**Инструкция по работе с git на проекте Mobile 1.0**

Уважаемые коллеги, наша группа разработки работает в <https://github.com/> репозиторий <https://github.com/natalysoliton/SF_DevOps_1> ветка мастер для разработки версии 1.0 мобильного продукта. Для разработки определенных фич вы можете делать отдельные фичи-бранчи из мастера и при успешной работе кода, аккуратно вливать ваш код в ветку мастер По правилам компании мы настоятельно просим вас называть ваши отбранчеванные ветки в формате user/name/feature/. Также в целях безопасности мы ввели правила приоритетности и доступности коллег к коду проекта. Кроме этого, у нас настроена политика аппрувов ваших комитов группой экспертов.

Мы подготовили для вас полезную информацию для быстрого, удобного совместного использования гит на вашем проекте.

Команда git config позволяет настроить инсталляцию Git (или отдельный репозиторий) из командной строки. С помощью этой команды можно установить любые настройки: от информации о пользователе до его предпочтений и характеристик репозитория. Ниже перечислены распространенные варианты конфигурации.

Git хранит варианты конфигурации в трех различных файлах, позволяющих ограничивать область видимости на уровне отдельных репозиториев (локальный), пользователя (глобальный) или всей системы (системный):

Локальный: /.git/config — настройки на уровне репозитория.

Глобальный: /.gitconfig — настройки на уровне пользователя.

Системный: $(prefix)/etc/gitconfig — настройки на уровне всей системы.

Укажите имя автора, которое будет использоваться для всех коммитов в текущем репозитории. Обычно для настройки параметров конфигурации для текущего пользователя используется флаг --global.

git config --global user.name <name>

Эта команда задает имя автора, которое будет использоваться для всех коммитов, выполненных текущим пользователем.

Добавление аргумента --local или выполнение команды без параметра уровня конфигурации приведет к установке значения user.name для текущего локального репозитория.

git config --local user.email <email>

Эта команда задает адрес электронной почты автора, который будет использоваться для всех коммитов, выполненных текущим пользователем.

git config --global alias.<alias-name> <git-command>

Создайте быстрые клавиши для команды Git. Это мощная возможность для создания собственных комбинаций клавиш для часто используемых команд Git. Ниже показан упрощенный пример:

git config --global alias.ci commit

**Пример:**

В первую очередь после установки Git требуется указать свое имя и адрес электронной почты, а также настроить некоторые параметры по умолчанию. Пример типичной начальной конфигурации показан далее.

Представьтесь репозиторию Git с помощью команды git config

git --global user.name "John Smith" git config --global user.email john@example.com

**Добавьте алиасы по типу SVN**

git config --global alias.st status

git config --global alias.co checkout

git config --global alias.br branch

git config --global alias.up rebase

git config --global alias.ci commit

Создастся файл ~ /.gitconfig

(<https://github.com/natalysoliton/SF_DevOps_1/blob/main/git%20config>)

Система Git имеет в своем составе возможность графического представления ветвления в репозитории. Причем, такое представление можно сделать даже в консоли, с помощью псевдографики.

Это можно сделать командой:

git log --oneline --abbrev-commit --all –graph

git log

**Для новых сотрудников мы рекомендуем придерживаться следующего воркфлоу проектной работы в гите:**

1\Сделайте клонирование рабочего репозитория (вы можете использовать разные IDE на ваше усмотрение):

git clone <https://github.com/natalysoliton/SF_DevOps_1> (сделайте git branch, чтобы убедиться, что вы в ветке master)

2\Отбранчуйте вашу ветку из ветки master:

git checkout –b user/name/feature1

3\Если вам надо добавить файл, используйте:

git add .

4\Добавляем комментарий:

git commit -m 'first launch'

5\Переносим из вашей ветки в ветку мастер:

git push -u origin/master user/name/feature1

6\Удалите вашу ветку:

git branch –d user/name/feature1

Если вам нужно внести изменение в уже существующий файл в ветке мастер, то не забудьте затянуть апдейты ваших коллег из ветки master:

git pull

Далее создаете свою ветку (см. п. 2).

Внесите ваши изменения в файл(ы).

Проделайте шаги 4, 5, 6 (см.выше).