	文件或文件夹	添加者	添加时间	操作
文件锁定	pom.xml		2020-12-26	删除
分支设置	.mvn/wrapper/	-	2020–12–26	删除
访问设置				
合并请求				
版本发布				
代码标签				



# 限制推送文件大小与仓库容量

最近更新时间: 2023-02-07 14:56:57

本文为您详细介绍如何创建仓库规范。

# 进入项目

1. 登录 CODING 控制台,单击团队域名进入 CODING 使用页面。

2. 单击团队首页左侧的**项目**,进入项目列表页,选择目标项目。

# 限制单次推送时的总文件大小

说明
 此功能适用于高级版团队。

如果一个代码仓库中包含大体积文件会显著降低拉取速度,大体积文件的正确处理方式应该是使用 Git LFS 方式进行管理。为提升每次代码的拉取速度,增强开 发效率,团队负责人或管理员可以在团队设置中心中开启所有仓库的文件推送大小限制。当单次推送时的总文件体积超出限制后将会被拦截,并且在团队内要求大 文件以 LFS 方式进行管理,规范团队内开发习惯。

单击左下角的**团队设置中心**,单击**功能设置**中的仓库设置,填写提交文件总大小限制。

<ul><li>● <!--</th--><th>代码仓库 / 仓库设置 仓库设置 统一设置您的团队中所有仓库的部分设置</th><th></th></li></ul>	代码仓库 / 仓库设置 仓库设置 统一设置您的团队中所有仓库的部分设置	
← 代码仓库	推送设置 开启后将会强制覆盖团队中所有仓库的相同设置点 □ 检查 Git 提交的提交者 (Committer) 和提交作者 (Author) 必须是已验证的邮箱	
仓库设置	✓ 单次提交的文件总大小不能超过 512 MB. Git LFS 文件除外。	
团队部署公钥 仓库规范	保存       版本创建页摘要       自定义版本创建页面右侧摘要内容       版本格式:       请遵循相应的版本规范。	
	77 / 保存	400

# 限制单仓库内提交文件大小

项目管理员进入目标仓库,单击仓库设置中的**推送设置**,勾选**开启单次提交文件总大小限制**并输入限制容量。例如此处限制了 512 MB 的体积限制,那么如果单 次推送过程中总文件体积超过了 512 MB,此次提交将会被代码仓库拒绝接收。



🔶 Angular-demo 🗸	浏览 提交 分支 合并请求 版本 标签 对比 设置
仓库设置	推送设置
基本设置	检查 Git 提交的提交者 (Committer) 和提交作者 (Author) 必须是已验证的邮箱
部署公钥	禁止强制推送 (Force Push) ② / / / / / / / / / / / / / / / / / /
文件锁定	512 MB 保存
分支设置	开启 Git 提交信息的格式校验。
访问设置	
推送设置	
合并请求	
版本发布	
代码标签	
仓库安全	
推送权限	

## 限制仓库容量

因 不同的团队版本 有着不同的总仓库容量限制,若不希望某个仓库占用过多体积而影响到其他代码仓库,项目管理员可以在代码仓库设置中将单个仓库总容量上 限修改为 1.00 GB 以下的任何数值,这样可以确保某个仓库不会占用过多的体积而影响到其他仓库。

仓库设置	基本设置	修改仓库容量上限	×
基本设置	仓库名称	您可以将仓库总容量上限修改为 1.00 GB 以下的任何数 值,若想高于 1.00 GB, <mark>请升级团队</mark>	仓库图标
部署公钥	HTTP + http://git.codingtest.coding.io/c	* 仓库容量	
文件锁定	<ul> <li>修改仓库名称将导致仓库的访问 URL 改变, git remote set-url origin http://git.codingt</li> </ul>	1.00 GB	U
分支设置			
访问设置	仓库描述	備定取消	更改图标
推送设置	请输入仓库描述		
合并请求			
版本发布	□ 设为自定义模板仓库		
代码标签	日定义候做它库可证用厂创建它库时以相同的目家结构和义	计生成新的它库,模倣它库当肌团队引用。	
仓库安全	<b>沐行</b>		
推送权限	仓库容量管理		
	已使用 16.44 MB / 总容量 1.00 GB 修改仓库容量上限		
	仓库清理		
	仓库清理使用 git gc 优化代码仓库存储空间, 压缩存储库对象,	减少存储库磁盘占用。	



# 文件推送限制

最近更新时间: 2023-02-07 14:57:02

本文为您详细介绍如何使用文件推送限制功能。

# 进入项目

1. 登录 CODING 控制台,单击团队域名进入 CODING 使用页面。

2. 单击团队首页左侧的项目,进入项目列表页,选择目标项目。

# 文件推送权限

文件推送控制功能主要用于限制团队成员对**特定文件路径**的推送行为。例如说当前代码仓库下的 /src 路径下存储着核心服务与依赖,当成员拉取至本地后有可能 会无意间修改其中的依赖文件并重新推送至远端仓库中。那么此时通过文件推送控制功能便可以杜绝这种情况的发生。 若要开启此功能,由项目管理员前往代码仓库的设置,在**推送权限**中添加文件路径。

<ul> <li>daily-sentence -</li> </ul>	浏览 提交 分支 合并请求 版本 标签 对比 设置		
仓库设置	<b>文件推送权限</b> 添加文件路径后,可以设置成员或者用户组是否具有推送权限。	+ 添	加文件路径
基本设置 部署公钥	/src 保存	+ 新增用户组/成员	<b>會</b> 删除
文件锁定	62) 所有人 (2) ① 九许推送		
分支设置			
访问设置			
推送设置			
台井请求			
代码标签			
仓库安全			
推送权限			
同步信息			

#### 放行特定的用户组或用户,允许他们修改此路径下的文件。

← daily-sentence -	浏览 提交 分支 合并请求 版本 标签 对比 设置			
仓库设置	<b>文件推送权限</b> 添加文件路径后,可以设置成员或者用户组是否具有推送权限。		+ >	忝加文件路径
基本设置	/src		+ 新增用户组/成员	● 删除
<b>立</b> 件 鄉 定	🙆 所有人 (2)	① 允许推送	+ 新增用户组	>
分支设置	22 开发 (1)	() 允许推送	+ 新増成员	>
访问设置		1 允许推送		
推送设置				
合并请求				
版本发布				
代码标签				
仓库安全				
推送权限				
同步信息				



# 仓库规范

最近更新时间: 2023-02-07 14:57:07

本文为您详细介绍如何创建仓库规范。

# 进入项目

1. 登录 CODING 控制台,单击团队域名进入 CODING 使用页面。

2. 单击团队首页左侧的项目,进入项目列表页,选择目标项目。

在仓库规范中可以设置分支命名规范与合并请求放行限制,起到规范开发行为与保持 Git 分支结构清晰的作用。项目管理员无需在团队内反复强调开发规范,分支 限制能够将不合格的分支提交行为直接杜绝在团队代码仓库之外,方便后续的代码维护工作。

项目管理员进入任一代码仓库的设置中,单击**仓库规范**进行设置。

□ 项目概览		<ul> <li>template-1 -</li> </ul>	浏览 提交 分支 合并	件请求 版本 标签 对比	北 设置	
☑ 项目协同		合库设置	仓库规范			
<小 代码仓库		SHKE	对代码仓库的分支名和分支间的	1合并方向进行规范,从而达到更	高效的开发工作流。	
③ 代码扫描 beta	>	基本设置	使用已有的仓库规范方案			
∞ 持续集成	>	部署公钥				
♣ 持续部署	>	文件锁定	分支类型			+ 添加分支类型
◎ 应用管理		分支设置	分支类型	分支名匹配规则	描述	操作
同 制品管理	>	仓库规范	主干分支	master	与仓库设置 > 分支设置中的默认分支保持一致。	
之档管理	>	访问设置	允许创建规定分支类型以外	的分支		
		推送设置	_			
		合并请求	<b>合并方向</b> 规范仓库分支间的合并方向,只允许	创建列表中规定方向的合并请求,列表	1为空则不会对仓库中的会并请求方向做限制。	+ 添加合并方向
		版本发布	源分支	目标分支	因示	操作
		代码标签			暂无添加任何合并方向限制,允许创建任意分支间的合并请求	
		仓库安全				
		推送权限				

# 分支规范

分支规范主要通过正则表达式限制分支命名行为,明晰各项分支的用途与场景。不符合命名格式的分支将无法上传至代码仓库中。 单击**添加分支类型**创建适合团队内自身需求的分支命名规范。

仓库设置	仓库规范 对代码仓库的分支名和分支	间的合并方向进行规范,	添加分支类型	×	
基本设置	✓ 使用已有的仓库规范方器	K	* 分支类型名称		
部署公钥			生产分支		
文件锁定	- 35 + 8		* 分支名匹配规则		
分支设置	t		dev/*		
仓库规范	分支类型		描述		
访问设置	分支类型	分支名匹配规则	请简要描述该分支类型的作用及意图等		
推送设置	主干分支	master			
合并请求	允许创建规定分支类型L	以外的分支			
版本发布					
代码标签	<b>合并方向</b> 规范仓库分支间的合并方向,只:	允许创建列表中规定方向的合			
仓库安全	源分支	目标分支	确 定 取 消		
推送权限				王意分支间的合并请求	

创建分支类型后,在分支页可以看到所有分支的分类。不符合命名规范的分支将会被标注。

← template-1 → 浏览 提交 分支 合并请求	求 版本 标签 对比 ì	受置	+ 创建代码分支
<b>活跃分支</b> 保护分支 只读分支 所有分支 过时分支	分支类型 ▼ 搜索分支	٩	批量删除 分支设置
ジ master-patch-1 不規范 更新子 10 分钟前	清餘已选 主干分支 master 开发分支 develop/*	1 0	0594db4 (引) 创建合并请求 设为只读 …
ジ master 試认分支 主干分支 更新于 3 个月前	发布分支 release/* 不规范的分支	٥ ٥	<b>4316a13 </b>
1-2个, 共2个			

# 合并方向





此功能用于限制代码分支间的合并方向,仅允许创建固定合并方向的合并请求。列表为空则表示不会对分支的合并方向做出限制。

仓库设置	<b>仓库规范</b> 对代码仓库的分支名和分支间	间的合并方向进行规范,.	添加合并方向		×		
基本设置	● 使用已有的仓库规范方案	1	* 源分支				
部署公钥			发布分支	*			
文件锁定	分支类型		* 目标分支				+ 添加分支类型
分支设置	分支类型	分支名匹配规则	主干分支	~			操作
仓库规范	主干分支	master	图示				
访问设置	开发分支	develop/*	release/*				编辑 删除
推送设置	发布分支	release/*					编辑 删除
合并请求		4.6575=	确 定 取 消				
版本发布	☑ 元件刨建规定方文失空以	TRIAL					
代码标签	<b>合并方向</b> 规范仓库分支间的合并方向,只允	计创建列表中规定方向的合并	请求,列表为空则不会对仓库中的合并请求方向做	限制。			+ 添加合并方向
仓库安全	源分支	目标分支	图示				操作
推送权限							

# 系统规范方案

系统内置了单分支规范、功能分支规范、Gitflow 分支规范与 GitLab 分支规范四种方案,不同的研发团队可以根据实际需求进行选择。

说明
 选择系统方案后无法手动创建新的分支。

#### 单分支规范



#### 仅提供一项主干分支,不对合并方向做出限制,适用于个人开发项目。

全部仓库规范方案 4 切换后,新方案将会覆盖原方案的仓库 规范设置,请谨慎操作。	<b>单分支规范</b> 支持 master 分支							
	分支类型							
<b>单分支规范</b> 系统 支持 master 分支	分支类型	分支名匹配规则	描述					
	主干分支	[主干分支]	与仓库设置 > 分支设置中的默认分支保持一致。					
<b>功能分支规范 系统</b> 支持 master、feature/* 分支	○ 允许创建规定分支类型以外	卜的分支						
	<b>合并方向</b> 规范仓库分支间的合并方向,只允许	F创建列表中规定方向的合并请求,列表	为空则不会对仓库中的合并请求方向做限制。					
Gitflow 分支规范 系统 支持 master、develop、feature/*、 release/*、hotfix/* 分支	源分支	目标分支	图示					
		暂无添加任何合并方向限制,允许	午创建任意分支间的合并请求					
支持 master、develop、feature/*、 release/*、hotfix/* 分支     暂无添加任何合并方向限制,允许创建任意分支间的合并请求       GitLab 分支规范 系统 支持 master、feature/*、*-stabl e、environment/* 分支     新加工								
	确认 取消							

功能分支规范



#### 提供主干分支与功能分支两种分支类型,同时限制仅允许功能分支单向合并至主干分支。 × 全部仓库规范方案 4 功能分支规范 支持 master、feature/\* 分支 切换后, 新方案将会覆盖原方案的仓库 规范设置,请谨慎操作。 分支类型 单分支规范 系统 分支类型 分支名匹配规则 描述 支持 master 分支 主干分支 [主干分支] 与仓库设置 > 分支设置中的默认分支保持一致。 功能分支规范 系统 功能分支 feature/\* 一般一个事项卡对应一个功能分支 支持 master、feature/\* 分支 ◎ 允许创建规定分支类型以外的分支 Gitflow 分支规范 系统 合并方向 规范仓库分支间的合并方向,只允许创建列表中规定方向的合并请求,列表为空则不会对仓库中的合并请求方向做限制。 支持 master、develop、feature/\*、 release/\*、hotfix/\* 分支 目标分支 图示 源分支 功能分支 主干分支 feature/\* → [主干分支] GitLab 分支规范 系统 支持 master、feature/\*、 \*-stabl e、environment/\* 分支 确认 取 消

#### Gitflow 分支规范

Gitflow 是业内常见的分支规范,将开发工作的重点划分为主干与开发两条线路。主干线路下划分为 release 与 hotfixes 两种场景,分别满足发版与热修复的 需要。而功能开发工作则集中在 feature 分支上,并且仅允许收归至 develop 分支上。经测试功能无误后再将 develop 分支合并至 release 分支中,等待最







#### 单击 Gitflow 分支规范后选择本方案并应用。

全部仓库规范方案 4 切换后,新方案将会覆盖原方案的仓库	<b>Gitflow 分支规范</b> <sub>支持 master、develop、</sub>	feature/*、 release/*、 hotfix	:/* 分支				
规范设置,请谨慎操作。	分支类型	分支类型					
<b>单分支规范 系统</b> 支持 master 分支	分支类型	分支名匹配规则	描述				
	主干分支	[主干分支]	与仓库设置 > 分支设置中的默认分支保持一致。				
功能分支规范 系统	开发分支	develop	平时开发用的主分支,永远是功能最全最新				
	功能分支	feature/*	一般一个事项卡对应一个功能分支				
Gitflow 分支规范系统	发布分支	release/*	一般一次新版本的发布对应一个发布分支				
支持 master、develop、feature/ *、 release/*、hotfix/* 分支	热修复分支	hotfix/*	从主干分支拉出,用于线上版本的 Bug 修复				
GitLab 分支规范 系統	○ 允许创建规定分支类	型以外的分支					
支持 master、feature/*、 *-stabl e、environment/* 分支	<b>合并方向</b> 规范仓库分支间的合并方向,	只允许创建列表中规定方向的合并请	f求,列表为空则不会对仓库中的合并请求方向做限制。				
	源分支	目标分支	图示				
	发布分支	主干分支	release/* → [主干分支]				
	热修复分支	主干分支	hotfix/* → [主干分支]				
	功能分支	开发分支	feature/* $\rightarrow$ develop				
	发布分支	开发分支	release/* $\rightarrow$ develop				
	确认 取消						

GitLab 分支规范



### 提供主干分支、功能分支、发布分支与环境分支四种类型,仅允许功能分支合并至主干分支。

全部仓库规范方案 4	GitLab 分支规范	I		×
切换后,新方案将会覆盖原方案的仓库 规范设置,请谨慎操作。	支持 master、feature	e/*、 *-stable、environment/* 分	支	
	分支类型			
<b>単分文规氾 <sup>永筑</sup></b> 支持 master 分支	分支类型	分支名匹配规则	描述	
	主干分支	[主干分支]	与仓库设置 > 分支设置中的默认分支保持一致。	
<b>功能分支规范 系统</b> 支持 master_feature/* 分支	功能分支	feature/*	一般一个事项卡对应一个功能分支	
	发布分支	*-stable	一般一次新版本的发布对应一个发布分支	
Gitflow 分支规范 系统	环境分支     env/*     每个环境对应一个分支,比如 env/uat、 env/production 等等			
release/*、hotfix/* 分支	○ 允许创建规定分支	支类型以外的分支		
GitLab 分支规范 系统 支持 master、feature/*、*-stabl	<b>合并方向</b> 规范仓库分支间的合并方	向,只允许创建列表中规定方向的合并请家	求,列表为空则不会对仓库中的合并请求方向做限制。	
e、environment/* 分支	源分支	目标分支	env/*     每个环境对应一个分支,比如 env/uat、 env/production 等等       分支        I建列表中规定方向的合并请求,列表为空则不会对仓库中的合并请求方向做限制。       目标分支     图示       主干分支     feature/* → [主干分支]	
	功能分支	主干分支	feature/* → [主干分支]	
	确认 取消			

# 团队级规范设置

团队负责人或管理员可以前往**团队设置中心 > 功能设置 > 仓库规范**中创建新的仓库规范方案,生效后团队内的所有仓库都能选用此仓库规范方案。

💦 团队设置中心 全局	设置 功能设置 Q 搜索	設置项			$\checkmark$
← 代码仓库	<sup>功能设置 / 代码仓库</sup> 创建仓库规范方案				
仓库设置	* 分支类型名称				
团队部署公钥	请输入分支类型名称,如生产	*分支、热修复分支等			
仓库规范	描述				
	请简要描述该仓库规范				
	分支类型				+ 添加分支类型
	分支类型	分支名匹配规则	描述		操作
	主干分支	[主干分支]	与仓库设置 > 分支设置中的默认分支保持一致		
	允许创建规定分支类型以外	卜的分支			
	<b>合并方向</b> 我范仓库分支间的合并方向,只允祥	〒创建列表中规定方向的合并请求,列8	为空则不会对仓库中的合并请求方向做限制。		+ 添加合并方向
	源分支	目标分支	图示		操作
			1	最无源加任何合并方向限制,允许创建任意分支间的合并请求	
	完成创建 取 消				

在基本设置中提供以下两个自动选项:

• 创建仓库时默认启用选定的仓库规范。



<ul> <li>代码仓库</li> <li>仓库设置</li> <li>团队邮署公钥</li> </ul>	功能设置/代码仓库 仓库规范 全部方案 基本设置 到建仓库时数认勾选仓库规范方	全部仓库规范方案 4 单分支规范 <u>系统</u> 支持 master 分支	<b>单分支规范</b> 支持 master 分支 <b>分支</b> 类型		
仓库规范	仓库创建者可以自行取消勾选。 单分支规范 系统 支持 master 分支	<b>功能分支规范 系统</b> 支持 master、feature/* 分支	<ul> <li>分支类型</li> <li>主干分支</li> <li>○ 允许创建规定分支类型</li> </ul>	分支名匹配规则 [主干分支] 以外的分支	<sup>描述</sup> 与仓库设置 > 分支设置中的默认分支保持一致。
	创建仓库时自动创建精确匹配分: 如创建仓库选择了 Gitflow 仓库规范时	Gitflow 分支规范 系统 支持 master、develop、feature/*、 release/*、hotftx/* 分支	<b>合并方向</b> 规范仓库分支间的合并方向,只 源分支	允许创建列表中规定方向的合并证 目标分支	真求,列表为空则不会对仓库中的合并请求方向做限制。 图示
		GitLab 分支规范 系统 支持 master、feature/*、*-stabl e、environment/* 分支		暂无添加任何合并方向限	限制,允许创建任意分支间的合并请求
			· 追 认 取 消		

• 创建仓库时自动创建规范中预设的分支。



# **Git** 通识 常用命令

最近更新时间: 2021-11-24 10:16:12

#### 本文为您详细介绍 Git 中的常用命令。

#### 创建版本库

### 修改和提交

\$ git status	#查看状态	٥	
\$ git diff	#查看变更内容		
\$ git add .	#跟踪所有改动过的文件		
\$ git add <file></file>	#跟踪指定的文件		
\$ git mv <old><new></new></old>	#文件改名		
\$ git rm <file></file>	#删除文件		
\$ git rmcached <file></file>	· #停止跟踪文件但不删除		
\$ git commit -m "comm	nit messages" #提交所有更新过的文件		
\$ git commitamend	#修改最后一次改动		

### 查看提交历史

\$ git log \$ git log -p <file> \$ git blame <file></file></file>	#查看提交历史 #查看指定文件的提交历史 #以列表方式查看指定文件的提交历史	0
¢ gie blaine silier		

#### 撤销

- \$ git reset --hard HEAD #撤销工作目录中所有未提交文件的修改内容 \$ git checkout HEAD <file> #撤销指定的未提交文件的修改内容
- \$ git revert <commit> #撤销指定的提交
- \$ git log --before="1 days" #退回到之前1天的版本

# 分支与标签

- \$ git branch #显示所有本地分支
- \$ git checkout <branch/tag> #切换到指定分支和标签 \$ git branch <new-branch> #创建新分支 \$ git branch -d <branch> #删除本地分支

- \$ git tag
   #列出所有本地标签

   \$ git tag <tagname>
   #基于最新提交创建标签
- \$ git tag -d <tagname> #删除标签

# 合并与衍合

\$ git merge <branch> \$ git rebase <branch></branch></branch>	# 合开指定分支到当前分支 # 衍合指定分支到当前分支	Ľ
远程操作		

#### ð \$ git remote -v #查看远程版本库信息

ð

ð



- \$ git remote show < remote > #查看指定远程版本库信息
- \$ git remote add <remote> <url> #添加远程版本库 \$ git fetch <remote> #从远程库获取代码
- \$ git pull <remote> <branch> #下载代码及快速合并 \$ git push <remote> <branch> #上传代码及快速合并
- \$ git push <remote\> :<branch/tag-name\> #删除远程分支或标签 #上传所有标签
- \$ git push --tags

更多内容请参见 Git 文档。



# LFS 大文件支持

最近更新时间: 2021-12-02 09:52:48

本文为您详细介绍如何使用 Git LFS 扩展。

# 功能介绍

CODING 支持 Git LFS (Git Large File Storage ) 扩展,使用 Git LFS 提交的大文件不占用 Git 仓库存储空间,理论上可以提交的单个文件大小无上限。

#### 安装

# Linux

1. 下载 git-lfs 安装包。

 curl -s https://packagecloud.io/install/repositories/github/git-lfs/script.deb.sh | sudo bash
 □

 2. 安装 git-lfs 。
 sudo apt-get install git-lfs

3. 将 LFS 工具部署到 Git 上。

|--|

### Mac

1. 安装 Homebrew。

	/usr/bin/ruby -e "\$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"	ð
2. ≴	安装 git-lfs 。	

3. 将 LFS 工具部署到 Git 上。

### Windows

- 1. 下载安装 Windows installer。
- 2. 运行 Windows installer。
- 3. 在命令行执行 git Ifs install 。

# 使用

有关 Git 操作请参见 Git 常用命令。

# 追踪文件

没有特别说明的情况下,Git LFS 不会处理大文件问题,使用 git lfs track 命令进行大文件追踪。

### 追踪单个文件

ð

Ð

ð

Ð

Ð

例如追踪一个名为 "coding.png" 的文件,使用 git lfs track "coding.png" 命令

#### 追踪同一后缀的所有文件

如果要追踪所有后缀为 "png" 的文件,使用 git lfs track "\*.png" 命令。运行此命令后,不但会追踪已存在的所有后缀为 "png" 的文件,也包括以后创建的 "png" 文件。

#### 查看正在追踪的文件模式 (patterns)

腾讯云

运行命令 git lfs track :

Listing	tracked patterns	
*.pna	(.gitattributes)	

#### 提交大文件

提交代码时需要将".gitattributes" 文件也提交到仓库,提交完成后,执行 git Ifs Is-files 命令可以查看 LFS 跟踪的文件列表。

f05131d24d \* cat.png 7db207c488 \* dog.png

将代码 push 到远程仓库后,LFS 跟踪的文件会以 "Git LFS" 的形式显示:

\$ git push origin master Git LFS: ( 2 of 2 files ) 12.58 MB / 12.58 MB Counting objects: 2, done. Delta compression using up to 8 threads. Compressing objects: 100% (5/5), done. Writing objects: 100% (5/5), 548 bytes | 0 bytes/s, done. Total 5 (delta 1), reused 0 (delta 0) To https://e.coding.net/coding/coding-manual.git 67fcf6a..47b2002 master -> master

#### 克隆包含 Git LFS 文件的远程仓库

使用 git Ifs clone 命令 clone 包含 "Git LFS" 文件的远程仓库到本地。

\$ git lfs clone https://e.coding.net/coding/coding-manual.git Cloning into 'coding-manual' remote: Counting objects: 16,done. remote: Compressing objects: 100% (12/12),done. remote: Total 16 (delta 3), reused 9 (delta 1) Receiving objects: 100% (16/16),done. Resolving deltas: 100% (3/3),done. Checking connectively...done. Git LFS: (4 of 4 files) 0 B / 100 B

#### () 说明

- 了解更多 Git LFS 的使用,可执行 git Ifs help 命令查看帮助。
- 如需将原有仓库的文件以 LFS 方式存储,请参见 参见教程。



# Go get 支持

最近更新时间: 2021-11-24 10:20:20

# 功能介绍

Golang Get 可以借助代码管理工具通过远程拉取或更新代码包及其依赖包,并自动完成编译和安装,使得整个过程就像安装应用程式一样简单。目前 CODING 代码仓库已支持在多仓库中使用 go get 功能,以下是快速上手指南。

# 快速开始

假如用户拥有 A 仓库,其代码仓库的 Git HTTPS 地址是: https://e.coding.net/{team}/{project}/{repo}.git (花括号中是变量,以实际情况为准),以 该仓库 https://e.coding.net/baulk/jackson/mux.git 为例: 用户可以通过以下命令设置好模块名。

go mod init e.coding.net/baulk/jackson/mux 口 如果要通过 go get 获得模块,可以运行如下命令:

go get e.coding.net/baulk/jackson/mux

另外多仓库也支持获取子仓库的模块:

go get e.coding.net/baulk/jackson/mux/dev

有一些存储库的 Git HTTPS 克隆地址是: https://e.coding.net/{team}/{project}.git ,用户可以通过如下命令设置好模块名:

go mod init e.coding.net/team/project

如果要通过 go get 获得模块,可以运行如下命令:

go get e.coding.net/team/project

#### ▲ 注意

此类存储库不完全支持直接获取子模块,您应该使用 e.coding.net/team/project/project 作为模块名、获得模块及其子模块。

ð

Ð

Ð

Ð