【Bash】一键自动提交 Git 代码，再也不怕忘记提交！

Original LabHub LabHub 2025年02月11日 21:41 广东

【Bash】一键自动提交 Git 代码，再也不怕忘记提交！

在日常开发中，我们经常需要手动执行 `git add .` 、 `git commit -m "xxx"` 、 `git push` 这些重复性操作，时间久了不仅麻烦，还容易忘记提交。有没有办法一键完成 Git 提交和推送？

今天，就给大家介绍一个 **Git 自动提交脚本**—— `auto_commit.sh` ，让你的代码管理更高效，再也不用担心忘记提交代码啦！

💡 auto_commit.sh 能做什么？

这个脚本可以 **自动完成 Git 提交和推送**，包括以下功能： ✅ **自动切换到脚本所在目录**，无论在哪执行都不会出错。 ✅ **检测未提交的更改**，如果没有变更，直接退出，避免无意义提交。 ✅ **自动暂存所有更改**，确保所有修改都被包含在提交中。 ✅ **拉取远程最新代码**，避免本地代码与远程代码冲突。 ✅ **恢复本地修改**，确保拉取代码后，仍然保留你的本地更改。 ✅ **自动提交并推送**，省去手动输入命令的麻烦。

📜 脚本完整代码

下面是 `auto_commit.sh` 的完整代码，每一步都有详细注释：

文章封面
文章摘要
净化链接
原始链接
编辑markdown
下载markdown
下载PDF
解析markdown
下载所有图片
帮助

```
#!/bin/bash

#
            auto_commit.sh
           - 自动提交并推送 Git 代码
set -e  # 遇到错误时立即退出

# 获取脚本所在目录并切换到该目录
SCRIPT_DIR="$(cd "$(dirname "${BASH_SOURCE[0]}")" && pwd)"
cd "$SCRIPT_DIR"

# 获取当前时间戳
TIMESTAMP=$(date "+%Y-%m-%d-%H_%M_%S")

# 检查是否有未提交的更改
if git diff --quiet && git diff --staged --quiet; then
    echo "[INFO] 没有检测到更改，跳过提交。"
    exit 0
fi

# 暂存所有更改
echo "[INFO] 暂存所有未提交的更改到 stash..."
git stash push -m "Auto stash before pull"

# 拉取最新代码并合并更改
echo "[INFO] 拉取远程最新代码..."
if ! git pull --rebase origin master; then
    echo "[ERROR] 拉取失败，请手动解决冲突。"
    exit 1
fi
```

```
# 恢复暂存的更改
echo "[INFO] 恢复之前的更改..."
git stash pop || echo "[WARNING] 没有需要恢复的暂存更改。"

# 添加所有新文件和修改
echo "[INFO] 添加所有更改到暂存区..."
git add .

# 进行 Git 自动提交
echo "[INFO] 提交更改: auto commit at $TIMESTAMP"
git commit -m "auto commit at $TIMESTAMP"

# 推送到远程 master 分支
echo "[INFO] 推送到远程仓库..."
git push origin master

echo "[INFO] 操作完成！"
```

📌 代码详解

1️⃣ 确保脚本在正确的目录下执行

```
SCRIPT_DIR="$(cd "$(dirname "${BASH_SOURCE[0]}")" && pwd)"
cd "$SCRIPT_DIR"
```

无论你在哪个目录执行脚本，它都会自动切换到脚本所在的 Git 仓库目录，避免因为路径问题导致 Git 命令执行失败。

2 检测是否有改动，避免无意义提交

```
if git diff --quiet && git diff --staged --quiet; then
    echo "[INFO] 没有检测到更改，跳过提交。"
    exit 0
fi
```

如果没有任何代码变更，脚本会直接退出，避免创建无意义的空提交。

3 解决拉取代码时的冲突问题

```
git stash push -m "Auto stash before pull"
git pull --rebase origin master
git stash pop || echo "[WARNING] 没有需要恢复的暂存更改。"
```

- `git stash push`：将当前未提交的修改存入暂存区，确保 `git pull` 时不会冲突。
- `git pull --rebase`：拉取远程最新代码，并以 `rebase` 方式合并，避免产生多余的合并记录。
- `git stash pop`：拉取完成后恢复之前的修改。

4 提交并推送代码

文章封面
文章摘要
净化链接
原始链接
编辑markdown
下载markdown
下载PDF
解析markdown
下载所有图片
帮助

```
git add .
git commit -m "auto commit at $TIMESTAMP"
git push origin master
```

- `git add .` ：自动添加所有新文件和修改的文件。
- `git commit -m "auto commit at $TIMESTAMP"` ：自动生成提交信息，避免手动输入。
- `git push origin master` ：推送到远程 `master` 分支。

🚀 如何使用?

使用超级简单，只需两步：

1️⃣ **给脚本赋予执行权限**（仅需执行一次）：

```
chmod +x
            auto_commit.sh
```

2️⃣ **运行脚本**：

```
./
            auto_commit.sh
```

执行后，脚本会自动完成 Git 提交和推送，让你的代码管理更加轻松！

🌟 适用场景

✅ **个人开发者**：避免忘记提交代码，提升效率。 ✅ **团队协作**：保持代码最新，减少冲突。
✅ **自动化部署**：配合 `crontab` 或 CI/CD，定期提交代码。

❤️ 结语

这个 `auto_commit.sh` 脚本可以帮你 **省去手动提交代码的麻烦**，极大提升开发效率！如果你觉得有用，欢迎 **点赞**👍、**评论**💬、**转发**📢、**关注**，支持我分享更多实用工具！ 🚀🚀🚀

技术分享 22　实用工具 22　命令行工具 22　Linux系统自动化管理 22

技术分享 · 目录

上一篇
一键删除磁盘分区：适用于物理磁盘 & 虚拟磁盘的全能清理脚本！

下一篇
一键创建虚拟磁盘：从零开始自动化创建、分区、格式化！

文章封面
文章摘要
净化链接
原始链接
编辑markdown
下载markdown
下载PDF
解析markdown
下载所有图片
帮助