

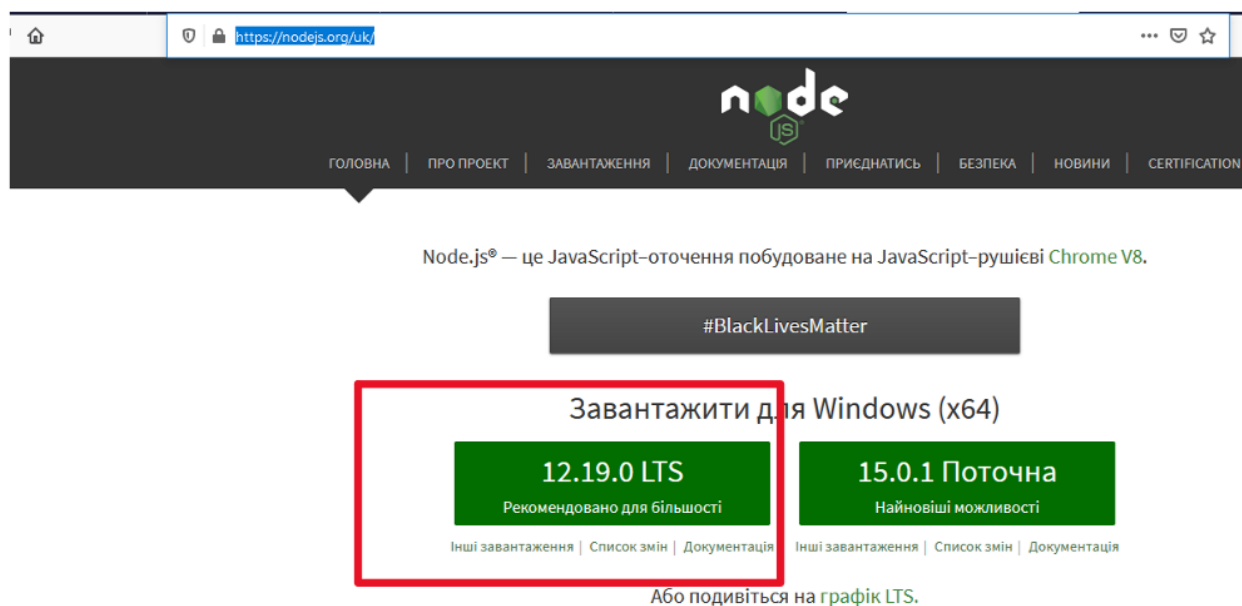
Пример первого приложения для проверки среды разработки

- [Установка необходимых компонент](#)
 - [Установка Node.js](#)
 - [Установка git](#)
 - [Установка Visual Studio Code](#)
- [Генерация шаблонного Node.js express приложения](#)
- [Поместить исходный код в локальный Git-репозиторий](#)
- [Отправить исходный код в remote Git-репозиторий](#)

Установка необходимых компонент

Установка Node.js

Для развёртывания Node.js необходимо скачать LTS (последнюю устойчивую) версию с сайта: [Node.js](https://nodejs.org/).
На данный момент там доступна 12

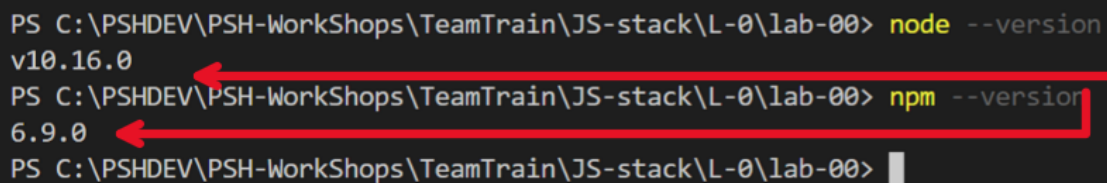


pic-1

Дальше соглашаемся со всем, что предлагает установка. Минимально проверить можно таким способом, результаты на pic-2 :

```
node --version
```

```
npm --version
```



```
PS C:\PSHDEV\PSH-WorkShops\TeamTrain\JS-stack\L-0\lab-00> node --version
v10.16.0
PS C:\PSHDEV\PSH-WorkShops\TeamTrain\JS-stack\L-0\lab-00> npm --version
6.9.0
PS C:\PSHDEV\PSH-WorkShops\TeamTrain\JS-stack\L-0\lab-00> 
```

pic-2

Запустить можно с консоли node.exe . Выход всегда CTRL + C

Установка git

У многих git уже есть. Для тех у кого нет Download and install using the following link <http://git-scm.com/download/win>

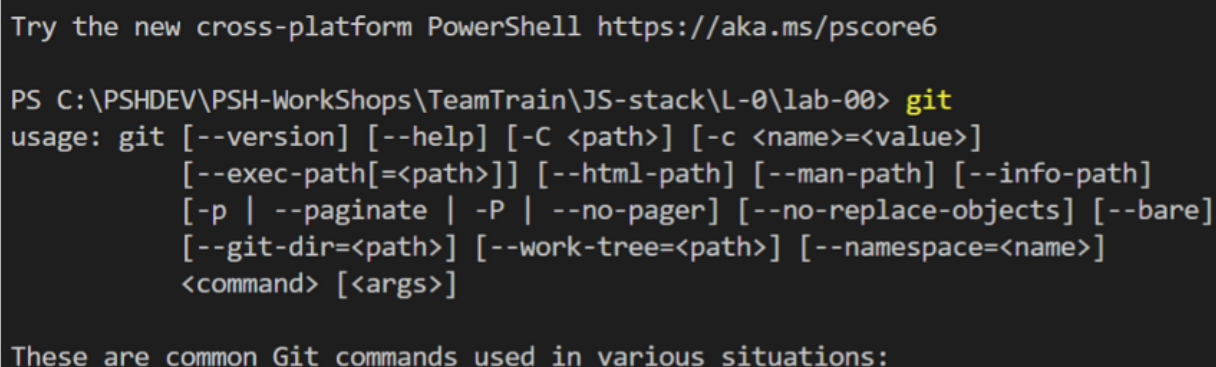
Выполнить все пожелания по умолчанию. Единственное, интеграция с windows не нужна. нам больше полезна git bash.

Для проверки установки git запустить с командной строки

```
git

git --version
```

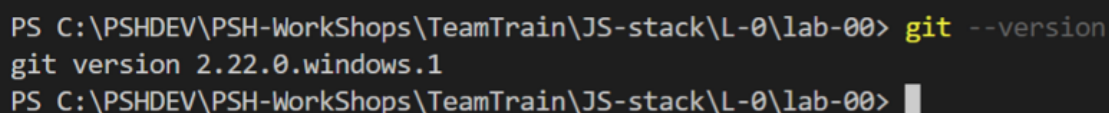
получим:



```
Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\PSHDEV\PSH-WorkShops\TeamTrain\JS-stack\L-0\lab-00> git
usage: git [--version] [--help] [-C <path>] [-c <name>=<value>]
      [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
      [-p | --paginate | -P | --no-pager] [--no-replace-objects] [--bare]
      [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
      <command> [<args>]

These are common Git commands used in various situations:
```

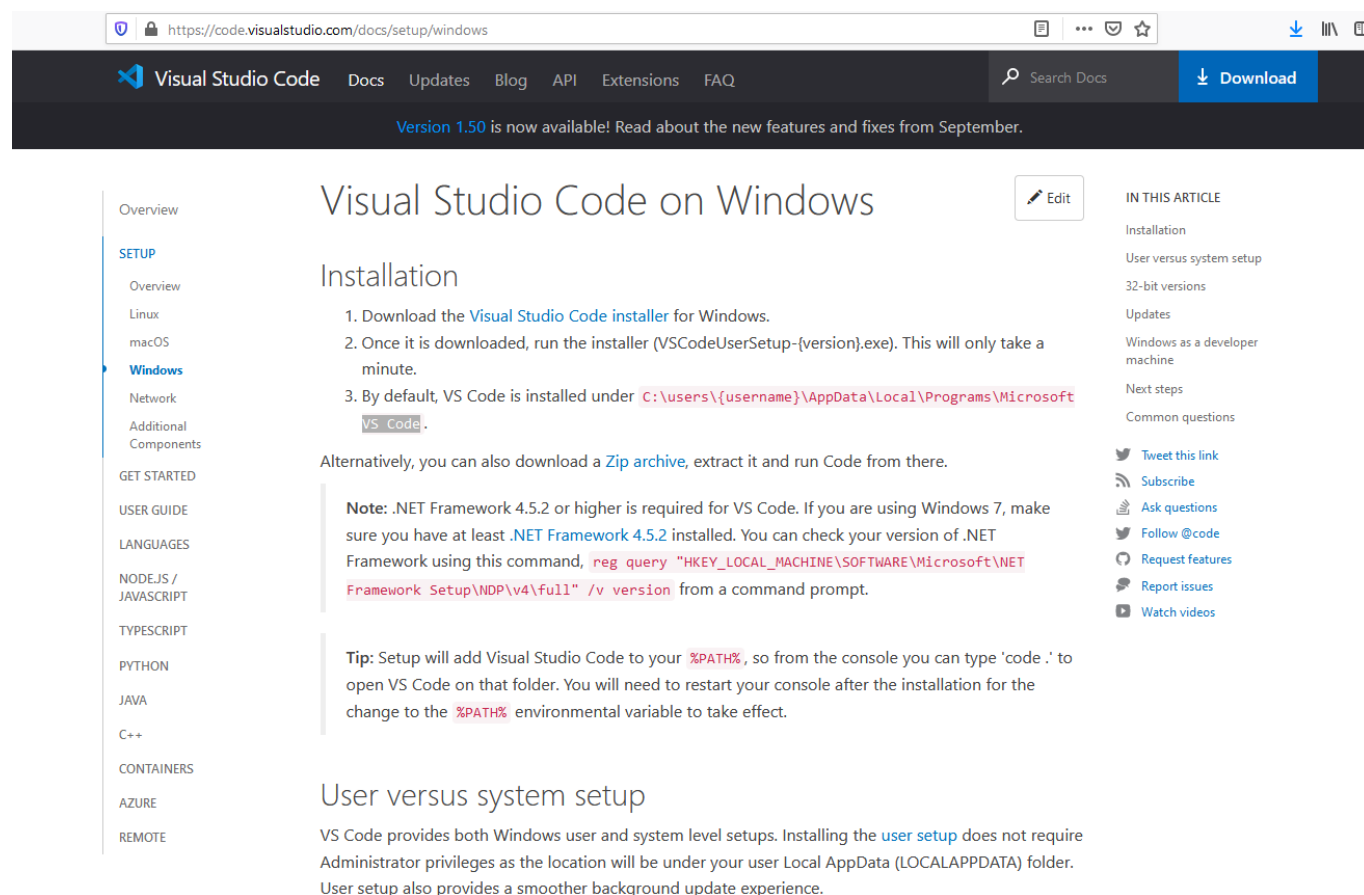


```
PS C:\PSHDEV\PSH-WorkShops\TeamTrain\JS-stack\L-0\lab-00> git --version
git version 2.22.0.windows.1
PS C:\PSHDEV\PSH-WorkShops\TeamTrain\JS-stack\L-0\lab-00> 
```

pic-5

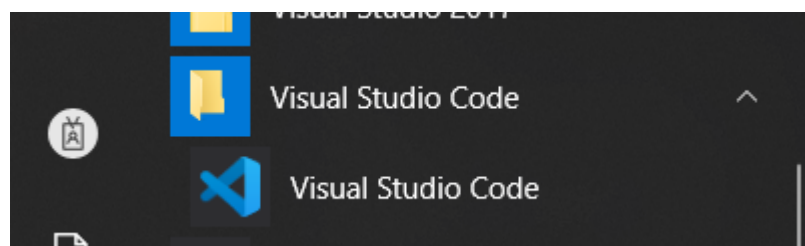
Установка Visual Studio Code

Скачать VS Code. Если будет спрашивать, выбирать для JavaScript.



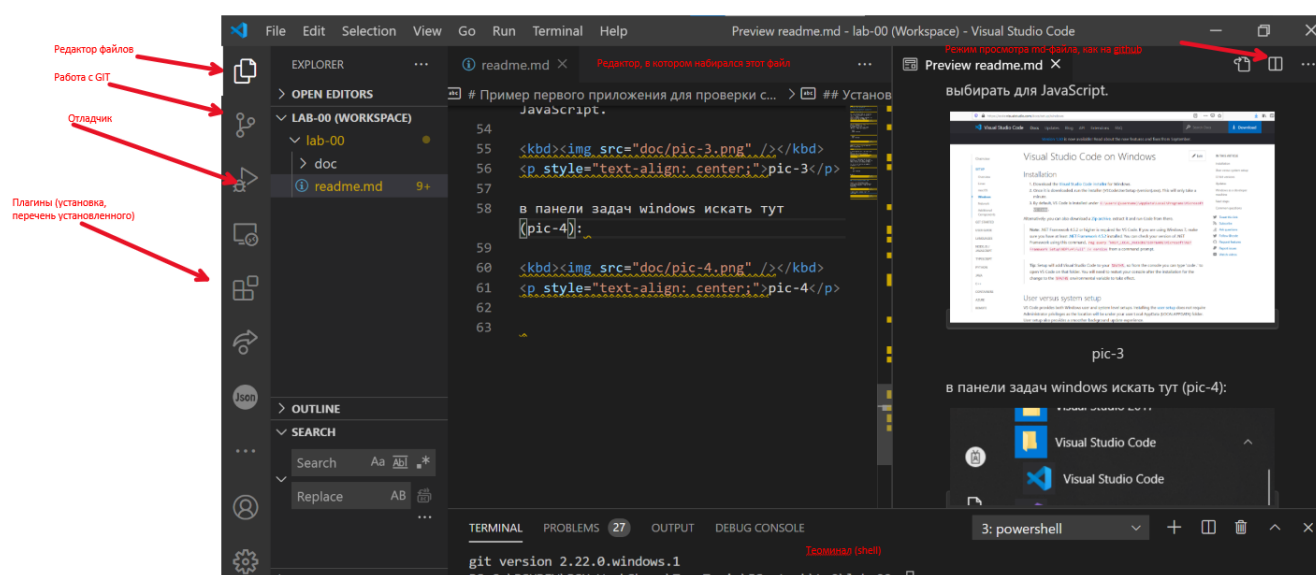
pic-3

в панели задач windows искать тут (pic-4):



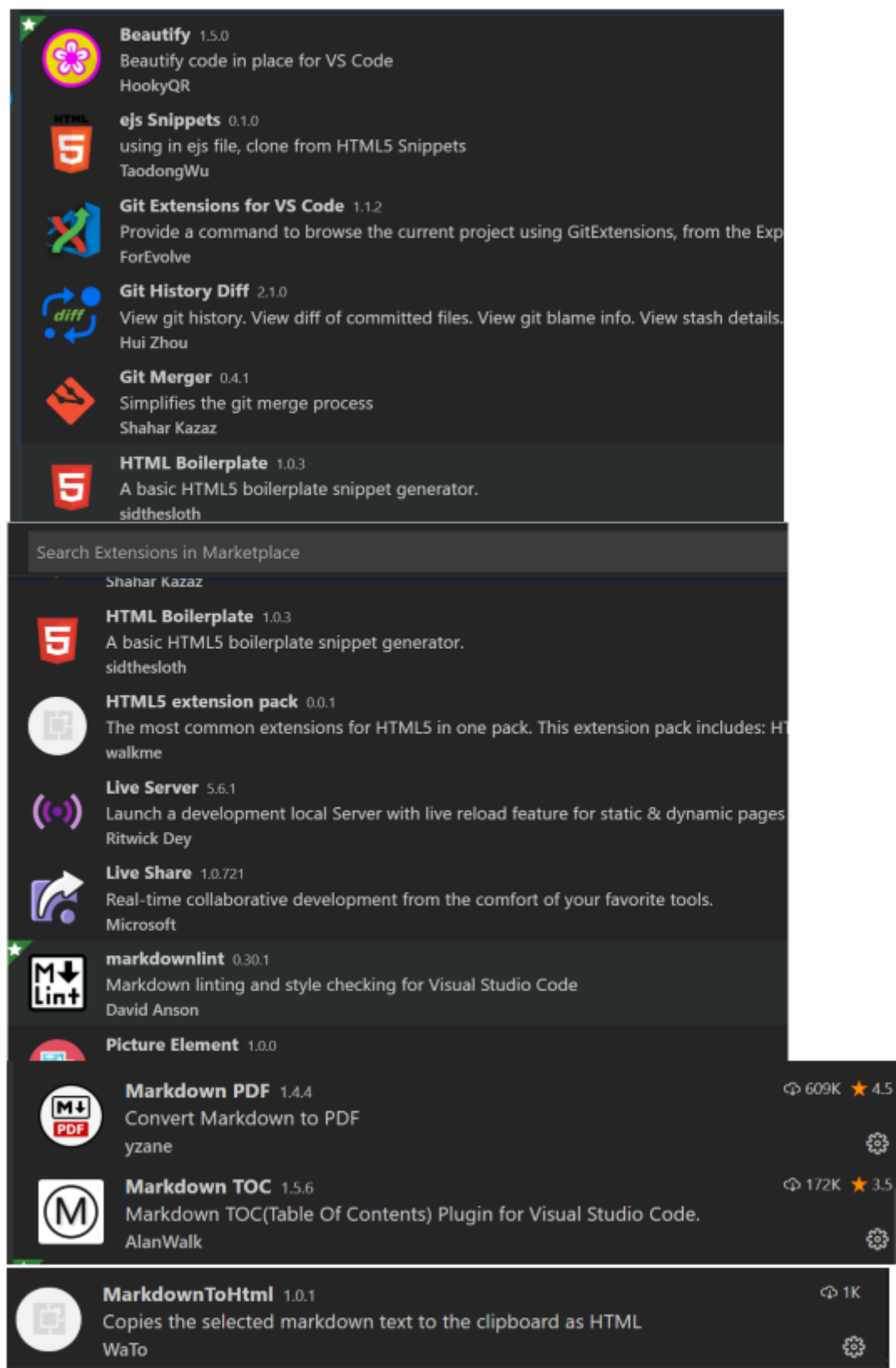
pic-4

Рабочий экран VS Code показан на pic-6



pic-6

Для комфортной работы желательно установить такие плагины pic-7:



pic-7

Во всяком случае плагины по работе с Git, Beautify для JS и по работе с MD - файлами - обязательны

Live Server и Live Share - это плагины, позволяющие расшарить ваш экран редактирования с коллегами для чтения или даже изменения кода. Авторизация по e-mail или github account (не github.ibm.com!).

Плагины ставятся в C:/users//.vscode и все написаны на Node.js

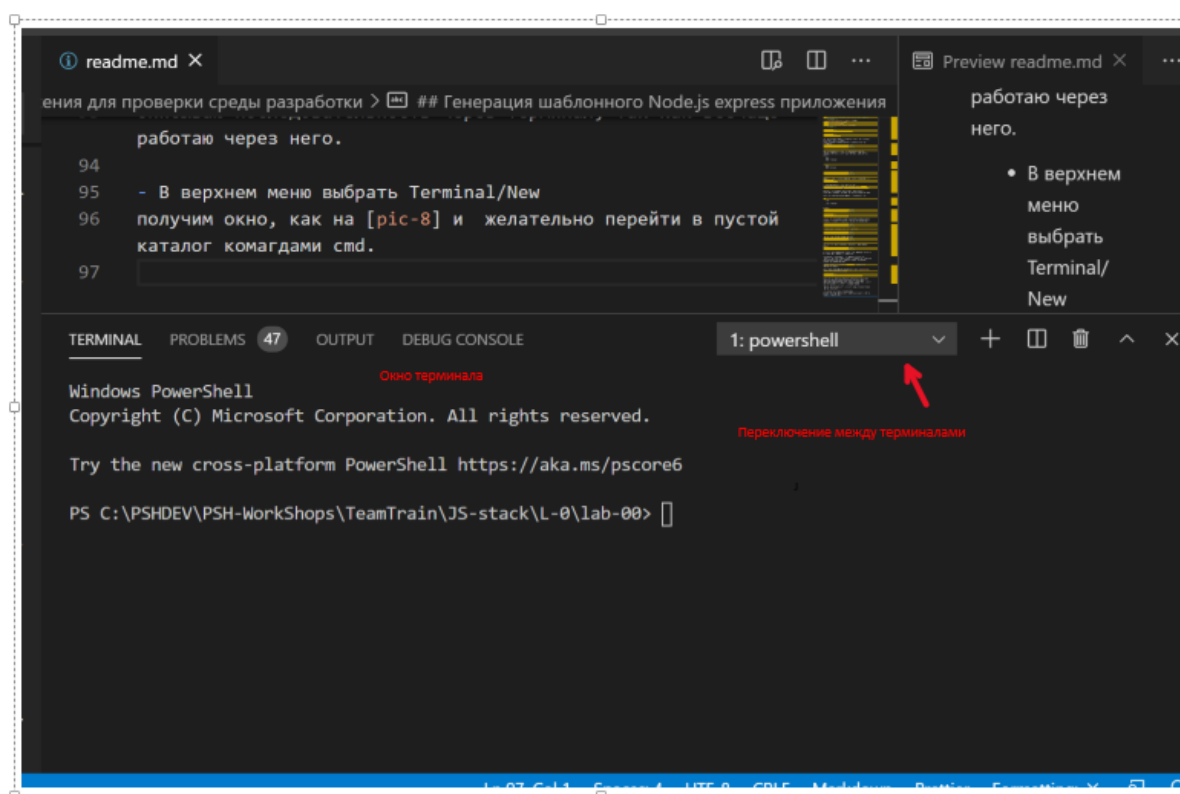
Да, а сама среда написана на Node.js фреймворке [electron](#)

Генерация шаблонного Node.js express приложения

Для проверки результата установки компонентов сгенерируем шаблонное Node.js express приложение. Для генерации используется последовательность команд. Можно с far, можно с cmd или winshell а можно с терминала VSCode.

Описываю последовательность через терминал, так как всеяче работаю через него.

- В верхнем меню выбрать Terminal/New получим окно, как на [pic-8] и желательно перейти в пустой каталог комагдами cmd.



pic-8

- С помощью менеджера пакетов npm установим глобально генератор express приложения (npm, это типа агалог maven для java)

```
npm install -g express-generator
```

На pic-9 показан результат работы команды, с той разницей, что у меня только обновление одного пакета, а там где его нет - там будет установка пакетов. Каталог установке виден на картинке

```
TERMINAL  PROBLEMS  54  OUTPUT  DEBUG CONSOLE  1: powershell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\PSHDEV\PSH-WorkShops\TeamTrain\JS-stack\L-0\lab-00> npm install -g express-generator
C:\Users\PavloShcherbukha\AppData\Roaming\npm\express -> C:\Users\PavloShcherbukha\AppData\Roaming\npm\node_
modules\express-generator\bin\express-cli.js
+ express-generator@4.16.1
updated 1 package in 0.977s
PS C:\PSHDEV\PSH-WorkShops\TeamTrain\JS-stack\L-0\lab-00> 
```

pic-9

- генерируем непосредственно приложение

Сгенерируем тестовое приложение testapp1 с генератором html страничек на стороне сервера [pug](#).

```
express testapp1 --view=pug
```

результат генерации показан ниже:

```
PS C:\PSHDEV\PSH-WorkShops\TeamTrain\JS-stack\L-0\lab-00> express testapp1 --
view=pug
```

```
create : testapp1\
create : testapp1\public\
create : testapp1\public\javascripts\
create : testapp1\public\images\
create : testapp1\public\stylesheets\
create : testapp1\public\stylesheets\style.css
create : testapp1\routes\
create : testapp1\routes\index.js
create : testapp1\routes\users.js
create : testapp1\views\
create : testapp1\views\error.pug
create : testapp1\views\index.pug
create : testapp1\views\layout.pug
create : testapp1\app.js
create : testapp1\package.json
create : testapp1\bin\
create : testapp1\bin\www
```

```
change directory:
> cd testapp1
```

```
install dependencies:
> npm install
```

```
run the app:
```

```
> SET DEBUG=testapp1:* & npm start
```

```
PS C:\PSHDEV\PSH-WorkShops\TeamTrain\JS-stack\L-0\lab-00>
```

Команда `dir` выводит список каталогов:

```
PS C:\PSHDEV\PSH-WorkShops\TeamTrain\JS-stack\L-0\lab-00> dir
```

```
Directory: C:\PSHDEV\PSH-WorkShops\TeamTrain\JS-stack\L-0\lab-00
```

Mode	LastWriteTime	Length	Name
d----	23.10.2020 13:47		doc
d----	23.10.2020 13:51		testapp1
-a----	23.10.2020 13:53	6391	readme.md

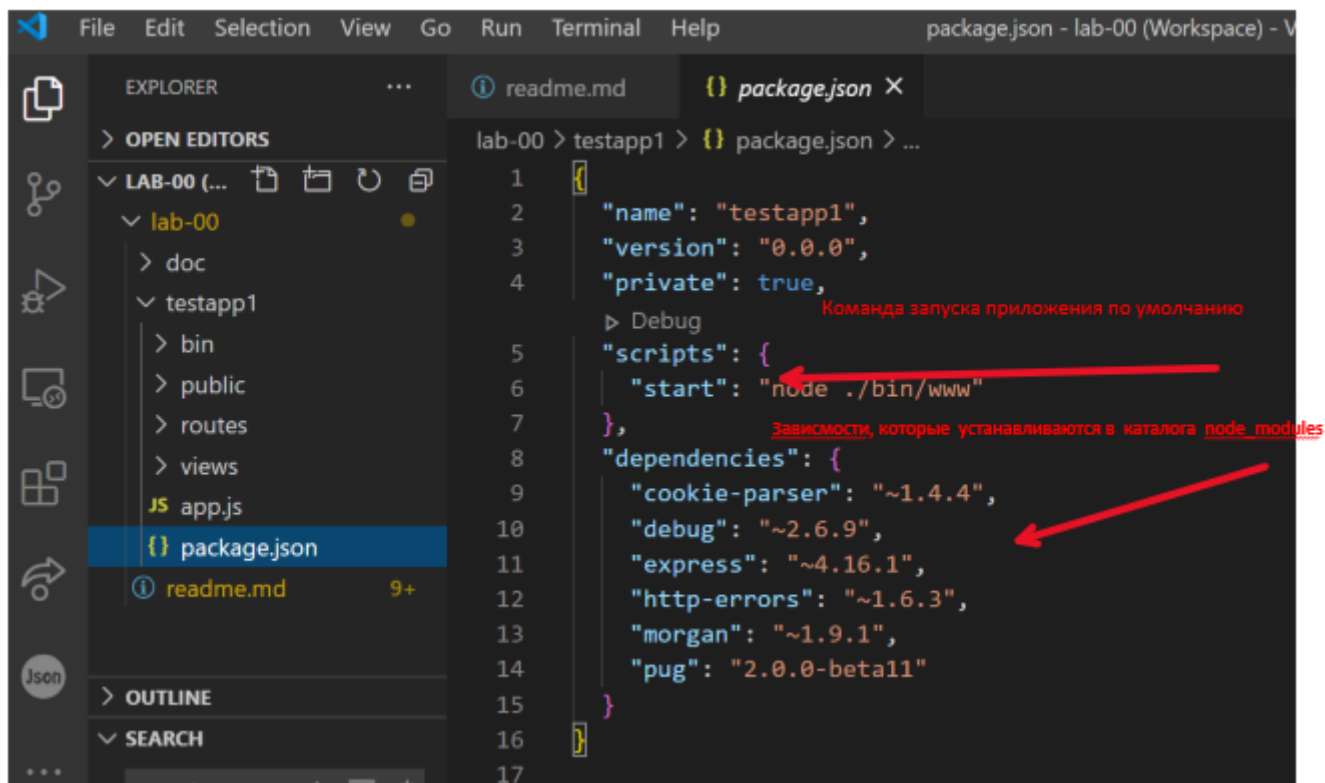
```
PS C:\PSHDEV\PSH-WorkShops\TeamTrain\JS-stack\L-0\lab-00> █
```

pic-10

В списке видим новый каталог `testapp1`.

- Скачиваем зависимости, необходимые для работы

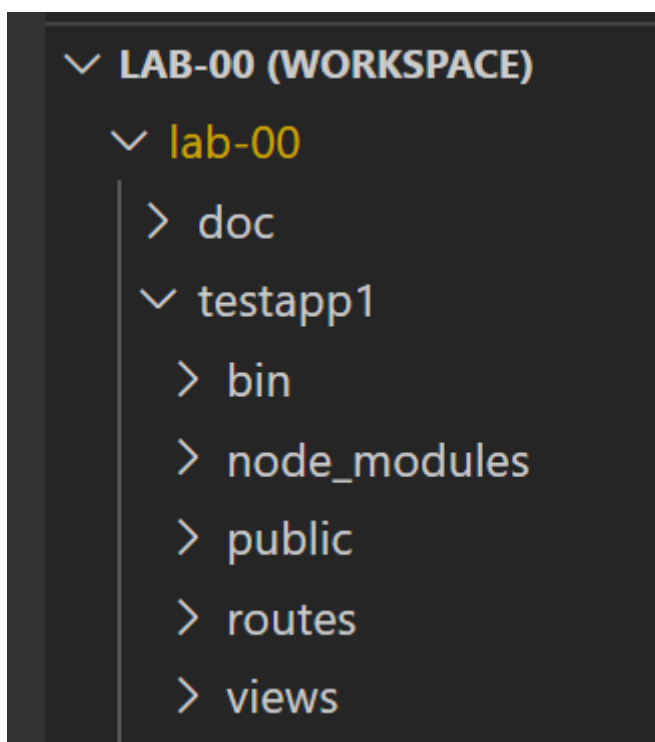
ЗависимостиЮ необходимые для работы описаны в файлах: `testapp1.package.json` [pic-11]



pic-11

Установка зависимостей выполняется командой менеджера пакетов npm

```
cd testapp1
npm install
```



В каталоге testapp1 появится node_modules

pic-12

- выполнить запуск приложения локально

Типичная команда запуска приложения

```
npm start
```

что аналогично `node ./bin/www`

после этого у вас стартует web приложение по порту 3000. Смотриите файл `./bin/www.js`, строка 15

```
/**
 * Get port from environment and store in Express.
 */

var port = normalizePort(process.env.PORT || '3000');
app.set('port', port);

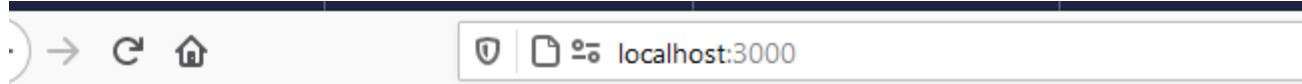
/**
 * Create HTTP server.
 */

var server = http.createServer(app);

/**
 * Listen on provided port, on all network interfaces.
 */

server.listen(port);
server.on('error', onError);
server.on('listening', onListening);
```

Переход по адресу `http://localhost:3000/` в browser выведет страничку



Express

Welcome to Express

pic-13

остановить приложение: CTRL-C в терминале

Поместить исходный код в локальный Git-репозиторий

Исходный код сначала помещается в локальный Git-репозиторий. Потом отправляется в удаленный (remote) репозиторий (github, gitlab,)

Не все файлы помещаются в git репозиторий. В git не пишутся логи, файлы конфигурации (особенно с API-KEY, паролями....) и node_modules. Для этого в корне каталога testapp1 необходимо создать файл **.gitignore**

```
node_modules
*.log
```

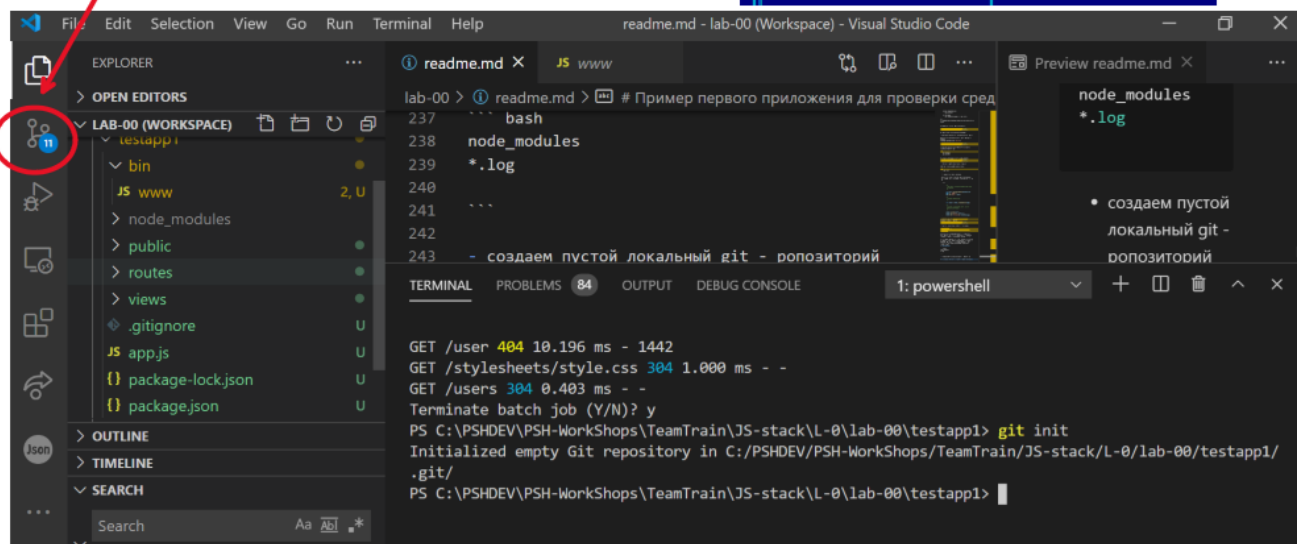
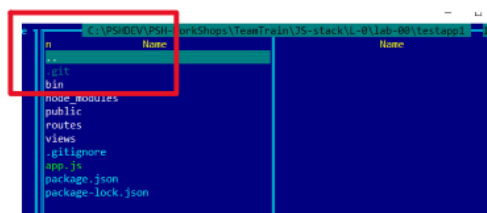
- создаем пустой локальный git - репозиторий

В терминале выполнить команду в папке testapp1

```
git init
```

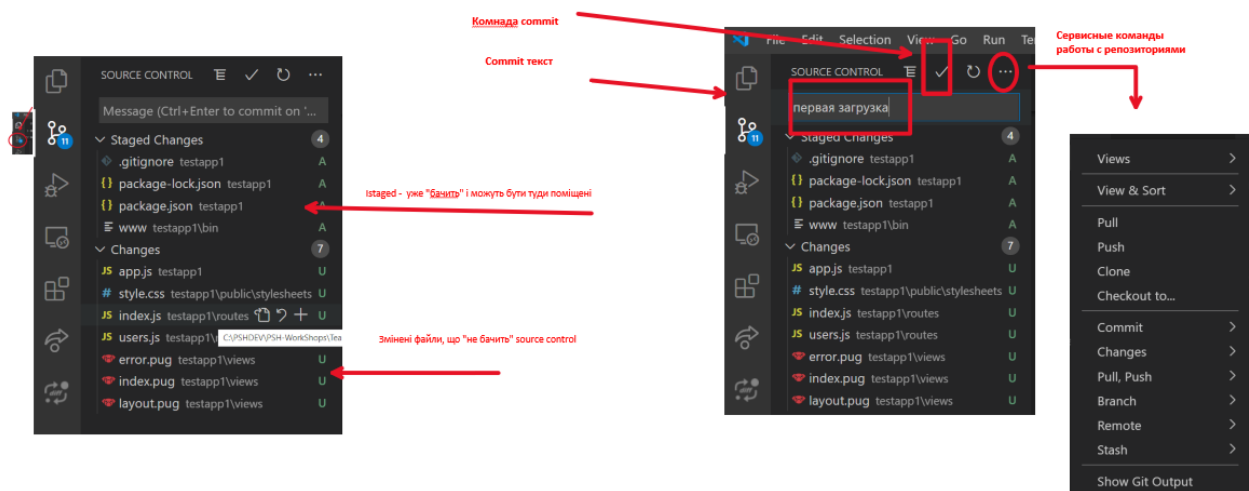
Результат работы [pic-14]:

Создлся скрытый каталог .git
В среде разработки видим 11 файлов - кандидатов для
помещения в репозиторий



pic-14

Краткие пояснения [pic-15]:



pic-15

Отправить исходный код в remote Git- репозиторий


- Создать на github.ibm пустой репозиторий

Create a new repository

A repository contains all project files, including the revision history.

Owner

Repository name *

 pvx-dev-team ▾

/ lab-00 ✓

Great repository names are short and memorable. Need inspiration? How about [jubilant-octo-train?](#)

Description (optional)

lab-00 Проверка установки среды

☒  **Public**

Any logged in user can see this repository. You choose who can commit.

☐  **Internal**

IBM - Office of the CIO [enterprise members](#) can see this repository. You choose who can commit.

☐  **Private**

You choose who can see and commit to this repository.

Skip this step if you're importing an existing repository.

☐ **Initialize this repository with a README**

This will let you immediately clone the repository to your computer.

Add .gitignore: **None** ▾

Create repository

pic-16

readme.md не включаем!

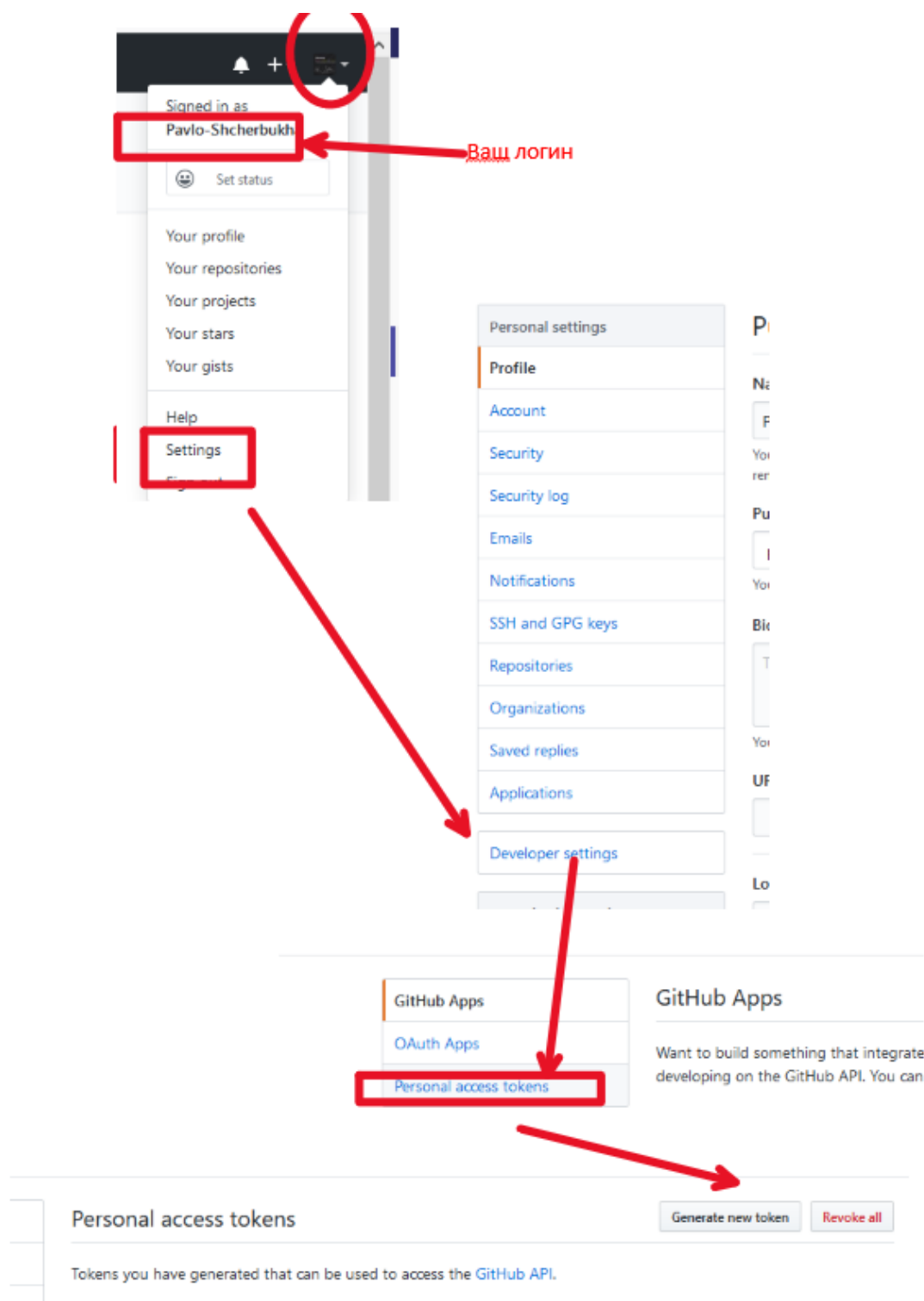
- скопировать команду git remote и выполнить ее в терминале

```
git remote add origin https://github.ibm.com/pvx-dev-team/lab-00.git
```

В результате, в файле `/.git/config` увидим лин на связанный репозиторий. Но, на github IBM в него нужно включить ваш логин и токен, который сгенерируете.

```
[remote "origin"]
  url = https://github.ibm.com/pvx-dev-team/lab-00.git
  fetch = +refs/heads/*:refs/remotes/origin/*
```

- Генерация токена Токе генерируется 1 раз, и дальше его помните. Шаги по генерации показаны на pic-17



pic-17

После этого трансформируем url репозитория:

было:

```
https://github.ibm.com/pvx-dev-team/lab-00.git
```

Стало:

```
https://login:token@github.ibm.com/pvx-dev-team/lab-00.git
```

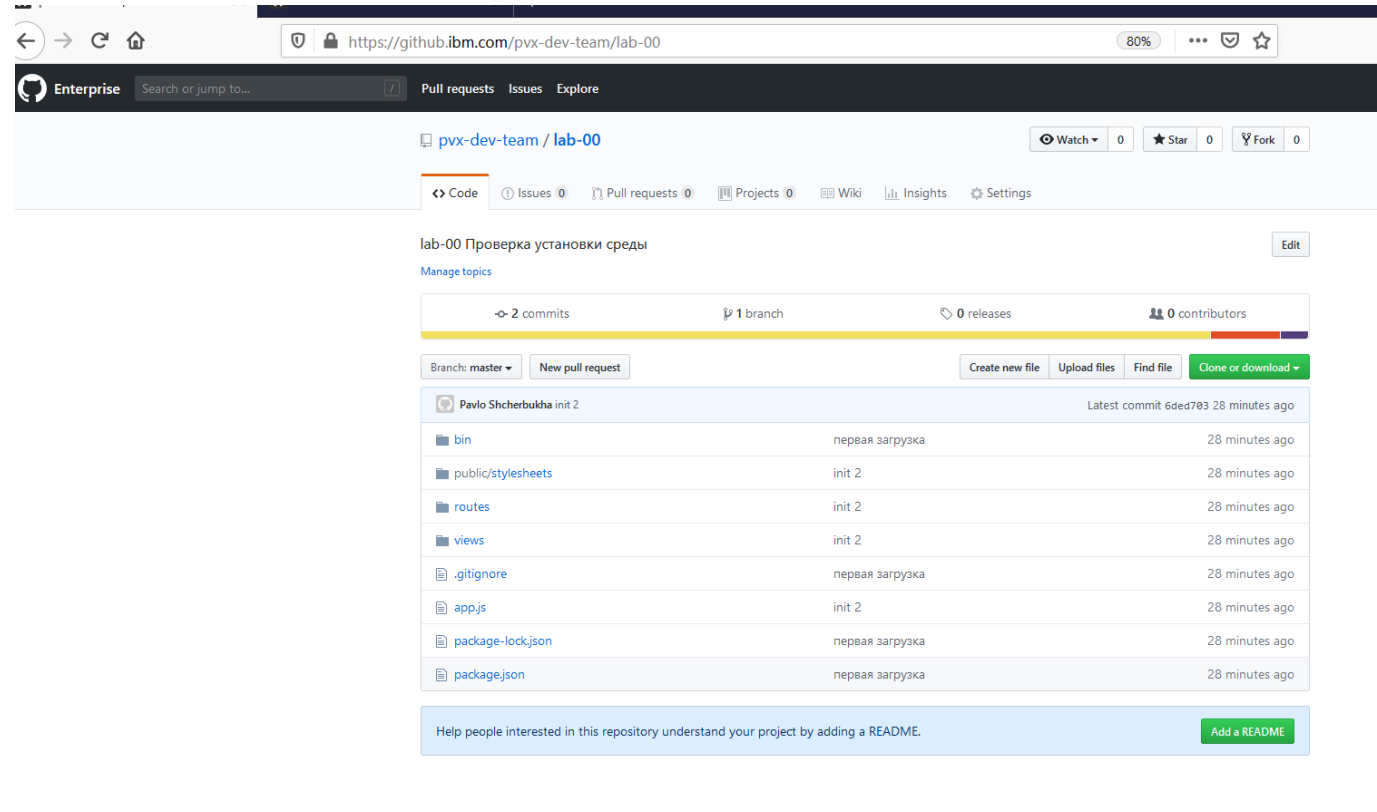
- Отправить данные из локального репозитория в remote

```
git push --set-upstream origin master
```

```
PS C:\PSHDEV\PSH-WorkShops\TeamTrain\JS-stack\L-0\lab-00\testapp1> git push --set-upstream origin master
Enumerating objects: 20, done.
Counting objects: 100% (20/20), done.
Delta compression using up to 12 threads
Compressing objects: 100% (16/16), done.
Writing objects: 100% (20/20), 12.09 KiB | 4.03 MiB/s, done.
Total 20 (delta 2), reused 0 (delta 0)
remote: Resolving deltas: 100% (2/2), done.
remote: detect-secrets-stream (beta) ver=164-1ac7e673b9acf31ed1ba86a8422cf59cb8a848be FAQ: https://ibm.biz/detect-secrets-stream-faq
remote:
remote: Successfully send push metadata.
remote: Push info collect pre-receive hook finished within 3 seconds.
To https://github.ibm.com/pvx-dev-team/lab-00.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
PS C:\PSHDEV\PSH-WorkShops\TeamTrain\JS-stack\L-0\lab-00\testapp1>
```

pic-18

И вуаля - на github.ibm появился исходный код



pic-19