**Лабораторная работа 1 Собственные репозитории**

В этом уроке мы научимся создавать свои собственные репозитории. *Настоятельно рекомендую представленные команды в этой лабораторной вбивать руками, этим вы убережете себя от многих ошибок связанных с распознаванием кавычек и тире, при работе с терминалом*  
1) Создайте учетную запись на сайте [https://bitbucket.org](https://bitbucket.org/) или авторизуйтесь под своей, если такая имеется  
2) Возможно вам потребуется рассказать о вашем опыте, в таком случае заполните как на рисунке

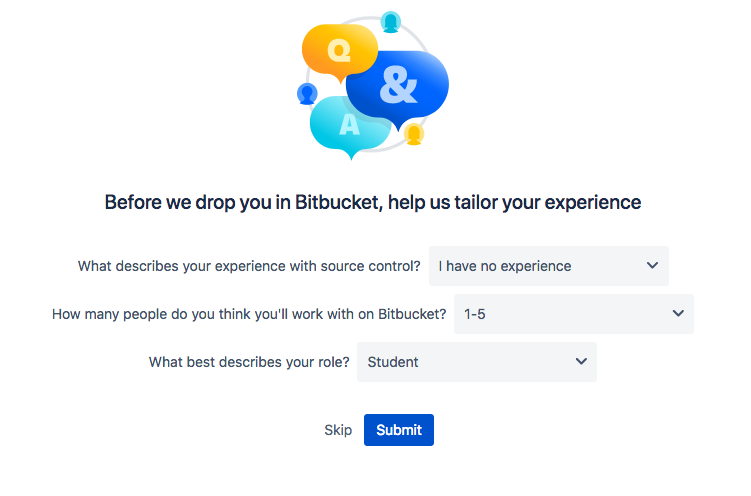


Рисунок 1

3) Создадим репозиторий (хранилище), на странице <https://bitbucket.org/dashboard/overview> во владке **Your work** нажмем на кнопку **Create repository**, и укажем имя хранилища в поле **Repository name** – “learn-01”, в **Access level** мы можем указать тип репозитория (приватный или публичный), все остальные настройки мы можем оставить без изменений  
4) На своем компьютере создайте каталог **learn-01** и откройте его в **Терминале**(GIT Bash) и введите последовательно команды:  
**git init** (создаем пустой репозиторий)

**git remote add origin https://\_путь\_до\_репозитория\_на\_битбакете\_(https)\_**

(привязываем репозиторий к битбакету)

путь до репозитория можно взять нажав на кнопку **Clone** на основной странице репозитория (см. рисунок 2)

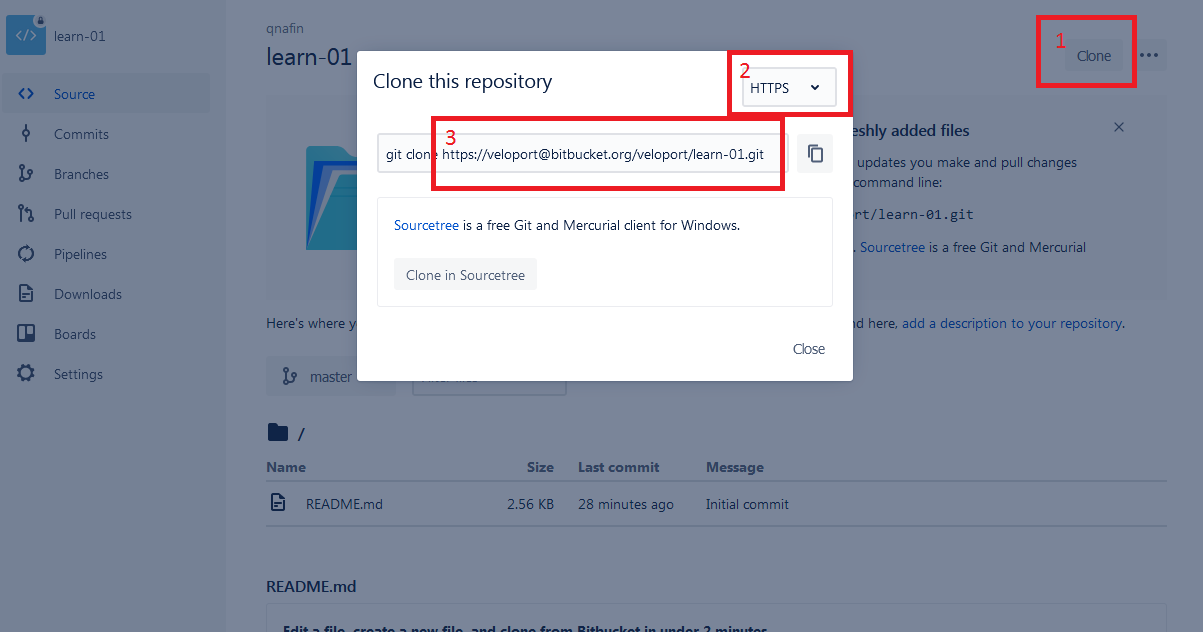


Рисунок 2

5) Теперь, когда ваша среда разработки (локальный репозиторий) связана с глобальным репозиторием на bitbucket, выполните команду

**git pull origin master** чтобы подтянуть первоначальные файлы, с глобального репозитория. Создайте в своем каталоге **learn-01**, произвольные кол-во **текстовых** **файлов** (рекомендую в формате .txt) и **папок**, 2 – 3 будет достаточно.

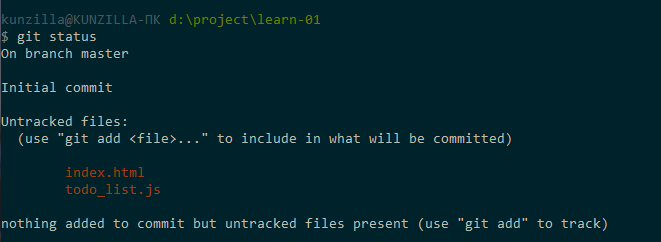
В терминале введем команду **git status**, которая покажет какие файлы и папки были созданы, либо изменены

Рисунок 3

Добавим изменения в наш репозиторий введя команду **git add** -**A**, ключом **-А** (all)мы говорим, что необходимо добавить все файлы, в противном случае нам бы пришлось добавлять каждый файл построчно.

Зафиксируем и опишем наши изменения командой **git commit -m «Добавлены файлы» -a**

И отправим в наш глобальный репозиторий расположенный на bitbucket, используя команду **git push origin master  
ВАЖНО!** При попытке отправить наши файлы командой **git push** в репозиторий, может возникнуть ошибка из-за того, что наш локальный репозиторий отличается от глобального, тогда нам необходимо будет выполнить команду

**git pull origin master,** чтобы подтянуть изменяя с глобального и повторно отправить командами

**git commit -m “merge file” -a**

**git push origin master**

6) Если все было сделано верно в вашем репозитории должны появиться ваши файлы

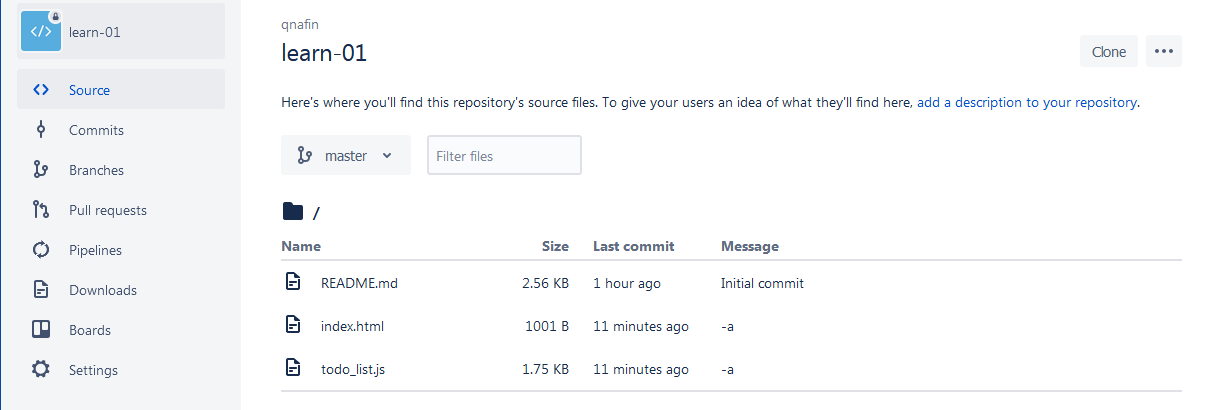


Рисунок 4

7) Историю изменений можно проследить во вкладке **Commits**

**Лабораторная 2 Командная работа в репозитории**

В этом уроке мы научимся использовать чужие репозитории, работать в команде и немножечко поиграемся в программистов. Для этого потребуется разделиться на команды по 2 человека.

*Настоятельно рекомендую представленные команды в этой лабораторной вбивать руками, этим вы убережете себя от многих ошибок связанных с распознаванием кавычек и тире, при работе с терминалом*  
1) Создайте новый репозиторий на bitbucket c именем learn-02, на своем компьютере создайте каталог **project**  
2) Находясь в директории **project** выполните команду

**git clone** [**https://bitbucket.org/veloport/todo\_list.git**](https://bitbucket.org/veloport/todo_list.git)данной командой вы подтяните мой репозиторий в свою папку. В вашей папке должна появиться новая с именем **todo\_list,** именно в ней мы и будем работать, зайдите через **терминал** в эту директорию и выполните команды

**git remote remove origin** это отвяжет ваш локальный репозиторий от моего

**git remote add origin** [**https://\_путь\_до\_вашего\_репозитория\_на\_битбакете\_(https)\_**](https://_путь_до_вашего_репозитория_на_битбакете_(https)_)

Добавьте любой произвольный текстовый файл и выполните команды:

**git pull origin master**

**git status**

**git add -A  
git commit -m “add file, fork learn-1”-a  
git push origin master**

Проверьте наличие файлов в вашем репозитории.

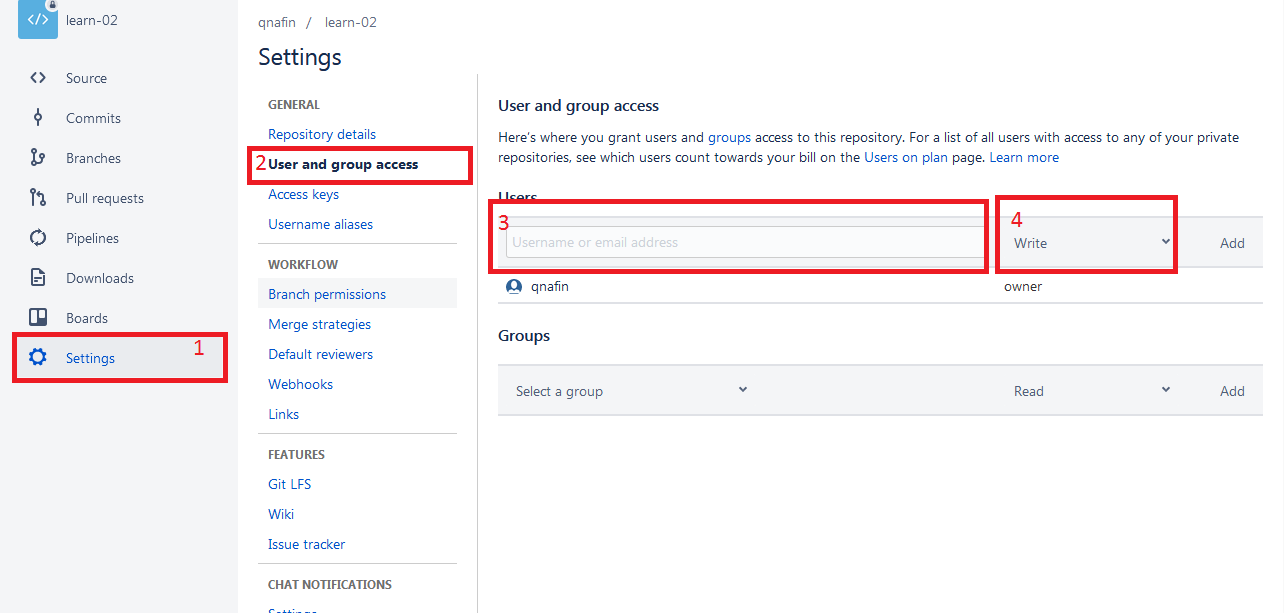
3) Теперь разделитесь на команды и 2 варианта, мы будем играть в командных разработчиков.  
Участник который взял на себя Роль 1, будет выступать в качестве владельца ПО и все изменения вы будите вносить в его репозиторий, за основу возьмите только что созданный репозиторий **learn-02** и добавьте того, кто выполняет Роль 2, обязательно установить для него права на Запись (Write), оповещение должно прийти к нему на почту

Рисунок 1

Для дальнейшей работы вам понадобится Браузер и текстовый редактор, например Notepad ++, SublimeText, блокнот и т.п.   
В качестве примера мы будем писать простенький Задачник на javascrip и html

|  |  |  |
| --- | --- | --- |
| Роль 1 | | Роль 2 |
| Создадим импровизированную базу данных 1) Создайте файл **tasks.json** со следующим содержанием:[  {  "id": 1,  "text": "Не выполнять свою работу",  "date": "13.02.2023"  },  {  "id": 2,  "text": "Придумать чем себя занять на 25 февраля",  "date": "13.02.2023"  }  ]  2) в файл **todo\_list.js** внесите следующие изменения, в класс **TodoList,** добавьте методы:  async fetchData () {  return await fetch('tasks.json')  .then(function(response) {  return response.json();  }).then(function(tasks) {  return tasks;  })  }  async componentDidMount() {  let tasks = await this.fetchData()  let result = []  Object.keys(tasks).map(function(objectKey, index) {  var value = tasks[objectKey];  result.push(value)  });    this.setState({items: result})    } | | После того как вам будет предоставлен доступ к репозиторию, он появится в списке ваших репозиториев на bitbucket, подтяните его, используя команду  **git clone** [**https://\_путь\_до\_репозитория**](https://_путь_до_репозитория)   1. Создайте файл **style.css** cо следующим содержанием:   body {  background-color: #ddd  }  li {  list-style-type: decimal;  }  .btn-primary { background-color: black }   1. Подключите файл **style.css** в файле **index.html** перед закрывающимся тегом **</body>,** добавьте код: <link rel="stylesheet" href="style.css" > |
| В терминале введите команду:  **git status** **git add -A  git commit -m “add db” -a git branch role\_1**  **git push origin role\_1** | | В терминале введите команду:  **git status** **git add -A  git commit -m “add style.css” -a git branch role\_2**  **git push origin role\_2** |
| Командой **git branch** каждый из вас создал свою ветку, в которой находятся только его изменения, зайдите в свой репозиторий на **bitbucket** во вкладку **Branches** и убедитесь в их наличии.  Подтяните ветки напарника командой **git fetch** Для перехода по веткам используйте команду **git checkout имя\_ветки**  Попереключайтесь между ветками role\_1, role\_2, master  И в заключении сольем наши ветки в главную **master,** для этого нам необходимо находиться в той ветке, в которую будем вливать изменения | | |
| Объединим ветки role\_2, role\_1 в master Перейдите на Bitbucket во вкладке **Branches** выберите поочередно ветки и в выпадающем списке выберите **merge** | Дождитесь пока этот шаг закончит ваш напарник | |
| По завершению работ обоих участников, перейдите в ветку **master** и обновите локальный репозиторий **git checkout master git pull origin master**  В результате вы должны увидеть изменения обоих участников в ветке master,  **ВАЖНО!** Можно принудительно залить все изменения с глобального репозитория для этого используйте команду **git reset** --**hard ,** однако будьте осторожны все не сохраненные результаты будут утеряны. Если английский интерфейс в bitbucket вызывает у вас отторжение, вы можете поменять язык на русский в настройках профиля | | |