仓库管理

目录
包命名规范
包版本管理
里程碑
标签
标签组
命令
范围
状态
命令
必需标签
Bug
上游错误
新手友好问题
提交
拉取请求
合并前提条件
无法合并的贡献
合并条件
版本发布
WP-CLI 版本发布
软件包发布
技巧与工具
GitHub 搜索

包命名规范

为官方 wp-cli GitHub 组织创建包名时，需提前与 维护团队 商议。

在创建新仓库前，务必先进行讨论。这有助于就代码库的用途及其在 WP-CLI 生态系统中的最佳归属达成共识。

WP-CLI 命令的包名必须以 -command 后缀结尾，以确保清晰性，并区别于非命令仓库。

在为命令创建新仓库前，需先决定该命令是应拥有独立仓库，还是应包含在现有仓库中。

示例： wp-cli/scaffold-package-command。

包版本管理

包版本遵循 语义化版本规范（ SemVer）。

用通俗的话来说，这通常意味着：

补丁版本 （如 1.0.x）用于错误修复、文档变更、小幅功能改进等
次版本 （如 1.x.0）用于向包添加新命令，或对现有命令进行重大功能增强
主版本 （如 x.0.0）用于框架变更以破坏性方式改变 WP-CLI API 的情况

里程碑
开放的里程碑通常指向即将发布的下一个版本。

由于发布说明是使用里程碑构建的，因此计划合并的拉取请求和议题都需要设置为即将到来的里程碑，以便这
些拉取请求成为发布说明的一部分。

标签

标签是传达工作状态与进度的重要组织工具。 代码提交者 需确保标签实时更新。

准确维护议题和拉取请求的标签，有助于我们整体把握问题追踪器的使用情况，并能更便捷地检索已关闭议题
及拉取请求中的历史讨论记录。

标签组

标签可以属于标签组，这一概念适用于所有软件包。可用的具体标签可能因实际软件包而异，但标签组始终具
有相同的语义角色。

命令

用于定义特定问题 /拉取请求所适用具体命令的标签均以 command: 为前缀。

示例： command:cli-update

范围

用于定义当前问题 /拉取请求所适用范围的标签均以 scope: 为前缀。

已使用的范围：

scope:bootstrap - 引导过程的一部分，负责加载 WP-CLI 和 WordPress 核心。 scope:distribution - 分发过程的一部分，涉及 Phar 文件的构建和版本发布。 scope:documentation - 手册、命令参考或内联帮助文档的一部分。
scope:framework - WP-CLI 框架本身的一部分，提供架构、 API 和辅助函数以实现命令功能。 scope:testing - 单元测试或功能测试的一部分。
scope:website - wp-cli.org 或 make.wordpress.org 网站基础设施的一部分。

状态

用于定义议题 /拉取请求所处状态的标签均以 state: 作为前缀。

使用的状态包括：

state:unconfirmed - 议题中的错误 /问题尚未能复现，或可能与报告者的环境有关。 state:unsupported - 该议题不属于错误报告的范畴，无法在 GitHub

上获得支持。应引导报告者前往 支持渠道 。

命令

用于定义特定问题 /拉取请求与哪个具体命令相关的标签均以 command: 作为前缀。

部分示例如下：

command:core - 与 wp core 父命令下的一个或多个子命令相关。
command:cli-check-update - 与 wp cli check-update 命令相关。 command:post-meta-update - 与 wp post-meta update 命令相关。

必需标签

部分标签具有特殊含义，并可能用于后续的自动化工作流程。所有官方软件包均需使用这些标签。

Bug

bug 标签用于标记已确认的代码缺陷，即代码未产生预期结果的情况。
缺陷特指现有功能出现的异常行为。功能缺失不被视为缺陷，而属于功能增强范畴。

上游错误

upstream-bug 标签表示该问题是上游软件（例如 WordPress 核心、 PHP 等）中的错误，不会在 WP-CLI 中修复。

新手友好问题

good-first-issue 标签表示该问题适合希望入门的新贡献者作为起点。
新手友好问题范围较小，不需要广泛的技术专长或项目历史知识即可开始处理。

提交
禁止直接向软件包的默认分支（ main / master）提交代码。所有代码变更必须通过拉取请求流程进行。

拉取请求
所有代码变更均需通过拉取请求流程。

每个提交的拉取请求都需要经过 代码审查 ，并且必须获得至少一位 提交者 的批准。

合并前提条件

重要的拉取请求应事先关联相关议题，该议题需明确待解决的问题，并允许在实际编写代码前讨论最合适的解
决方案。

若拉取请求由 提交者 之一提交，且需要一般性代码审查，提交者应将拉取请求的 "审阅者 "设置为 wp-cli/committers ；若需要 /希望特定人员的专业知识，则应设置为一位或多位特定提交者个人资料。

若拉取请求由外部贡献者提交， 提交者 应积极响应并提供快速反馈，以鼓励进一步贡献。

无法合并的贡献

有时，无论付出多少额外努力，某个拉取请求可能都无法合并（例如超出范围等）。在这种情况下，最好尽可能温和而坚定地拒绝贡献者，这既能鼓励未来的参与，也能避免引发争论。

请确保：

1. 感谢贡献者付出的时间和努力。
2. 充分解释关闭该拉取请求的决定背后的理由。
3. 尽可能链接到相关的支持文档。

如果您想参考一个模板：

> 感谢 _____ 在此拉取请求上花费的时间。
>
> 我关闭此拉取请求是因为 。进一步说明， 。
>
> 更多详情，请参阅 _____ 和 _____ 。

合并条件

除了需要获得批准外，拉取请求在合并前还必须通过所有测试。当这两个条件都满足时，任何 提交者____ 都可以通过以下步骤合并拉取请求：

1. 确保已设置所有适用的标签。
2. 确保已设置正确的里程碑。
3. 确保合并后删除分支（如适用）。
4. 根据需要调整拉取请求的标题。理想的标题应能直接复制粘贴到发布说明中。

拉取请求合并后，提交者应确保所有相关问题也已关闭并分配到正确的里程碑。

版本发布

WP-CLI 版本发布

WP-CLI 的版本发布计划由 维护团队_____提前制定并公布。

当版本准备就绪时，将由一位 维护者____根据 发布检查清单_____进行准备和执行。

软件包发布

软件包发布按需进行，由 维护者团队 执行。在以下情况下，软件包会被标记为发布版本：

积累了合理数量的变更（已存在数周或更久），且这些变更纳入夜间构建版本将带来益处。
包含值得进入夜间构建版本进行预发布验证的重大功能增强（例如新增命令）。

如果存在适合纳入当前版本的开放拉取请求，则应推迟（数日至数周）标记发布版本。

软件包发布说明应包含每个已合并拉取请求的主题和链接。无需包含次要文档、编码规范和测试套件的变更。
可自由编辑主题以提高清晰度。示例可参考 wp-cli/scaffold-command v1.0.6。

重要的是，标记软件包发布版本也是进行最终产品质量审查的机会：

每个已合并的拉取请求都适合发布，且具备充分的测试覆盖和文档。
软件包的 README.md 文件已更新至最新状态。
除非明确计划，否则未引入破坏性变更。

新软件包版本被标记后， wp-make-coffee 机器人会在稍后向 wp-cli/wp-cli 提交一个包含 composer update 结果的拉取请求。当此拉取请求被合并时，软件包发布版本将最终进入 WP-CLI 的夜间构建版本。

技巧与工具

GitHub 搜索
GitHub 提供了许多实用的搜索方式。这些搜索可以保存为书签，或作为链接嵌入页面。

以下是一些示例：

所有 WP-CLI 仓库中按最后更新排序的未解决问题和拉取请求
所有 WP-CLI 仓库中无里程碑且按最后更新排序的已合并拉取请求
所有 WP-CLI 仓库中无标签且按最后更新排序的已关闭问题和拉取请求

版本号 #2
由 壳壳虫 创建于 2026-03-26 17:08:46 UTC
由 壳壳虫 更新于 2026-03-26 17:08:46 UTC