**Практическая работа**

**Тема:** настройка работы системы контроля версий. Разработка и интеграция модулей проекта.

**Цель:** закрепить навыки настройки системы контроля версий и работы в ней.

**Задание 1:**  **ПРАКТИЧЕСКИЙ БЛОК**

**Условие**

1. Зарегистрируйтесь на GitHub, если еще не сделали это, и подтвердите свою почту.

2. Настройте **SSH** или **HTTPS** подключение (на ваш выбор). Мы будем работать по **HTTPS**.

3. Теперь перейдите на страницу <https://github.com/smartiqaorg/geometric_lib/> и создайте свой форк.

4. Склонируйте к себе свой форк репозитория **geometric\_lib**.

5. Настройте локального пользователя Git

6. Измените файл **docs/README.md**. Сделайте коммит.

7. Выполните пуш ваших изменений в свой удаленный репозиторий. Заметьте, что вам не нужно добавлять удаленный репозиторий: он уже был добавлен под именем **origin** во время клонирования.

8. Откройте страницу вашего репозитория на GitHub. Можете убедиться, что изменения были успешно загружены в удаленный репозиторий, просмотрев историю коммитов на главной странице репозитория.

9. Создайте пулл-реквест. В нем подробно опишите внесенные изменения. На странице предпросмотра пулл-реквеста проверьте, что вы не затронули файлы на других ветках и еще раз перепроверьте код

Задание 2: Ответьте на вопросы:

1. Для чего нужен сервис GitHub?
2. Что такое токен? Для чего нужен?
3. Что такое форк (fork)?
4. Что такое пулл-реквест (pull-request)?

Теория.

**7** GitHub. Работа с репозиторием, создание форков и пулл-реквестов

Научившись работать с локальным репозиторием, освоим основы работы с GitHub. Итак, если коротко, то GitHub – это, наверное, самый популярный сервис бесплатного хостинга удаленных репозиториев с множеством дополнительных функций. Среди них есть, например, создание **issue**– запросов, в которых можно сообщить разработчикам об ошибках, создание репозиториев-форков и пулл-реквестов. Кроме того, на GitHub можно подписаться на обновления какого-то конкретного пользователя или включить отслеживание репозитория вашего любимого проекта. Все это превращает GitHub в настоящую социальную сеть для разработчиков по всему миру. Давайте приступим к изучению основ работы с этим сервисом.

**7.0 Настройка подключения по HTTPS**

Раньше можно было подключаться по **HTTPS**, используя имя пользователя и пароль от аккаунта GitHub. Но потом эту возможность отключили в целях безопасности. Сейчас вместо пароля нужно использовать персональный **токен**. Давайте разберемся, как создать такой токен.

Итак, чтобы создать токен персонального доступа, следуйте инструкции:

1. Подтвердите свой email-адрес, который вы использовали при регистрации аккаунта GitHub (если он не подтвержден

2. Кликните на свою аватарку в правом верхнем углу, в открывшемся окне выберите **Settings** (Настройки):

3. Перед вами откроются настройки. В меню слева выберите **Developer settings**(Настройки разработчика)

4. Перед вами откроются Настройки разработчика. В меню слева выберите **Personal access tokens** (токены персонального доступа).

5. Нажмите **Generate new token** (сгенерировать новый токен)

6. Придумайте имя для своего токена. Имя должно описывать, зачем токен был создан.

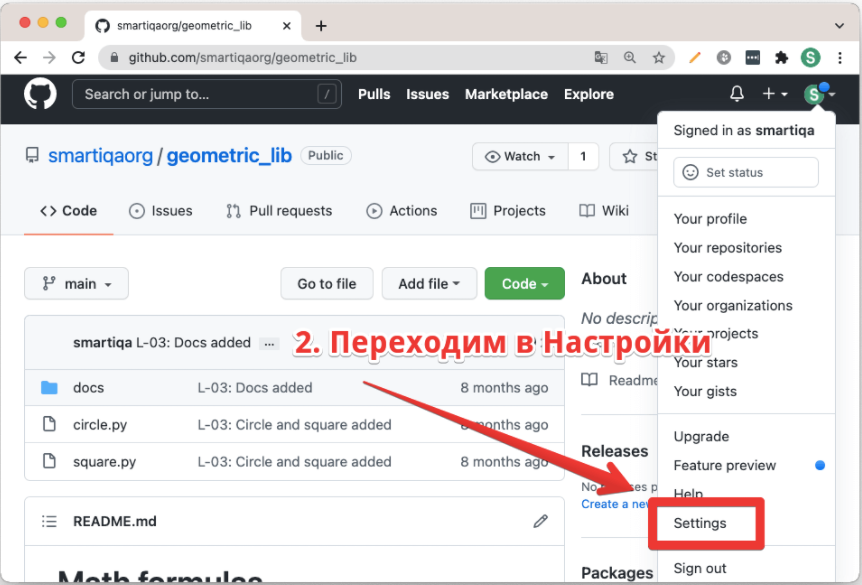
7. Дайте токену разрешения. Пользователь токена сможет выполнять с удаленным репозиторием только то, что вы указали в этих разрешениях. Чтобы дать токену доступ к управлению удаленным репозиторием из командной строки, выберите **repo**.

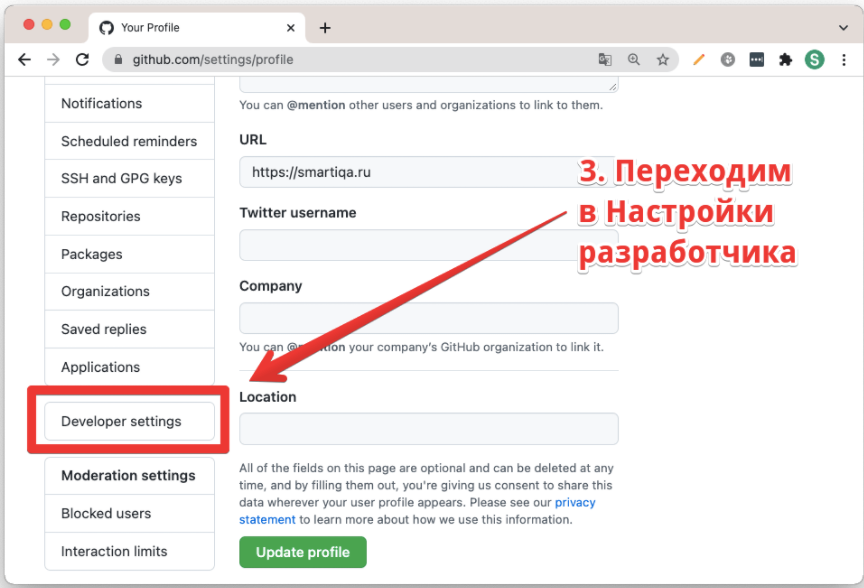
8. Нажмите **Generate token** (сгенерировать токен).

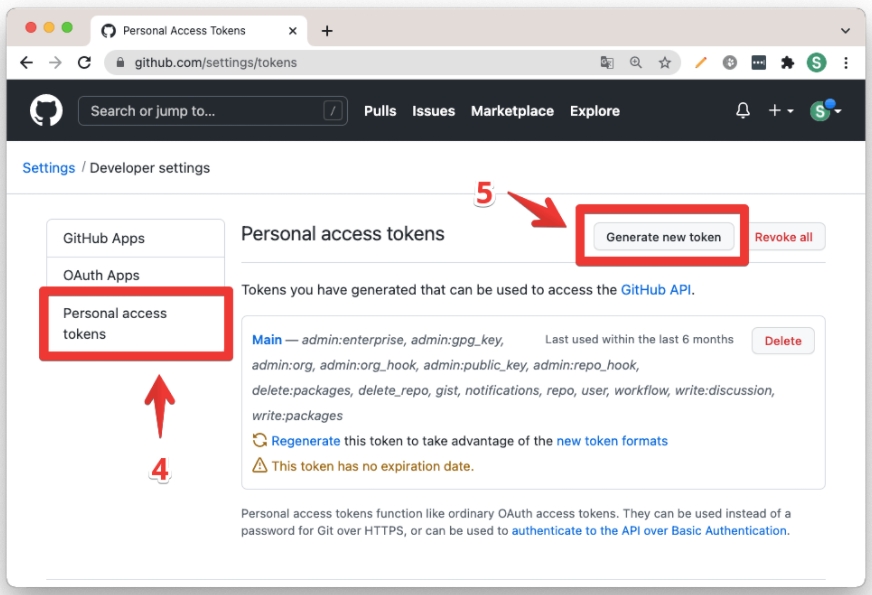
**Пример токена**

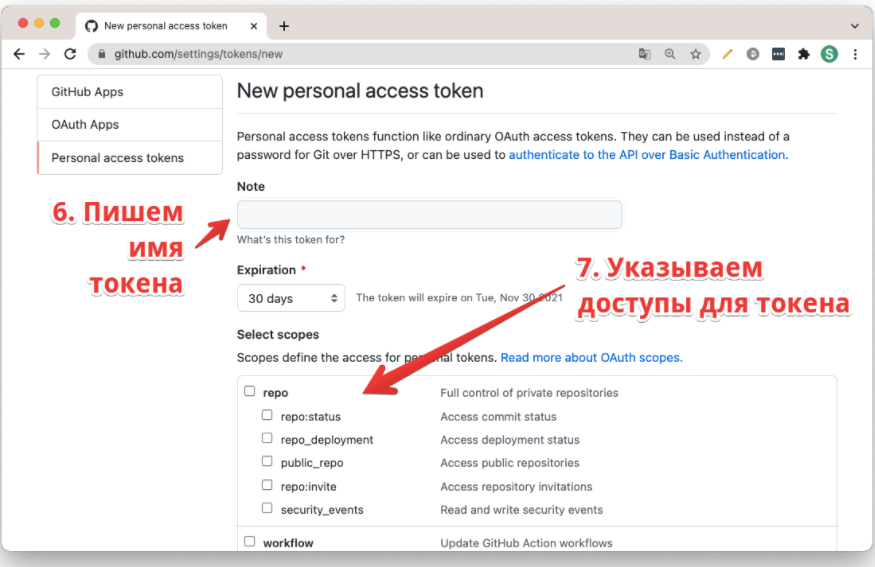
ghp\_VgHsXfkTbIdx5tsIu1vMJTLutA74BS1f0As0

9. Скопируйте токен. В целях безопасности, как только вы покинете страницу создания токена, вы больше не сможете просмотреть этот токен.

* 





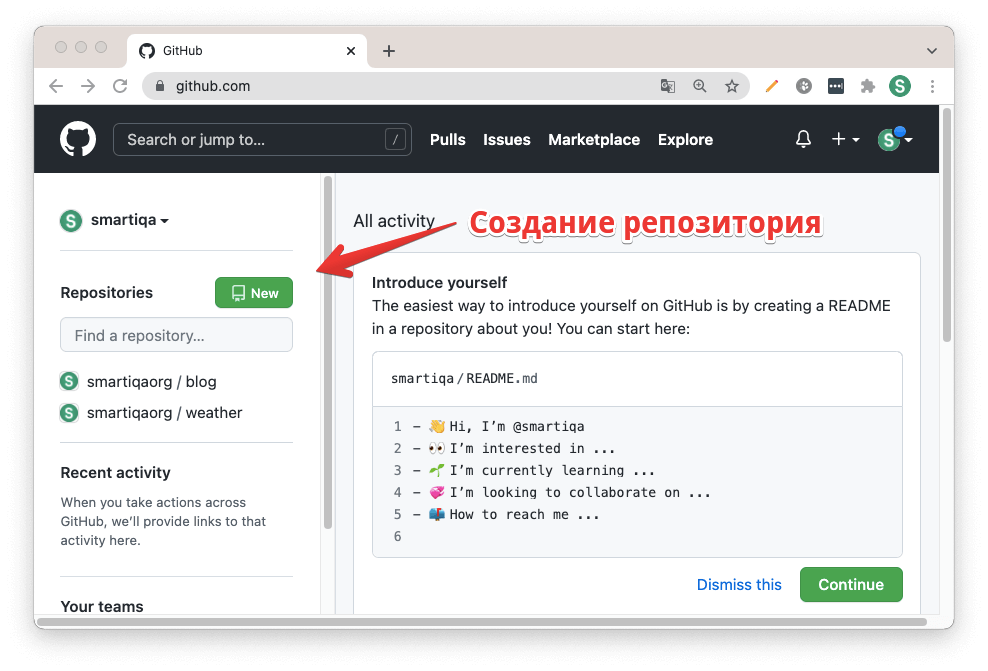


Отлично, ваш токен готов! При первой загрузке/скачивании изменений из удаленного репозитория, вас попросят ввести имя пользователя на GitHub и пароль. Нужно будет ввести свое имя пользователя, а вместо пароля вставить этот токен.

### 7.1 Создание репозитория на GitHub

Прежде всего вам необходимо зарегистрироваться на GitHub, но это довольно тривиальный процесс, так что его мы здесь опустим. После регистрации вы попадете на главную страницу. На ней будут отображаться действия людей, на которых вы подписались и обновления в репозиториях, которые вы добавили в избранное.

Чтобы создать свой репозиторий, нажмите на зеленую кнопку **New**, как показано на рисунке.

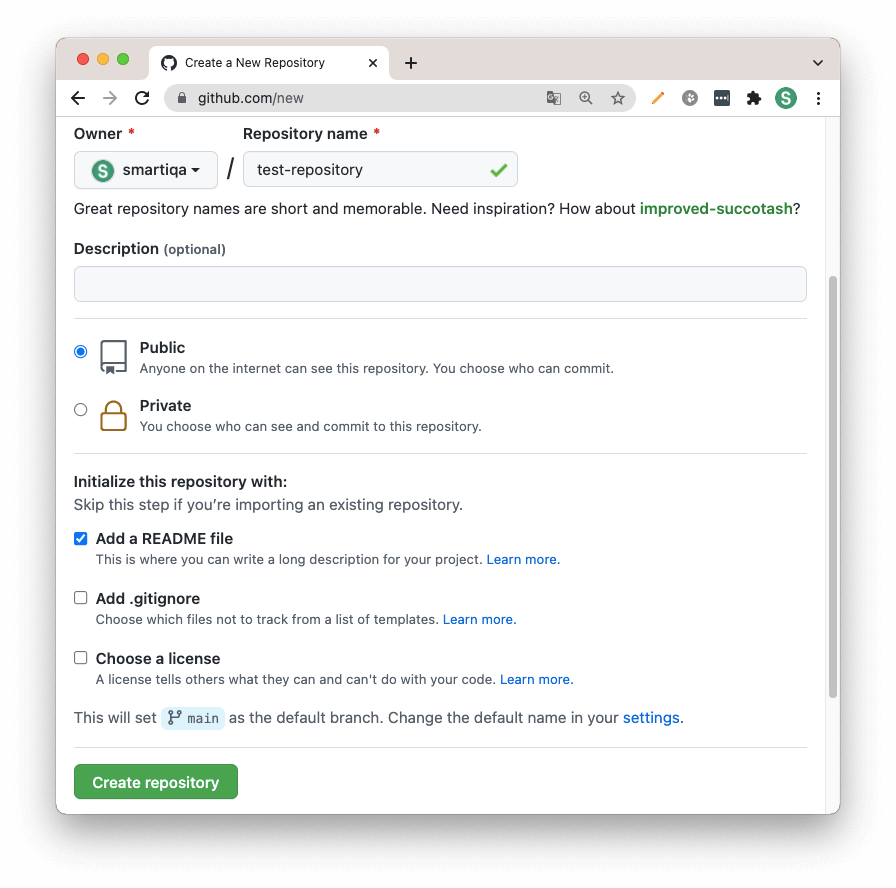


Создание репозитория - кнопка "New"

Перед вами откроется страница создания репозитория. Давайте разберем, что за**поля**нам предлагают заполнить.

1. Итак, первое поле **Repository name** – имя репозитория. Здесь все просто, вам нужно придумать имя, которое будет отображаться на странице вашего репозитория. Здесь нет никаких ограничений, но старайтесь давать как можно более содержательные имена своим репозиториям.
2. Второе поле – **Description**– описание. Его заполнять необязательно. Но другим пользователям, которые попали на страницу вашего репозитория, будет проще понять, что перед ними, если вы заполните графу описания.
3. Затем вы можете выбрать, будет ли репозиторий открытым, то есть доступным абсолютно всем пользователям GitHub, или закрытым, то есть доступным только вам и людям, которым вы предоставите доступ.
4. Последние три поля предлагают нам добавить, соответственно, **README-файл**, **.gitignore** файл и выбрать лицензию для нашего проекта.

После заполнения полей страница выглядит примерно так.



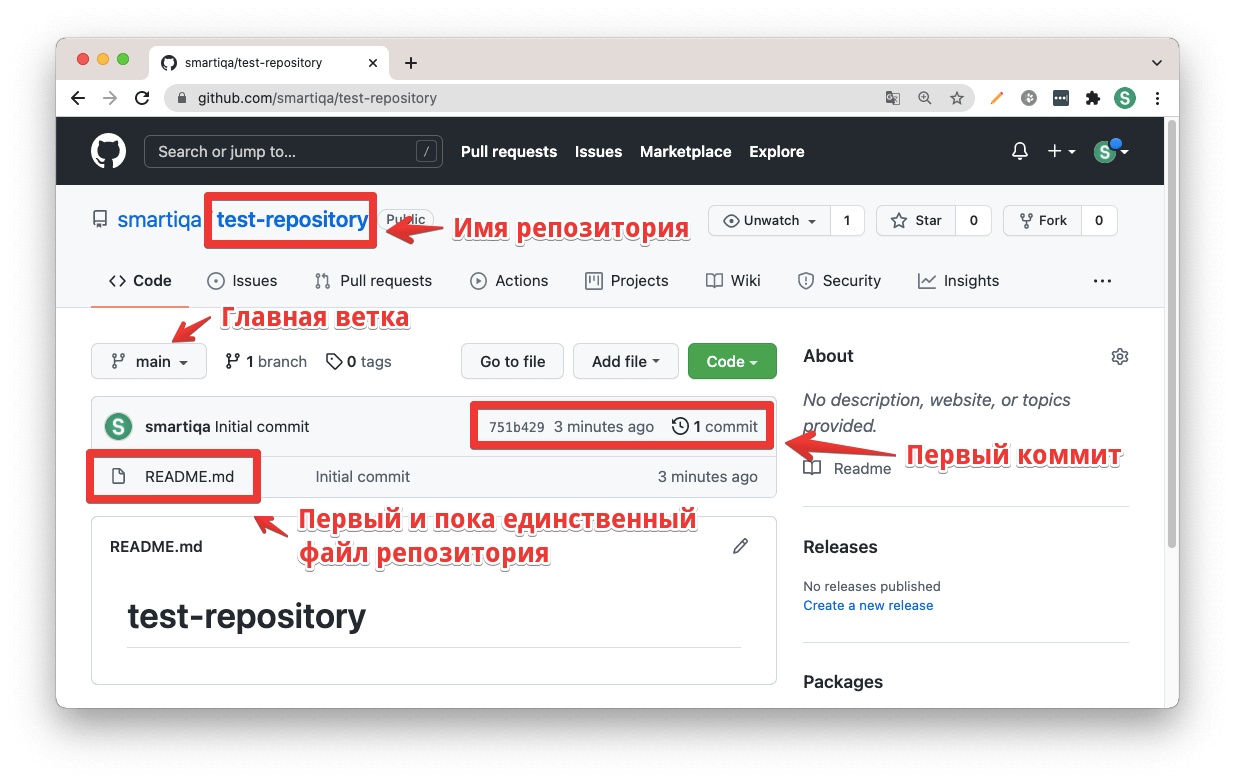
Заполненная страница создания репозитория

Заметьте, мы не стали вводить описание репозитория, поскольку решили добавить в него **README-файл**, хотя иногда и можно продублировать самую основную информацию из **README**в описании.

Завершим процесс создания репозитория, нажав кнопку **Create repository**.

**7.2 Страница репозитория на GitHub.**

После создания репозитория, мы попадем на его страницу на GitHub. В нашем случае эта страница будет выглядеть так.



Страница репозитория на GitHub

Как видно из рисунка, GitHub автоматически создал первый коммит, добавив в него файл **.gitignore** и файл **README**.

Кстати, можно заметить, что содержимое файла **README**выводится под рабочей копией репозитория. Это одна из особенностей GitHub. Вы в любое время можете создать файл с именем **README.md** и запушить его в свой удаленный репозиторий на GitHub. Тогда содержимое этого файла будет отображаться прямо на странице вашего репозитория.

В верхнем меню мы видим **9 разных вкладок**. Давайте разберем их по порядку.

1. Вкладка **Code**. Сейчас открыта именно она. В ней содержится рабочая копия нашего репозитория (по центру), описание (справа), вывод файла **README**(под рабочей копией), история коммитов, а также кнопки для клонирования репозитория и просмотра файлов.  
2. Вкладка **Issues**. В этой вкладке будут отображаться все запросы, сделанные другими пользователями. Как правило, пользователи используют запрос, чтобы сообщить о найденном баге, либо чтобы задать какой-то вопрос о вашем приложении.

3. Вкладка **Pull-requests**. На этой вкладке будут отображаться все пулл-реквесты, сделанные другими пользователями. О том, что такое пулл-реквесты, мы поговорим ниже.

4-5. Вкладки **Actions**и **Project**относятся скорее к системе **CI/CDI**, которую предоставляет GitHub, в этом курсе мы не будем затрагивать их.

6. Вкладка **Wiki**открывает вам доступ к созданию и размещению документации о собственном проекте.

7. На вкладке **Security**содержатся различные настройки безопасности вашего проекта. Там же можно включить инспекцию вашего кода, чтобы узнать, если вы случайно загрузите какой-нибудь секретный токен на GitHub.

8. Вкладка **Insight**содержит различную информацию и статистические данные об активности репозитория. Там вы сможете посмотреть на зависимость количества коммитов в репозитории от времени или на процент коммитов, сделанных вами.

9. Последняя вкладка – **Settings**. В ней находятся различные настройки вашего репозитория. Там вы можете поменять видимость репозитория, сделав его частным, или вовсе удалить репозиторий.

**7.3. Создание форка репозитория на GitHub. Пулл-реквесты.**

Итак, одной из самых важных частей GitHub является создание форков.

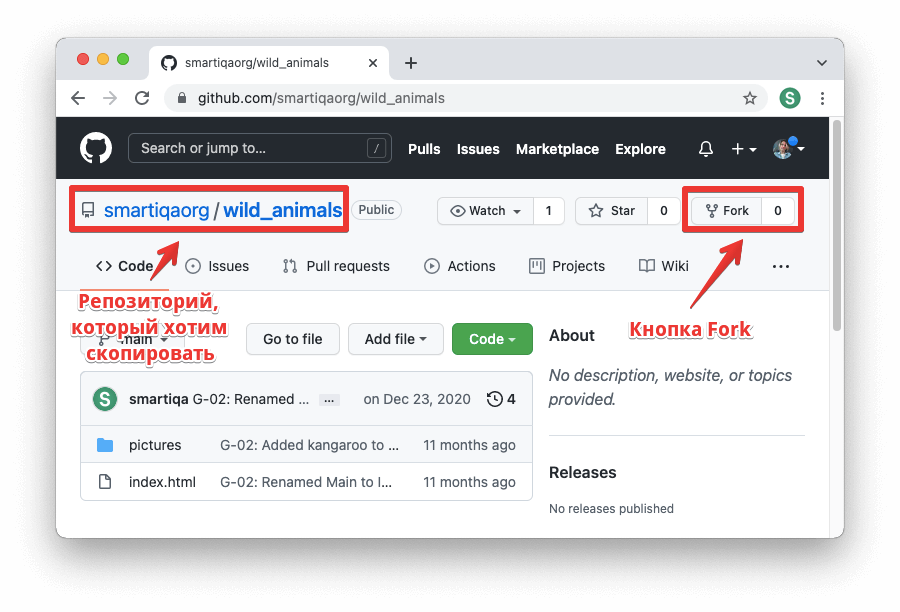
**Форк**(от англ. **fork**– вилка) – точная копия репозитория, но в вашем аккаунте. Форки нужны, чтобы вносить свои изменения в проект, к репозиторию которого у вас нет прямого доступа.

**Пулл-реквест**(от англ. **pull-request** – запрос **pull**) – функция GitHub, позволяющая попросить владельца репозитория, от которого мы сделали форк, загрузить наши изменения обратно в свой репозиторий.

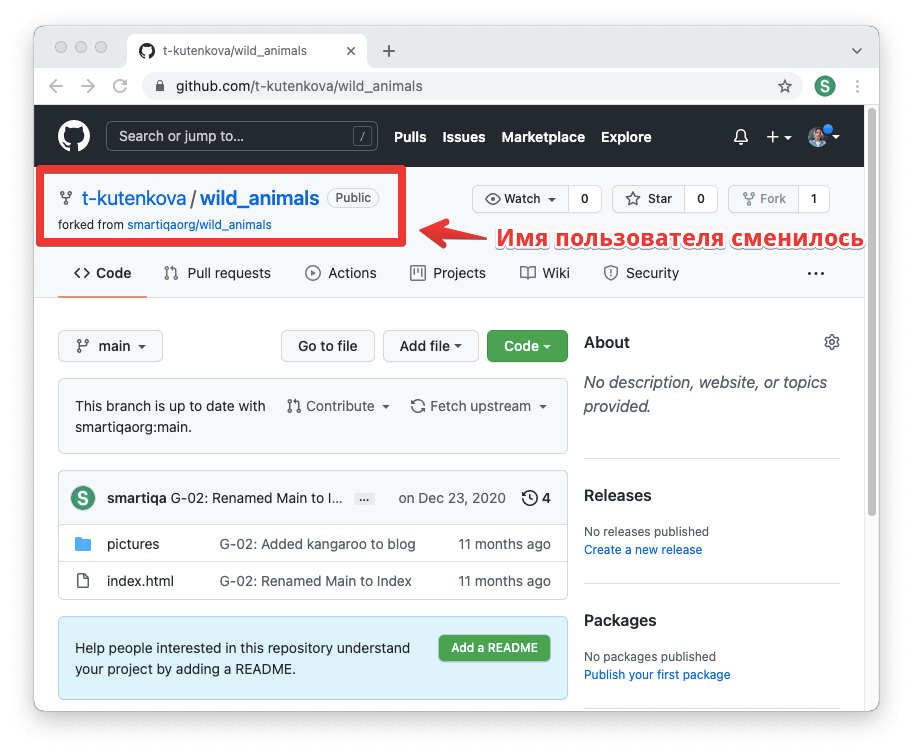
Если коротко, форки и пулл-реквесты нужны, чтобы любой пользователь мог внести свой вклад в любой открытый проект, репозиторий которого есть на GitHub. Кроме того, перед тем как влить ваши изменения в основной репозиторий, ответственные обязательно проверят ваш код на наличие ошибок и уязвимостей. Таким образом, даже если ваши изменения не примут, вы получите первоклассный **code-review**с указанием всех неточностей.

Теперь давайте рассмотрим **пайплайн контрибуции** (англ. **contribution**– внесение вклада) на примере реального репозитория. То есть научимся вносить свой вклад в разработку проектов с открытым исходным кодом на GitHub.

1. Для начала зайдем на страницу репозитория проекта. Нажимаем на кнопку **Fork**, как показано на картинке. После этого Git создаст точную копию этого репозитория в вашем аккаунте.



Кнопка Fork

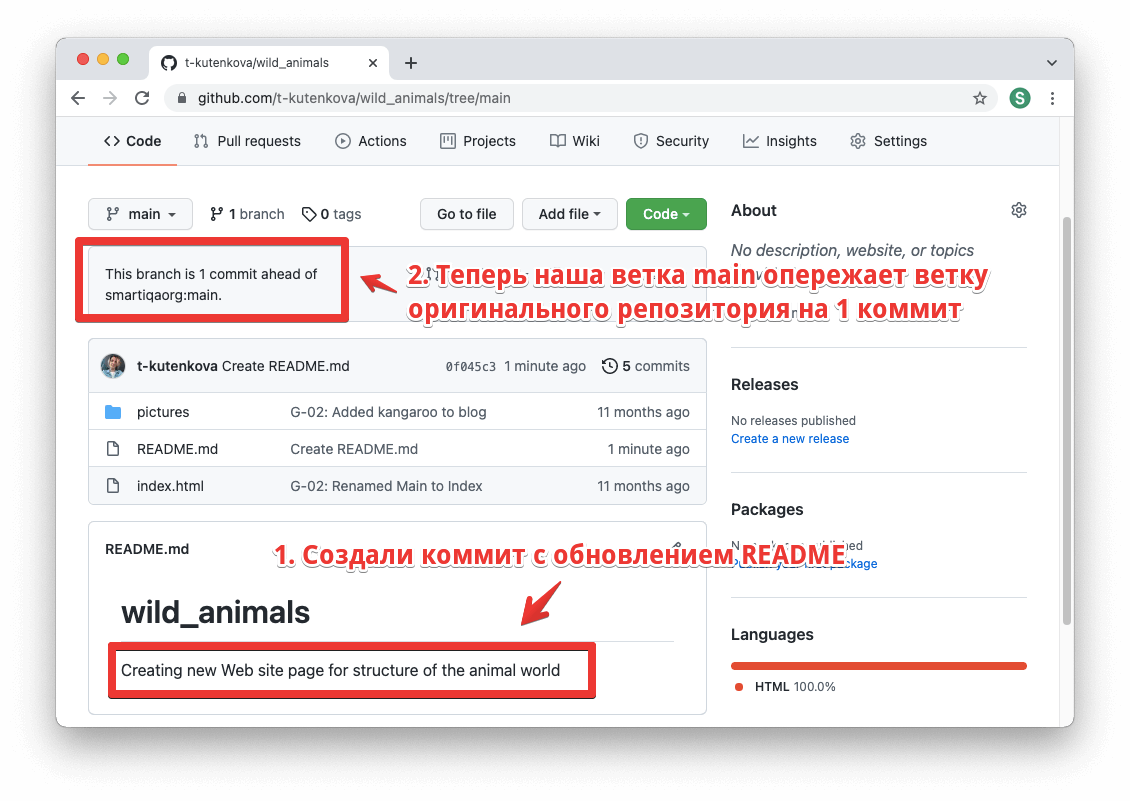


Fork-нутый репозиторий

2. Клонируем репозиторий к себе на компьютер командой **git clone**. Создадим файл **README.md**с описанием проекта, чтобы другим пользователям было понятно, в чем отличие этой реализации от остальных.

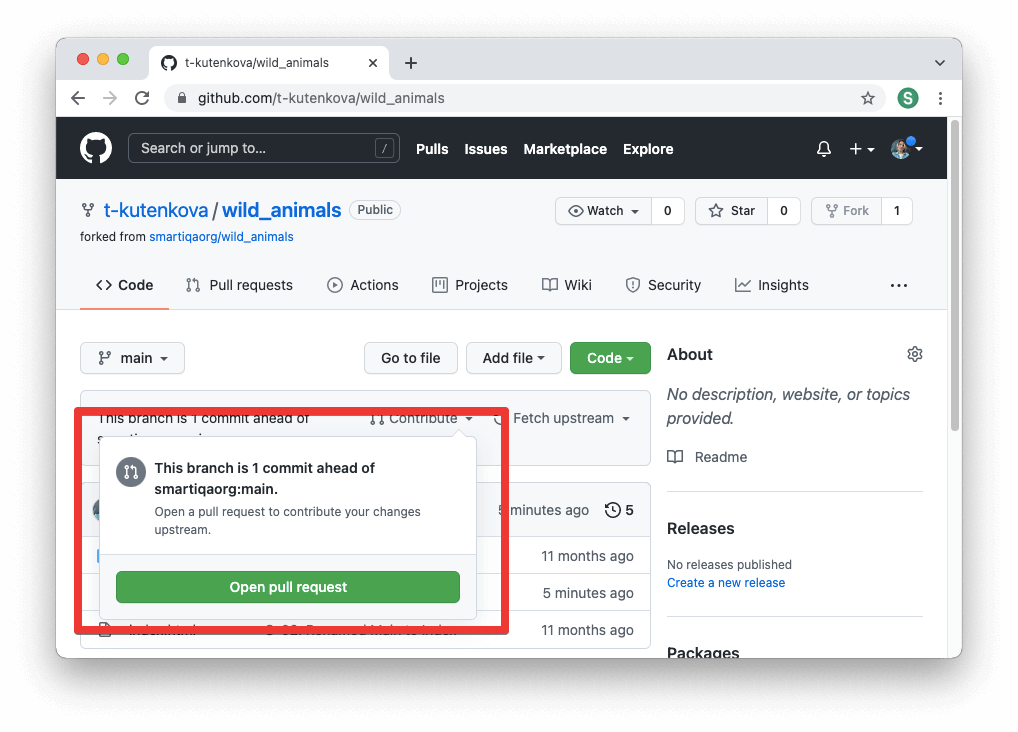
3. Сделаем коммит и выполним **git push**, чтобы загрузить наши изменения в удаленный репозиторий.

4. Теперь GitHub подсказывает нам, что наша ветка опережает ветку исходного репозитория на один коммит и предлагает сделать пулл-реквест.



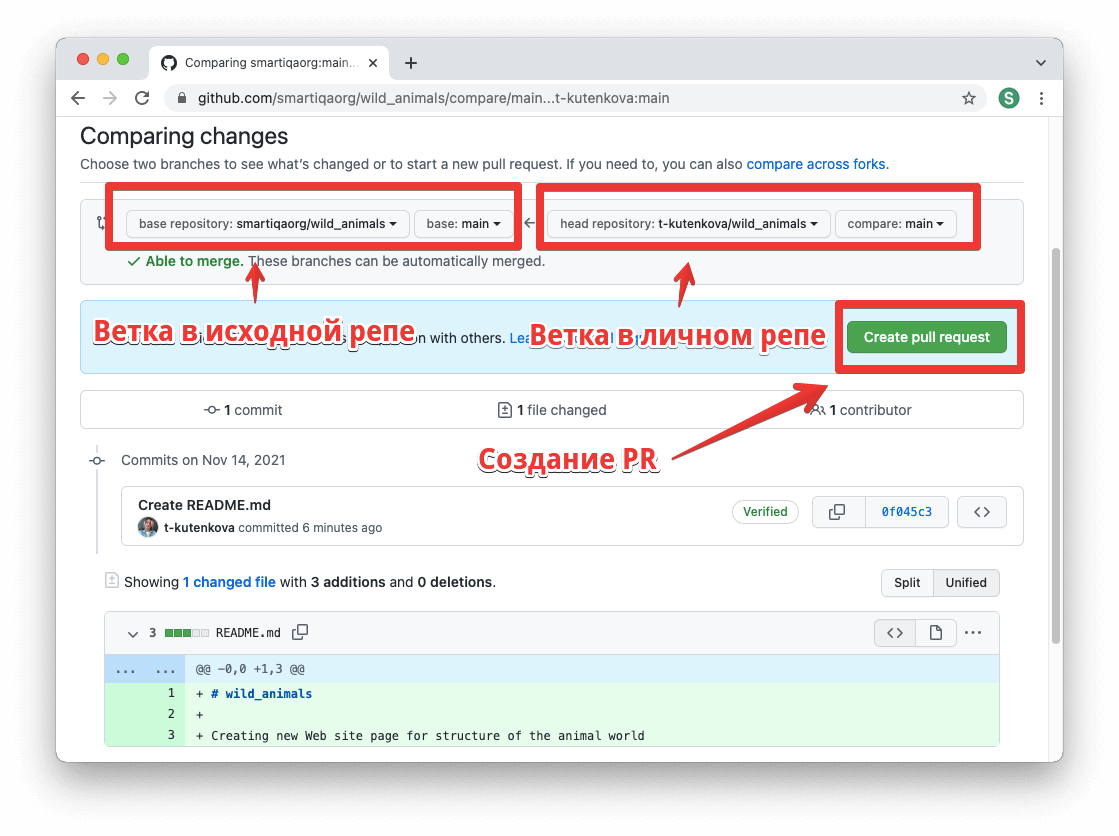
Подсказка GitHub

5. Нажимаем на кнопку **Compare** на подсказке GitHub, либо переходим на вкладку **Pull Requests** и нажимаем **New pull request**.



Создание пулл-реквеста

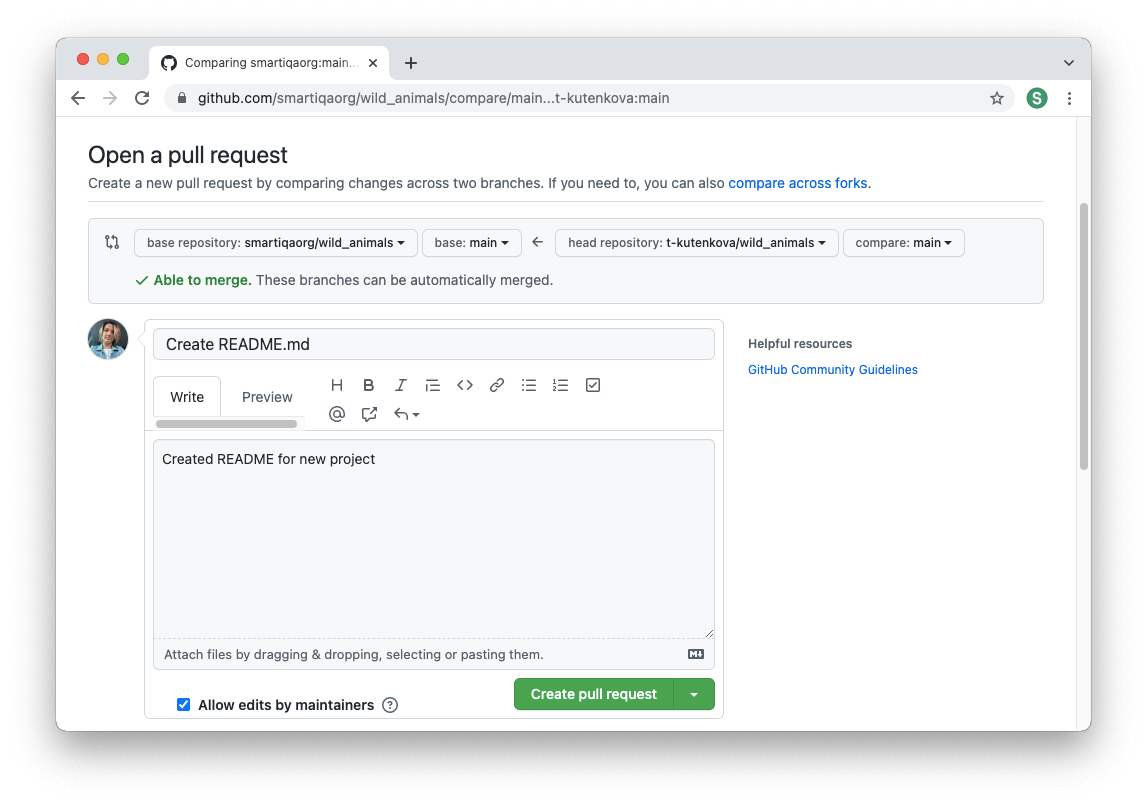
6. Перед нами откроется страница создания пулл-реквеста.



Страница пулл-реквеста

Здесь мы можем просмотреть внесенные изменения и выбрать две ветки: одну в исходном репозитории, на нее будут залиты наши изменения, вторую – в нашем репозитории, с нее будут скачаны изменения. Как только мы выбрали ветки и убедились, что не внесли никаких лишних изменений, нажимаем кнопку **Create pull request**.

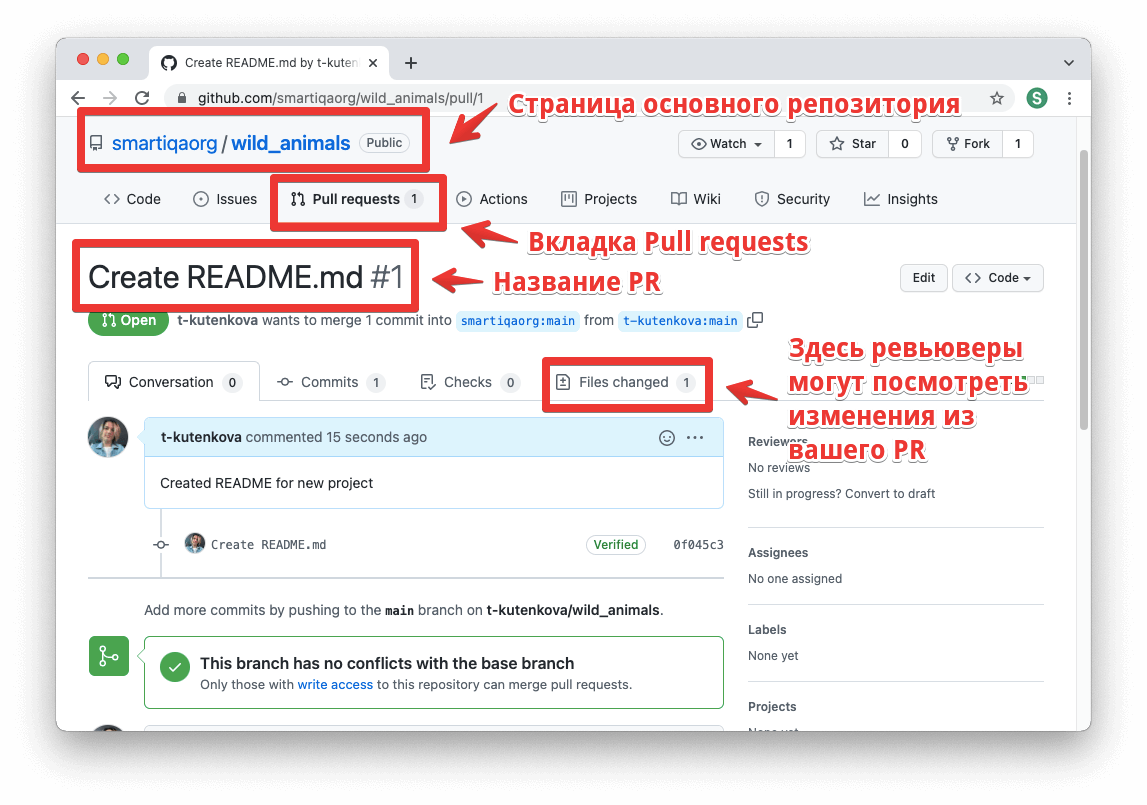
7. Теперь мы попадаем на страницу описания наших изменений.



Описание пулл-реквеста

Здесь необходимо описать, что за изменения вы внесли и почему они были необходимы. Сообщение, которое оставили мы, видно на картинке. Оно отражает суть и необходимость внесенных изменений. Как только мы закончили с описанием, можно нажимать кнопку **Create pull request**.

8. Теперь мы попадаем на страницу уже созданного пулл-реквеста в изначальном репозитоии. В нашем случае он выглядит так.



Страница созданного пулл-реквеста

Именно так будет выглядеть наш пулл-реквест и для владельца репозитория. На этой странице он сможет писать комментарии, указывая на ошибки или задавая вопросы. После того, как владелец репозитория просмотрит наши изменения и убедится, что они не имеют вредоносный характер, он сможет принять наш пулл-реквест. Тогда все изменения, добавленные в этот пулл-реквест нами, будут залиты в исходный репозиторий.

**Подытожим**

Таким образом, вы можете вносить свой вклад в абсолютно любые приложения с открытым программным кодом. Даже репозиторий самого Git хранится на GitHub, и вы в любой момент можете сделать форк и придумать новую команду (правда не факт, что ответственный примет ваш пулл-реквест). Вы можете внести свой вклад в разработку ядра Linux, в популярный редактор **Visual Studio Code**, в ядро криптовалюты **Bitcoin**, в языки программирования **Python**, **Go**, **Ruby**– все эти проекты имеют открытые репозитории на GitHub.