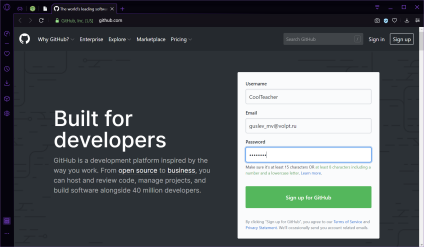
**GitHub** - хостинг для удалённых репозиториев, позволяет создавать репозитории на удалённом сервере.

*Важно!*

*Если инструкция не предоставит ответы на все ваши вопросы, то не ленитесь самостоятельно гуглить информацию для решения возникающих проблем.*

**1 Регистрация на GitHub**

**1.1 Авторизация**

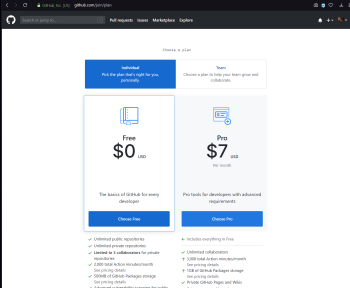


Заметка:

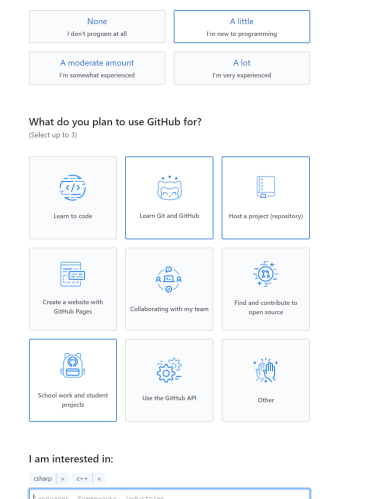
Если не удаётся выполнить регистрацию на главной странице, то нажмите в верхней панели кнопку «Sign up» и в открывшейся странице повторите ввод данных для авторизации.

**1.2 Выбор плана**

Воспользуемся бесплатным планом - Free Plane



**1.3 Проходим опрос (если есть желание)**



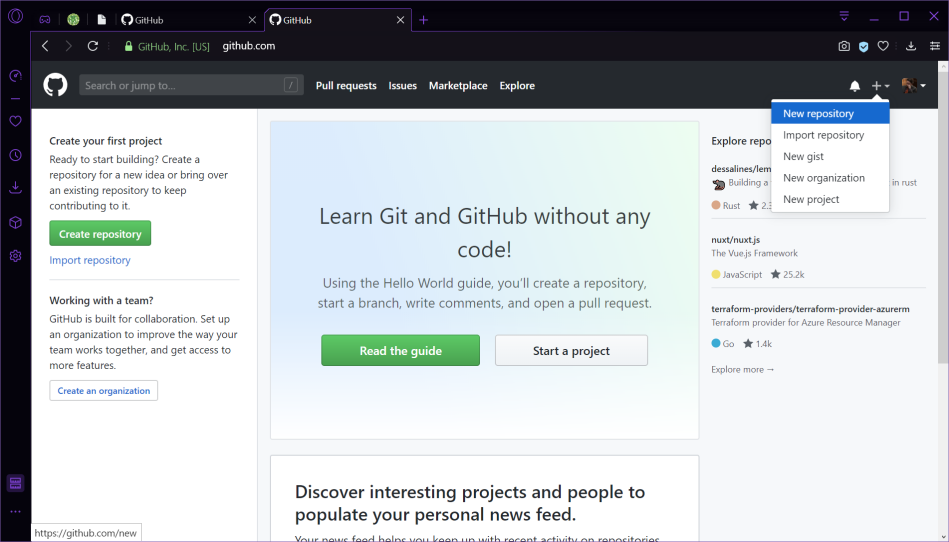
**1.4 Подтверждаем аккаунт**

Заходим на почту и открываем письмо от GitHub.  
Переходим по ссылке в письме для подтверждения созданного аккаунта.

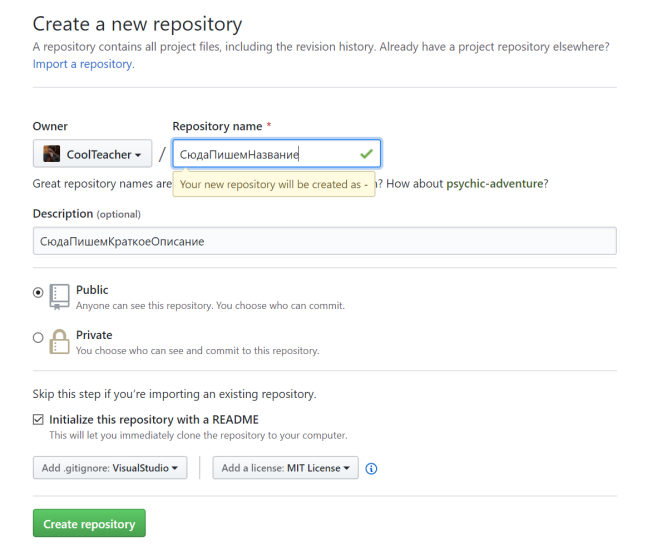
Далее будет открыта страница с предложением создать новый репозиторий. Если есть желание, то можно создать репозиторий, продолжив выполнять инструкции, описанные на странице, или можно перейти на главную страницу своего аккаунта и тем самым отменить создание репозитория. Для перехода на главную страницу аккаунта, следует нажать в верхней панели на логотип GitHub.

**2 Создание и настройка удалённого репозитория**

В верхней панели кликаем по кнопке «+» и в открывшемся меню выбираем «New repository»:



В открывшейся странице выполняем настройку репозитория



Указываем владельца (Owner), название репозитория (Repository name) и его краткое описание (Description, если нужно).

Заметка: в названии репозитория следует использовать только латинские буквы, цифры, тире и подчёркивание. Название репозитория следует делать идентичным названию проекта (но необязательно).

Далее следует

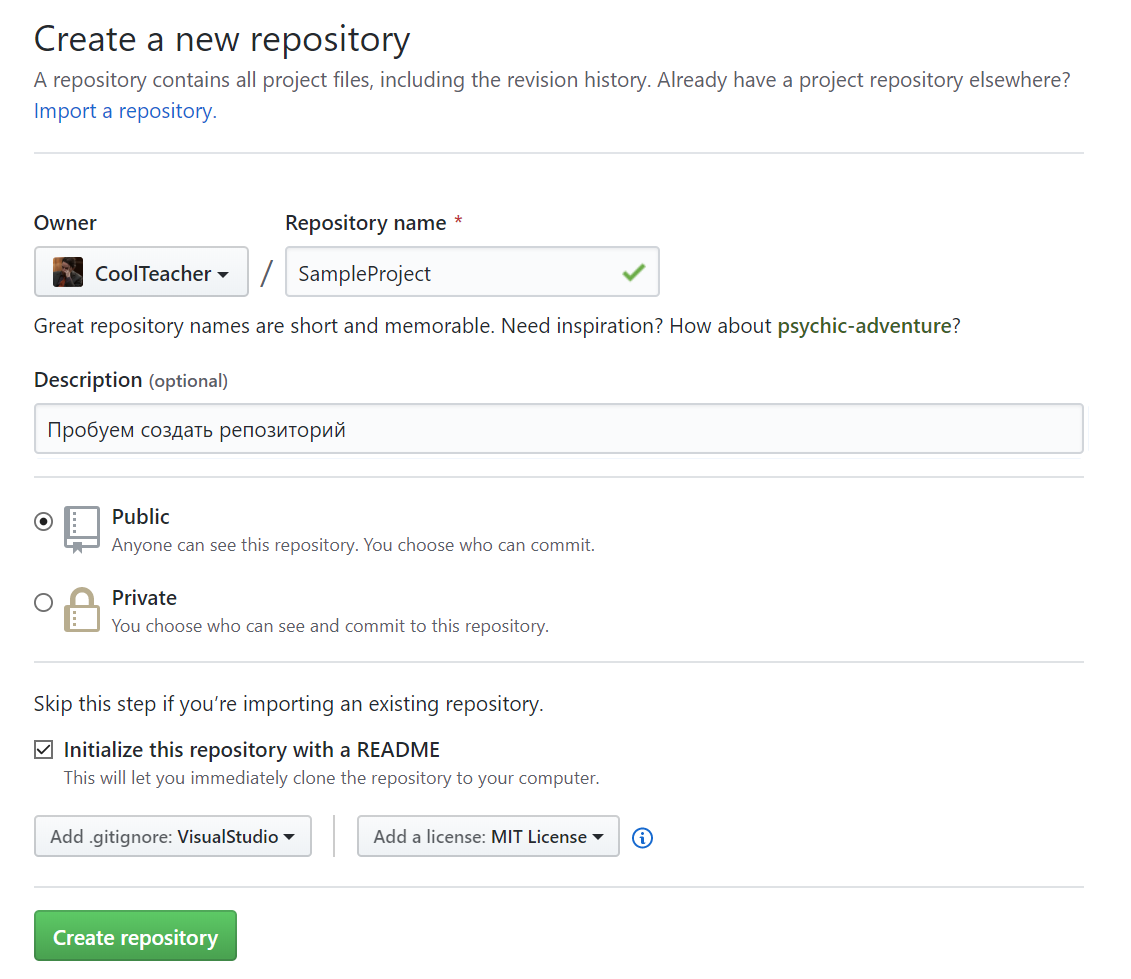
- выбрать тип репозитория - Public, т.е. доступный для всех.

- установить «Initialize this repository with a README» для автоматического создания файла с описанием содержимого репозитория

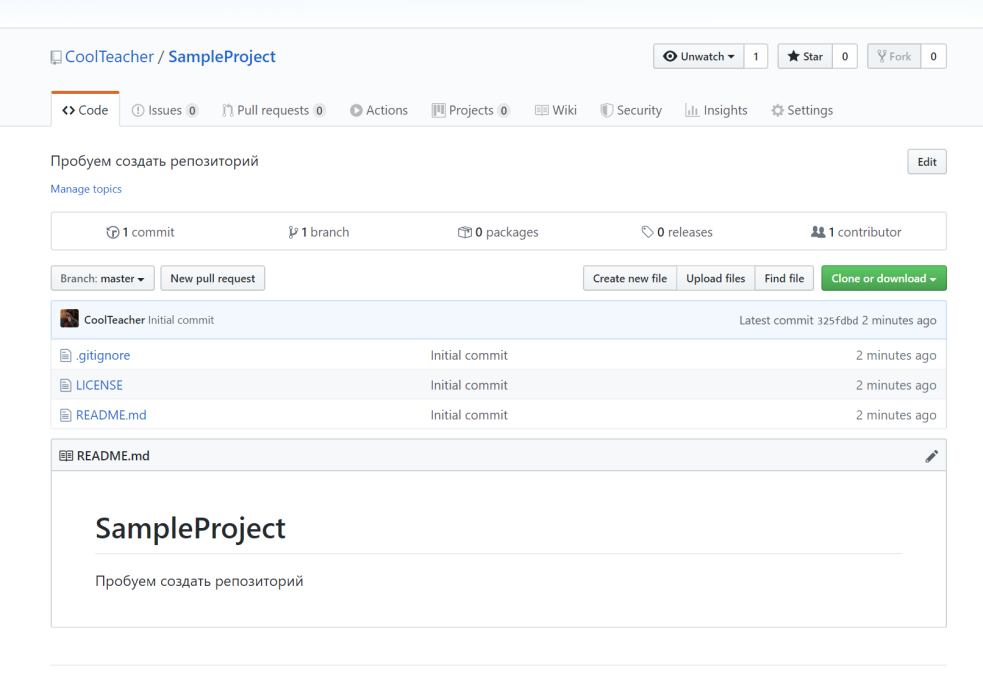
- настроить файл .gitignore под Visual Studio

- указать лицензию MIT

Пример

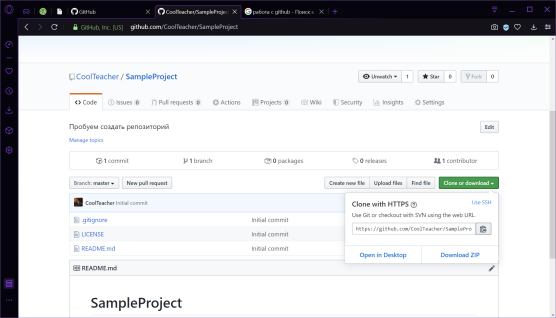


Подтвердив создание мы попадаем на страницу репозитория:



**3 Клонирование удалённого репозитория**

Чтобы начать работу с репозиторием его нужно клонировать себе на компьютер, после чего наполнить файлами проекта и отправлять все фиксации изменений проекта обратно на удалённый репозиторий.

1. Определяемся с каталогом в который будет производиться клонирование содержимого удалённого репозитория. Важно учитывать тот факт, что репозиторий клонируется с собственным каталогом, т.е. в выбранном локальном каталоге при клонировании будет создан подкаталог (название которого совпадает с названием репозитория) в котором и разместится содержимое репозитория.
2. Открываем GitBush и выполняем переход *(cd path\_to\_dir)* к выбранному локальному каталогу.
3. Со страницы репозитория копируем https ссылку на репозиторий:  
   
4. В GitBush выполняем команду git clone <https\_ссылка> и ожидаем завершения клонирования
5. Готовы наполнять каталог файлами проекта, редактировать и фиксировать изменения

**4 Отправка локальной истории изменений в удалённый репозиторий**

Для того, чтобы попрактиковаться в отправке локальных изменений в удалённый репозиторий, внесём изменения в файл README.md.

Открываем в любом текстовом редакторе файл README.md и изменяем его содержимое на:

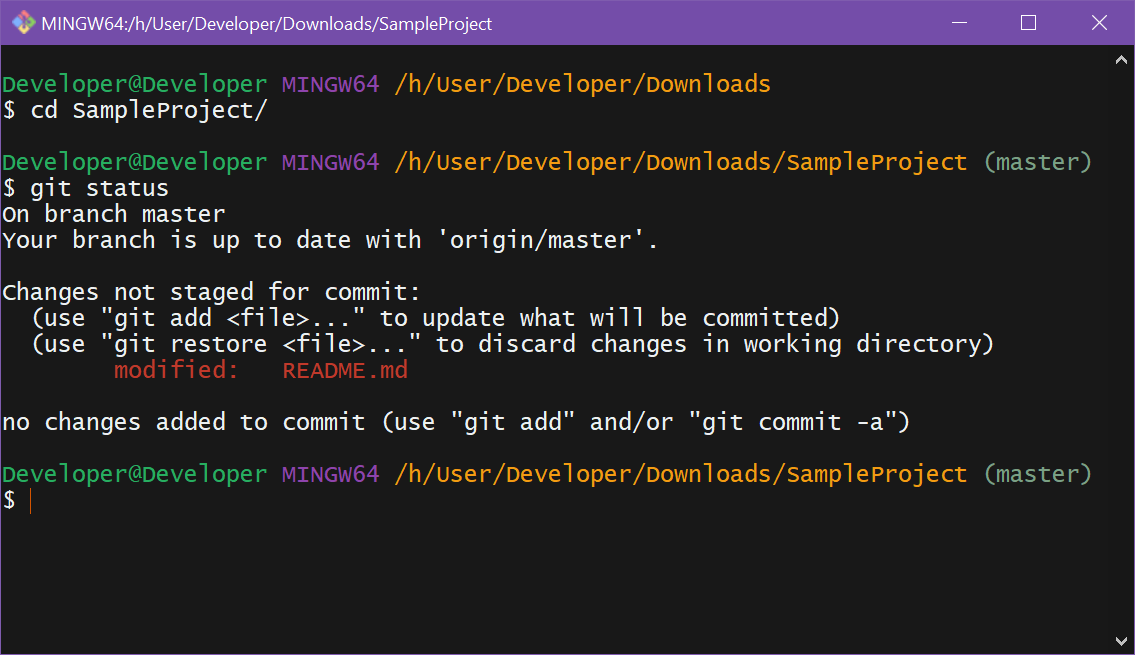
Привет!

Это мои первые попытки работы с удалённым репозиторием.

Сохраним изменения файла и закроем редактор.

Откроем GitBush и перейдём к каталогу репозитория (нужно перейти в каталог репозитория!).

Проверим состояние репозитория, выполнив команду git status, получим следующий результат:



Строка «modified: README.md» сообщает о том, что файл был модифицирован.

Выполним индексацию изменений этого файла командой git add README.md.

Зафиксируем изменения, создадим коммит с комментарием «added test changes to the readme file», выполнив команду git commit -m "added test changes to the file" .

Заметка! следующий этап может не потребоваться или потребуется один раз для каждого локального репозитория:

Теперь можно отправлять локальные изменения на удалённый репозиторий, но для этого нужно связать локальный репозиторий с удалённым, для этого выполним команду:

git remote add origin <https\_адрес\_репозитория>

Этот этап точно потребуется выполнять каждый раз:

Отправляем изменения на удалённый репозиторий:

git push -u origin master

Вводим логин и пароль для проведения операции. Локальные изменения были залиты на удалённый репозиторий.