JSON

1. Создать внешний репозиторий c названием JSON.

На github.com создаем новый репозиторий “JSON”.

2. Клонировать репозиторий JSON 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/ JSON.git

3. Внутри локального JSON создать файл “new.json”.

touch new.json

4. Добавить файл под гит.

git add new.json

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.json

А после уже коммитем:

git commit -m "add new.json"

6. Отправить файл на внешний GitHub репозиторий.

git push

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

Используем команду для редакции файла:

nano new.json

Заполняем наш файл формате JSON:

{

"name": "Afanasev Vasiliy Pavlovich",

"ege": 29,

"pets": 6,

"desired\_salary": 40000

}

Сохраняем ctrl+s, закрываем ctrl+x.

8. Отправить изменения на внешний репозиторий.

git add .

git commit -m "update new.json"

git push

9. Создать файл preferences.json

touch preferences.json

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

nano preferences.json

Заполняем наш файл формате JSON:

{

"favorite\_movie": "The Pianist"

"favorite\_tv\_show": "Game of Thrones"

"favorite\_fodd": "Borsch"

"favorite\_season": "Spring"

"desired\_travel\_destination": "China"

}

Сохраняем ctrl+s, закрываем ctrl+x.

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

touch skills.json

nano skills.json

Заполняем наш файл формате JSON:

{

"skills":[

{

"title": "Базовая теория, SDLC, STLC"

},

{

"title": "Что такое клиент-серверная архитектура"

},

{

"title": "HTTP Методы запросов на сервер"

},

{

"title": "Коды ответов HTTP сервера"

},

{

"title": "Структуры HTTP запросов и ответов"

},

{

"title": "Что такое JSON, XML. Их структура"

},

{

"title": "Тестирование API через Postman"

},

{

"title": "Снятие и чтение логов c внешнего сервера"

},

{

"title": "Снифинг http web трафика через Charles и Fiddler"

},

{

"title": "Dev Tools веб браузеров (Google Chrome, FireFox)"

},

{

"title": "Как работает VPN, Зачем нужен VPN, Как использовать VPN, Варианты инструментов VPN"

},

{

"title": "Мобильное тестирование"

},

{

"title": "Особенность iOS, Android, гайдлайны"

},

{

"title": "Сборка iOS приложений на XCode"

},

{

"title":  "Сборка Android приложений на Android Studio"

},

{

"title": "ADB (управление андройд девайсами)"

},

{

"title": "Настройка прокси и vpn на iOS и Android"

},

{

"title": "Перехват (сниффинг) мобильного трафика через Charles и Fiddler на iOS и Android"

},

{

"title": "Командная строка Linux, Копирование файлов на серверах без графического интерфейса, Создание файлов на серверах без графического интерфейса, Просмотр файлов на серверах без графического интерфейса"

},

{

"title": "Основы bash скриптинг, автоматизация рутинных задач на сервере"

},

{

"title": "Доступ к удалённым серверам"

},

{

"title": "Основы SQL (Create, Delete, Drop, Insert Into, Select, From, Where, Join)"

},

{

"title": "База данных Postgres (установка, настройка и использование)"

},

{

"title": "Нереляционная база данных Redis (установка, настройка и использование)"

},

{

"title": "Нагрузочное тестирование в Jmeter"

},

{

"title": "Методология разработки Scrum"

},

{

"title": "Python. Изучение основ. Создание клиент серверного приложения"

}

]

}

Сохраняем ctrl+s, закрываем ctrl+x.

12. Отправить сразу 2 файла на внешний репозиторий.

git add .

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

Git push

13. На веб интерфейсе создать файл bug\_report.json.

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

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

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

15. На веб интерфейсе модифицировать файл bug\_report.json, добавить баг репорт в формате JSON.

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

Заполняем:

"title": "Навигация по содержанию во вкладке 'Terms and Policies' при нажатии на заголовок пролистывает в позицию ниже требуемого",

"enviroment": "Win 11 Pro, Chrome Version 112",

"severity": "Minor",

"steps": [

"1. Перейти на сайт website.com",

"2. Опуститься в конец сайта",

"3. В правом нижнем углу выбрать 'Terms and Policies'",

"4. Выбрать из содержания 'SV Investments Limited (CySEC)'",

"5. Аналогично со следующими элементами содержания"

  ],

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

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

"License": "SCB, CYSEC, FCA",

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

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

17. Синхронизировать внешний и локальный репозиторий JSON

Переходим в терминал Git Bash и прописываем:

git pull