

Инструкция по работе с репозиторием (в Bitbucket)

i Для заведения нового приложения в каталоге необходимо запросить доступ на редактирование репозитория <https://bitbucket.lesta.group/projects/CONFIGS/repos/applications/>

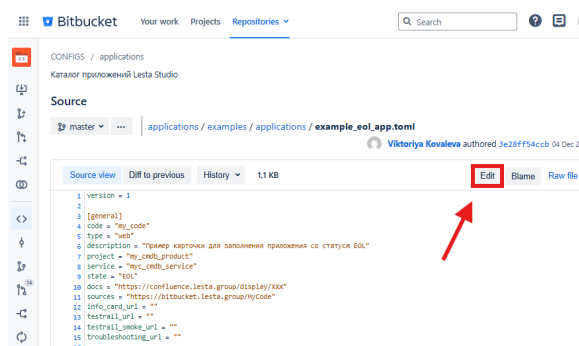
- 1. Структура репозитория
- 2. Маленькие правки
- 3. Большие изменения, новые файлы
 - 3.1. Клонирование репозитория на ПК
 - 3.2. Работа с локальной копией репозитория
 - 3.2.1. Создание рабочей ветки
 - 3.2.2. Создание нового файла
 - 3.3. Загрузка изменения в оригинальный репозиторий
- 4. Обработка комментариев от аппруверов
- 5. Мерж ветки в Master

1. Структура репозитория

| Путь | Описание директорий / файлов |
|----------------------|--|
| ./applications | Каталог приложений, структурированный по ответственным командам/оргструктуре |
| ./examples | Примеры приложений и сервисов |
| ./projects | Каталог проектов |
| ./services | Каталог сервисов |
| ./tags | Каталог тегов |
| ./tech | |
| ./tech/_dependencies | Инвертированные externals (заполняется автоматически) |
| ./tech/_releases | Информация о релизах приложения (заполняется автоматически) |
| ./tech/_tech_stack | Информация о техническом стеке приложений (заполняется автоматически) |
| ./technologies.toml | Описание технологий для составления технологического радар |

2. Маленькие правки

Если требуется внести небольшие правки в существующие файлы, можно сделать это прямо в Bitbucket, с помощью встроенных в него инструментов



Bitbucket:

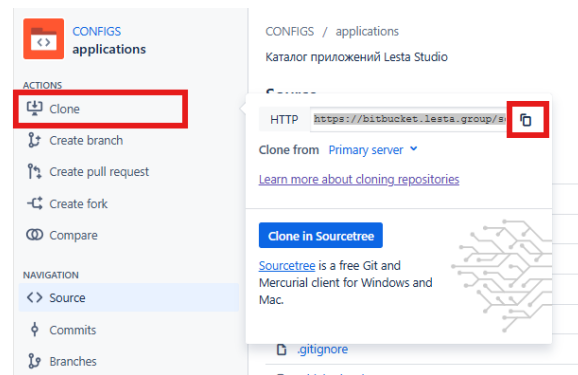
3. Большие изменения, новые файлы

3.1. Клонирование репозитория на ПК

Получите доступ на Read-Write в репозиторий APP Catalog + просто доступ к Bitbucket.
Авторизуйтесь в Bitbucket

Если вы хотите работать с репозиторием локально на своем ПК (без интернета, удаленно или просто в удобном для вас инструменте):

- создайте на своем ПК папку, куда хотите скачать репозиторий
- в Bitbucket в нужном репозитории выберите функцию Clone и скопируйте ссылку
- скачайте Git Bush - <https://git-scm.com/install/windows>, откройте его и введите команды:
 - `cd <>`
 - `git clone < >`
- вам откроются окна для авторизации - введите свои рабочие логин (без почтового домена) и пароль (если у вас корректные доступы, все файлы репозитория будут скопированы на ваш ПК в указанную директорию)
- теперь вы можете открыть репозиторий в любом удобном для вас инструменте (например, VS Code) или продолжить работу в консоли Git Bush, а файлы создать и наполнить через блокнот



Git Bush:

```
v_kovaleva@RU4-WL-1070 MINGW64 ~
$ cd Documents/rep

v_kovaleva@RU4-WL-1070 MINGW64 ~/Documents/rep (master)
$ git clone https://bitbucket.lesta.group/scm/configs/applications.git
Cloning into 'applications'...
fatal: Bitbucket DC OAuth Client ID must be defined
remote: Enumerating objects: 16792, done.
remote: Counting objects: 100% (156/156), done.
remote: Compressing objects: 100% (155/155), done.
remote: Total 16792 (delta 81), reused 2 (delta 0), pack-reused 16636 (from 1)
Receiving objects: 100% (16792/16792), 2.32 MiB | 7.93 MiB/s, done.
Resolving deltas: 100% (12100/12100), done.
Updating files: 100% (2366/2366), done.
```

3.2. Работа с локальной копией репозитория

На примере работы в VS Code

- откройте репозиторий в программе VS Code "File" (верхнее меню) "Open Folder..." выберите папку скачанного репозитория "Select Folder"
- откройте панель командной строки - Ctrl + J

3.2.1. Создание рабочей ветки

- выполните команду для добавления новой ветки (название ветки не может содержать пробелы):

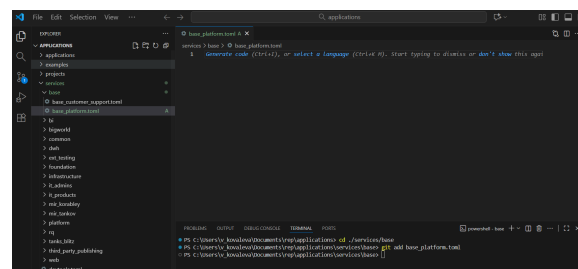
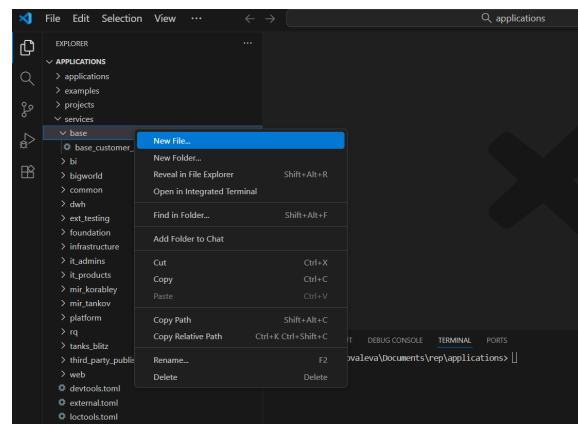
```
git checkout -b "< >"
```

Например, можно указать номер задачи в Jira, чтобы автоматически связать их:

```
git checkout -b "BASE-2002"
```

- после создания ветки, вы автоматически переключитесь на нее
 - чтобы переключиться на другую ветку (если работаете с несколькими):

```
git checkout < >
```



3.2.2. Создание нового файла

- создайте файл с расширением .toml в необходимом для вас разделе (кликните по выбранной папке правой кнопкой мыши и выберите из выпадающего меню "New File..." задайте название файла с расширением toml)
 - вы можете создать необходимый файл с помощью стандартного блокнота и поместить его через проводник в нужную папку (без VS Code)
- в командной строке перейдите в необходимую папку и добавьте созданный файл в систему контроля версий (чтобы вы могли сами отслеживать изменения в нем - пока никто кроме вас его не увидит)

```
cd < >
```

```
git add < >
```

- Заполните созданный файл необходимой информацией (воспользуйтесь предложенными шаблонами - папка "examples")

- Файлы, где внесены изменения подсвечиваются и отмечаются буквой "M"

- Сохраните внесенные изменения:

- Сохранить изменения в конкретных файлах:

```
git add < >
```

Чтобы сохранить все внесенные изменения, можно использовать команду:

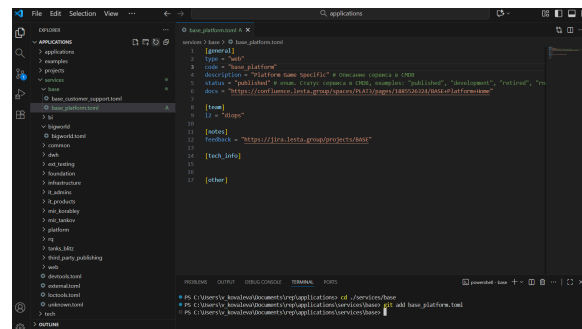
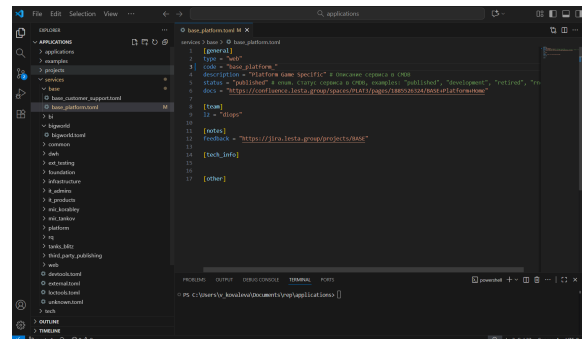
```
git add .
```

- Зафиксировать версию изменений ветки:

```
git commit -m "< >"
```

- Например, указать номер задачи и что конкретно было сделано:

```
git commit -m "<BASE-2002| >"
```

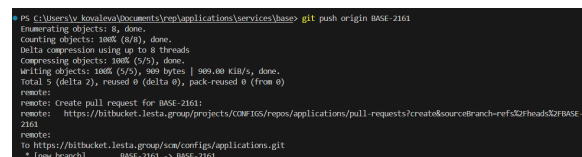


3.3. Загрузка изменения в оригинальный репозиторий

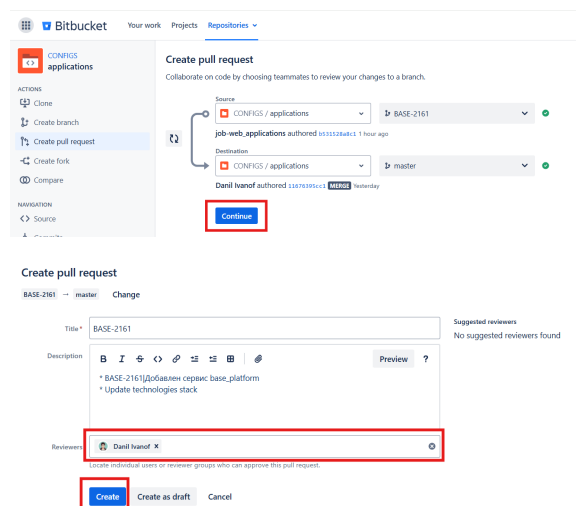
Чтобы загрузить изменения из своей локальной копии в оригинальный репозиторий, необходимо

- Отправить на согласование предложенные изменения:

```
git push origin < >
```



2. В консоли вам отобразится ссылка на pull request - необходимо перейти по ней в Bitbucket
3. В Bitbucket открыть Pull Request
 - a. выберите ветку, в которую хотите добавить изменения (master)
 - b. нажмите "Continue"
 - c. добавьте ревьюеров
 - d. нажмите "Create"



4. Обработка комментариев от аппруверов

При проведении ревью, аппруверы могут оставлять комментарии - необходимо на них ответить и/или обработать указанные замечания:

- ответить можно прямо в Bitbucket (см. рисунок)
- чтобы внести изменения/исправления, необходимо добавить их новым коммитом:
 - Подтяните другие коммиты вашей ветки (которые автоматически добавляются или кем-то из коллег) в свою локальную копию:

```
git pull origin < >
```

- Внести необходимые изменения
- Сохранить изменения в конкретных файлах:

```
git add < />
```

Например:

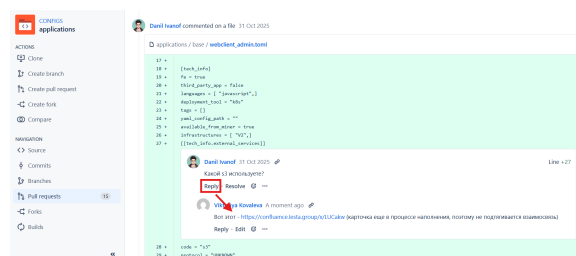
```
git add ./applications/base
```

- Зафиксировать версию изменений ветки:

```
git commit -m "< >"
```

- Загрузить изменения в оригинальный репозиторий:

```
git push origin < >
```



5. Мерж ветки в Master

После получения всех аппрувов, сmergeйте ветку: откройте свой PR и нажмите кнопку "Merge".

Теперь обновите свою локальную копию репозитория, чтобы в него подтянулись все внесенные вами и другими коллегами изменения:

- Переключитесь на master-ветку:

```
git checkout master
```

- Актуализируйте свою локальную копию с удаленным репозиторием (скачайте все изменения):

```
git pull
```

BASE-2161 — master OPEN

BASE-2161

Overview Diff Commits Builds

Viktoriya Kovalyeva created a pull request 2 days ago

- BASE-2161: добавлен сервис base_platform
- Update technologies stack

1 build

Applications cards validation

BASE-2161

Start review

Merge pull request

Commit message

Pull request #209: BASE-2161

Merge in CONFIGS/applications from BASE-2161 to master

BASE-2161

master

☐ Delete BASE-2161 after merging

Merge commit Merge Close