**GitHub. HW\_1**

Для выполнения задания у вас должен быть установлен для Windows - Git Bash.

Создан аккаунт в GitHub

Все шаги сценария выполняйте в терминале Git Push, Terminal, в папке под гитом.

Как отправить ДЗ на проверку.

1. Создайте текстовый файл как в первом ДЗ по Terminal.

2. Сценарий перенесите в этот файл.

3. Напротив каждого действия - напишите команду в Git Bash

Файл со сценарием и ссылку на свой гит GitHub отправляйте менторам на проверку.

**JSON**

4. Создать внешний репозиторий c названием JSON.

**- В аккаунте GitHub в меню Repositories нажать кнопку New, прописать имя “JSON”, проставить необходимые настройки, нажать Create repository.**

**- Создание SSH ключа:**

**В GitBush ввести команду ssh-keygen -t rsa -C "user.name@mail.domain" Сохранить сгенерированный ключ в предложенном файле (C:\Users\%username%\.ssh\id\_rsa.pub). Задать пароль. Открыть файл id\_rsa.pub в текстовом редакторе, скопировать текст.**

**В аккаунте GitHub->Settings->SSH and GPG keys->New ssh key->вставить ключ, сохранить.**

**Открыть страницу репозитория-> скопировать “ссылка ssh”**

5. Клонировать репозиторий JSON на локальный компьютер.

**- git clone <ssh-ссылка репозитория>**

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

**- touch new.json**

7. Добавить файл под гит.

**- git add new.json**

8. Закоммитить файл.

**- git commit -m ‘new file’**

9. Отправить файл на внешний GitHub репозиторий.

**- git push**

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

**- vim new.json:**

**{“person”:**

**{**

**“name”: “Ksenia”,**

**“age”: 33,**

**“count\_pet”: 1,**

**“salary”: 1000**

**}**

**}**

*\*vim:Режим ввода i (insert)*

*Возврат в обычный режим ESC*

*:q! - выйти без сохранения;*

*:wq - записать файл и выйти;*

11. Отправить изменения на внешний репозиторий.

**- git add new.json**

**- git commit -m 'new file edited'**

**- git push**

12. Создать файл preferences.json

**- touch preferences.json**

13. 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.

**- vim preferences.json**

**{**

**"film": "ocean",**

**"serial": "game of the throns",**

**"food": "sushi",**

**"season": "summer",**

**"country": "serbia"**

**}**

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

**- touch skills.json**

**- vim skills.json**

*\* одной командой: cat >> skills.json*

**{"Skills":**

**[**

**"Theory of testing",**

**"Knowledge about testing tools",**

**"Knowledge about Python coding"**

**]**

**}**

15. Отправить сразу 2 файла на внешний репозиторий.

**- git add .**

**- git commit -am 'add two file'**

**- git push**

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

**- в меню репозитория на GitHub выбрать Add file \ Create new file**

**- прописать название файла bug\_report.json**

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

**- нажать кнопку “Commit changes…”, в поле ввода “Extended description” ввести описание коммита,** **нажать кнопку “Commit changes”.**

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

**- открыть файл bug\_report.json и нажать иконку Edit this file**

**- добавить содержимое файла:**

**{**

**"ID": "C001"**

**"title": "the notification is not displayed",**

**"severity": "major",**

**"priority": "high",**

**"precondition": "open website",**

**"step\_to\_reproduce":**

**[**

**"1. log in to your personal account",**

**"2. open the personal data tab"**

**]**

**"actual result": "the notification is not displayed",**

**"expected result": "the notification is displayed"**

**}**

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

**- нажать кнопку “Commit changes…”, в поле ввода “Extended description” ввести описание коммита,** **нажать кнопку “Commit changes”.**

20. Синхронизировать внешний и локальный репозиторий JSON

**в Git Bash прописать команду:**

**- git pull**

**XML**

21. Создать внешний репозиторий c названием XML.

**- В аккаунте GitHub в меню Repositories нажать кнопку New, прописать имя “XML”, проставить необходимые настройки, нажать Create repository.**

**- Создание SSH ключа:**

**В GitBush ввести команду ssh-keygen -t rsa -C "user.name@mail.domain" Сохранить сгенерированный ключ в предложенном файле (C:\Users\%username%\.ssh\id\_rsa.pub). Задать пароль. Открыть файл id\_rsa.pub в текстовом редакторе, скопировать текст.**

**В аккаунте GitHub->Settings->SSH and GPG keys->New ssh key->вставить ключ, сохранить.**

**Открыть страницу репозитория-> скопировать “ссылка ssh”**

22. Клонировать репозиторий XML на локальный компьютер.

**- git clone <ssh-ссылка репозитория>**

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

**- touch new.xml**

24. Добавить файл под гит.

**- git add new.xml**

25. Закоммитить файл.

**- git commit -m ‘new file’**

26. Отправить файл на внешний GitHub репозиторий.

**- git push**

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

**- vim new.xml**

**<?xml version="1.0" encoding="UTF-8"?>**

**<person>**

**<name>Ksenia</name>**

**<age>33</age>**

**<count\_pet>1</count\_pet>**

**<future\_salary>1000</future\_salary>**

**</person>**

28. Отправить изменения на внешний репозиторий.

**- git add new.xml**

**- git commit -m 'new file edited'**

**- git push**

29. Создать файл preferences.xml

**- touch preferences.xml**

30. 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.

**- vim preferences.xml**

**<?xml version="1.0" encoding="UTF-8"?>**

**<preferences>**

**<film>Ocean</film>**

**<serial>Game of the thrones</serial>**

**<food>sushi</food>**

**<season>summer</season>**

**<country>Serbia</country>**

**</preferences>**

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

**- touch skills.xml**

**- vim skills.xml**

*\* одной командой: cat >> skills.xml*

**<?xml version="1.0" encoding="UTF-8"?>**

**<skills>**

**<skill\_1>json</skill\_1>**

**<skill\_2>sql</skill\_2>**

**<skill\_3>git\_bush</skill\_3>**

**<skill\_4>git\_hub</skill\_4>**

**<skill\_5>devtools</skill\_5>**

**<skill\_6>js</skill\_6>**

**<skill\_7>theory</skill\_7>**

**</skills>**

32. Сделать коммит в одну строку.

**- git add . | git commit -m 'new files'**

33. Отправить сразу 2 файла на внешний репозиторий.

**- git push**

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

**- в меню репозитория на GitHub выбрать Add file \ Create new file**

**- прописать название файла bug\_report.xml**

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

**- нажать кнопку “Commit changes…”, в поле ввода “Extended description” ввести описание коммита,** **нажать кнопку “Commit changes”.**

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

**- открыть файл bug\_report.xml и нажать иконку Edit this file**

**- добавить содержимое файла:**

**<?xml version="1.0" encoding="UTF-8"?>**

**<Bug\_repport>**

**<ID>C001</ID>**

**<title>the notification is not displayed</title>**

**<severity>major</severity>**

**<priority>high</priority>**

**<precondition>open website</precondition>**

**<steps\_to\_reproduse>1. log in to your personal account**

**2. open the personal data tab**

**</steps\_to\_reproduse>**

**<actual-result>the notification is not displayed</actual-result>**

**<expected-result>the notification is displayed</expected-result>**

**</Dug\_repport>**

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

**- нажать кнопку “Commit changes…”, в поле ввода “Extended description” ввести описание коммита,** **нажать кнопку “Commit changes”.**

38. Синхронизировать внешний и локальный репозиторий XML

**в Git Bash прописать команду:**

**- git pull**

**TXT**

1. Создать внешний репозиторий c названием TXT.

**- В аккаунте GitHub в меню Repositories нажать кнопку New, прописать имя “TXT”, проставить необходимые настройки, нажать Create repository.**

**- Создание SSH ключа:**

**В GitBush ввести команду ssh-keygen -t rsa -C "user.name@mail.domain" Сохранить сгенерированный ключ в предложенном файле (C:\Users\%username%\.ssh\id\_rsa.pub). Задать пароль. Открыть файл id\_rsa.pub в текстовом редакторе, скопировать текст.**

**В аккаунте GitHub->Settings->SSH and GPG keys->New ssh key->вставить ключ, сохранить.**

**Открыть страницу репозитория-> скопировать “ссылка ssh”**

2. Клонировать репозиторий TXT на локальный компьютер.

**- git clone <ssh-ссылка репозитория>**

3. Внутри локального TXT создать файл “new.txt”.

**- touch new.txt**

4. Добавить файл под гит.

**- git add new.txt**

5. Закоммитить файл.

**- git commit -m ‘new txt file’**

6. Отправить файл на внешний GitHub репозиторий.

**- git push**

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

**- vim new.txt**

**Name - Ksenia**

**Age - 33**

**Count pet - 1**

**Future salary - 1000**

8. Отправить изменения на внешний репозиторий.

**- git add new.txt**

**- git commit -m 'new file txt was created'**

**- git push**

9. Создать файл preferences.txt

**- touch preferences.txt**

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

**- vim preferences.txt**

**Film - Ocean**

**Serial - Game of the thrones**

**Food - sushi**

**Season - summer**

**Country - Serbia**

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

**- touch skills.txt**

**- vim skills.txt**

*\* одной командой: cat >> skills.txt*

**skill\_1 - json**

**skill\_2 - sql**

**skill\_3 - git\_bush**

**skill\_4 - git\_hub**

**skill\_5 - devtools**

**skill\_6 - js**

**skill\_7 - theory**

12. Сделать коммит в одну строку.

**- git add . | git commit -m 'new files'**

13. Отправить сразу 2 файла на внешний репозиторий.

**- git push**

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

**- в меню репозитория на GitHub выбрать Add file \ Create new file**

**- прописать название файла bug\_report.txt**

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

**- нажать кнопку “Commit changes…”, в поле ввода “Extended description” ввести описание коммита,** **нажать кнопку “Commit changes”.**

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

**- открыть файл bug\_report.txt и нажать иконку Edit this file**

**- добавить содержимое файла:**

**ID - C001**

**title - the notification is not displayed**

**severity - major**

**priority - high**

**precondition - open website**

**step\_to\_reproduce:**

**step\_1 - log in to your personal account**

**step\_2 - open the personal data tab**

**actual result - the notification is not displayed**

**expected result - the notification is displayed**

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

**- нажать кнопку “Commit changes…”, в поле ввода “Extended description” ввести описание коммита,** **нажать кнопку “Commit changes”.**

18. Синхронизировать внешний и локальный репозиторий TXT

**в Git Bash прописать команду:**

**- git pull**