

Инструкция по работе с репозиторием (в Bitbucket)



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 оригиналный репозиторий
- 4. Обработка комментариев от апруверов
- 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 радара

2. Маленькие правки

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

The screenshot shows the Bitbucket interface for editing a file named 'example_eol_app.toml'. The 'Source' tab is selected, showing the file's content in a code editor. At the top right of the editor, there is a red box highlighting the 'Edit' button. A red arrow points from the text above to this 'Edit' button.

```
version = 1
[general]
type = "code"
type = "eol"
description = "Пример картонки для заполнения приложения со статусом EOL."
project = "my_code_product"
service = "my_code_service"
state = "tot"
docs = "https://confluence.lesta.group/display/XXX"
info_card_url = ""
testrail_url = ""
testrail_smoke_url = ""
troubleshooting_url = ""
```

3. Большие изменения, новые файлы

3.1. Клонирование репозитория на ПК

Bitbucket:

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):

1. создайте на своем ПК папку, куда хотите скачать репозиторий
2. в Bitbucket в нужном репозитории выберите функцию Clone и скопируйте ссылку
3. 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)

4. 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

The screenshot shows the Bitbucket repository page for 'CONFIGS / applications'. In the 'ACTIONS' sidebar, the 'Clone' button is highlighted with a red box. Below it are other actions like 'Create branch', 'Create pull request', 'Create fork', 'Compare', and 'Source' (which is selected). On the right, there's a 'Clone in Sourcetree' section with a 'Clone' button and a 'gitignore' file icon.

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

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

3.2.1. Создание рабочей ветки

1. 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"
```

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

```
git checkout < >
```

The screenshot shows the VS Code interface with the 'applications' folder selected in the Explorer sidebar. A context menu is open over the 'base' folder, with 'New File...' highlighted. Other options include 'New Folder...', 'Reveal in File Explorer', 'Open in Integrated Terminal', 'Find in Folder...', 'Add Folder to Chat', 'Cut', 'Copy', 'Paste', 'Copy Path', 'Copy Relative Path', 'Rename...', and 'Delete'.

The screenshot shows the VS Code interface with the 'applications' folder selected in the Explorer sidebar. The terminal at the bottom shows command-line history related to the 'base' folder.

3.2.2. Создание нового файла

1. 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)
2. 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 < >

3. Заполните созданный файл необходимой информацией (воспользуйтесь предложенными шаблонами - папка "examples")
 - Файлы, где внесены изменения подсвечиваются и отмечаются буквой "M"
4. 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:

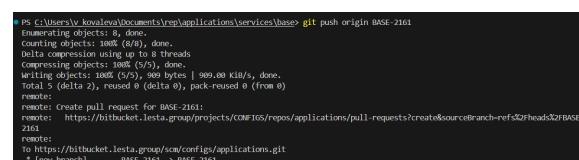
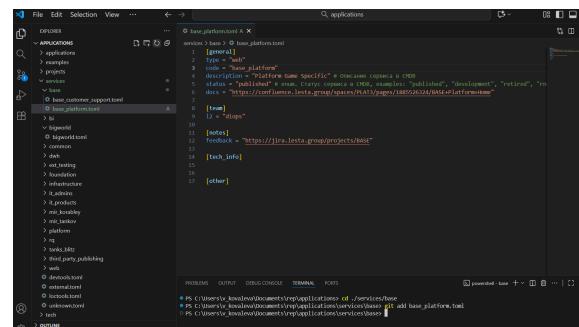
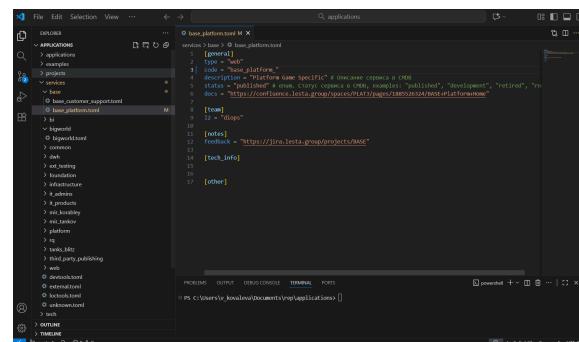
```
git commit -m "<BASE-2002| >"
```

3.3. Загрузка изменения в оригинальный репозиторий

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

1. Отправить на согласование предложенные изменения:

```
git push origin < >
```



```
PS C:\Users\lv_kovaleva\Documents\rep\applications\services\base> git push origin BASE-2161
Enumerating objects: 8, done
Compressing objects: 100% (7/7), done.
Delta compression using up to 8 threads
Writing objects: 100% (8/8), 909 bytes | 909.00 KiB/s, done.
Total 8 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote: create pull request for BASE-2161:
remote: https://bitbucket.esta-group/scm/configs/applications/pull-requests?create&sourceBranch=refs%2Fheads%2Fbase-2161
remote:
To https://bitbucket.esta-group/scm/configs/applications.git
 * [new branch]      BASE-2161 -> BASE-2161
```

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 реviewеров
 - d. нажмите "Create"

The first screenshot shows the 'Create pull request' dialog in Bitbucket. It has two main sections: 'Source' (set to 'CONFS / applications') and 'Destination' (set to 'master'). A 'Reviewers' field at the bottom contains 'Danil Ivanof'. The 'Continue' button is highlighted with a red box.

The second screenshot shows the 'Create pull request' form. It includes fields for 'Title' (set to 'BASE-2161'), 'Description' (containing 'BASE-2161|Добавлен сервис base_platform' and 'Update technologies stack'), and a 'Reviewers' section with 'Danil Ivanof' selected. The 'Create' button is highlighted with a red box.

The third screenshot shows the pull request list. It lists a single pull request titled 'BASE-2161' with a status of 'MERGED'. The 'Create' button for this pull request is highlighted with a red box.

4. Обработка комментариев от апруверов

При проведении реview, апруверы 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 оригиналый репозиторий:

```
git push origin < >
```

A screenshot of a GitHub pull request interface. It shows a comment from 'Danil Ivanof' dated '31 Oct 2025' with the message 'Кто-то якобы?'. Below the comment is a 'Reply' button, which is highlighted with a red box.

5. Мерж ветки в Master

После получения всех аппровов, смержите 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
```

The screenshot shows a GitHub pull request interface for a repository named 'BASE-2161'. The 'master' branch is selected. At the top right, there is a blue 'Merge' button with a red box drawn around it. Below the button, there are status indicators: '1 build', 'Applications cards validation', and 'BASE-2161'. The main area shows a 'Merge pull request' form with a commit message field containing 'Pull request #209: BASE-2161' and a note below it stating 'Merge in CONFIGS/applications from BASE-2161 to master'. A merge conflict graph shows two branches merging: 'BASE-2161' (blue) and 'master' (green). A checkbox labeled 'Delete 'BASE-2161' after merging' is present. At the bottom right of the form, there is another 'Merge' button and a 'Close' link.