wp user application-password exists

目录

- 选项
- 示例
- 全局参数

检查指定应用程序的应用密码是否存在。

选项

有关所用语法约定的详细说明，请参阅 参数语法 参考文档。

<user>：要检查应用密码存在的用户的登录名、邮箱或用户 ID。

<app-name>：要检查应用密码存在的应用程序名称。

示例

```
# 检查指定应用程序的应用密码是否存在
$ wp user application-password exists 123 myapp
$ echo $?
1


# 检查应用密码是否存在，若不存在则创建的 Bash 脚本
if ! wp user application-password exists 123 myapp; then
    PASSWORD=$(wp user application-password create 123 myapp --porcelain)
fi
```

全局参数

这些 全局参数 在所有命令中具有相同的行为，并影响 WP-CLI 与 WordPress 的交互方式。

参数	描述
`--path=<path>`	WordPress 文件的路径。
`--url=<url>`	模拟请求来自指定 URL。在多站点中，此参数用于指定目标站点。
`--ssh=[<scheme>:][<user>@]<host|container>[:<port>][<path>]`	通过 SSH 对远程服务器（或使用 "docker"、"docker-compose"、"docker-compose-run"、"vagrant" 方案的容器）执行操作。
`--http=<http>`	通过 HTTP 对远程 WordPress 安装执行操作。
`--user=<id|login|email>`	设置 WordPress 用户。
`--skip-plugins[=<plugins>]`	跳过加载所有插件，或逗号分隔的插件列表。注意：mu-plugins 仍会被加载。
`--skip-themes[=<themes>]`	跳过加载所有主题，或逗号分隔的主题列表。
`--skip-packages`	跳过加载所有已安装的包。
`--require=<path>`	运行命令前加载 PHP 文件（可多次使用）。
`--exec=<php-code>`	运行命令前执行 PHP 代码（可多次使用）。
`--context=<context>`	在指定上下文中加载 WordPress。
`--[no-]color`	是否对输出进行着色。
`--debug[=<group>]`	显示所有 PHP 错误并增加 WP-CLI 输出的详细程度。内置组包括：bootstrap、commandfactory 和 help。
`--prompt[=<assoc>]`	提示用户输入所有命令参数的值，或指定为逗号分隔值的子集。
`--quiet`	抑制信息性消息。

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