Лабораторная работа № 4.

Ветвления в репозитории git

Цель работы: познакомить студентов с понятиями branching (ветвление), тег и с приемами работы в git с ветвями и тегами.

Задание

1. Войти в систему со своим логином/паролем, в браузере открыть сайт https://gitea.pivt.spbgut.ru, ввести логин/пароль и найти репозиторий, созданный в предыдущей работе.

2. Перейти на вкладку Code, через ссылку Commits (Рис. 1) перейти к списку коммитов (Рис. 2) и отобразить коммиты в виде дерева (кнопка Commit Graph).

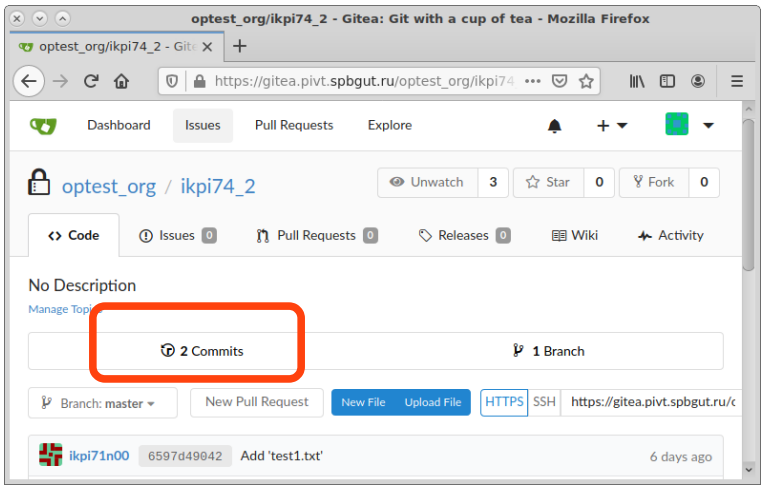

Рис. 1 Вкладка Code со списком файлов.

3. Изучить информацию о коммитах, представленную в дереве коммитов, в справочном материале прочитать пояснение к понятиям HEAD, master, хеш.

4. Отредактировать один из текстовых файлов в Gitea. В списке коммитов найти предыдущий коммит, переключиться на него и открыть тот же файл. Сравнить тексты файлов в этих коммитах.

5. Вернуться к последнему коммиту и обозначить его как релиз (выпуск) с тегом ver1.

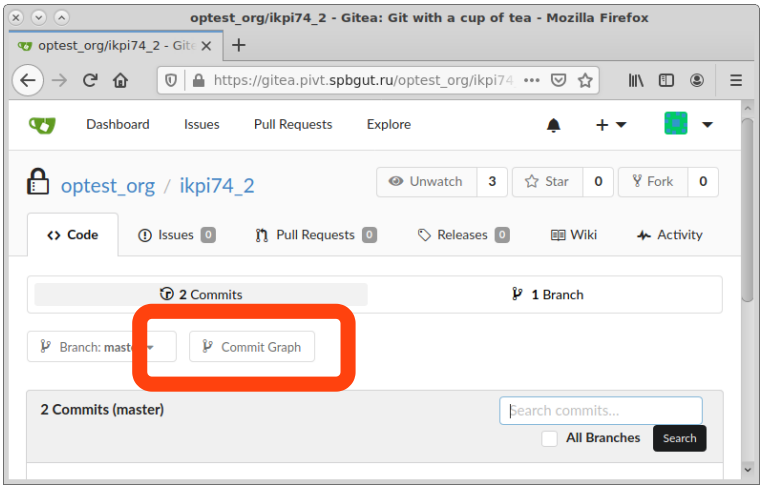

Рис. 2 Список коммитов и кнопка перехода к просмотру дерева коммитов

6. Открыть эмулятор терминала и перейти в корневой каталог вашего локального репозитория. Если локальный репозиторий отсутствует, клонировать удалённый репозиторий.

7. Проверить состояние локального репозитория, вывести список ветвей (git branch), список тегов (git tag).

8. Создать ветку branch1 (git branch branch1). Перейти в ветку branch1 (сделать ее активной) (git checkout branch1). Проверить, какие объекты в ней находятся. Изменить один из файлов и зафиксировать в локальном репозитории.

9. Перейти в ветку master и проверить ее состояние. Просмотреть список коммитов (логи) в виде дерева (git log - -graph). Какие ветки представлены в дереве? Появились ли изменения в файле, который был измёнен в ветке branch1. Сделать копию экрана.

10. Выполнить команду git log - -graph - -all и сравнить с результатом выполнения предыдущей команды. Сделать копию экрана.

11. Выполнить изменение файла и его фиксацию в локальном репозитории в ветке master, затем в branch1.

12. Выполнить слияние **активной** ветки branch1 и ветки master (git merge). Проверить, какая версия файла находится на вершине каждой ветки. Сравнить тексты (git diff). Выполнить синхронизацию текстов.

13. Вывести на экран содержимое репозитория в виде дерева и найти точку слияния веток. Через хеш перейти в коммит, принадлежащий branch1 и предшествующий слиянию, убедиться, что в рабочем каталоге находится предыдущая версия файла. Затем вернуться к последнему коммиту. Операции зафиксировать копиями экранов.

14. Создать тег tag1 для текущего коммита (git tag -a <имя тега>). Вывести список тегов.

15. Выполнить по два изменения в каждой из веток, каждое изменение зафиксировать в локальном репозитории, затем в удалённом репозитории.

16. Перейти в коммит через тег tag1, убедиться в доступности соответствующих версий файлов, затем вернуться в последний коммит.

17. Перейти в Gitea и проверить, получены ли изменения из локального репозитория (тег, коммиты).

18. Выполнить слияние веток в Gitea (вкладка Pull Request): в ветку master добавить изменения, сделанные в branch1. Если при слиянии возникнут конфликты, разрешить их любым доступным способом.

19. Создать в Gitea тег tag2 для коммита, зафиксировавшего слияние.

20. Создать в Gitea ветку branch2. Отредактировать непосредственно в Gitea в ветках master и branch2 один и тот же файл.

21. Перейти в локальный репозиторий, получить изменения из удалённого репозитория, в т.ч. ветку branch2, внести изменения в файлы веток master и branch2, зафиксировать в локальном репозитории.

22. Выполнить слияние веток master и branch2 в локальном репозитории (активная ветка master), в случае конфликта разрешить конфликт и завершить слияние. Для выполненного коммита создать тег tag3. Зафиксировать список тегов в копии экрана.

23. Зафиксировать изменения в удалённом репозитории. Перейти в Gitea и проверить доступность созданных тегов, ветвей, коммитов.

24. Сделать снимок экрана дерева коммитов репозитория (в локальном репозитории или в Gitea — по желанию).

25. Перейти в локальный репозиторий, удалить ветку branch2 из локального репозитория и проверить удаление. Затем удалить из удалённого (remote) репозитория ветку branch2.

26. Перейти в Gitea и проверить результат удаления ветки.

27. Результаты показать преподавателю.

Справочный материал

Словарь

Хеш — 16-ричное число, состоящее из 40 цифр (0-9, a-f), уникальное для каждого состояния репозитория (коммита). Вычисляется как контрольная сумма данных перед каждым сохранением в репозитории. Используется для обращения к конкретному коммиту (команды diff, checkout. В небольшом репозитории для обращения к коммиту можно использовать первые 7 цифр хеша.

HEAD — указатель на активный (текущий) коммит.

Ветвление (Branching) — отклонение от основной линии разработки.

branch — ветка, ответвление в репозитории для проведения экспериментов и устранения ошибок. Каждая ветка имеет имя.

master - название первой ветки проекта, создаваемой по умолчанию.

origin — название по умолчанию удалённого репозитория для его локального клона, обращение к удалённому репозиторию из локального.

Тег (tag) — метка, устанавливаемая на коммиты, которые считаются релизами (выпусками). Тег используется для быстрого перехода к коммиту.

tree – дерево каталога репозитория.

Команды git

git log - -pretty=”%h %s” - -graph - вывести объекты репозитория в виде дерева.

git log - -oneline - -decorate - -graph - -all – история коммитов в виде дерева с разветвлениями.

git branch - вывести список имеющихся в репозитории ветвей. Активная ветка будет отмечена символом *.

git branch <имя ветки> - создание ветки с именем …..

git checkout <имя ветки> - переход на указанную ветку.

git checkout -b <имя ветки> - создание новой ветки и переход на неё.

git pull origin <имя ветки> - получение изменений из указанной ветки удалённого репозитория.

git push origin <имя ветки> - сохранение изменений в указанной ветке удалённого репозитория

git merge <имя ветки>- слияние текущей ветки репозитория с веткой, указанной в команде. Текущая ветка принимает данные, а ветка, указанная в команде, является источником данных.

git tag -a <имя тега> - создание метки (аннотированный тег).

git tag - просмотр списка меток (тегов).

git diff <хеш 1 коммита> <хеш 2 коммита> <имя файла> - сравнение двух версий некоторого файла.

git diff <ветка> - сравнение текущей ветки и ветки, указанной в команде.

git branch -d <ветка> - удаление ветки в локальном репозитории.

git push origin - -delete <ветка> - удаление ветки из удаленного репозитория.

git push origin <тег или ветка> - отправка тега или ветки из локального репозитория в удалённый.

git push -u - выводит список отслеживаемых веток.

git fetch origin <ветка> - скачивает из удалённого репозитория в локальный новую ветку; ветка будет доступна после выполнения команды git checkout <ветка>.

Вопросы к защите

1. Что такое «ветка» в репозитории?
2. Какие операции можно выполнять над ветками?
3. Как представить ветвление в репозитории графически?
4. Что такое тег и как его можно использовать?
5. Как перейти к выбранной версии файла?