

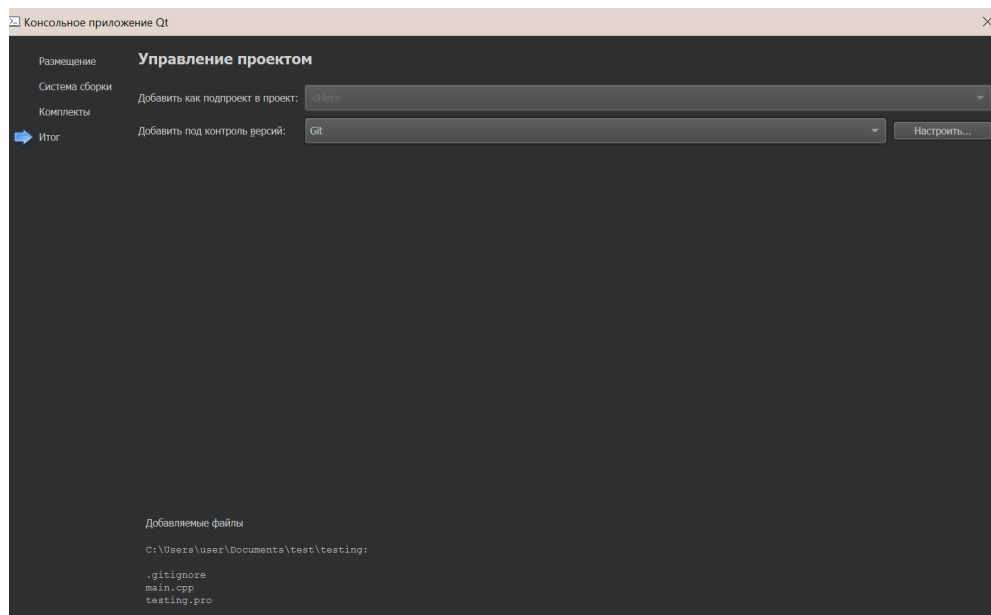
Составил: Каваллини Э. Д. 181-331

Дополнил:

Подключение git'а к Qt Creator

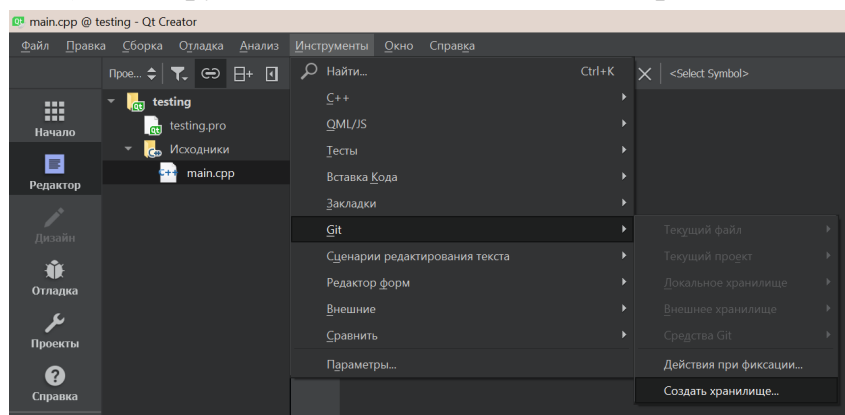
Создание хранилища в Qt Creator

При создании проекта требуется выбрать в качестве контроля версий: *git*.



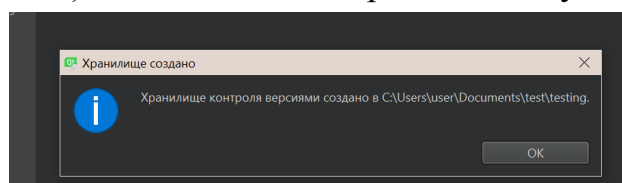
Если у вас уже был создан проект без контроля версий, то выполните следующие шаги:

1) *Инструменты -> Git -> Создать хранилище*



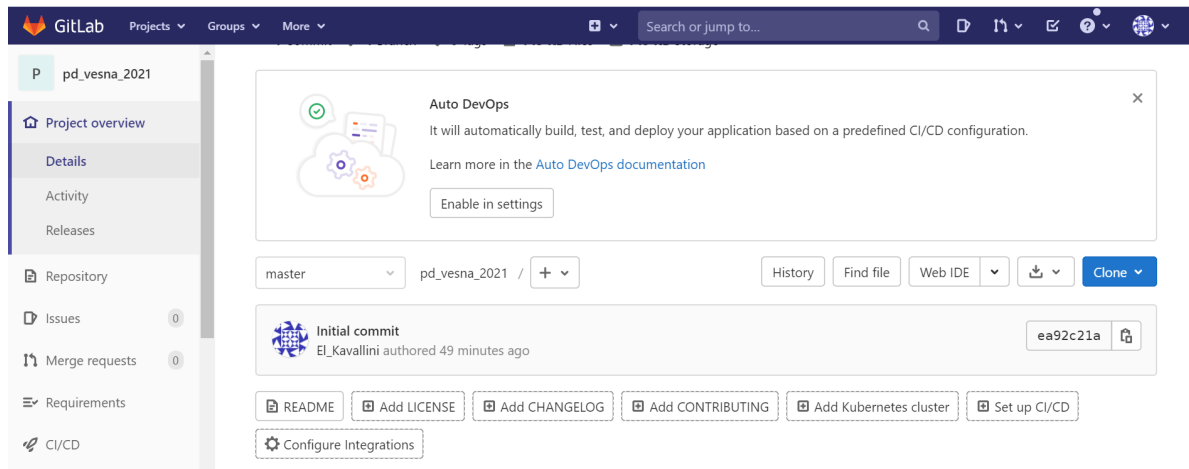
2) Вам будет предложена папка проекта по умолчанию. Выберите её.

3) Появится подтверждение об успешном создании хранилища:



Создание репозитория на GitLab

Теперь создадим репозиторий на gitlab.com, например. Репозиторий может быть пустым или же содержать README.md - это не принципиально.



Если вы используете GitHub или BitBucket, то можете воспользоваться данной инструкцией: <https://exlmoto.ru/git-and-qt-creator/>

Установка Git и работа с Git Bash

Далее, если у вас не установлен Git, это следует сделать, после чего открываем «Git Bash» и вводим такие команды:

```
git config --global user.email "ваш_email@mail.com"
git config --global user.name "ваш_никнейм"
```

```
user@DESKTOP-706PBIK MINGW64 ~
$ git config --global user.email "el_kavallini@list.ru"

user@DESKTOP-706PBIK MINGW64 ~
$ git config --global user.name "El_Kavallini"
```

Создание SSH-ключа

Один из важных этапов настройки является создание ssh-ключа. Для этого вводим в «Git Bash» следующую команду:

```
cd ~/.ssh
```

Если папка «.ssh» уже существует, сохраните резервную копию её содержимого следующей командой:

```
tar -cvf backup.tar .
```

Если такой папки нет, выполняем генерацию SSH-ключа командой:

```
ssh-keygen -t rsa -C "это поле для комментария"
```

На все вопросы, возникающие при выполнении этой команды, необходимо просто нажимать «Enter».

После того, как мы сгенерировали наш уникальный SSH-ключ, скопируем его в буфер обмена командой:

```
clip < ~/.ssh/id_rsa.pub
```

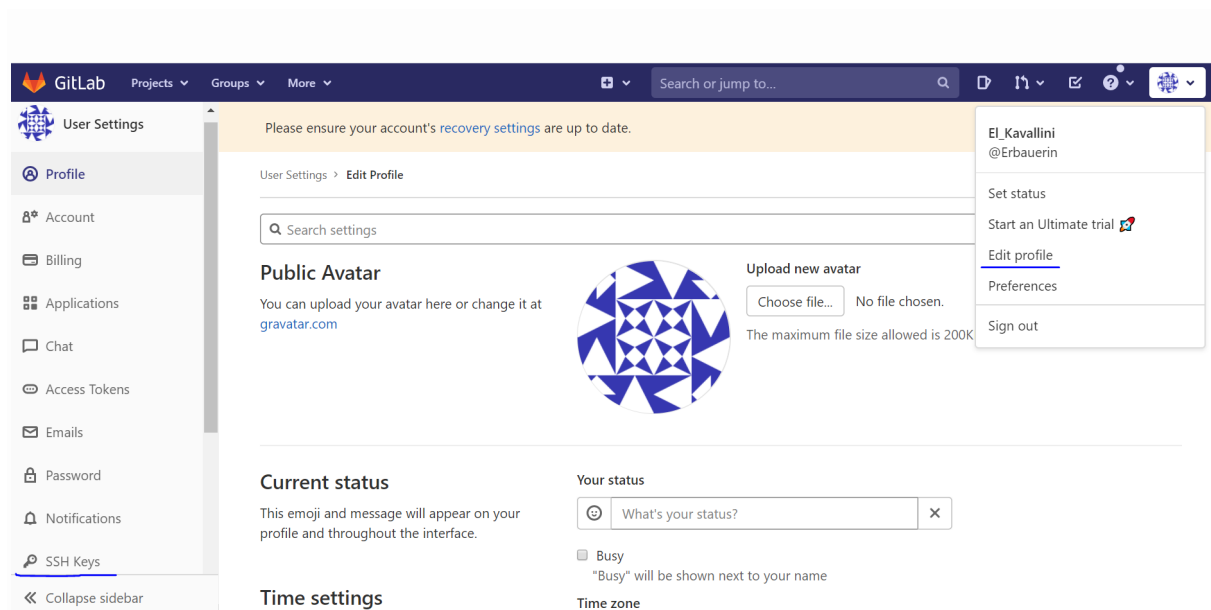
Ниже приведён ход выполнения данных команд. Стоит отметить, что если вам понадобятся файлы с ключами позже, вы сможете найти их по указанному командой пути.

```
user@DESKTOP-706PBIK MINGW64 ~/.ssh
$ ssh-keygen -t rsa -C "this_is_commit"
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/user/.ssh/id_rsa):
/c/Users/user/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/user/.ssh/id_rsa.
Your public key has been saved in /c/Users/user/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:fHJLAKA1/vJdUCkgzyajE+gJep/xKGI1HSYm5LMazyk this_is_commit
The key's randomart image is:
+---[RSA 2048]---+
|..=.O..|
|++==....|
|E+*o+o+ o.|
|=. *o+=.. o|
|.++oo.. S =|
|..O.B . * .|
|... + o . .|
|.. .|
+-----[SHA256]-----+
user@DESKTOP-706PBIK MINGW64 ~/.ssh
$ clip < ~/.ssh/id_rsa.pub
```

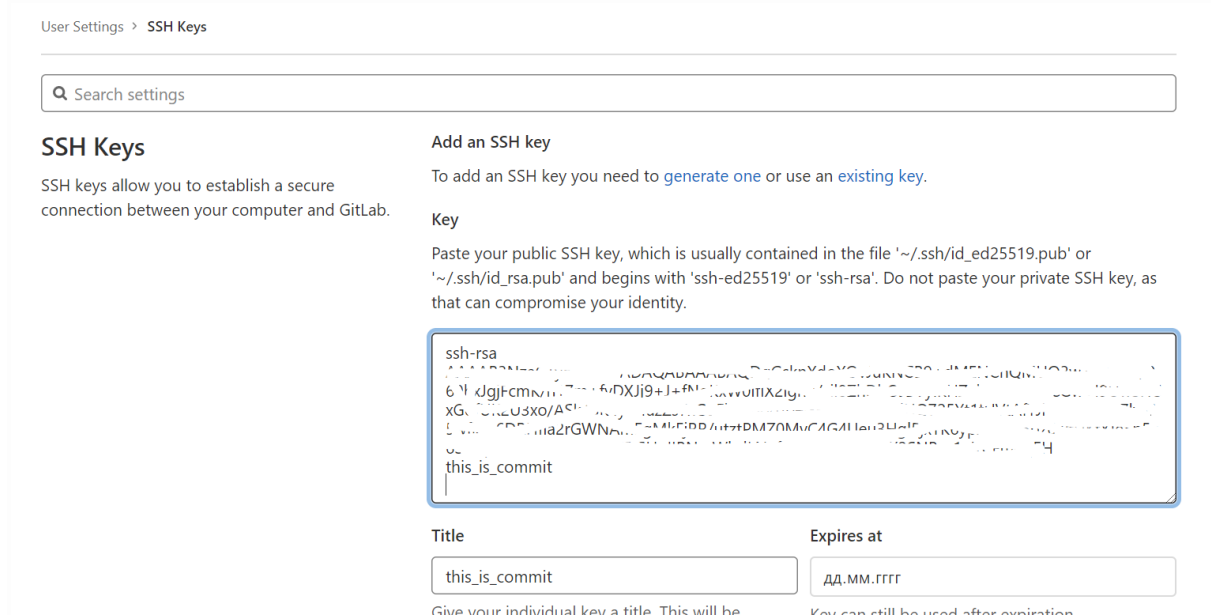
Добавление SSH-ключа в аккаунт удаленного репозитория

Следующим этапом будет - добавление ssh-ключа в наш аккаунт gitlab.

Для этого в меню аккаунта выбираем «Edit profile», а далее в левом боковом меню «SSH Keys».



Теперь в соответствующее поле вставляем скопированный (при помощи команды `clip < ~/.ssh/id_rsa.pub`) ключ из буфера обмена:



Нажимаем кнопку «Add key» для добавления ssh-ключа.

Настройка связи локального и удаленного репозитория

Осталось лишь связать наш проект с удаленным репозиторием. Для этого требуется для начала создать локальный репозиторий.

Запускаем «Git Bash» и переходим в директорию с исходным кодом проекта:

```
user@DESKTOP-706PBIK MINGW64 ~/.ssh
$ cd /c/Users/user/Documents/test/testing
```

Заметьте, что в «Git Bash» диск C:// пишется как /c/, аналогичное будет происходить и с остальными дисками.

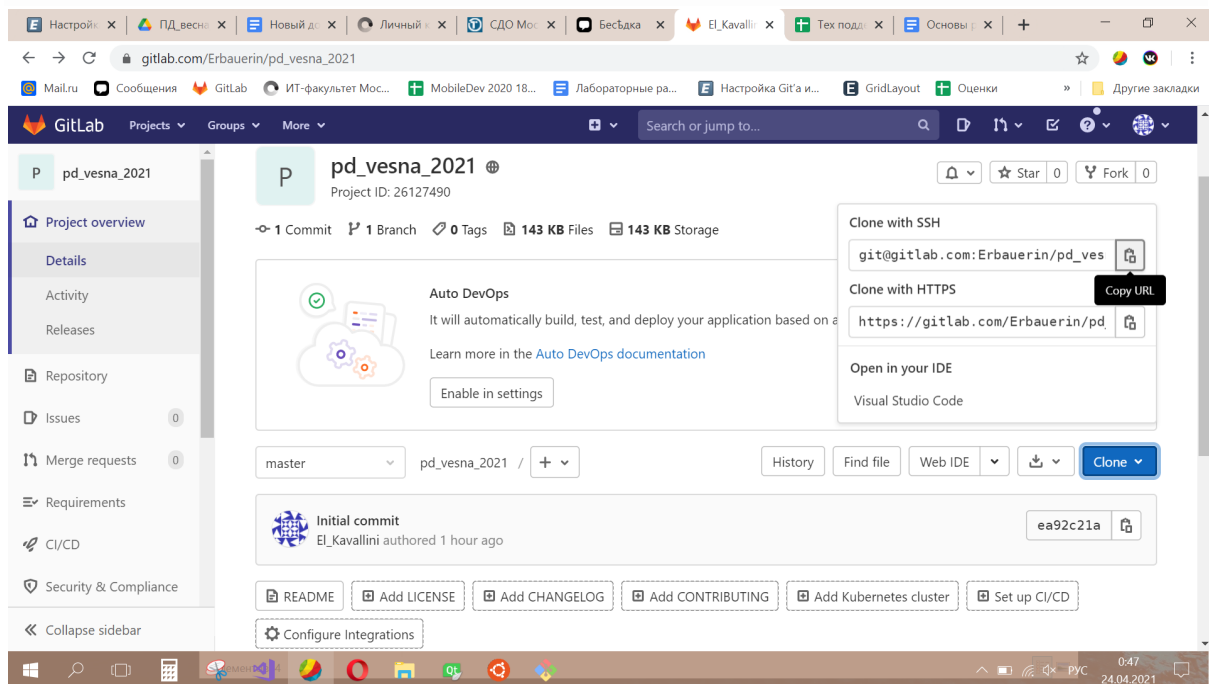
Далее стандартная процедура инициализации локального репозитория:

`git init` # инициализация пустого локального репозитория

`git add .` # добавление всех файлов, содержащихся в текущей папке, в репозиторий

`git commit -m "First commit"` # создаем первый коммит, записываем изменения

Важно! Для работы в Qt Creator требуется указать именно SSH-доступ в команде «`git remote add origin`», **SSH-ссылку** на ваш проект можно получить здесь:



`git remote add origin SSH-ссылка*` # устанавливаем связь с нашим удаленным репозиторием
если появится ошибка о том, что origin уже существует, то можно применить команду: `git remote set-url origin SSH-ссылка*`

`git pull origin master` # примените эту команду, если ваш удаленный репозиторий не пустой, это позволит вашему локальному хранилищу сначала получить эту информацию

`git push -u origin master` # отправляем исходники в удаленный репозиторий

```

user@DESKTOP-7O6PBIK MINGW64 ~/Documents/test/testing (master)
$ git config --global user.name "El_Kavallini"

user@DESKTOP-7O6PBIK MINGW64 ~/Documents/test/testing (master)
$ git config --global user.email "smert-s-kosoi@list.ru"

user@DESKTOP-7O6PBIK MINGW64 ~/Documents/test/testing (master)
$ git init
Reinitialized existing Git repository in C:/Users/user/Documents/test/testing/.git/

user@DESKTOP-7O6PBIK MINGW64 ~/Documents/test/testing (master)
$ git remote add origin git@gitlab.com:Erbauerin/pd_vesna_2021.git
fatal: remote origin already exists.

user@DESKTOP-7O6PBIK MINGW64 ~/Documents/test/testing (master)
$ git remote set-url origin git@gitlab.com:Erbauerin/pd_vesna_2021.git

user@DESKTOP-7O6PBIK MINGW64 ~/Documents/test/testing (master)
$ git add .

user@DESKTOP-7O6PBIK MINGW64 ~/Documents/test/testing (master)
$ git commit -m "Initial commit"
On branch master
Your branch and 'origin/master' have diverged,
and have 2 and 3 different commits each, respectively.
(use "git pull" to merge the remote branch into yours)

nothing to commit, working tree clean


```

```

user@DESKTOP-7O6PBIK MINGW64 ~/Documents/test/testing (master)
$ git push -u origin master
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 4 threads
Compressing objects: 100% (9/9), done.
Writing objects: 100% (9/9), 3.70 KiB | 474.00 KiB/s, done.
Total 9 (delta 2), reused 0 (delta 0)
To gitlab.com:Erbauerin/pd_vesna_2021.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.

```

Таким образом, мы успешно отправили файлы в удаленный репозиторий:



This my first commit

El_Kavallini authored 14 minutes ago

19f2896d

Upload File

Add README

Add LICENSE




Add CHANGELOG

Add CONTRIBUTING

Add Kubernetes cluster

Set up CI/CD

Configure Integrations

Name	Last commit	Last update
 main.cpp	This my first commit	14 minutes ago
 testing.pro	First commit	43 minutes ago
 testing.pro.user	This my first commit	14 minutes ago

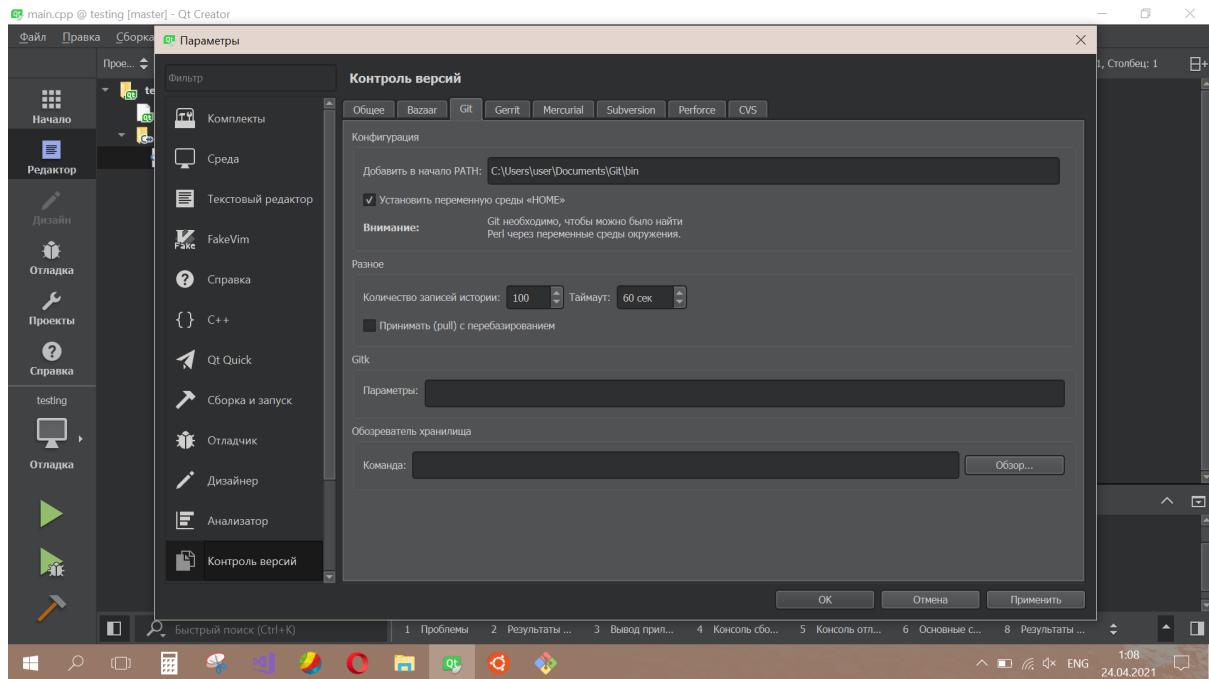
Конечная настройка Qt Creator'a и работа с Git'ом в нём

Возвращаемся в Qt Creator.

Переходим по пути:

Инструменты -> Параметры -> Контроль версий -> Git

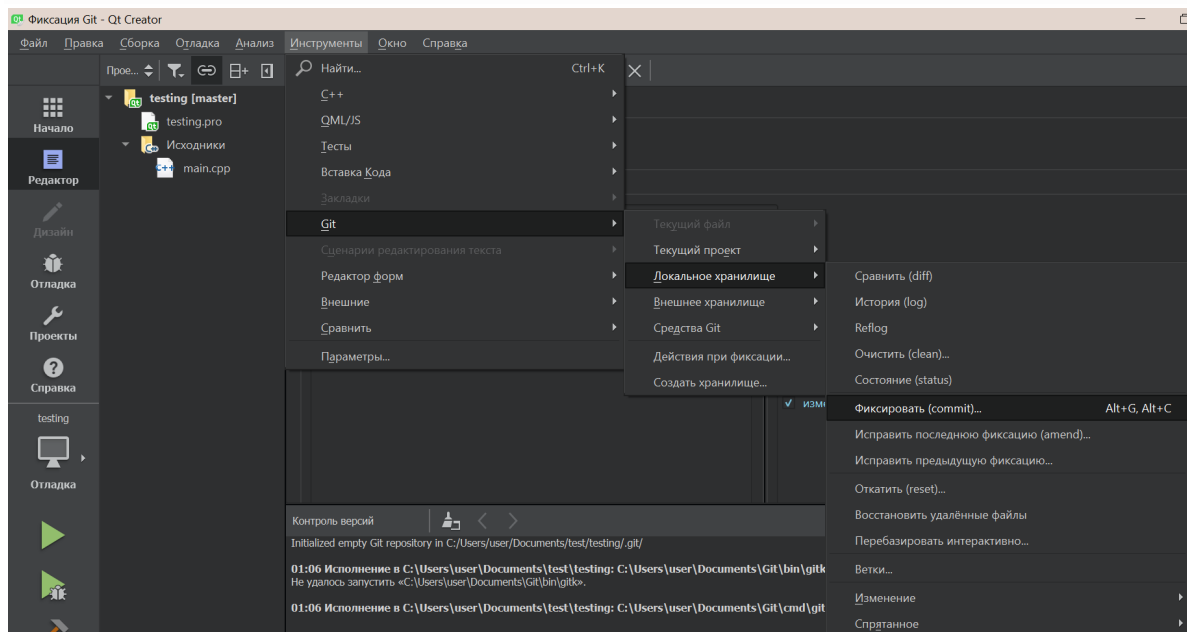
Здесь должен быть прописан путь до исполняемого файла git.exe и проставлена галочка напротив среды HOME.



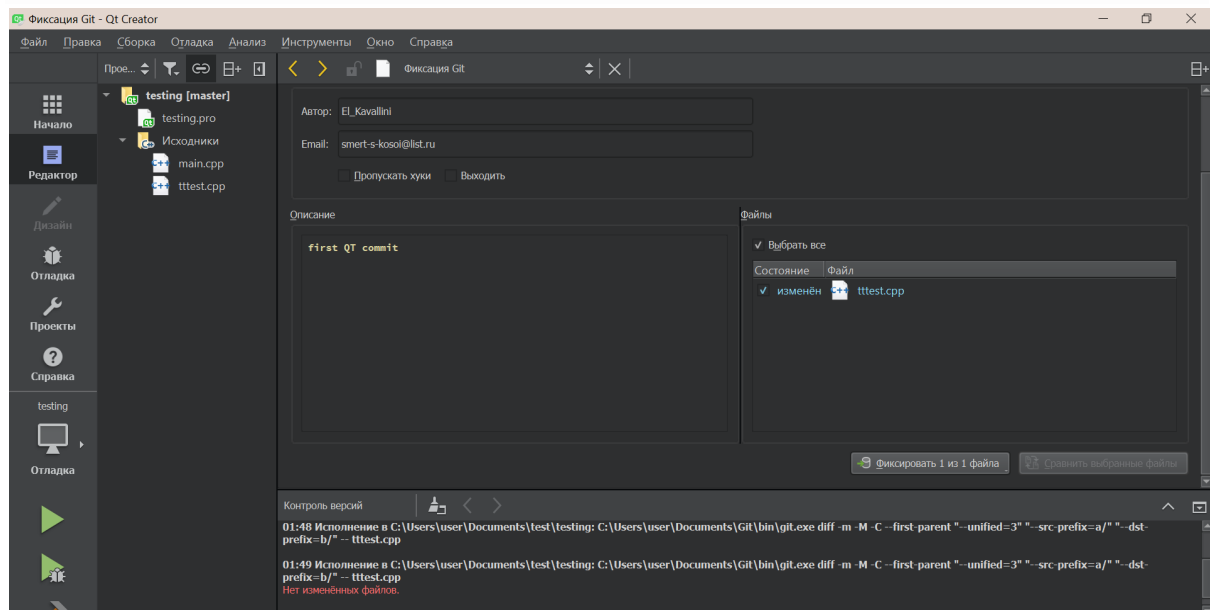
После этого следует перезапустить Qt Creator.

Чтобы зафиксировать изменения в файлах вашего проекта требуется выбрать:

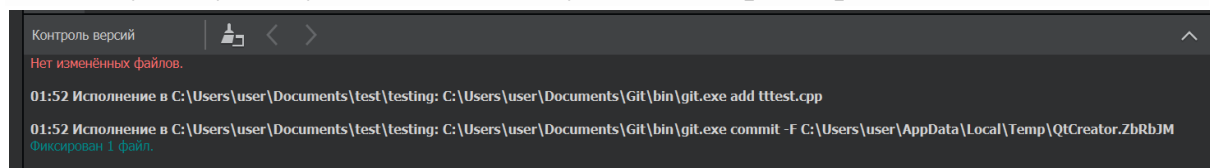
Инструменты -> Git -> Локальное хранилище -> Фиксировать



Выбираем нужные файлы, прописываем комментарий и нажимаем кнопку «Фиксировать»:

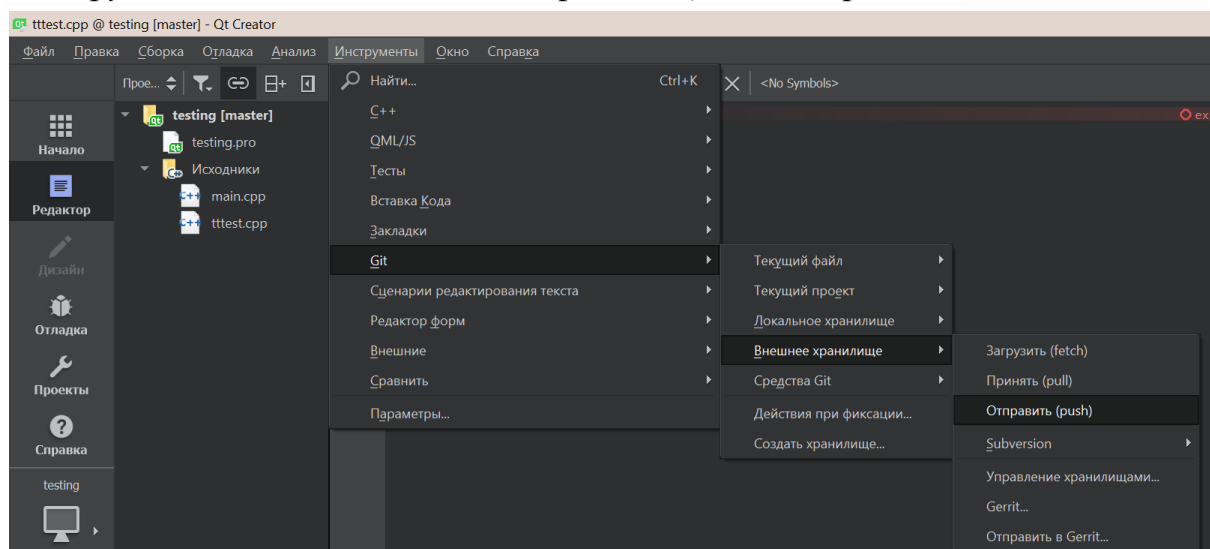


По итогу получим уведомление об успешном фиксировании:

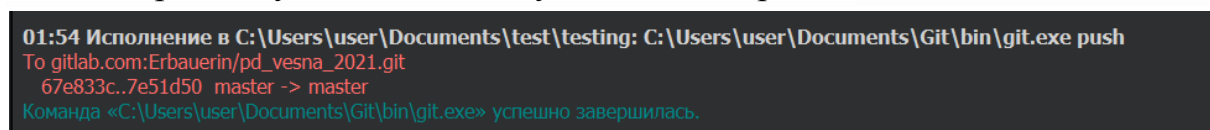


Для отправки закоммиченных изменений на удаленный репозиторий выбираем:


Инструменты -> Git -> Внешнее хранилище -> Отправить




И вот очередное уведомление об успешной отправке:







А вот и наш комментарий на gitlab:

 **first QT commit**
El_Kavallini authored 2 minutes ago

7e51d503 

[Upload File](#) [Add README](#) [Add LICENSE](#) [Add CHANGELOG](#) [Add CONTRIBUTING](#) [Add Kubernetes cluster](#)

[Set up CI/CD](#) [Configure Integrations](#)

Name	Last commit	Last update
 main.cpp	2c	8 minutes ago
 testing.pro	2c	8 minutes ago
 testing.pro.user	2c	8 minutes ago
 ttest.cpp	<u>first QT commit</u>	2 minutes ago

Теперь вы сможете комментировать и отправлять изменения своего проекта всего двумя кликами!

Наше кредо: «Делайте что-то лёгкое и повседневное ещё проще.»