2.1 Основы Git - Создание Git-репозитория

Если вы хотите начать работать с Git, прочитав всего одну главу, то эта глава — то, что вам нужно. Здесь рассмотрены все базовые команды, необходимые вам для решения подавляющего большинства задач, возникающих при работе с Git. После прочтения этой главы вы научитесь настраивать и инициализировать репозиторий, начинать и прекращать контроль версий файлов, а также подготавливать и фиксировать изменения. Мы также продемонстрируем вам, как настроить в Git игнорирование отдельных файлов или их групп, как быстро и просто отменить ошибочные изменения, как просмотреть историю вашего проекта и изменения между отдельными коммитами (commit), а также как отправлять (push) и получать (pull) изменения в/из удалённого (remote) репозитория.

**Создание Git-репозитория**

Для создания Git-репозитория вы можете использовать два основных подхода. Во-первых, импорт в Git уже существующего проекта или директории. Во-вторых, клонирование существующего репозитория с другого сервера.

В обоих случаях вы получите готовый к работе Git репозиторий на вашем копьютере.

**Создание репозитория в существующей директории**

Если у вас уже есть проект в директории, которая не находится под версионным контролем Git, то для начала нужно перейти в эту директорию. Если вы не делали этого раньше, то для разных операционных систем это выглядит по-разному:

для Linux:

$ cd /home/user/my\_project

для macOS:

$ cd /Users/user/my\_project

для Windows:

$ cd C:/Users/user/my\_project

а затем выполните команду:

$ git init

Эта команда создаёт в текущей директории новую поддиректорию с именем .git, содержащую все необходимые файлы репозитория — структуру Git-репозитория. На этом этапе ваш проект ещё не находится под версионным контролем. (Подробное описание файлов содержащихся в только что созданной вами директории .git приведено в главе [[ch10-git-internals]](https://git-scm.com/book/ru/v2/ch00/ch10-git-internals))

Если вы хотите добавить под версионный контроль существующие файлы (в отличие от пустого каталога), вам стоит добавить их в индекс и осуществить первый коммит изменений. Добиться этого вы сможете запустив команду git add несколько раз, указав индексируемые файлы, а затем выполнив git commit:

$ git add \*.c

$ git add LICENSE

$ git commit -m 'initial project version'

Мы разберем, что делают эти команды чуть позже. Теперь у вас есть Git-репозиторий с отслеживаемыми файлами и начальным коммитом.

**Клонирование существующего репозитория**

Для получения копии существующего Git-репозитория, например, проекта, в который вы хотите внести свой вклад, необходимо использовать команду git clone. Если вы знакомы с другими системами контроля версий, такими как Subversion, то заметите, что команда называется "clone", а не "checkout". Это важное различие – вместо того, чтобы просто получить рабочую копию, Git получает копию практически всех данных, которые есть на сервере. При выполнении git clone с сервера забирается (pulled) каждая версия каждого файла из истории проекта. Фактически, если серверный диск выйдет из строя, вы можете использовать любой из клонов на любом из клиентов, для того, чтобы вернуть сервер в то состояние, в котором он находился в момент клонирования (вы можете потерять часть серверных хуков (server-side hooks) и т.п., но все данные, помещённые под версионный контроль, будут сохранены, подробнее об этом смотрите в главе [Установка Git на сервер](https://git-scm.com/book/ru/v2/ch00/r_git_on_the_server)).

Клонирование репозитория осуществляется командой git clone <url>. Например, если вы хотите клонировать библиотеку libgit2, вы можете сделать это следующим образом:

$ git clone https://github.com/libgit2/libgit2

Эта команда создаёт директорию libgit2, инициализирует в ней поддиректорию .git, скачивает все данные для этого репозитория и извлекает рабочую копию последней версии. Если вы зайдёте в новую директорию libgit2, то увидите в ней файлы проекта, готовые для работы или использования. Для того, чтобы клонировать репозиторий в директорию с именем, отличающимся от libgit2, необходимо указать желаемое имя, как параметр командной строки:

$ git clone https://github.com/libgit2/libgit2 mylibgit

Эта команда делает всё то же самое, что и предыдущая, только результирующий каталог будет назван mylibgit.

В Git реализовано несколько транспортных протоколов, которые вы можете использовать. В предыдущем примере использовался протокол https://, вы также можете встретить git:// или user@server:path/to/repo.git, использующий протокол передачи SSH. В главе [Установка Git на сервер](https://git-scm.com/book/ru/v2/ch00/r_git_on_the_server) мы познакомимся со всеми доступными вариантами конфигурации сервера для обеспечения доступа к вашему Git репозиторию, а также рассмотрим их достоинства и недостатки.