

[Г л а в н а я](#)

[И н с т р у к ц и и](#)

К а к п о л ь з о в а т ь с я GitHub

# К а к п о л ь з о в а т ь с я GitHub

[И н с т р у к ц и и](#) А в г у с т 25, 2017 [5admin](#)

Github — это очень известная платформа для хранения, распространения и управления исходным кодом открытых проектов. Этот сервис используют множество разработчиков по всему миру, среди которых есть и крупные компании, такие как Microsoft, RedHat и множество других, также сотни разработчиков многих популярных проектов.

Платформа предоставляет возможности не только по просмотру кода и его распространения, но также историю версий, инструменты совместной разработки, средства для предоставления документации, выпуска релизов и обратной связи. И самое интересное, что вы можете размещать на Github как открытые, так и приватные проекты. В этой статье мы рассмотрим как пользоваться Github для размещения своего проекта. Так сказать, github для начинающих.

Содержание статьи:

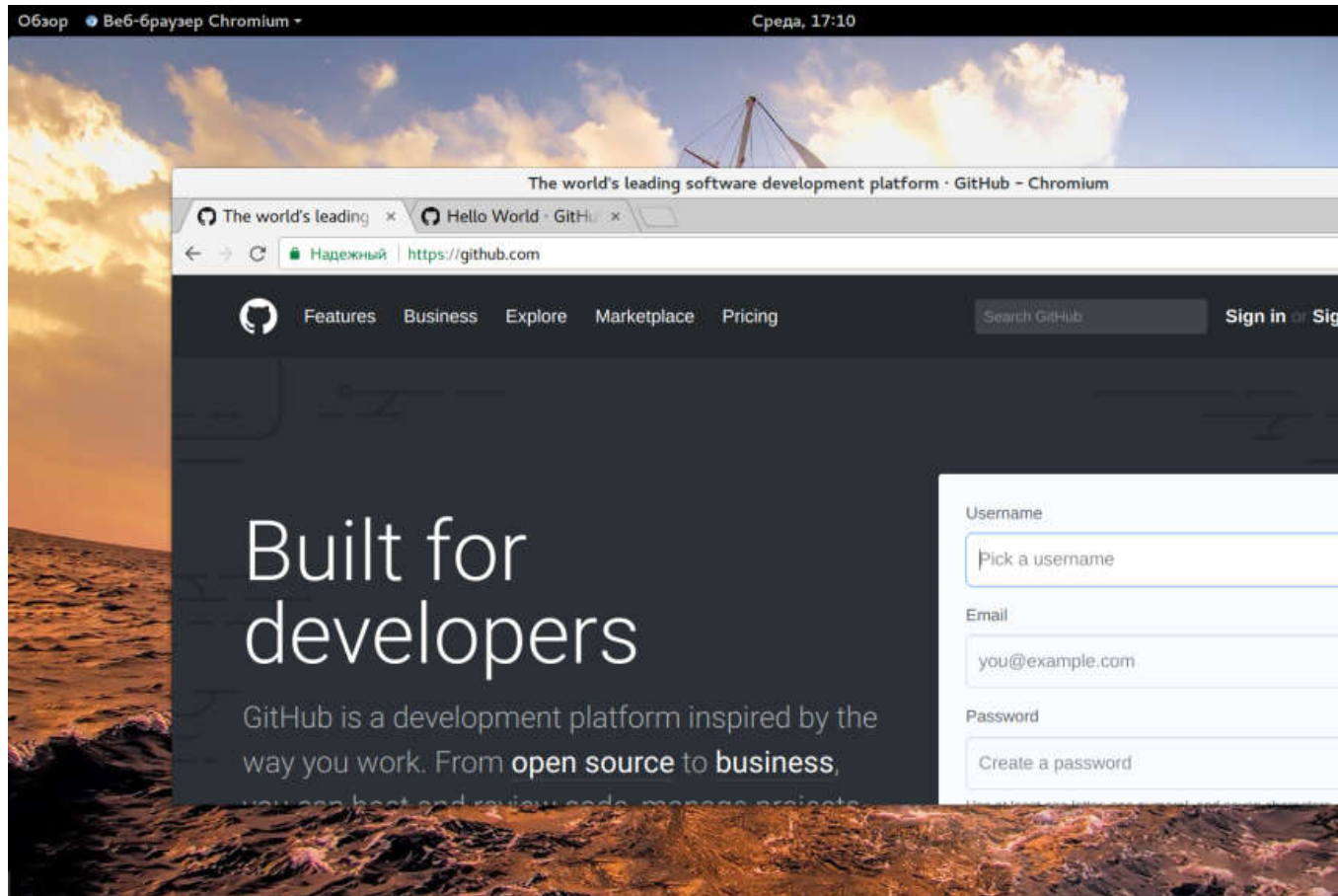
- [Как пользоваться GitHub](#)
  - [1. Создание аккаунта](#)
  - [2. Создание репозитория](#)
  - [3. Добавление веток](#)
  - [4. Изменение файлов и коммиты](#)
  - [5. Создание запросов слияния \(Pull Request\)](#)
  - [6. Просмотр и одобрение запросов на слияние](#)
  - [7. Отчеты об ошибках](#)
  - [8. Релизы](#)
- [Выводы](#)

## *К а к п о л ь з о в а т ь с я GitHub*

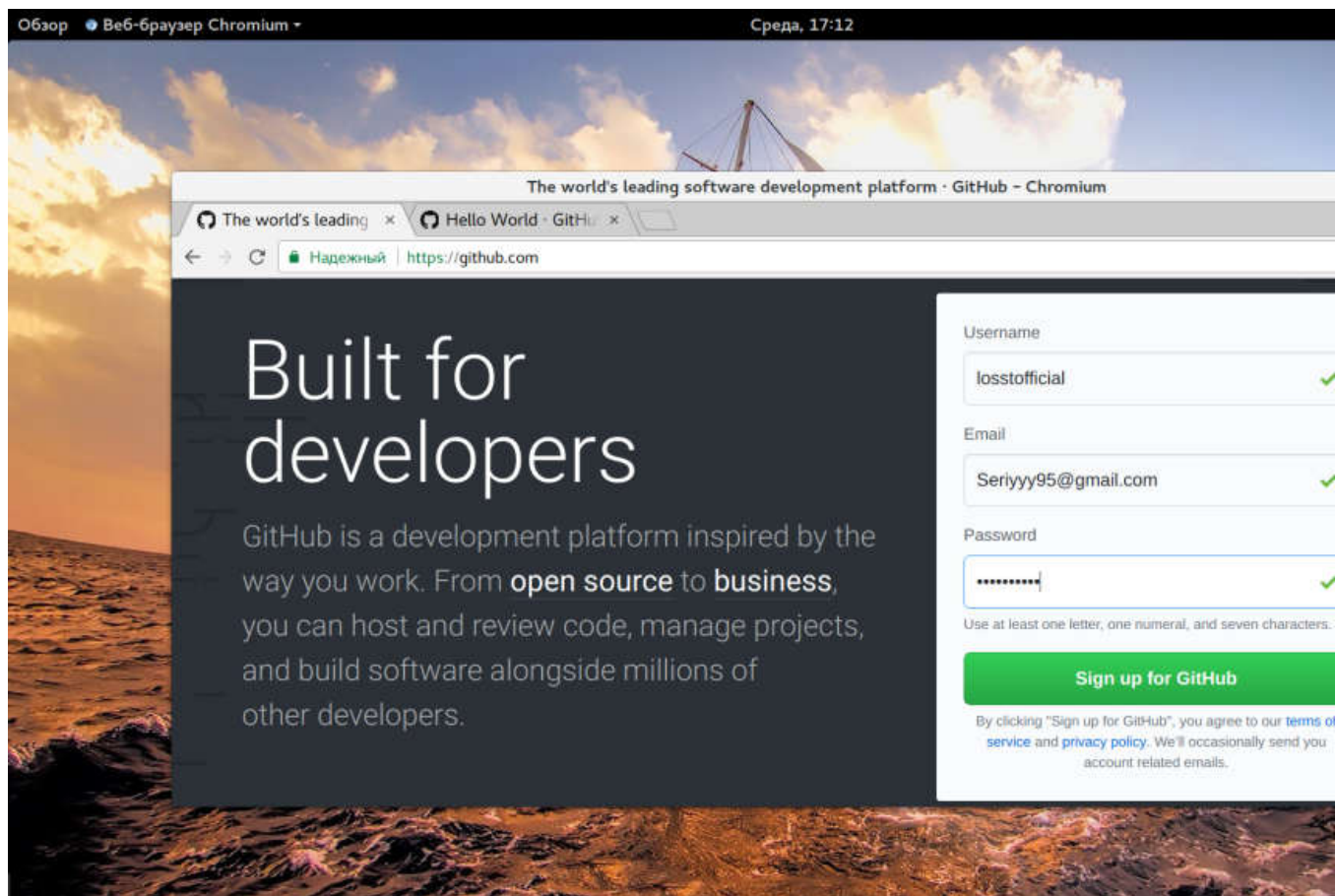
Итак, допустим, у вас есть свой проект и вы хотите разместить его код на Github в открытом доступе чтобы другие пользователи могли его посмотреть и участвовать в разработке. Первое что вам нужно сделать — создать аккаунт.

## 1. Создание аккаунта

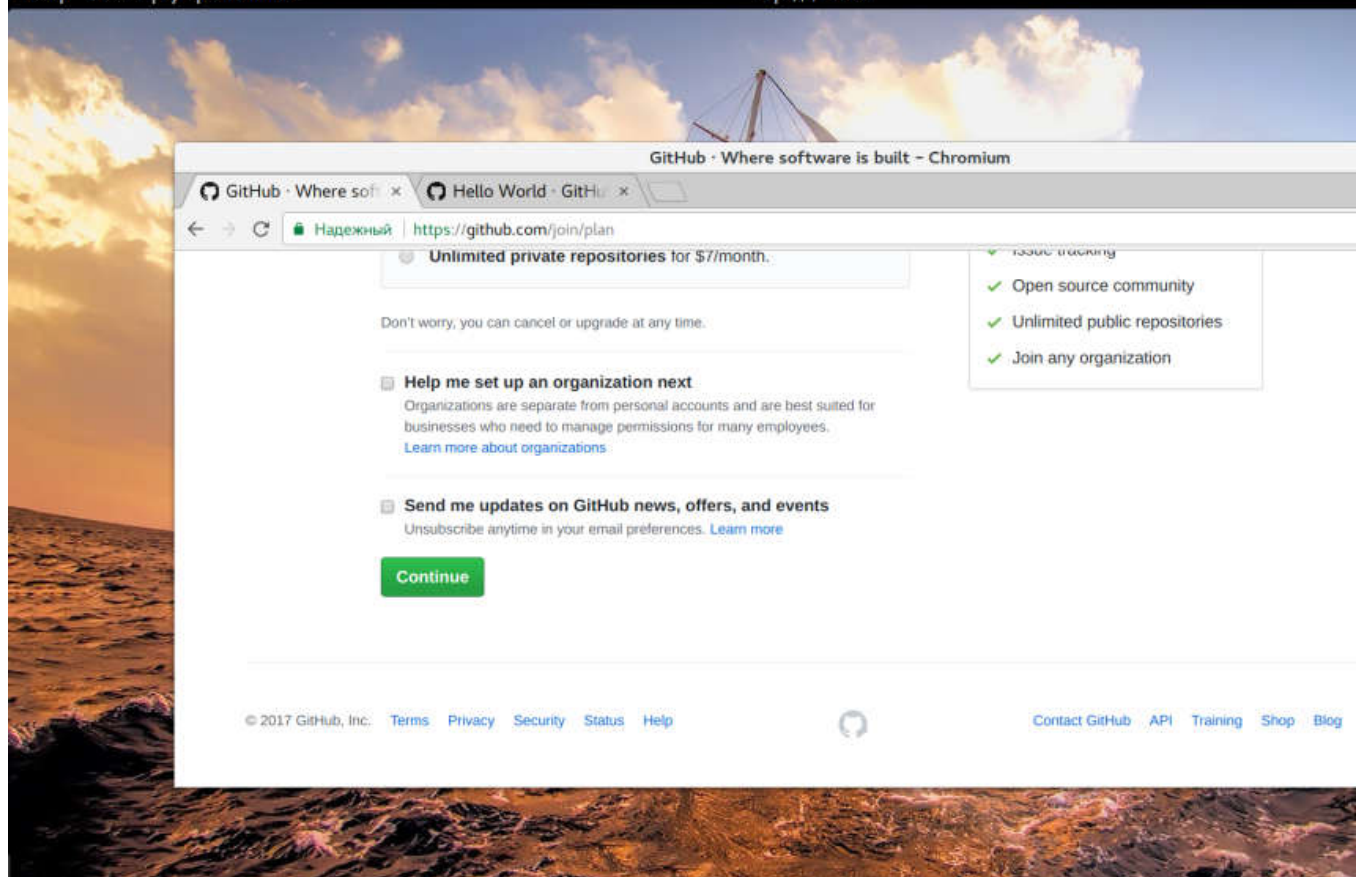
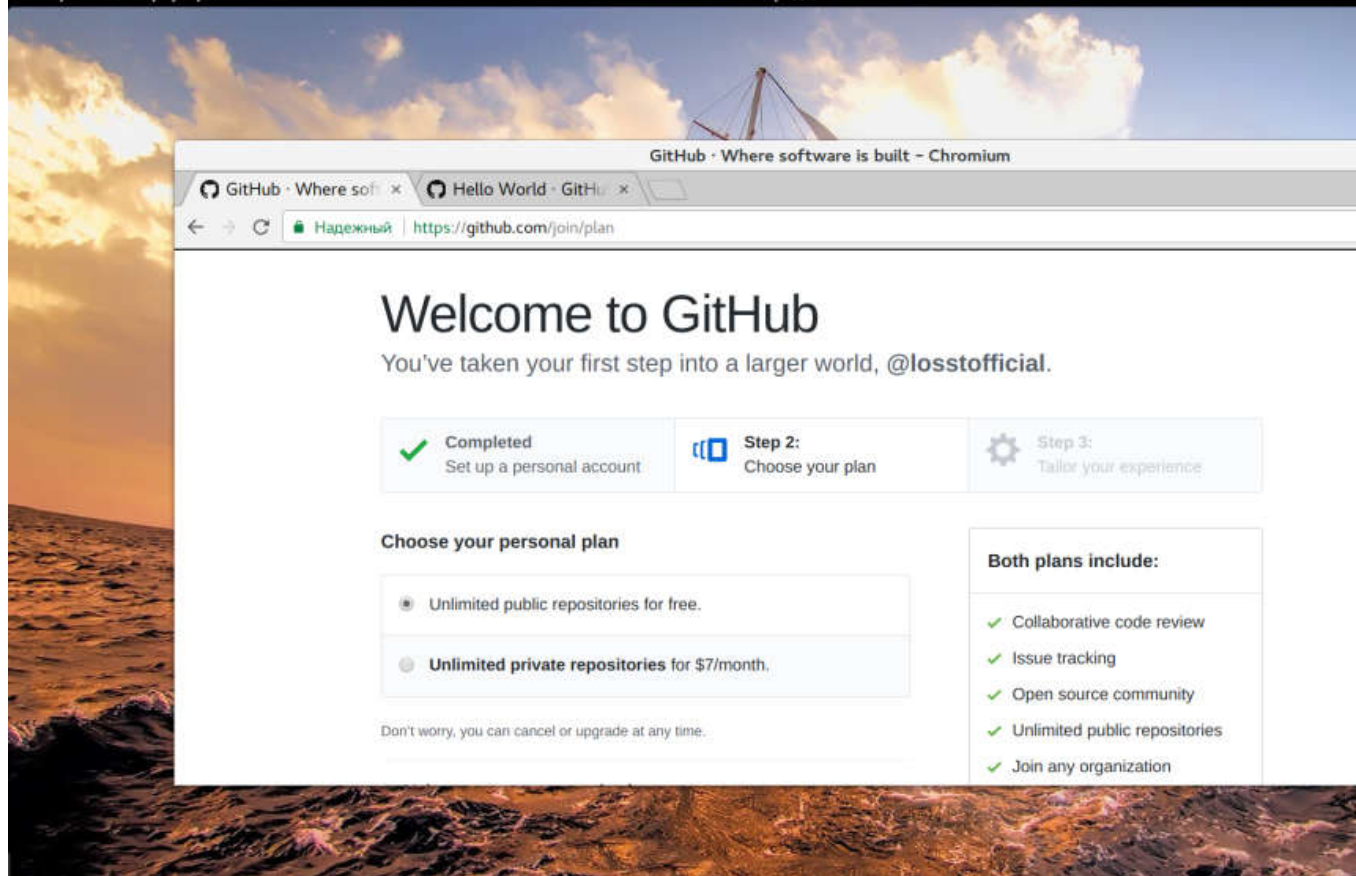
Чтобы создать новый аккаунт на сайте откройте главную страницу [GitHub](https://github.com) и тут же сразу вы можете ввести данные для новой учетной записи. Вам нужно указать имя пользователя, Email и пароль:



Когда завершите ввод, нажмите кнопку «Sign Up Free»:



На следующем шаге вам нужно выбрать тип репозитория. Публичные репозитории бесплатны, но если вы хотите создать частный репозиторий, код из которого будет доступен только вам, то придется платить \$7 в месяц.



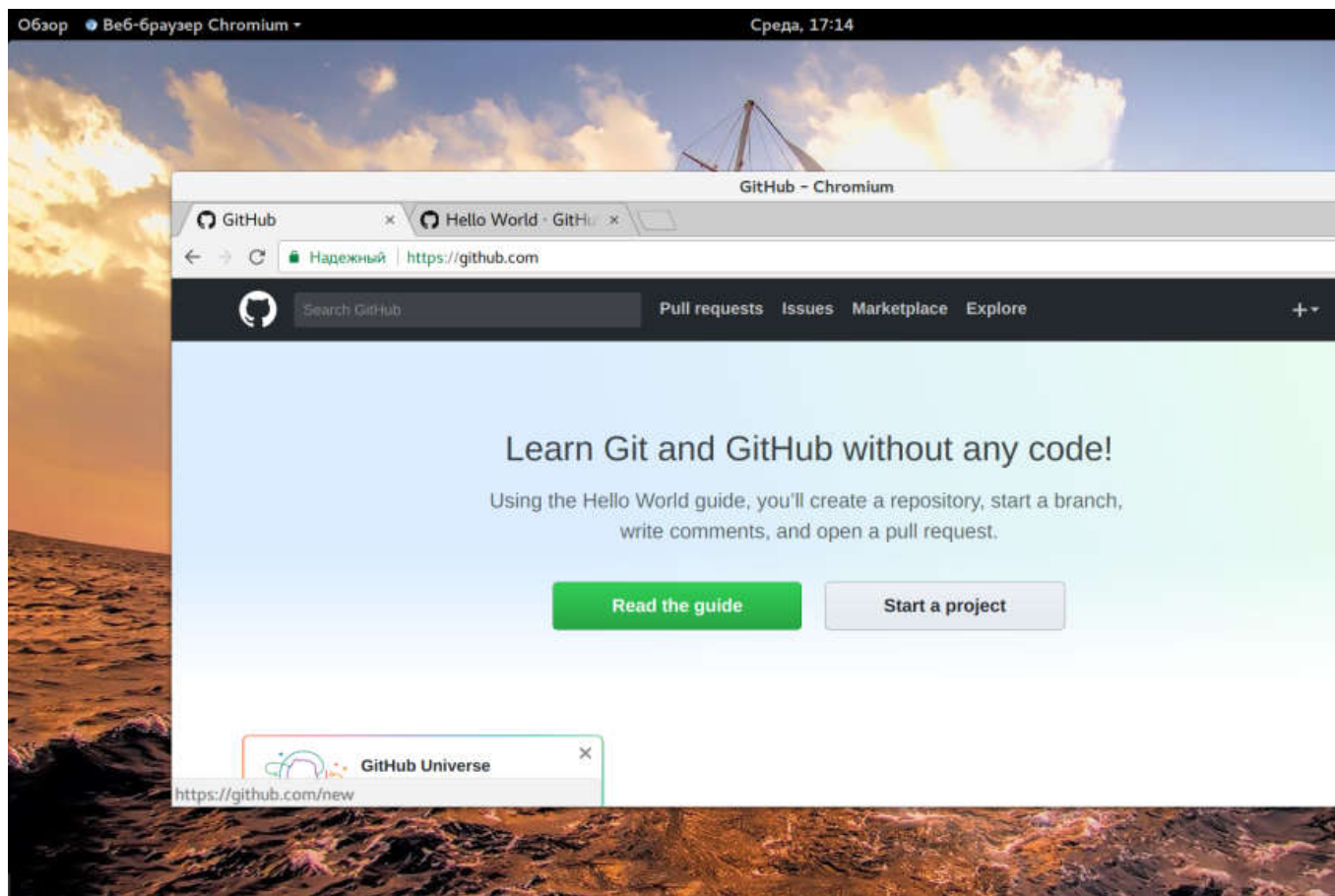


Аккаунт готов, и вы будете перенаправлены на страницу, где сможете создать свой первый проект. Но перед тем как вы сможете это сделать, нужно подтвердить свой Email адрес. Для этого откройте ваш почтовый ящик и перейдите по ссылке в письме от Github.

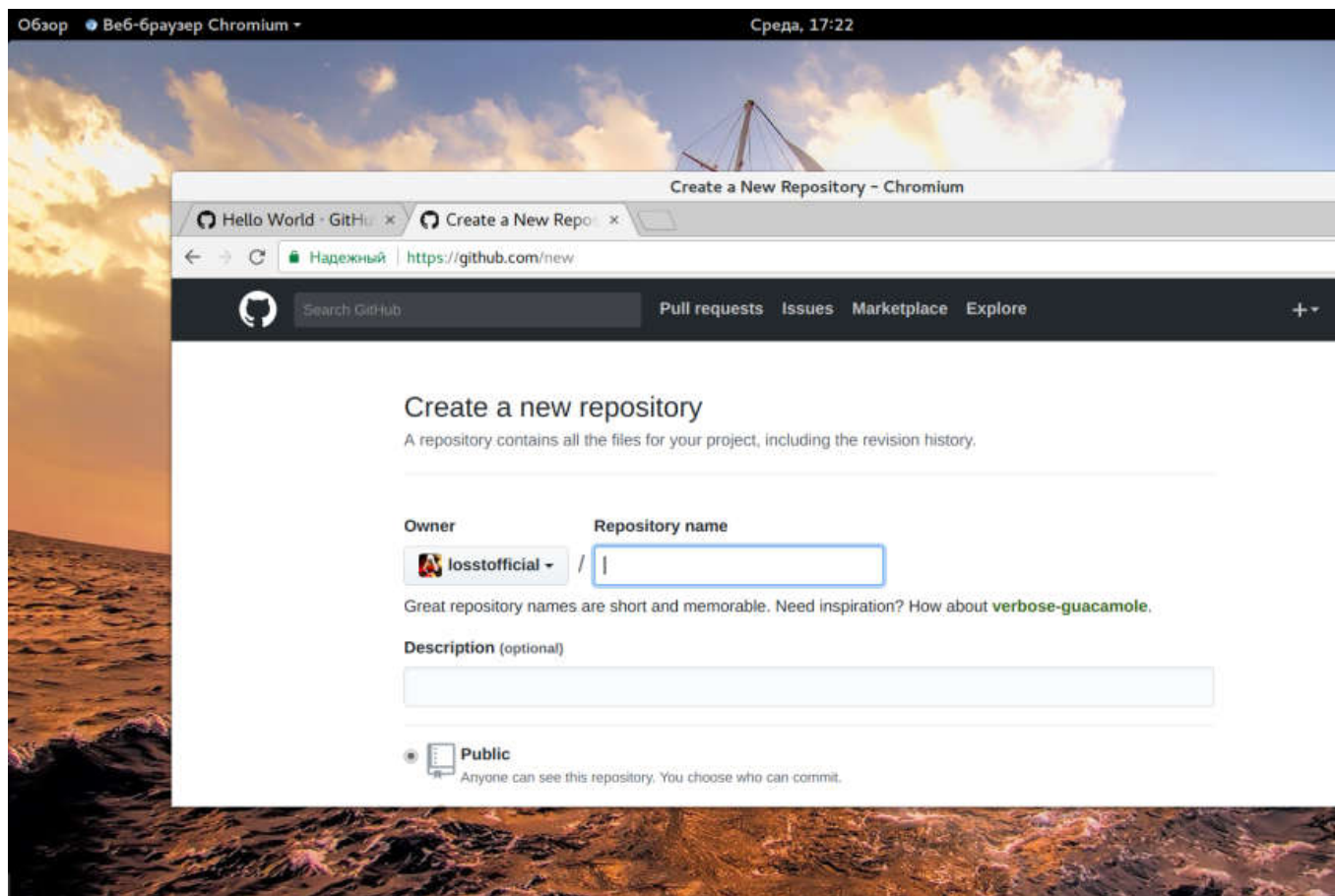
Никакая настройка github не нужна, достаточно лишь несколько кликов мышкой.

## 2. Создание репозитория

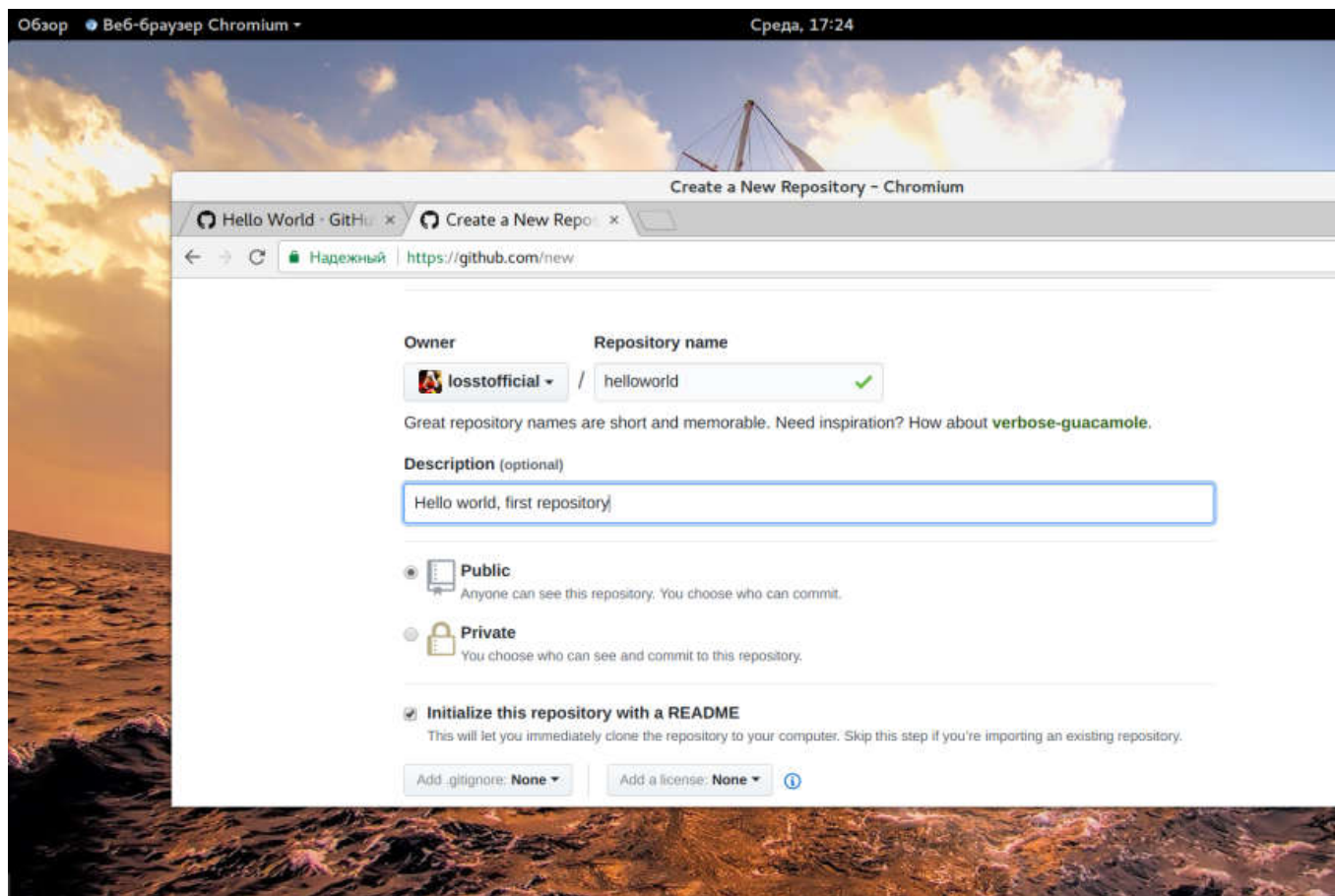
На открывшейся странице, это главная страница для авторизованных пользователей, нажмите кнопку «**Start a project**»:



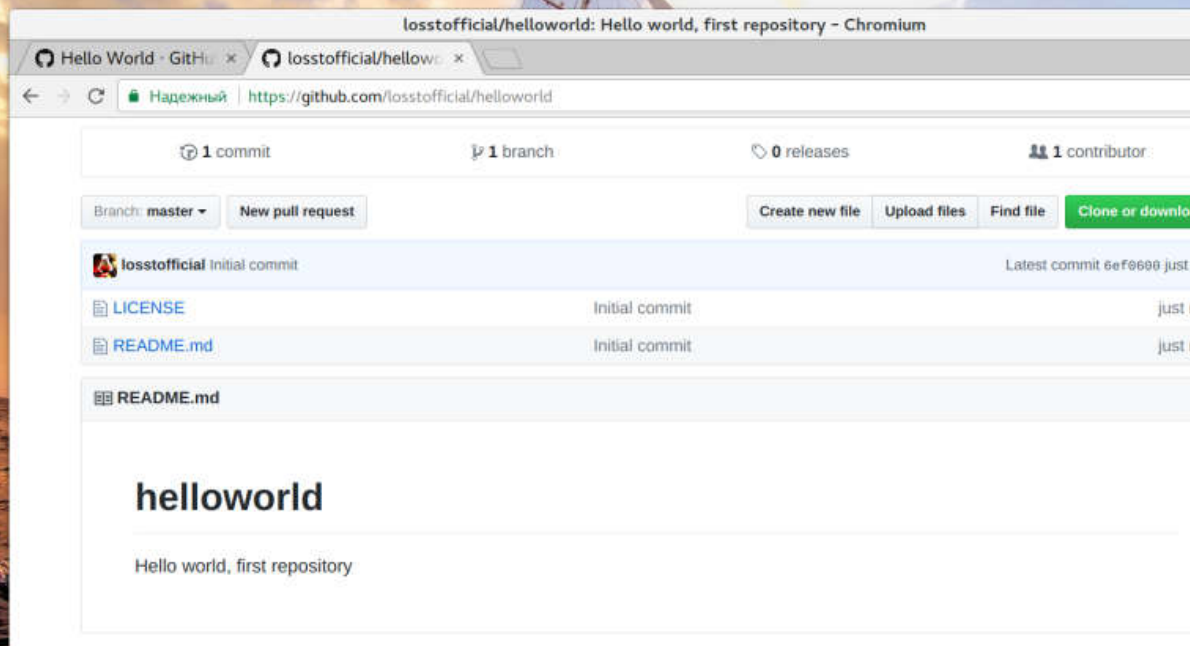
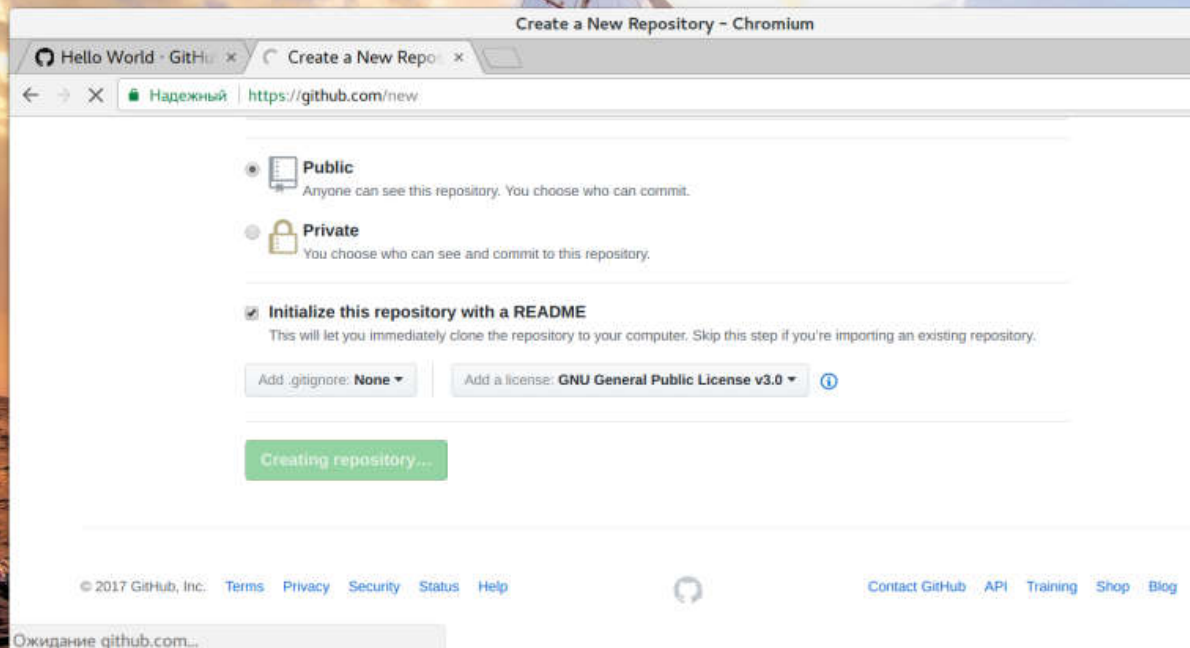
Дальше введите имя и описание будущего репозитория:



Вы можете сразу же инициализировать репозиторий, создав файл Readme, для этого нужно отметить галочку **«Initialize this repository with a README»** внизу страницы. Также можно выбрать лицензию:



Когда все будет готово, выберите «**Create project**», будет создан новый проект с файлом README, в котором находится описание и файлом лицензии.

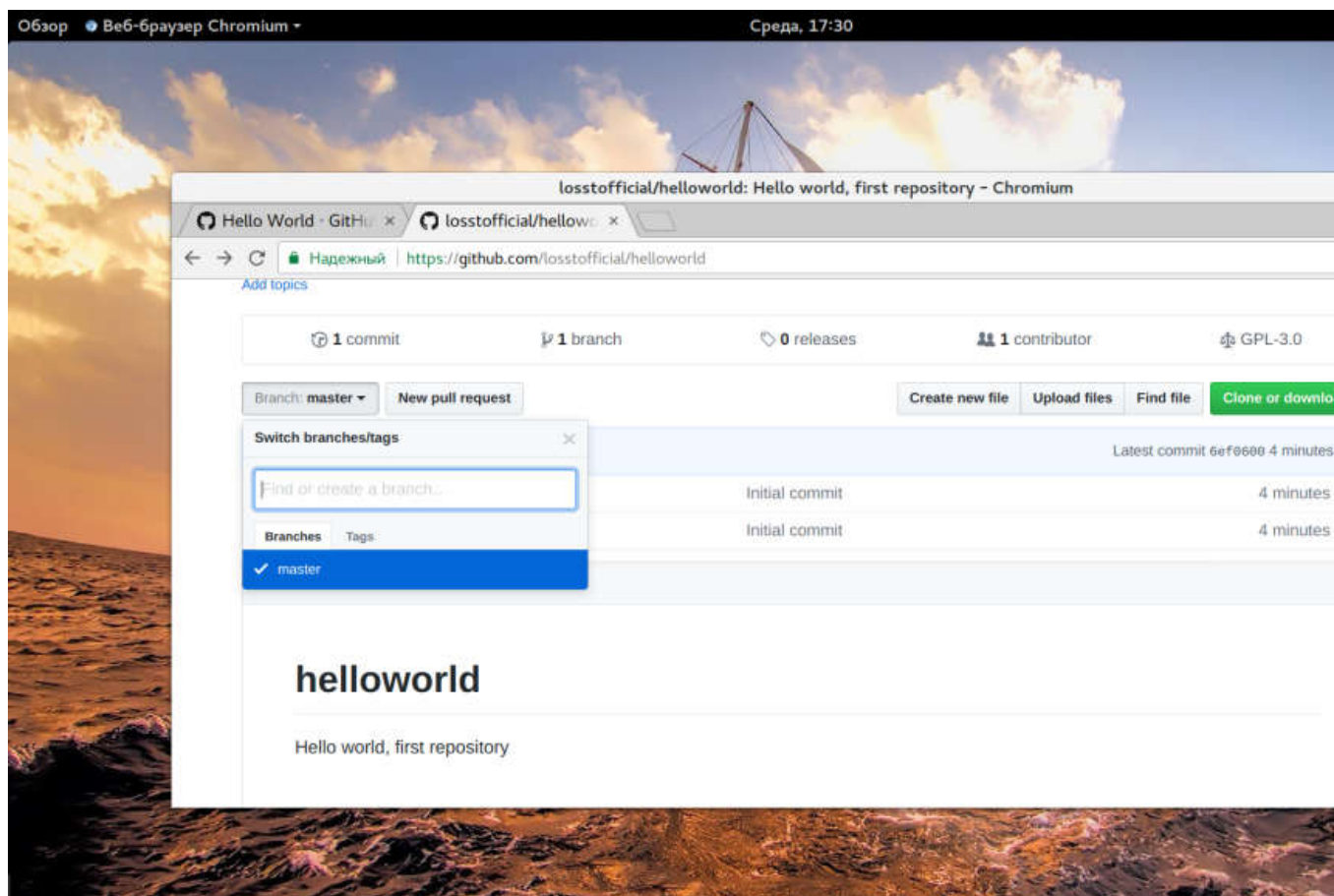




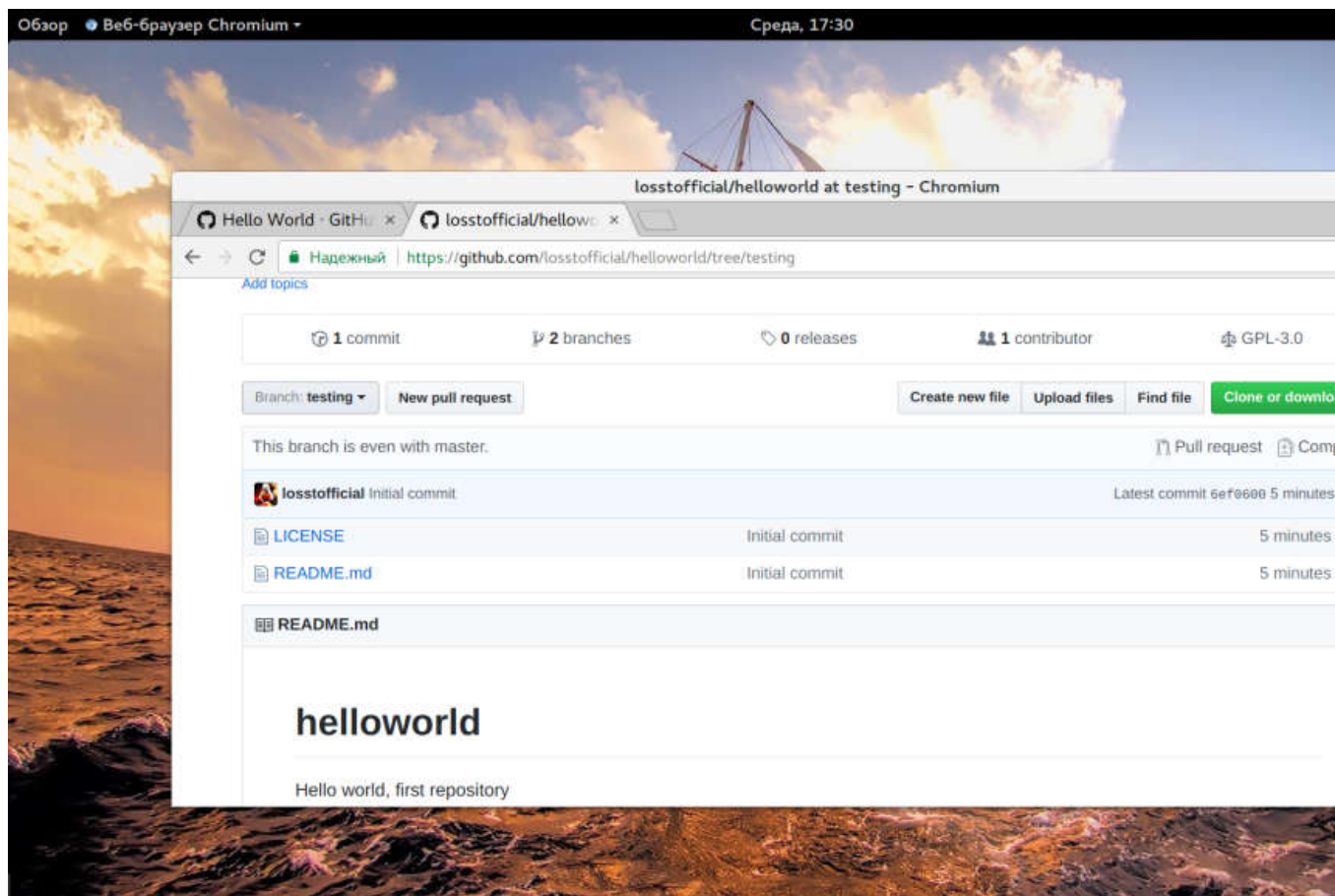
Дальше все самое интересное как работать с github.

### 3. Д о б а в л е н и е в е т о к

Ветки Github позволяют работать с несколькими версиями проекта одновременно. По умолчанию при создании репозитория создается ветка master, это основная рабочая ветка. Можно создать дополнительные ветки, например, для того, чтобы тестировать программное обеспечение перед тем, как оно будет опубликовано в ветке master. Таким образом, можно одновременно разрабатывать продукт и предоставлять пользователям стабильную версию. Также можно создавать отдельные ветки для версии программы для разных систем.



Текущая ветка обозначена в верхнем левом углу после слова «**Branch**». Чтобы создать новую ветку просто разверните этот список и начните набирать ее имя:

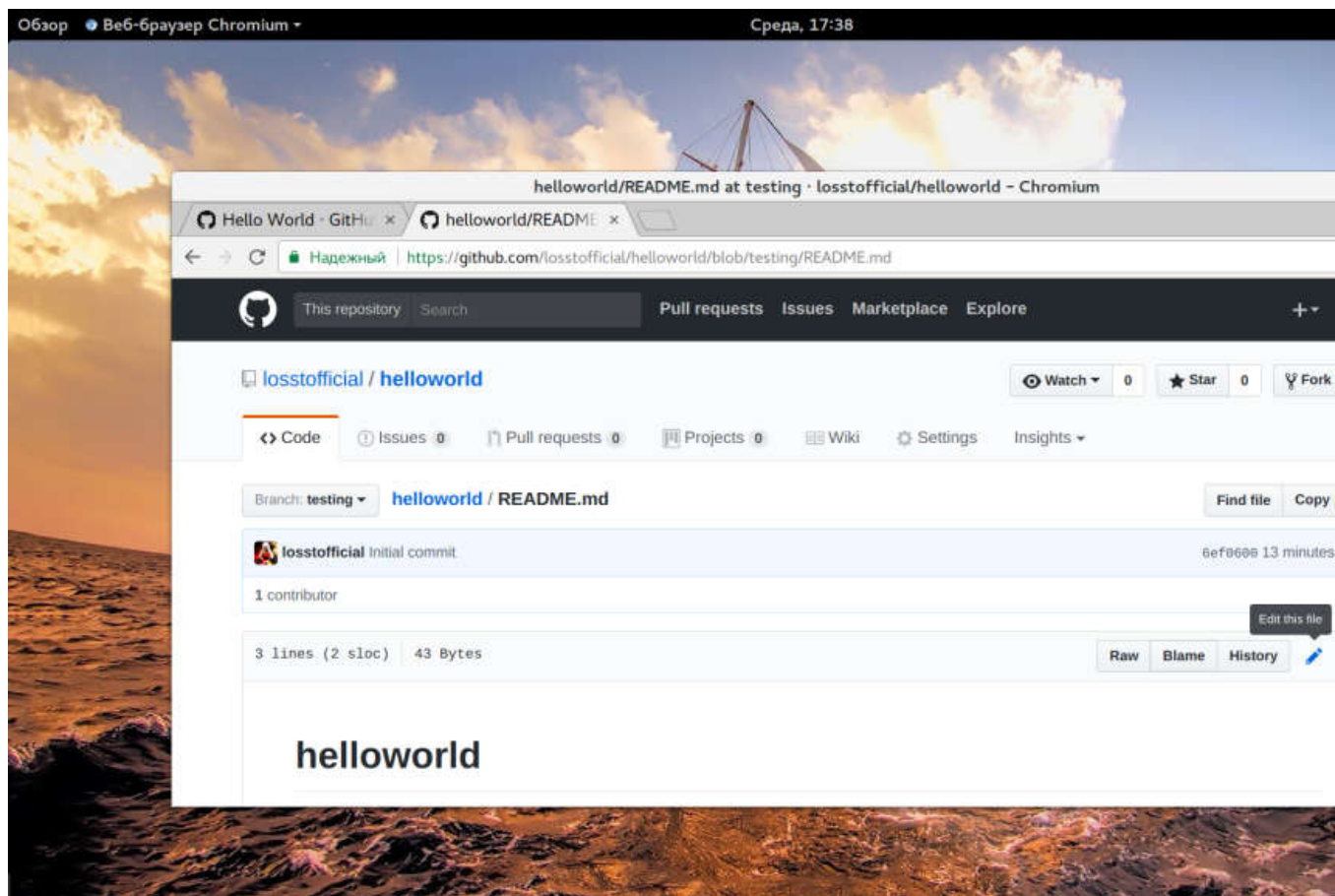


Сайт сам предложит вам создать новую ветку, выберите **«Create branch»**.

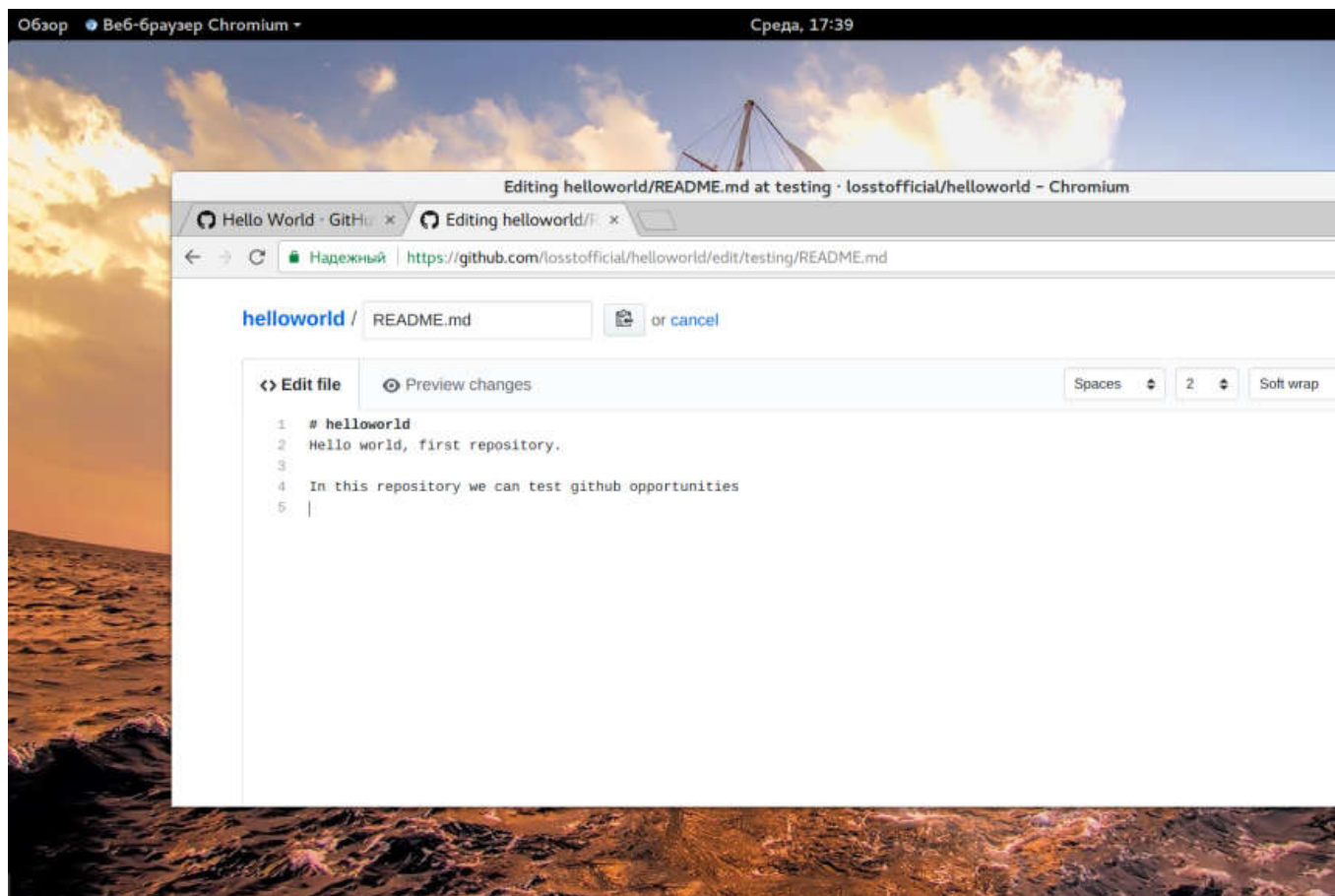
Сразу же после создания вы будете работать с только что созданной веткой.

#### 4. Изменение файлов и коммиты

Любые изменения файлов на Github делаются с помощью коммитов. Коммит выполняется путем внесения самих исправлений и описания этих исправлений. Это необходимо для того, чтобы вы знали что и когда вы меняли, а также позволяет легко отслеживать работу команды. Слово коммит можно перевести как «фиксировать». То есть мы можем внести изменения в несколько файлов, а затем их зафиксировать. Давайте для примера изменим файл README. Для этого найдите в правой стороне панели кнопку с кисточкой и нажмите на нее:

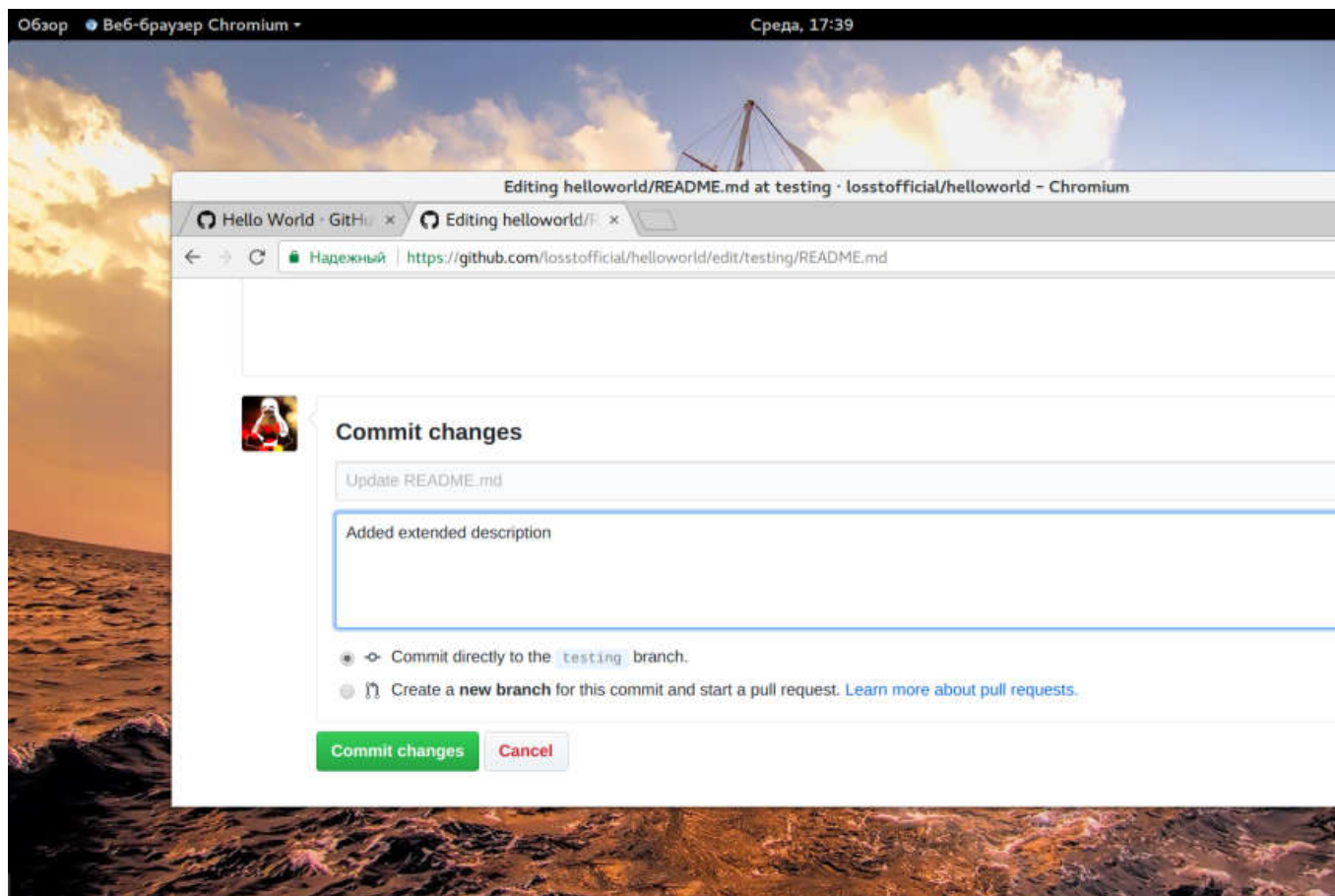


Откроется текстовый редактор, где вы можете ввести нужные вам исправления:

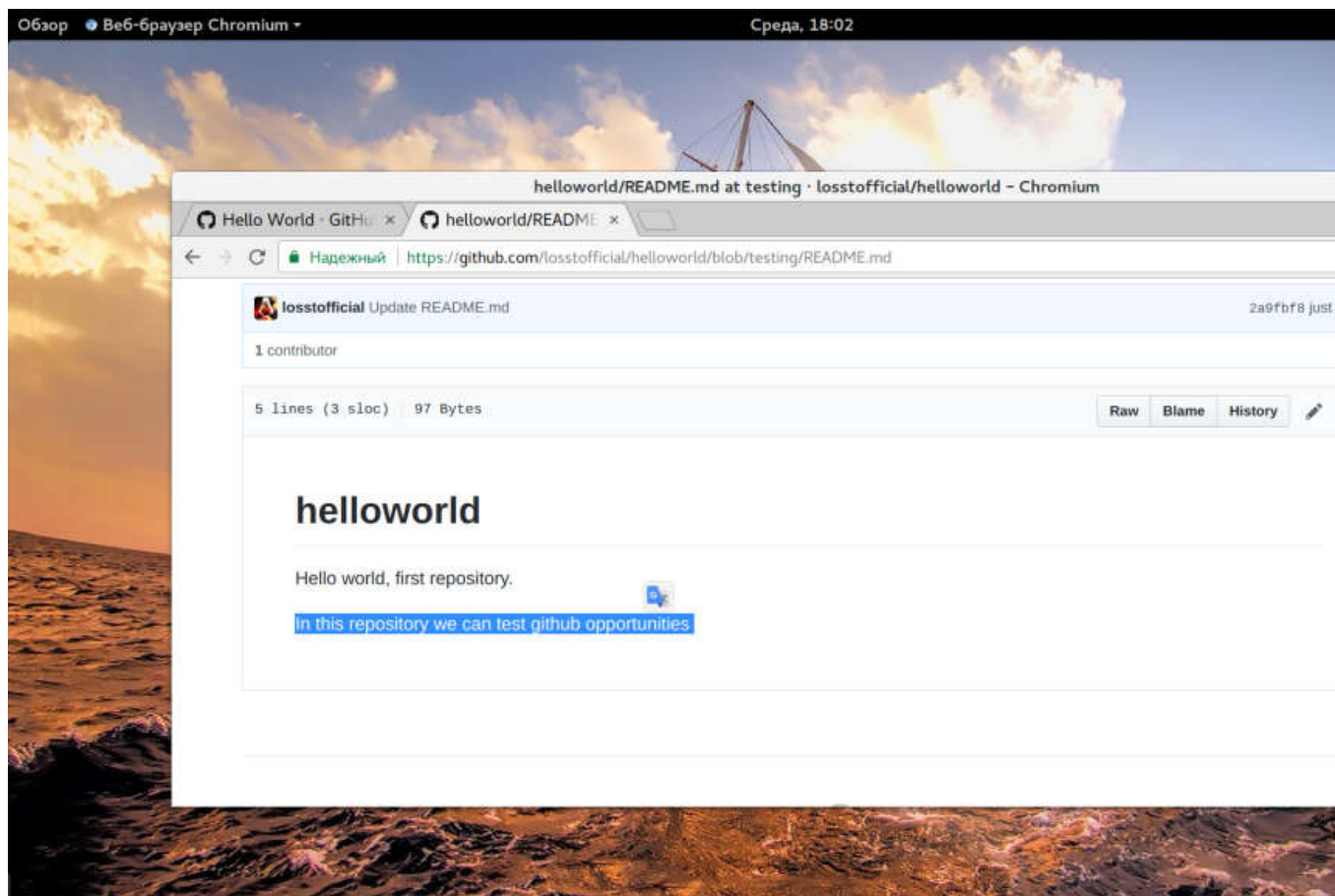


После того как вы сделаете все что вам нужно, необходимо заполнить поле «**Commit**» внизу страницы. Кратко опишите что было изменено, а затем нажмите кнопку «**Commit changes**»:





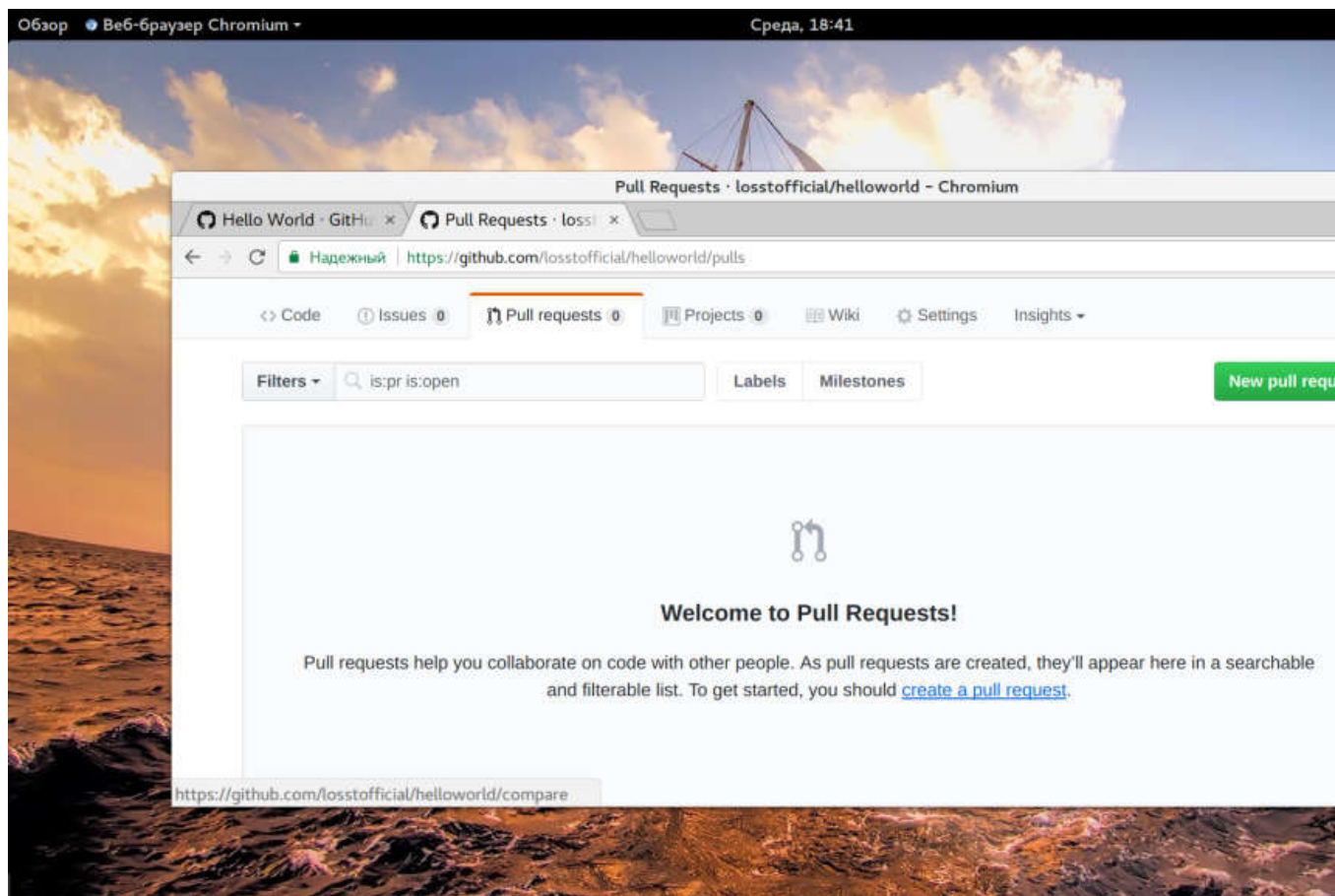
Эти изменения будут внесены в текущую ветку проекта, поскольку мы сейчас работаем с `testing`, то и изменения будут отправлены именно туда.



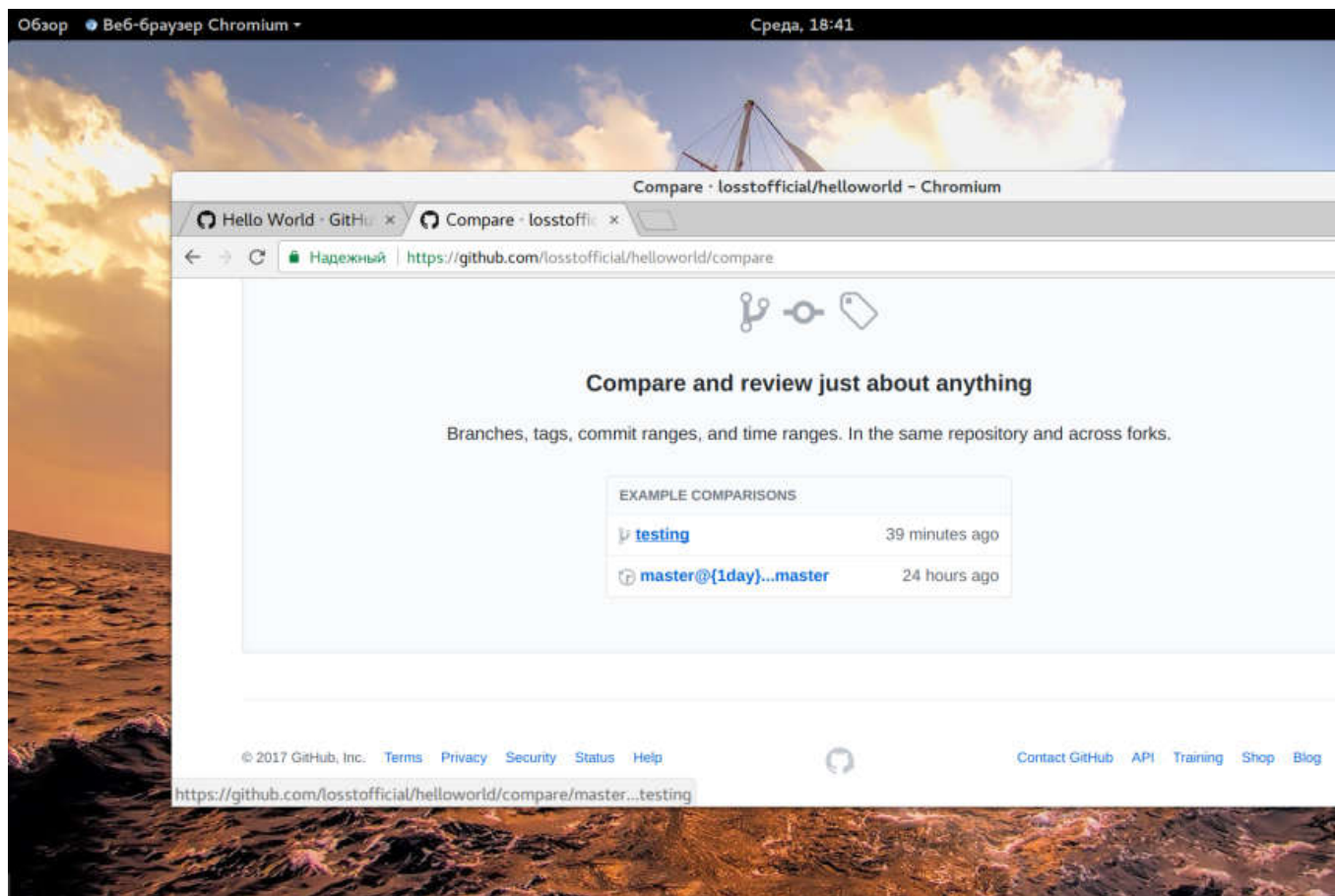
## 5. Создание запросов слияния (Pull Request)

GitHub для начинающих может показаться очень сложным именно из-за таких возможностей, но это очень удобно если разобраться. Запрос слияния или Pull Request — это возможность, благодаря которой любой разработчик может попросить другого, например, создателя репозитория просмотреть его код и добавить его в основной проект или ветку. Инструмент работы с запросами слияния использует инструмент сравнения diff, поэтому вы можете увидеть все изменения, они будут подчеркнуты другим цветом. Pull Request можно создать сразу же после создания коммита. Давайте отправим Pull Request из нашей testing ветки в основную. Сначала откройте вкладку «Pull Request».

Здесь нажмите кнопку «Create Pull Request»:

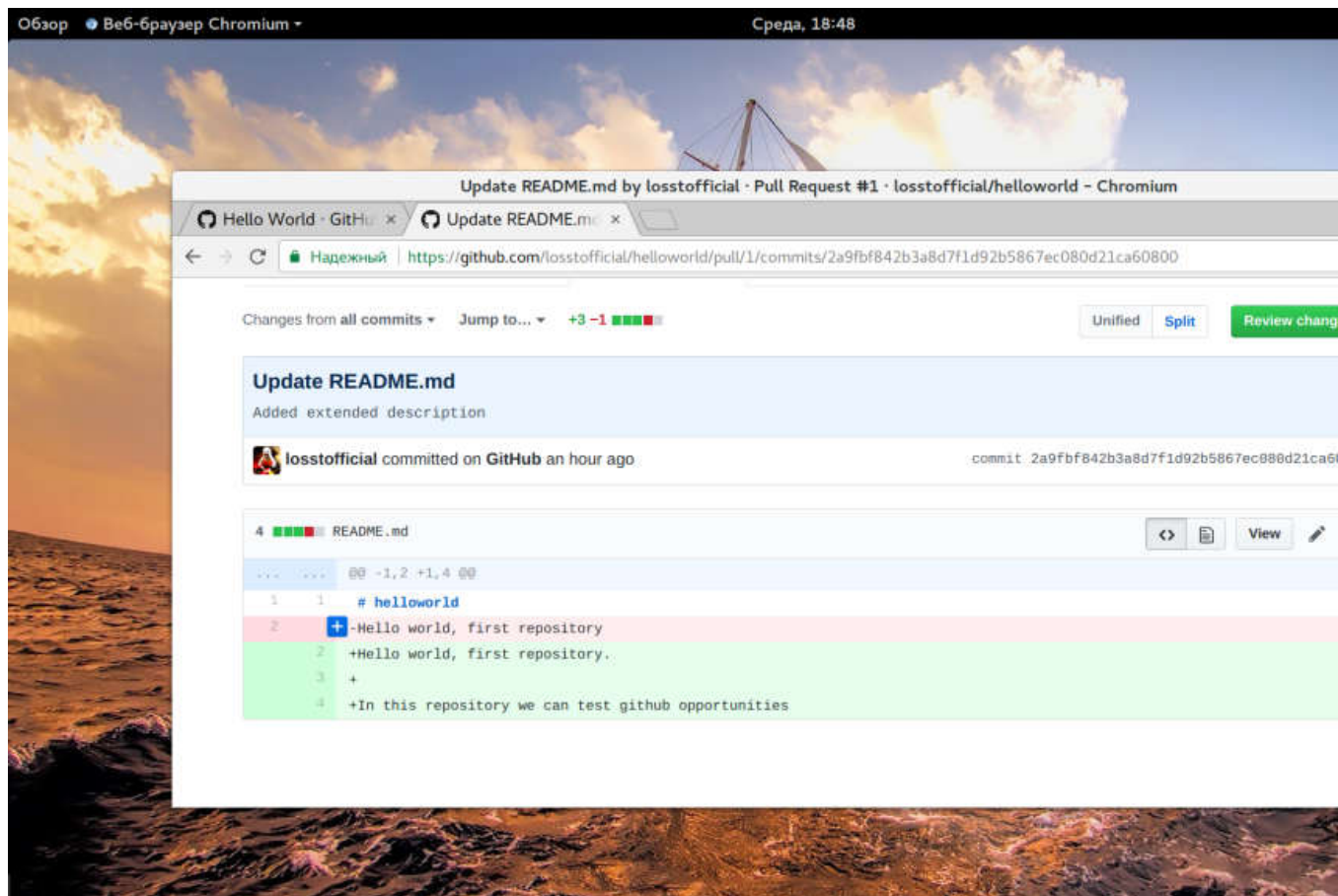
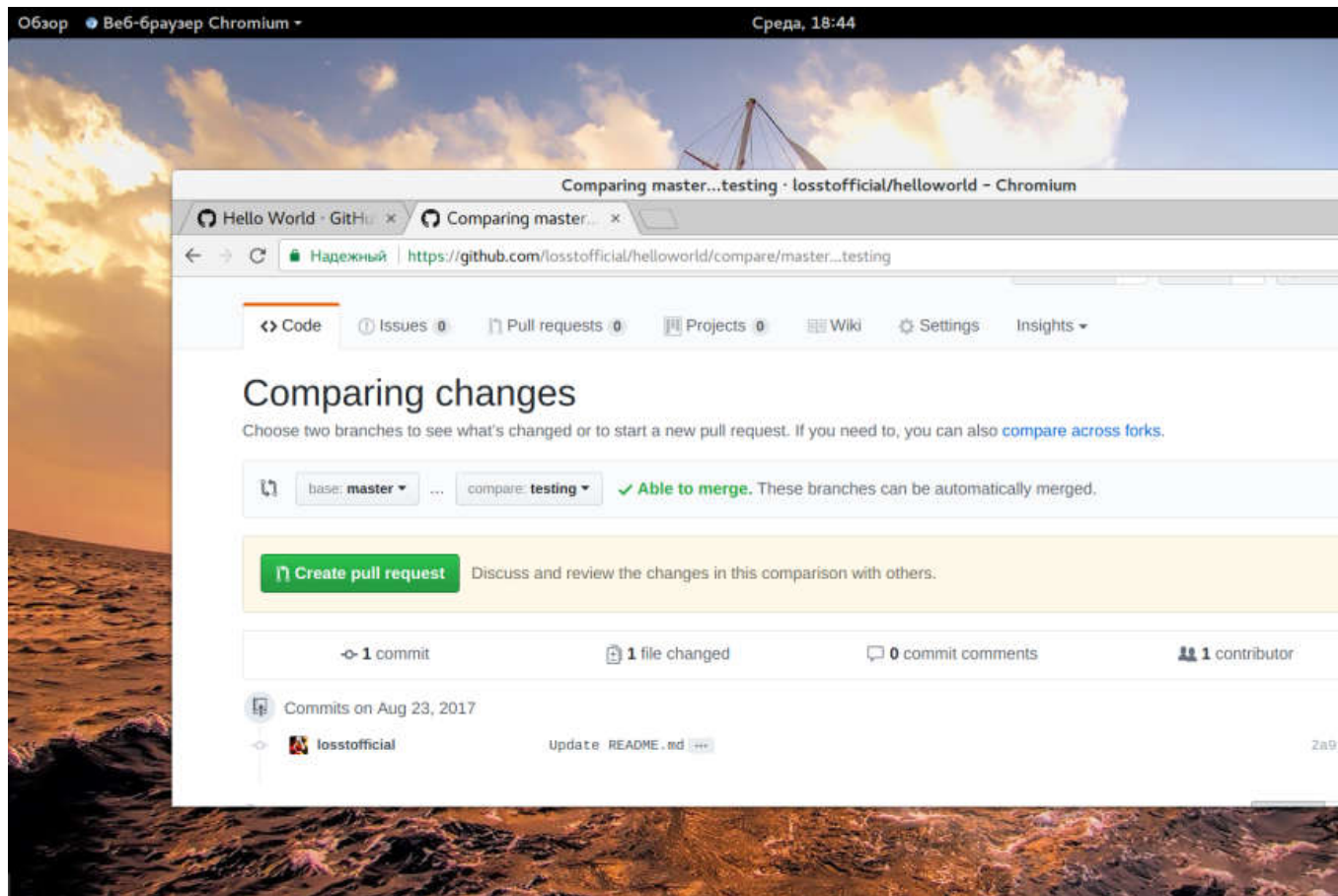


Дальше вам нужно будет выбрать ветку, которую нужно слить с основной, в нашем случае «**testing**».

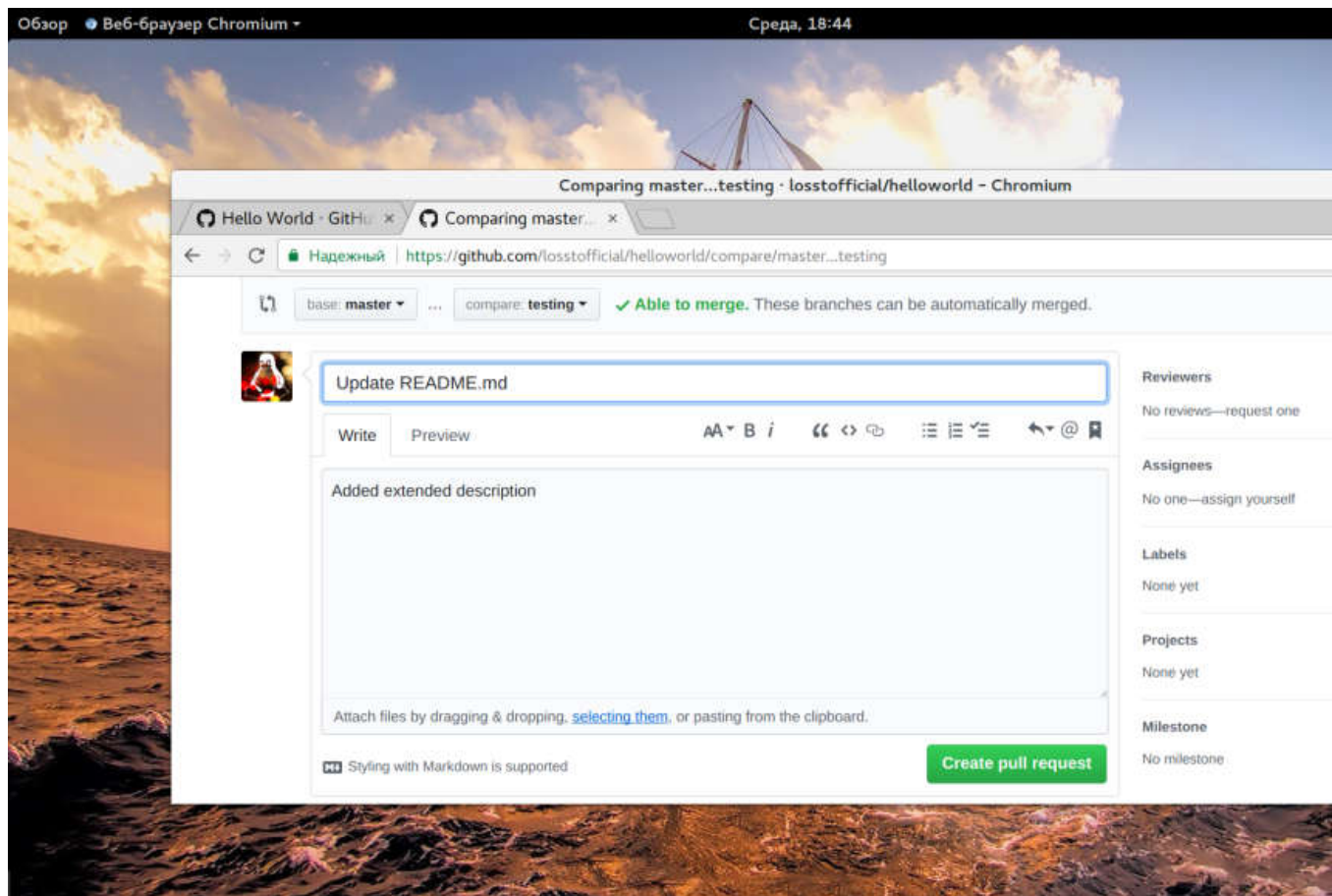


В этом окне вы можете просмотреть все изменения, сейчас мы видим, что была добавлена строка:



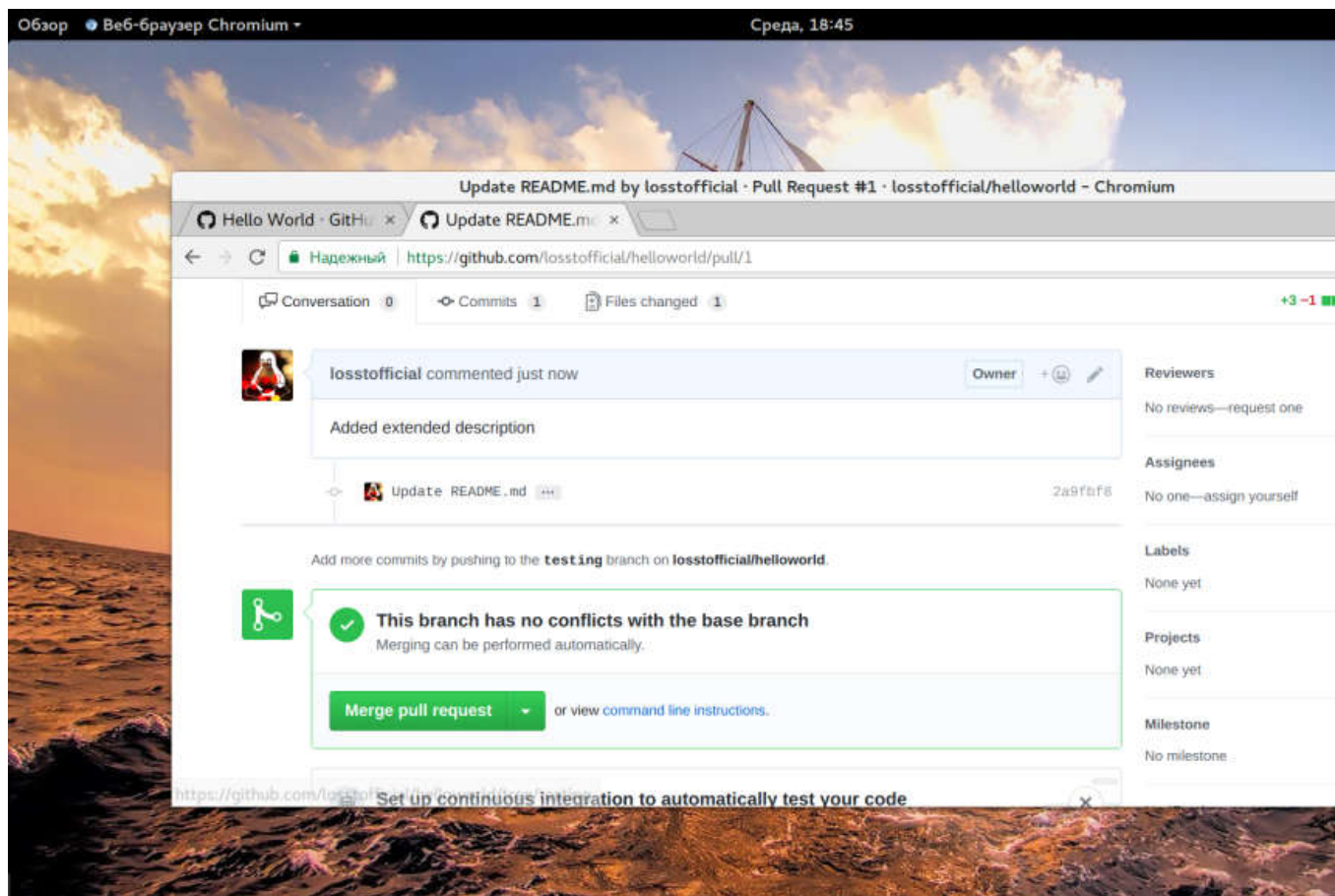


Дальше нажмите зеленую кнопку «**Create Pull Request**» и введите описание, как и для коммита:

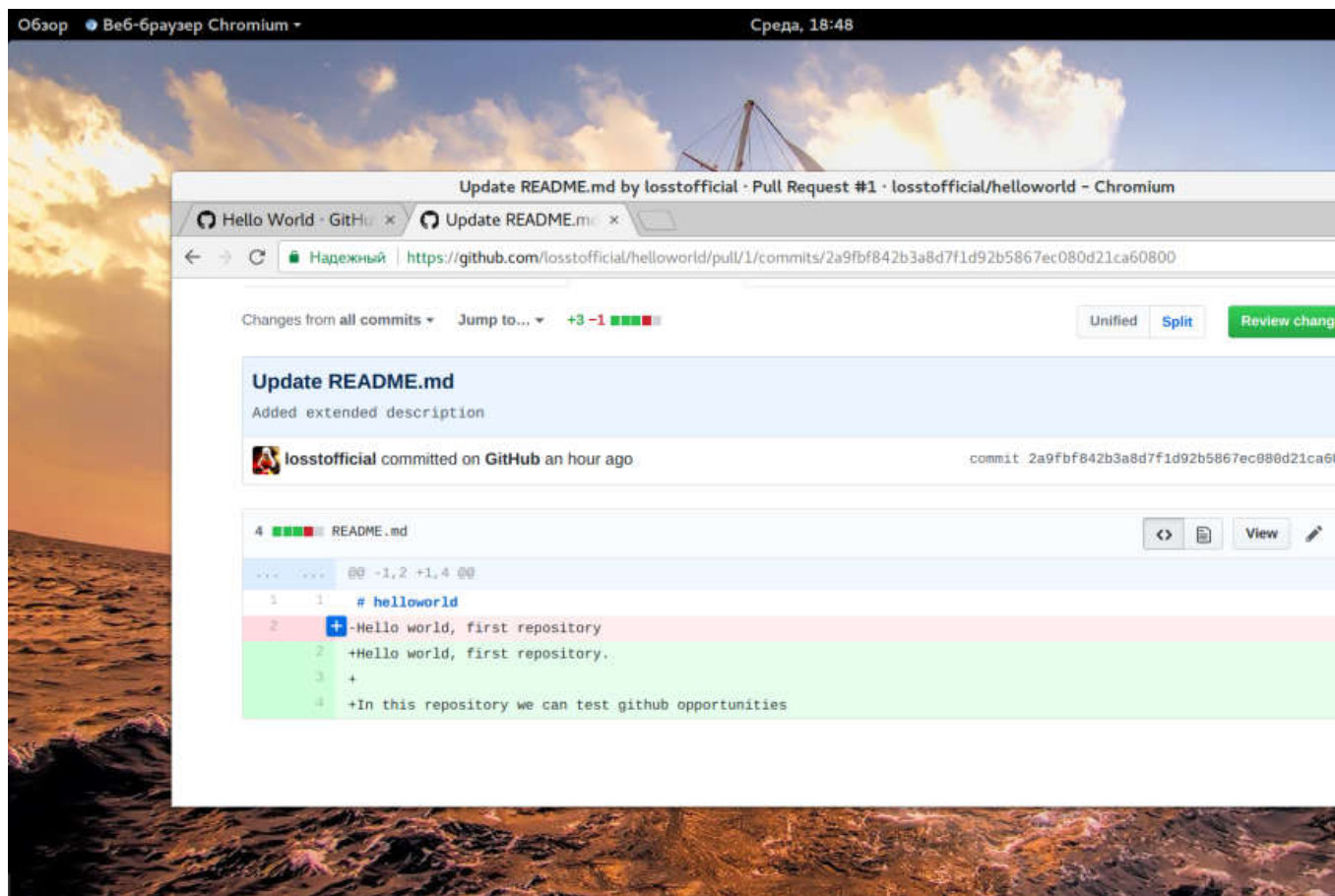


## 6. П р о с м о т р и о д о б р е н и е з а п р о с о в н а с л и я н и е

Теперь, на той же вкладке Pull Requests мы видим только что созданный запрос на слияние и нам остается только принять его нажав «**Merge Pull Request**»:

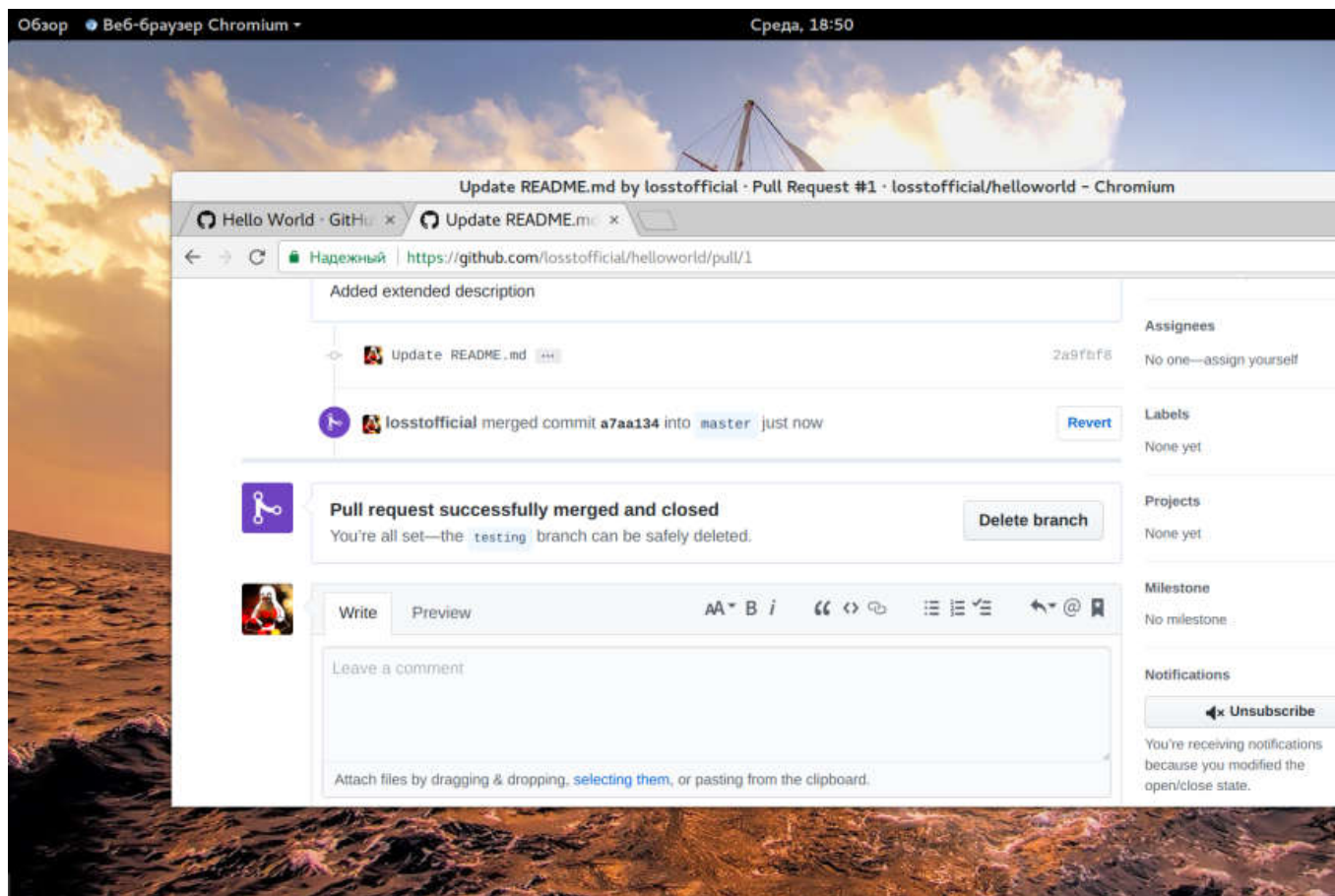


Но если этот запрос пришел от другого человека, вы должны проверить что он там изменил и нужно ли это, для этого просто нажмите на описание запроса и увидите уже знакомое окно просмотра изменений:



Дальше можно подтвердить Pull Request:

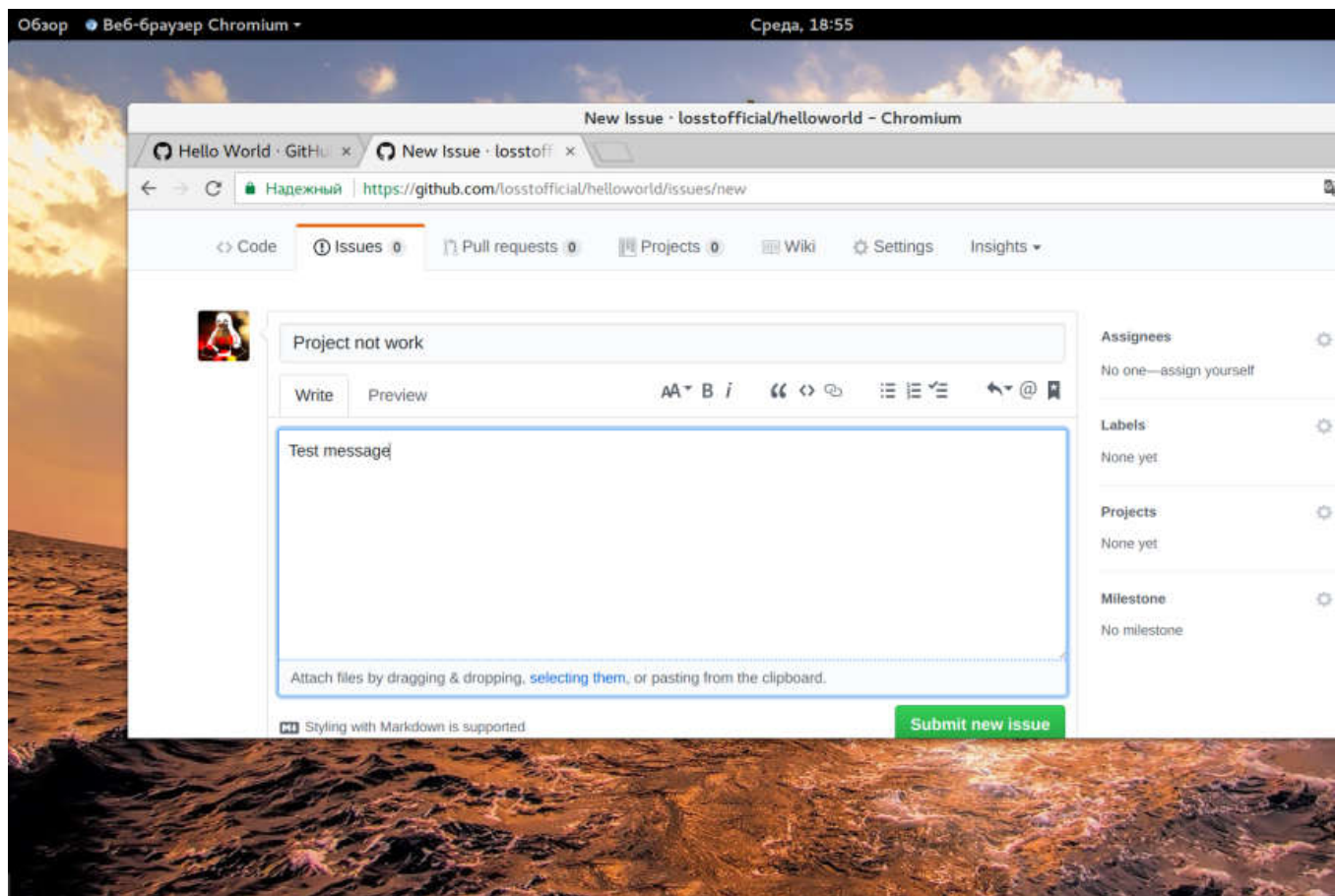




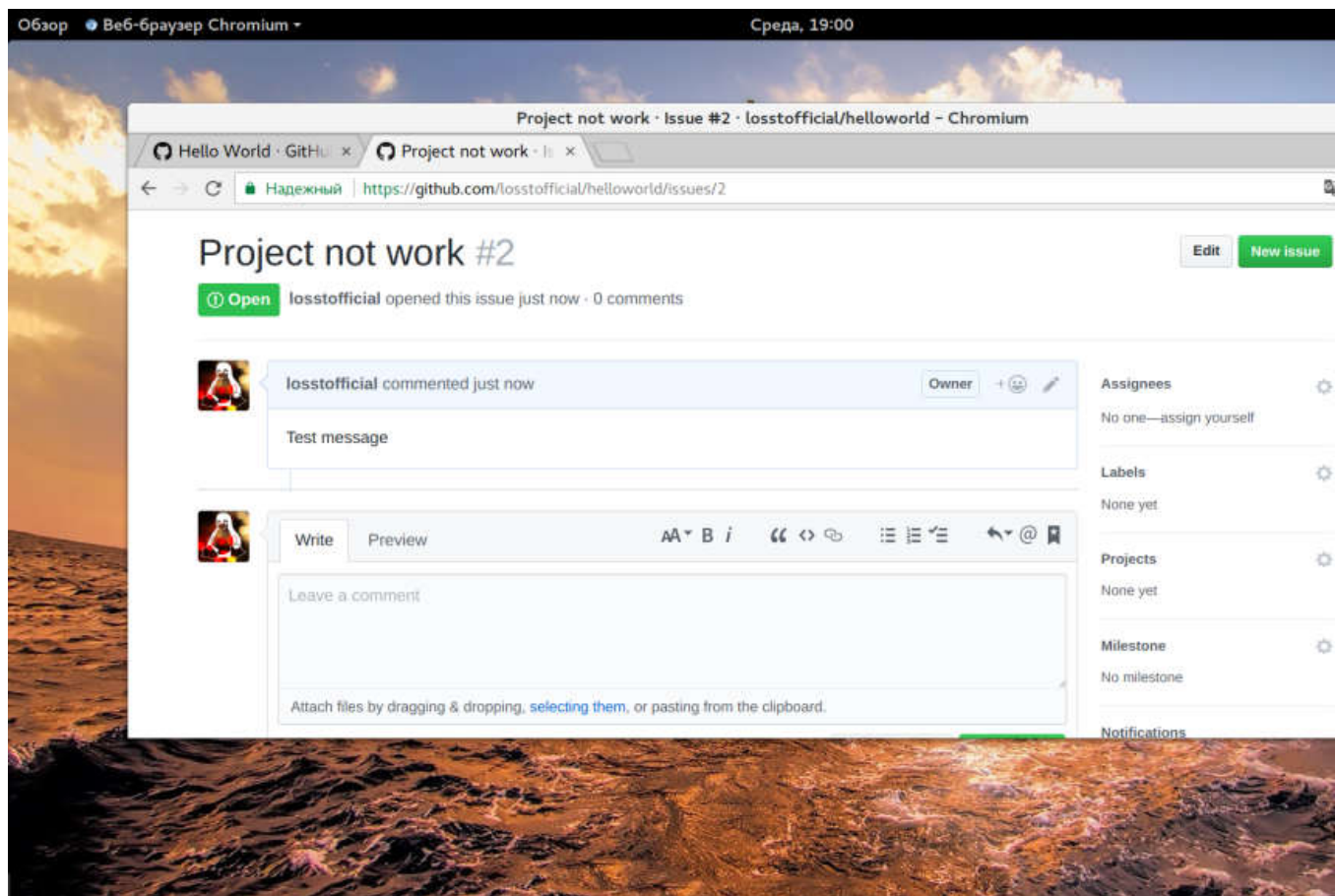
Затем код будет импортирован в основную ветку, а ветка `testing` может быть безопасно удалена.

## 7. О т ч е т ы о б о ш и б к а х

Удобно еще то, что возможно использование GitHub не только для разработки и управления кодом, но и для обратной связи с пользователями. На вкладке «**Issue**» пользователи могут оставлять сообщения о проблемах, с которыми они столкнулись при использовании вашего продукта. Откройте вкладку «**Issues**», и нажмите на кнопку «**New issue**»:

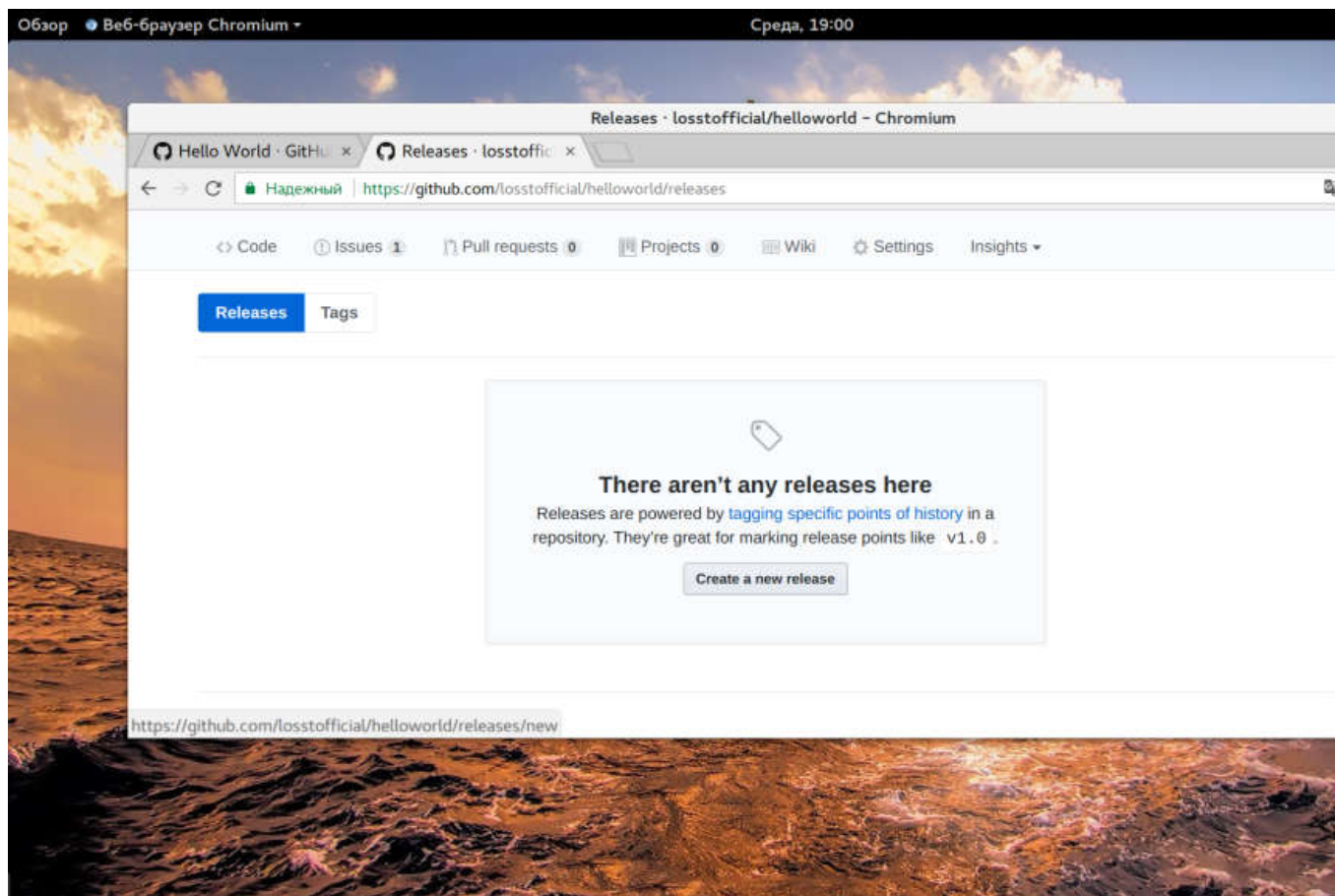


Дальше вам осталось ввести заголовок, текст и нажать «**Create new issue**».



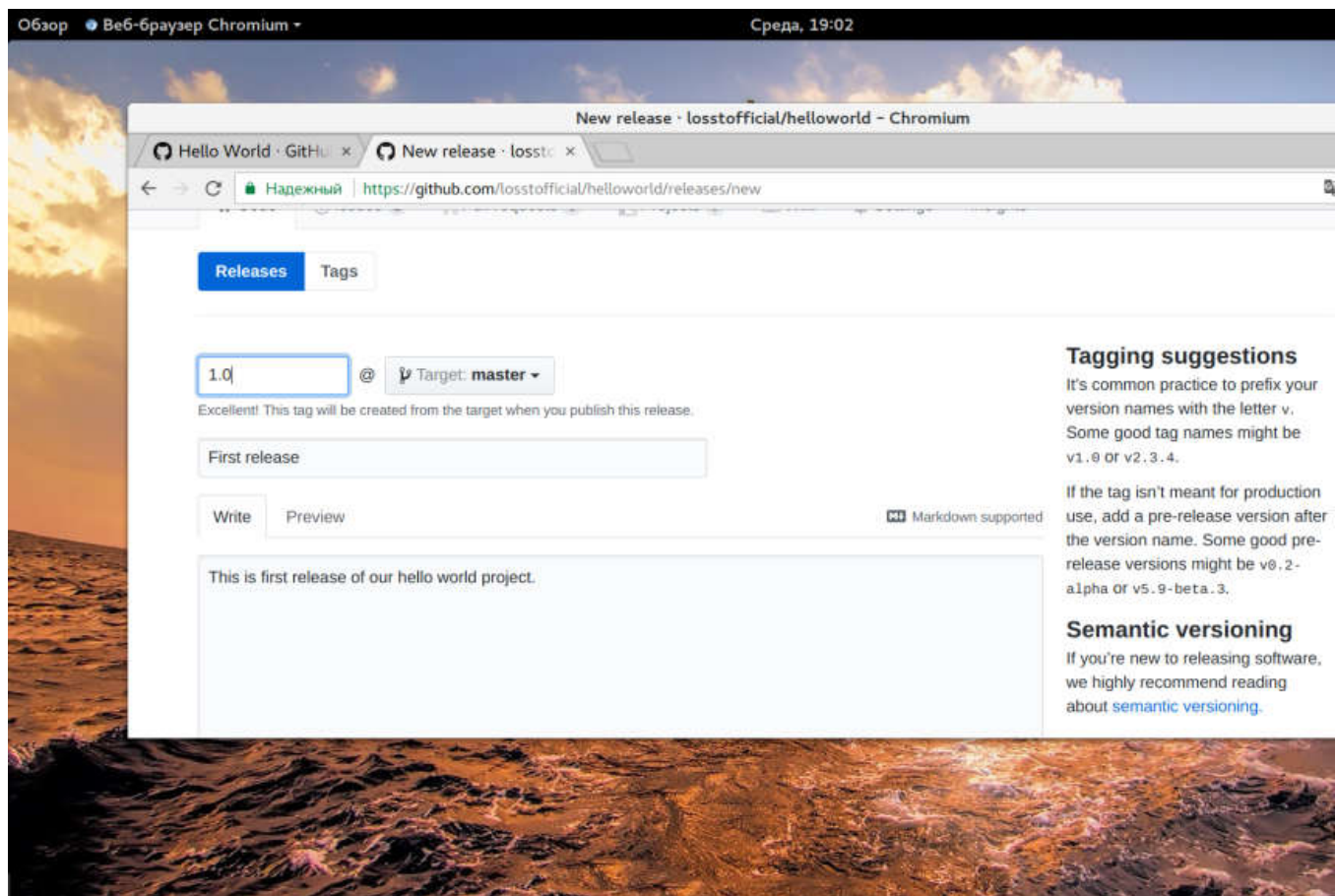
## 8. Р е л и з ы

Последнее что мы сегодня рассмотрим — это релизы. Когда продукт достиг определенной стадии можно выпустить релиз, чтобы пользователи и вы могли быть уверены что там все стабильно и никто ничего не сломал неверным Pull Request в Master. Сначала нужно перейти на главную страницу проекта, затем на вкладку «**Releases**»:

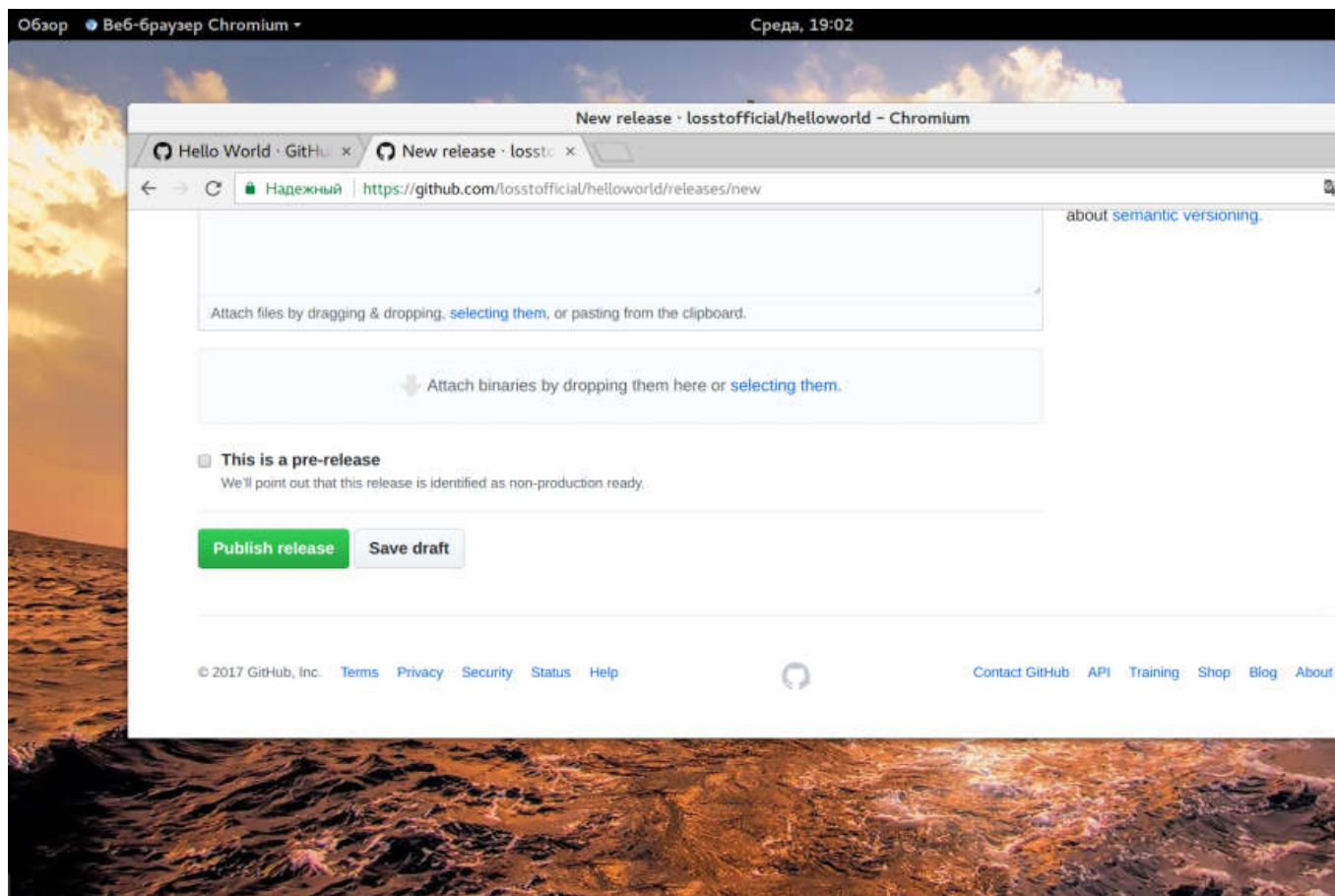


Дальше нажмите кнопку «**Create New Release**»:

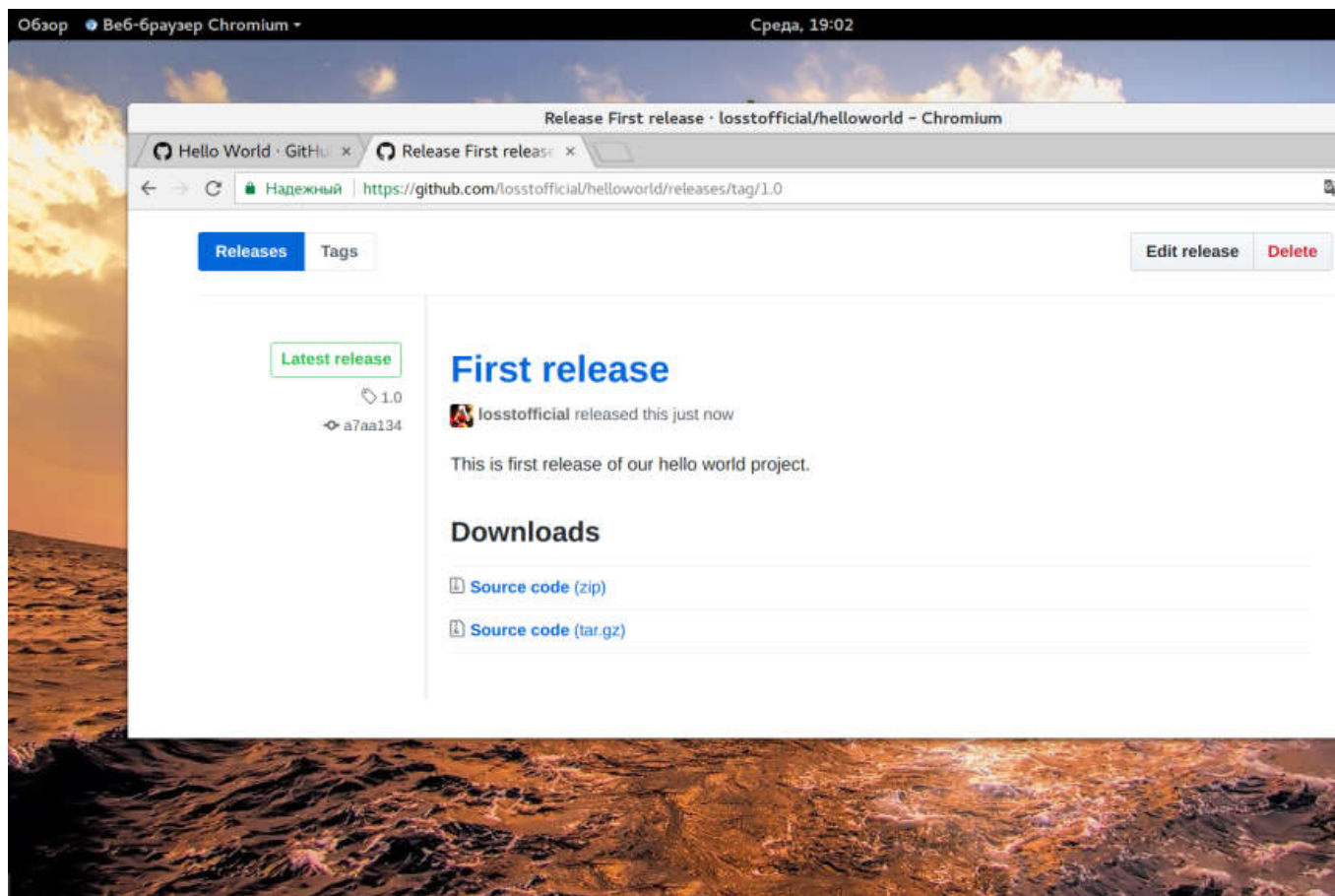




На этой странице нужно указать версию в поле «**Tag Version**», затем имя релиза и небольшое описание. Если у вас есть скомпилированные архивы с бинарниками то их тоже нужно прикрепить сюда. Затем нажмите «**Create Release**»:



После создания релиза будет создана такая страничка:



Ссылки на исходный код в tar.gz и zip будут созданы автоматически, все остальные файлы вам придется добавлять вручную.

## *В ы в о д ы*

В этой статье мы рассмотрели как пользоваться GitHub для размещения и управления своим проектом. Вся система на английском, так что базовые знания языка очень желательны но даже без них работа с github будет не очень трудной. Надеюсь, эта информация была полезной для вас. Если вас интересует как работать с Git из командной строки, смотрите статью [как пользоваться git](#) для начинающих.

## *П о х о ж и е з а п и с и :*

Нет похожих записей

## *О ц е н и т е с т а т ь ю :*

★★★★★ (6 оценок, среднее: 4,00 из 5)

[1 С о х р а н и т ь](#)

## Об авторе



[admin](#)

Основатель и администратор сайта losst.ru, увлекаюсь открытым программным обеспечением и операционной системой Linux. В качестве основной ОС сейчас использую Ubuntu. Кроме Linux интересуюсь всем, что связано с информационными технологиями и современной наукой.

Ваше имя тоже может быть здесь. Начните писать статьи для Losst. Это просто!

Смотрите подробнее как начать писать статьи – [Пишите для нас](#)

5 комментариев

1.



abo Август 25, 2017 [Ответить](#)

2.

этот сайт окончательно скатился в унылое говно с статьями «как удалить каталог» и «как посмотреть версию линуха»

3.





Г а н н и б а л Л е к т о р А в г у с т  
26, 2017 [О т в е т и т ь](#)

○

Скатился окончательно здесь только ты, запомни это, друг мой.  
А сайт пусть пишет всё шо угодно, интересно же, и я уверен, шо  
80% пользователей данного ресурса черпают с данного ресурса  
что-то полезное с любой статьи, если читают её, конечно же.

○

4.



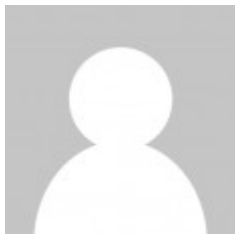
Д А в г у с т 27, 2017 [О т в е т и т ь](#)

5.

Лучше бы видео сделали.

6.

7.



В и т а л и й М а р т 31, 2018 [О т в е т и т ь](#)

8.

Очень хорошая статья, особенно для тех кто не пользовался до этого  
ГитХабом. Я к примеру пользовался иными продуктами. Всяко легче  
прочсть вот такое дружеское описание за 10 мин на русском, чем читать

документацию. Автору спасибо. Не то чтобы у меня мир разделился на «до прочтения статьи и после» , но определенную помощь статья оказала.

9.

10.



И в а М а й 21, 2018 [О т в е т и т ь](#)

11.

Коротко , ёмко, ясно и понятно изложено всё о ресурсе! Автору респект и уважение!

12.

**О т в е т и т ь**