

Припрятывание в git

Желательные цели

Обозначения стадий выполнения

- (-) — еще не внесено в план
- (+) — внесено в план
- (v) — выполнено в bat
- (!) — работает
- (?) — вроде работает, но есть вопросы
- (~) — есть подпункты разной стадии

Если все подпункты имеют одинаковую стадию выполнения, то содержащему их (под)пункту присваивается та же стадия

Например:

1. (+) — (...)
 - a. (+) — (...)
 - b. (+) — (...)

Цели со стадиями

1. (~) — припрятывание
 - a. (!) — выполнить припрятывание из разных веток
 - b. (!) — посмотреть, как "коллекция" припрятываний выглядит в
 - i. (!) — git status
image::status.PNG[]
 - ii. (!) — git log
image::log.PNG[]
 - iii. (?) — Git Extensions (*сделать паузу на этом моменте в bat*)
 - c. (!) — найти способ увидеть, какие именно изменения хранятся в каждом припрятывании
image::stash_list.PNG[]
2. (~) — восстановление припрятывания
 - a. (!) — из того же места
 - b. (!) — из другого места (та же ветка, другая ветка и т.п.)
 - c. (?) — постараться получить конфликт при восстановлении (посмотреть как

выглядит, найти, как исправить)

Описание тестов

Смысловая разница двух тестов ниже

1. припрятанная "коллекция" в **stage_test1.bat** возвращается в свою ветвь каждая
2. припрятанная "коллекция" в **stage_test2.bat** возвращается в другую ветвь каждая

Смысловая разница двух ветвей в первом тесте **stage_test1.bat**

1. в главной ветви **stage_test1.bat** файл добавляется в stage
2. в ветви **branch_2 stage_test1.bat** файл **НЕ** добавляется в stage

План для **stage_test1.bat**

1. создать директорию для репозитория
2. инициализировать репозиторий
3. создать файл (file1.txt) в главной ветке
4. сделать первый коммит (иначе ругается stash)
5. редактировать file1.txt
6. положить в stage
7. припрятать коммит из главной ветви
 - a. посмотреть, как выглядит массив припрятанных "коммитов" (данных) через
 - i. status
 - ii. log
 - iii. Extensions (**сделать паузу на этом моменте в bat**)
 - b. вернуть припрятанный "коммит" (данные) из главной ветки в неё же

План для **stage_test2.bat**

1. создать директорию для репозитория
2. инициализировать репозиторий
3. создать файл (file1.txt) в главной ветке
4. припрятать коммит из главной ветви
 - a. посмотреть, как выглядит массив припрятанных "коммитов" (данных) через
 - i. status

- ii. log
 - iii. Extensions (*сделать паузу на этом моменте в bat*)
5. создать ветку branch_2
 6. перебранчеваться из главной ветки в branch_2
 7. создать файл (file2.txt) в ветке branch_2
 8. припрятать "коммит" (данные) из новой ветки branch_2
 - a. посмотреть, как выглядит массив припрятанных "коммитов" (данных) через
 - i. status
 - ii. log
 - iii. Extensions (*сделать паузу на этом моменте в bat*)
 - b. найти данные из branch_2
 9. вернуть и закоммитить припрятанный "коммит" (данные) из главной ветки в branch_2
 10. перебранчеваться из branch_2 в главную ветвь
 11. вернуть и закоммитить припрятанный "коммит" (данные) из branch_2 в главную ветвь

Текущие проблемы и вопросы

Обозначения стадий проблем и вопросов

- (-) — не решено
- (!) — решено

Проблемы и вопросы со стадиями

1. (-) — не отображаются цветные комментарии в консоли
2. (-) — "The system cannot find the batch label specified - log"
//вероятно, именно из-за этоо не работает п.1
3. (-) — Возвращает ли stash данные из stage, и, если да, куда: в stage или только рабочую папку?
Вроде бы работает так: из stage кладет в рабочую директорию