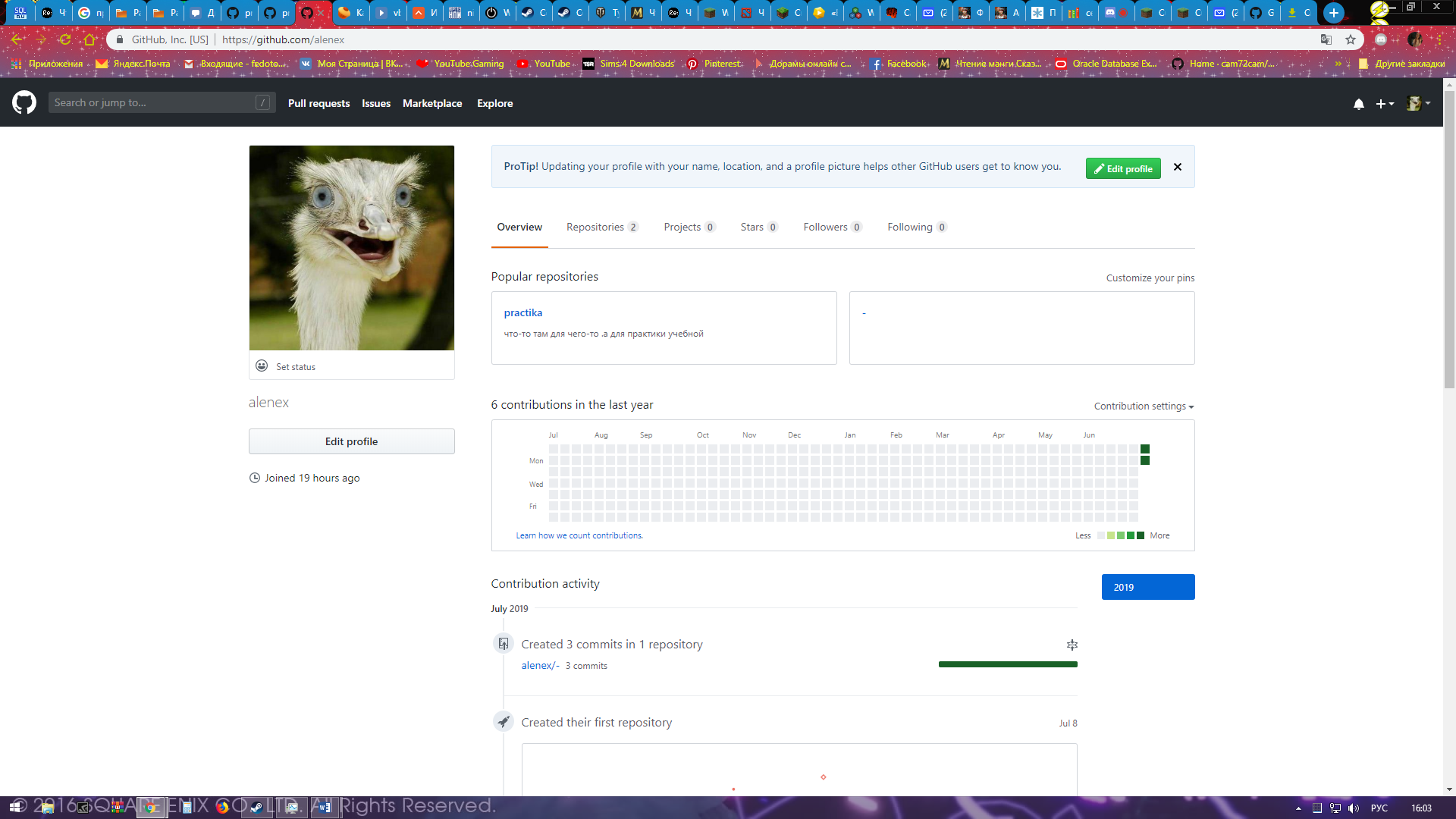
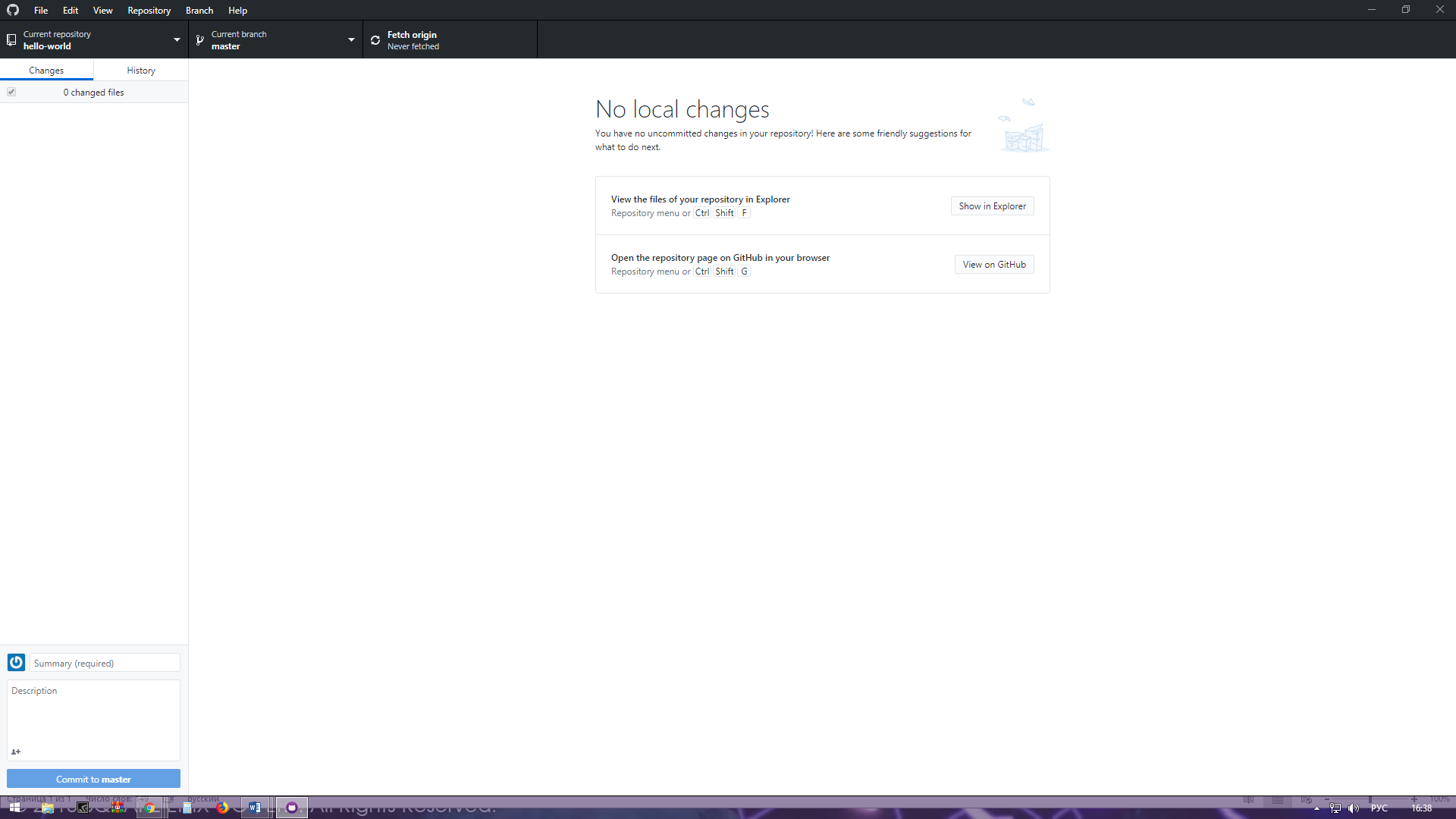
Регистрация на сайте



Установка приложения

клиента GitHub Deskop



GitHub - это платформа для размещения кода для контроля версий и совместной работы. Это позволяет вам и другим работать вместе над проектами из любой точки мира.

Из этого туториала вы узнаете основы GitHub, такие как репозитории , ветки , коммиты и запросы на извлечение . Вы создадите свой собственный репозиторий Hello World и изучите рабочий процесс GitHub's Pull Request, популярный способ создания и просмотра кода.

Хранилище обычно используется для организации одного проекта.

Репозитории могут содержать папки и файлы, изображения, видео, электронные таблицы и наборы данных - все, что нужно вашему проекту.

Ветвление - это способ работы с разными версиями хранилища одновременно.

На GitHub сохраненные изменения называются коммитами . Каждый коммит имеет соответствующее сообщение коммита , которое представляет собой описание, объясняющее, почему было сделано конкретное изменение. Сообщения фиксации фиксируют историю ваших изменений, чтобы другие участники могли понять, что вы сделали и почему.

Основные команды git

Всего несколько команд нужно для базового варианта использования Git для ведения истории изменений.

git add

Команда git add добавляет содержимое рабочей директории в индекс (staging area) для последующего коммита. По умолчанию git commit использует лишь этот индекс, так что вы можете использовать git add для сборки слепка вашего следующего коммита.

Это одна из ключевых команд Git, мы упоминали о ней десятки раз на страницах книги. Ниже перечислены наиболее интересные варианты использования этой команды.

git status

Команда git status показывает состояния файлов в рабочей директории и индексе: какие файлы изменены, но не добавлены в индекс; какие ожидают коммита в индексе. Вдобавок к этому выводятся подсказки о том, как изменить состояние файлов.

git diff

Команда git diff используется для вычисления разницы между любыми двумя Git деревьями. Это может быть разница между вашей рабочей директорией и индексом (собственно git diff), разница между индексом и последним коммитом (git diff --staged), или между любыми двумя коммитами (git diff master branchB).

git difftool

Команда git difftool просто запускает внешнюю утилиту сравнения для показа различий в двух деревьях, на случай если вы хотите использовать что-либо отличное от встроенного просмотрщика git diff.

git commit

Команда git commit берёт все данные, добавленные в индекс с помощью git add, и сохраняет их слепок во внутренней базе данных, а затем сдвигает указатель текущей ветки на этот слепок.

git reset

Команда git reset, как можно догадаться из названия, используется в основном для отмены изменений. Она изменяет указатель HEAD и, опционально, состояние индекса. Также эта команда может изменить файлы в рабочей директории при использовании параметра --hard, что может привести к потере наработок при неправильном использовании, так что убедитесь в серьёзности своих намерений прежде чем использовать его.

git rm

Команда git rm используется в Git для удаления файлов из индекса и рабочей директории. Она похожа на git add с тем лишь исключением, что она удаляет, а не добавляет файлы для следующего коммита.

git mv

Команда git mv — это всего лишь удобный способ переместить файл, а затем выполнить git add для нового файла и git rm для старого.

git clean

Команда git clean используется для удаления мусора из рабочей директории. Это могут быть результаты сборки проекта или файлы конфликтов слияний.