**XML**

1. Создать внешний репозиторий c названием XML.

На github.com создаем новый репозиторий “XML”.

2. Клонировать репозиторий XML на локальный компьютер.

В Git Bash заходим в нашу локальную папку с репозиториями:

cd it/git

Клонируем наш репозиторий с github:

Git clone <https://github.com/Vasiliy-Afanasev/XML.git>

Заходим в него:

cd xml

3. Внутри локального XML создать файл “new.xml”.

touch new.xml

4. Добавить файл под гит.

git add new.xml

5. Закоммитить файл.

Перед этим обычно проверяю добавился ли наш файл:

git status

ответ:

On branch main

Your branch is up to date with 'origin/main'.

Changes to be committed:

(use "git restore --staged <file>..." to unstage)

new file: new.xml

А после уже коммитем:

git commit -m "add new.xml"

6. Отправить файл на внешний GitHub репозиторий.

git push

7. Отредактировать содержание файла “new.xml” - написать информацию о себе (ФИО, возраст, количество домашних животных, будущая желаемая зарплата). Всё написать в формате XML.

Используем команду для редакции файла:

nano new.xml

Заполняем наш файл формате XML:

<info>

<name>Afanasev Vasiliy Pavlovich</name>

<age>29</age>

<pets>6</pets>

<desired\_salary>40000</desired\_salary>

</info>

Сохраняем ctrl+s, закрываем ctrl+x.

8. Отправить изменения на внешний репозиторий.

git add .

git commit -m "update new.xml"

git push

9. Создать файл preferences.xml

touch preferences.xml

10. В файл preferences.xml добавить информацию о своих предпочтениях (Любимый фильм, любимый сериал, любимая еда, любимое время года, сторона которую хотели бы посетить) в формате XML.

nano preferences.xml

Заполняем наш файл формате XML:

<preferences>

<favorite\_movie>The Pianist</favorite\_movie>

<favorite\_tv\_show>Game of Thrones</favorite\_tv\_show>

<favorite\_food>Borsch</favorite\_food>

<favorite\_season>Spring</favorite\_season>

<desired\_travel\_destination>China</desired\_travel\_destination>

</preferences>

Сохраняем ctrl+s, закрываем ctrl+x.

11. Создать файл skills.xml добавить информацию о скиллах которые будут изучены на курсе в формате XML

touch skills.xml

nano skills.xml

Заполняем наш файл формате XML:

<skills>

<name>Базовая теория</name>

<name>Что такое клиент-серверная архитектура</name>

<name>HTTP Методы запросов на сервер</name>

<name>Коды ответов HTTP сервера</name>

<name>Структуры HTTP запросов и ответов</name>

<name>Что такое JSON, XML. Их структура</name>

<name>Тестирование API через Postman</name>

<name>Снятие и чтение логов c внешнего сервера</name>

<name>Снифинг http web трафика через Charles и Fiddler</name>

<name>Dev Tools веб браузеров (Google Chrome, FireFox)</name>

<name>Как работает VPN, Зачем нужен VPN, Как использовать VPN, Варианты инструментов VPN</name>

<name>Мобильное тестирование</name>

<name>Особенность iOS, Android, гайдлайны</name>

<name>Сборка iOS приложений на XCode</name>

<name>Сборка Android приложений на Android Studio</name>

<name>ADB (управление андройд девайсами)</name>

<name>Настройка прокси и vpn на iOS и Android</name>

<name>Перехват (сниффинг) мобильного трафика через Charles и Fiddler на iOS и Android</name>

<name>Командная строка Linux, Копирование файлов на серверах без графического интерфейса, Создание файлов на серверах без графического интерфейса, Просмотр файлов на серверах без графического интерфейса</name>

<name>Основы bash скриптинг, автоматизация рутинных задач на сервере</name>

<name>Доступ к удалённым серверам</name>

<name>Основы SQL (Create, Delete, Drop, Insert Into, Select, From, Where, Join)</name>

<name>База данных Postgres (установка, настройка и использование)</name>

<name>Нереляционная база данных Redis (установка, настройка и использование)</name>

<name>Нагрузочное тестирование в Jmeter</name>

<name>Методология разработки Scrum</name>

<name>Python. Изучение основ. Создание клиент серверного приложения</name>

</skills>

Сохраняем ctrl+s, закрываем ctrl+x.

12. Сделать коммит в одну строку.

git add .

git commit -m "add skill.xml, preferences.xml" skills.xml preferences.xml

13. Отправить сразу 2 файла на внешний репозиторий.

Git push

14. На веб интерфейсе создать файл bug\_report.xml.

Заходим в наш репозиторий на github.com, жмем “Add file” далее “Create new file” и вбиваем в строку “ bug\_report.xml ”.

15. Сделать Commit changes (сохранить) изменения на веб интерфейсе.

Сверху можем оставить наш комментарий и жмем внизу “Commit new file”.

16. На веб интерфейсе модифицировать файл bug\_report.xml, добавить баг репорт в формате XML.

Заходим в наш репозиторий на github.com, выбираем файл “ bug\_report.xml”

Заполняем:

<bug\_report>

</title>Навигация по содержанию во вкладке 'Terms and Policies' при нажатии на заголовок пролистывает в позицию ниже требуемого</title>

<enviroment>Win 11 Pro, Chrome Version 112</enviroment>

<severity>Minor<severity>

<steps>

<steps number="1">Перейти на сайт website.com</steps>

<steps number="2">Опуститься в конец сайта</steps>

<steps number="3">В правом нижнем углу выбрать 'Terms and Policies'</steps>

<steps number="4">Выбрать из содержания 'SV Investments Limited (CySEC)'</steps>

<steps number="5">Аналогично со следующими элементами содержания</steps>

</steps>

<expected result>При выборе заголовка из содержания 'SV Investments Limited (CySEC)', страница пролистывается на позицую заголовка, аналогично и с другими заголовками</expected result>

<Actual Result>При выборе заголовка из содержания 'SV Investments Limited (CySEC)', страница пролистывается ниже заголовка, аналогично и с другими заголовками</Actual Result>

<License>SCB, CYSEC, FCA</License>

</bug\_report>

17. Сделать Commit changes (сохранить) изменения на веб интерфейсе

Сверху можем оставить наш комментарий и жмем внизу “Commit new file”.

18. Синхронизировать внешний и локальный репозиторий XML

Переходим в терминал Git Bash и прописываем:

git pull