

Git & IntelliJIdea Overview

Regular meeting for get knowledge and tech in
EDUC

- 1 Введение
- 2 Git.exe (установка)
- 3 IntelliJIdea (настройка)
- 4 Выкачивание проекта из Git
- 5 код->локальный/удаленный- > репозиторий.

Введение

ЧТО ?

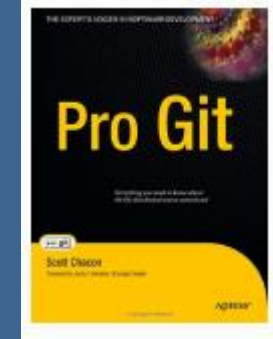
GitHub - это веб-сервис, на котором размещены миллиарды строк кода.

ЗАЧЕМ ?

GitHub нужен для размещения вашего кода или для совместной работы над кодом других людей.

Установка Git

0 Pro Git - <https://git-scm.com/book/ru/v1/> электронный учебник.

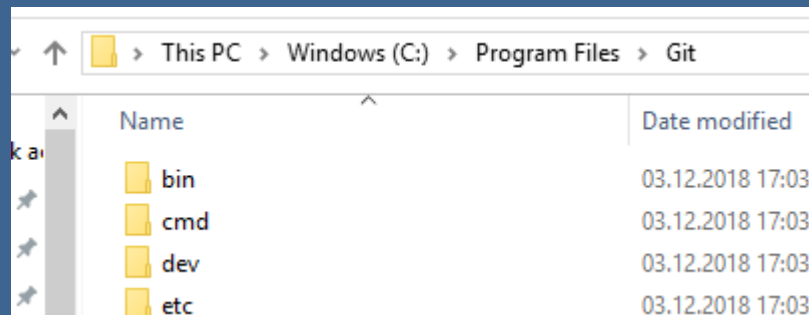


1 скачать утилиту GIT [Git-2.19.1-64-bit.exe] можно с <https://gitforwindows.org/>

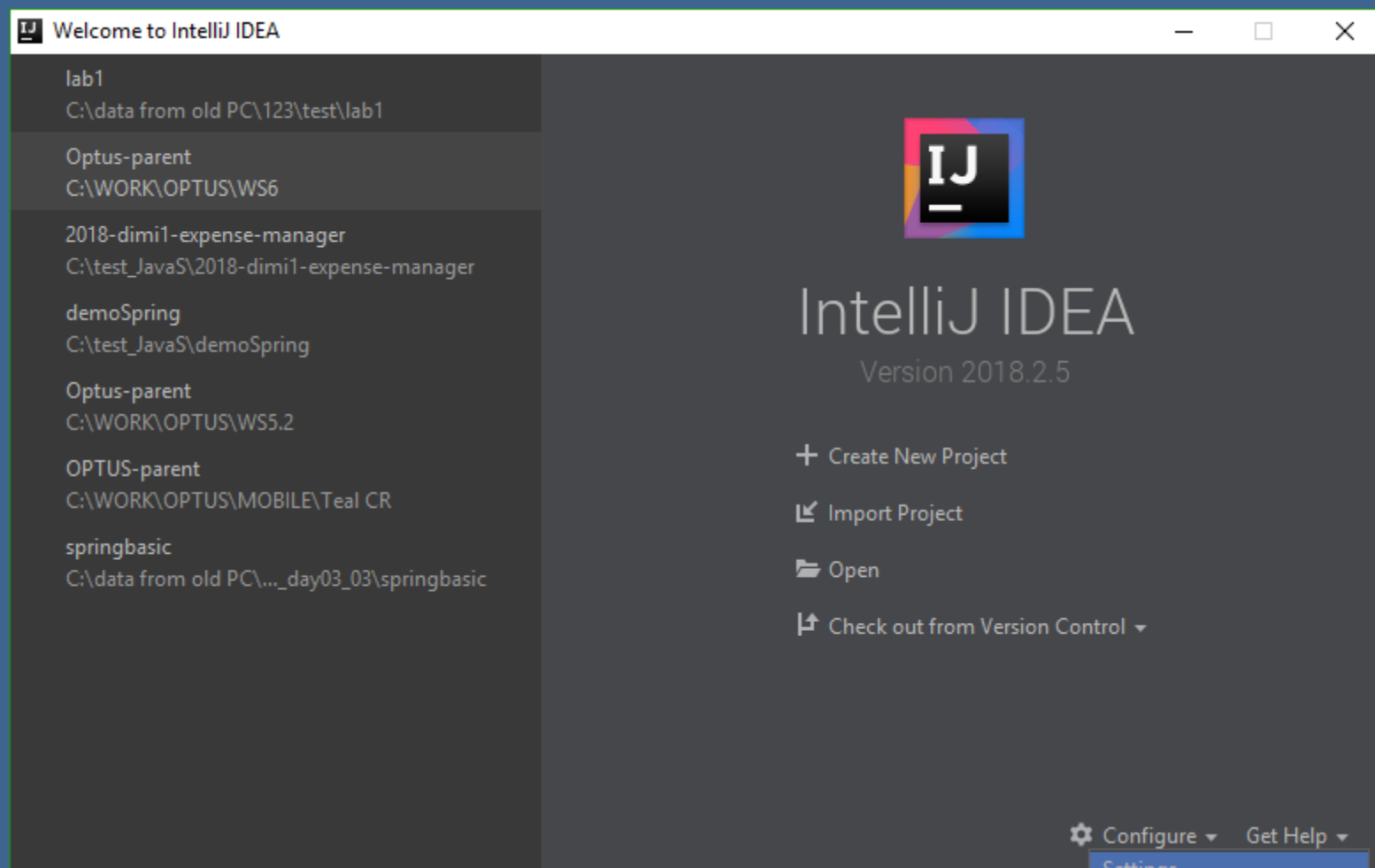


2 Установить

3 Проверить

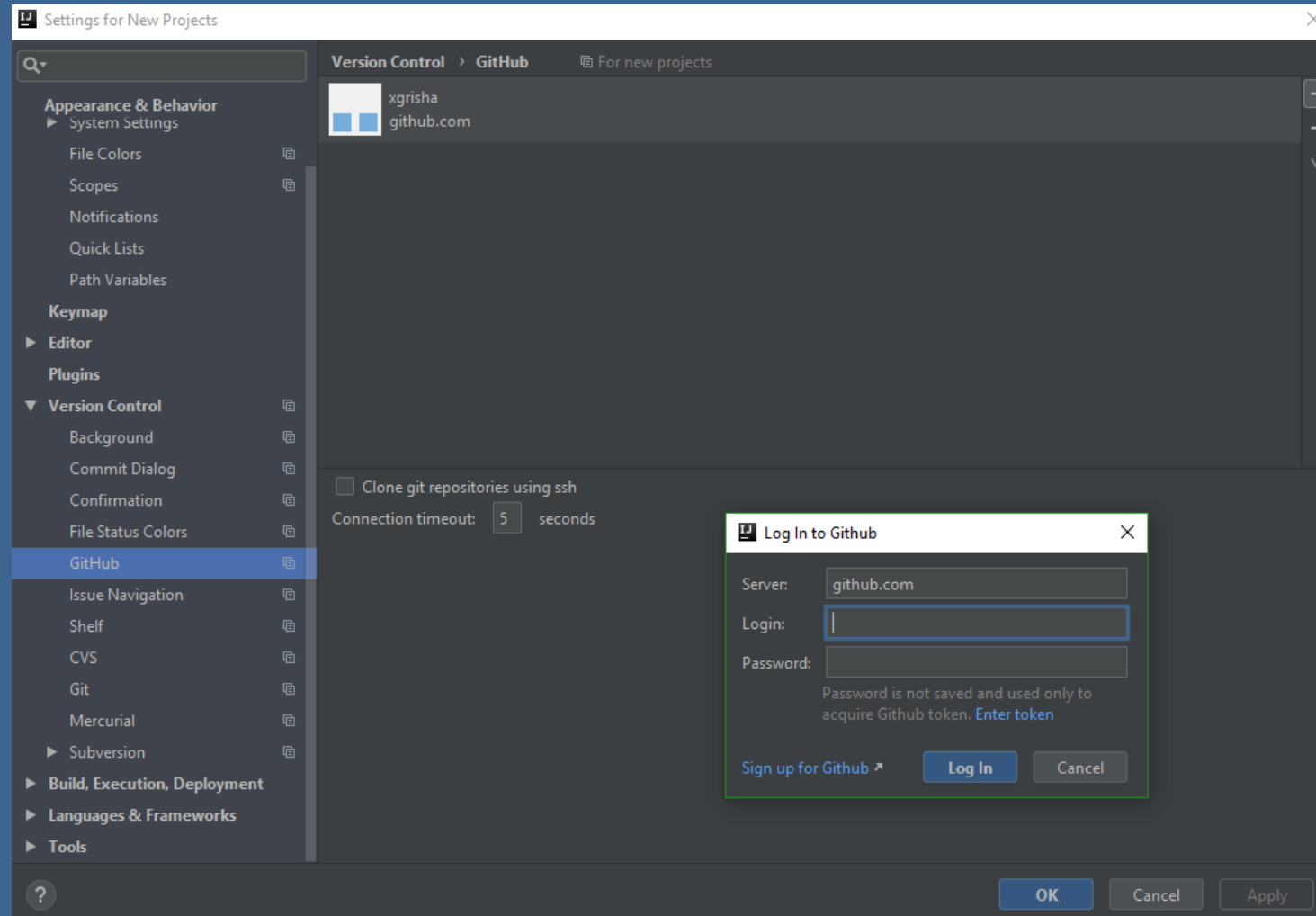


IntelliJ Idea (настройка)

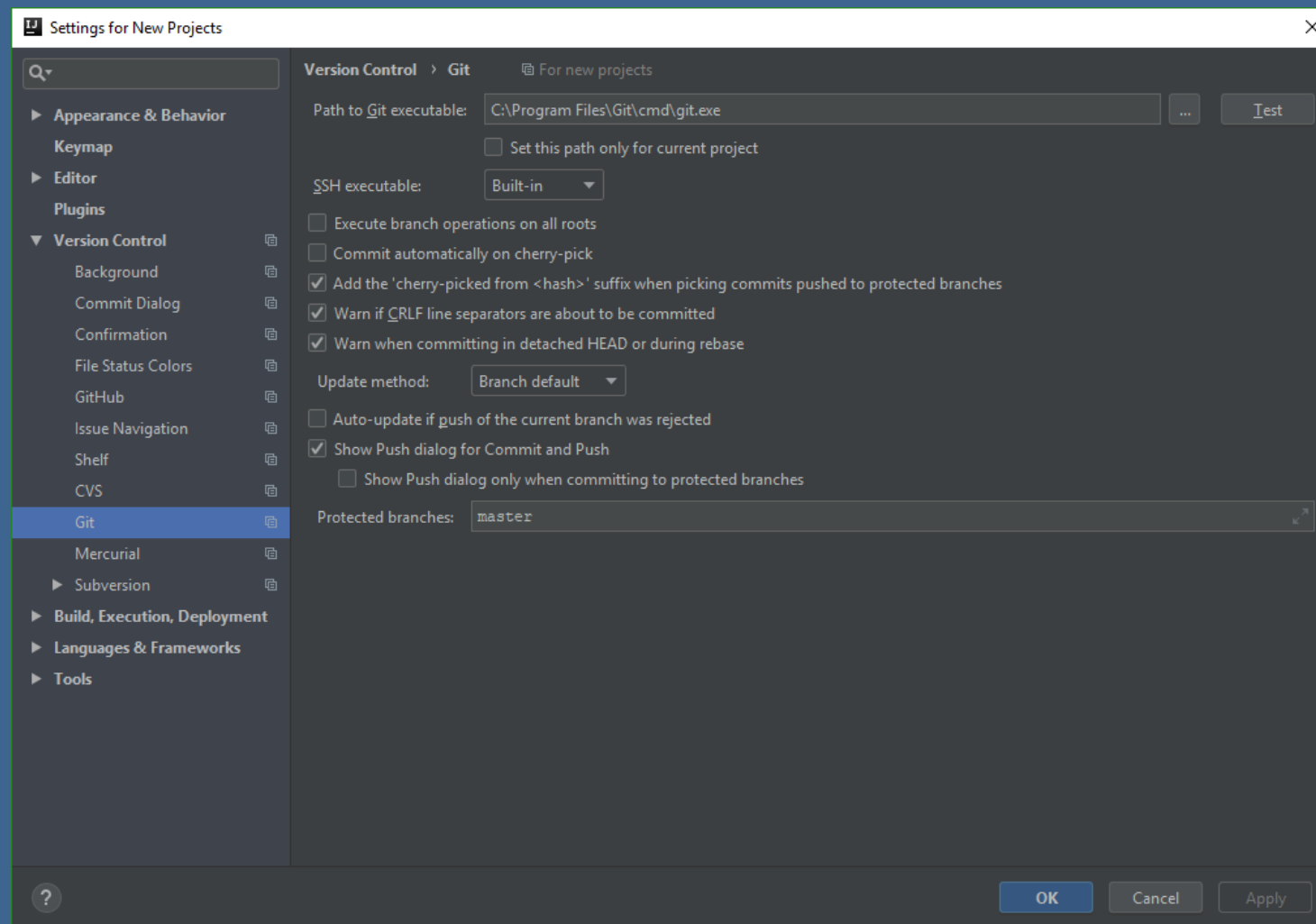


IntelliJ Idea (настройка)

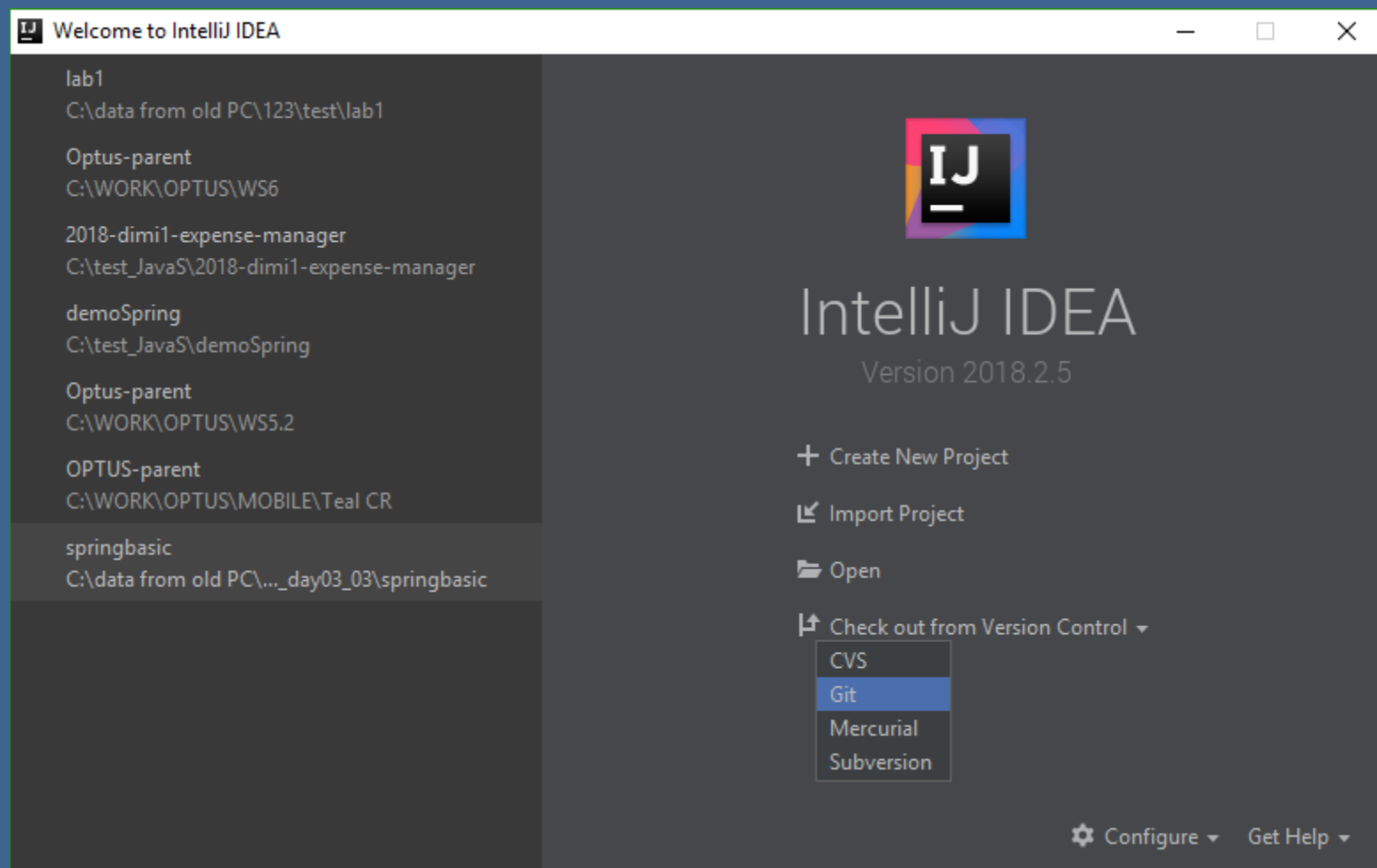
2
column



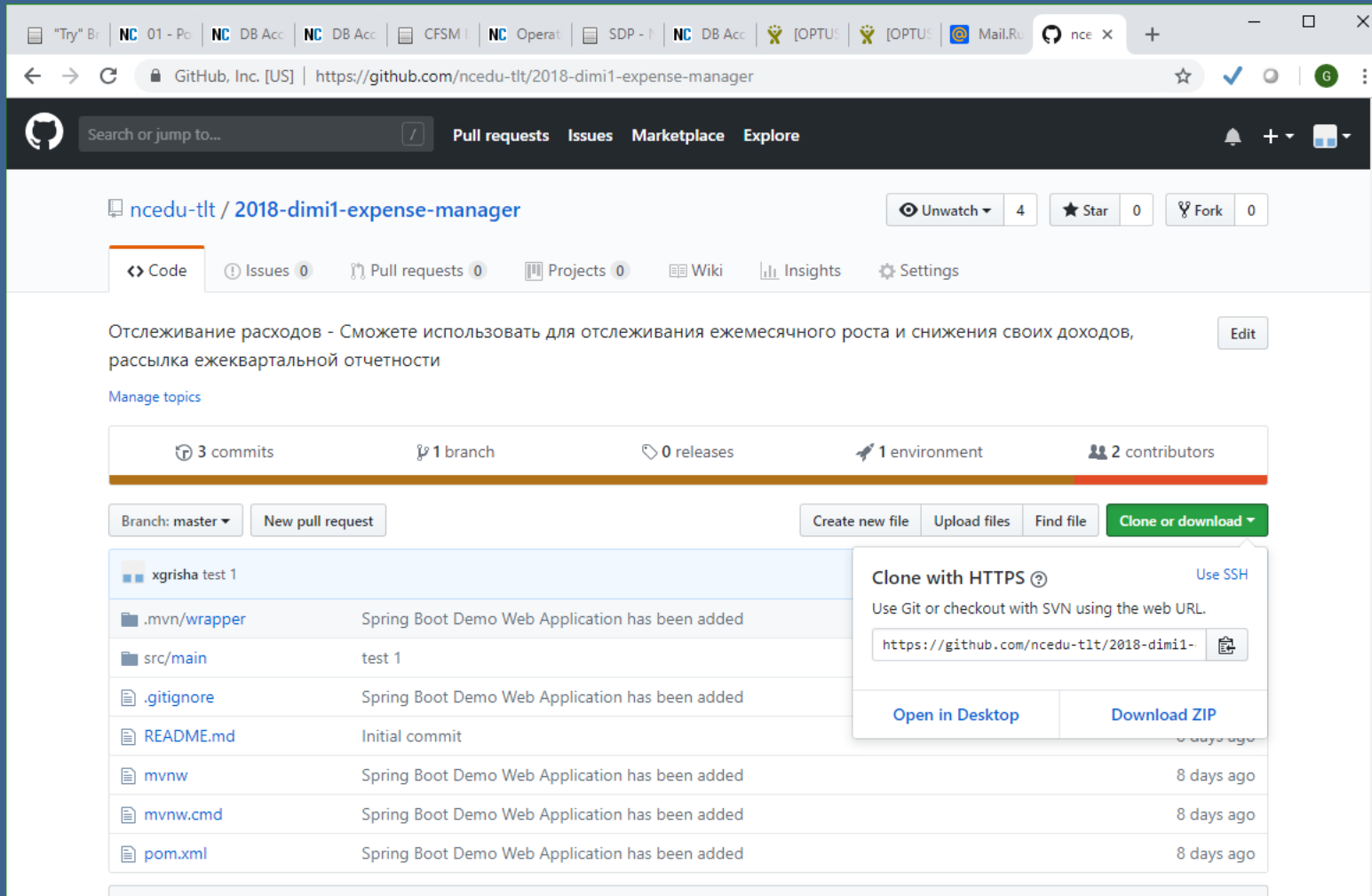
IntelliJ Idea (настройка)



IntelliJ Idea (настройка)

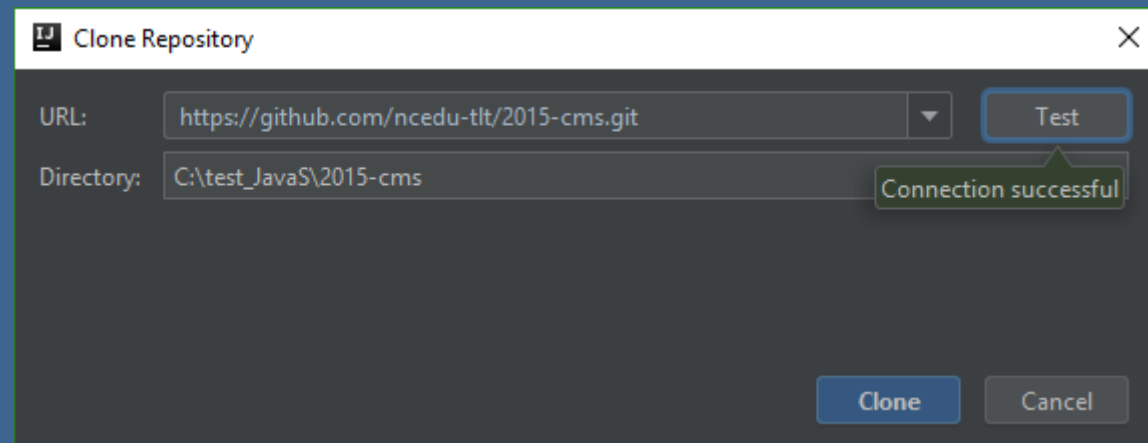


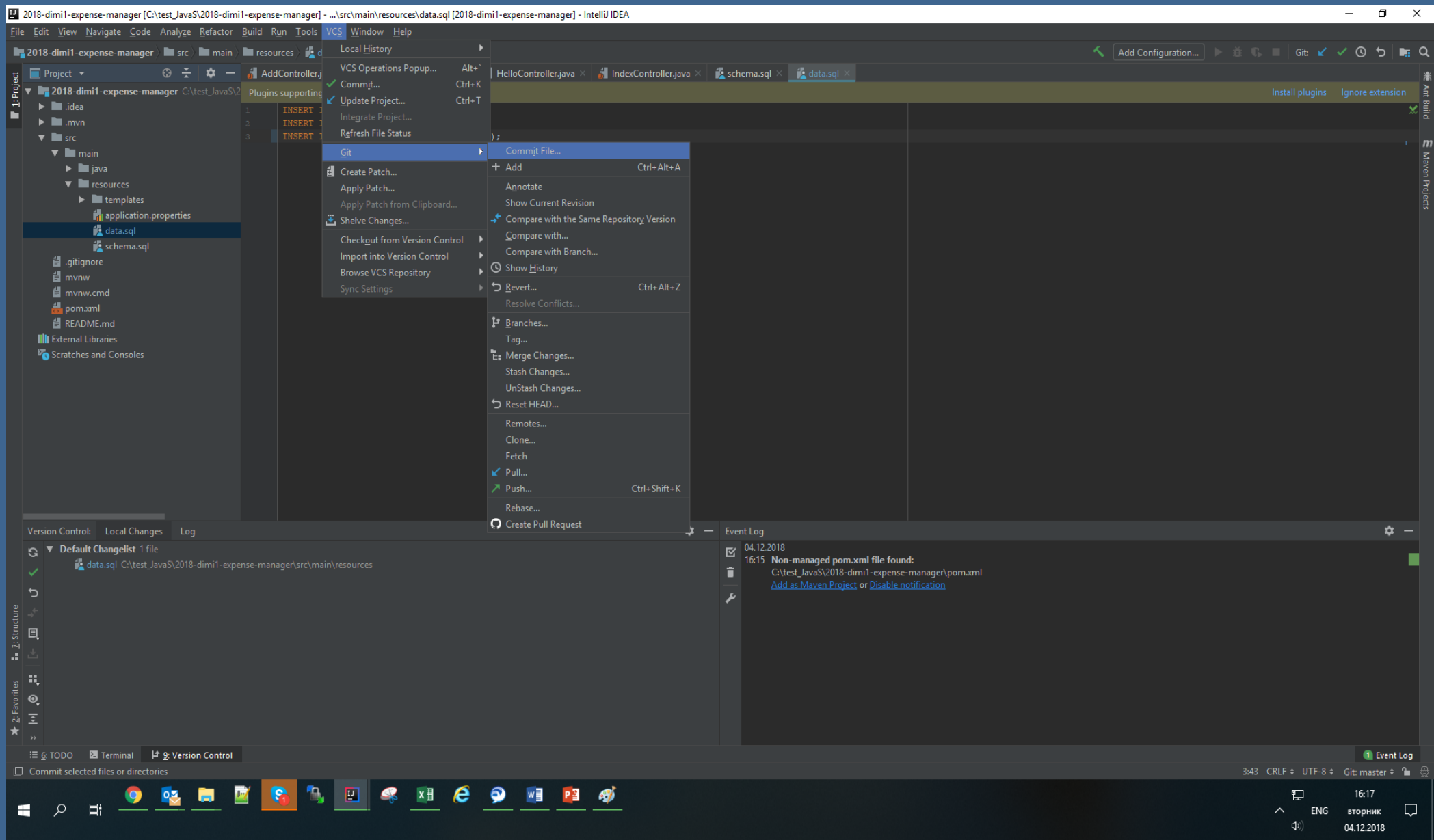
IntelliJ IDEA (настройка)



Для импорта проекта в IntelliJ IDEA потребуется адрес, его можно получить со странички на GitHub, для этого достаточно нажать на зеленую кнопку [Clone or download] и у вас отобразится выпадающее окошко в котором будет адресная строка.

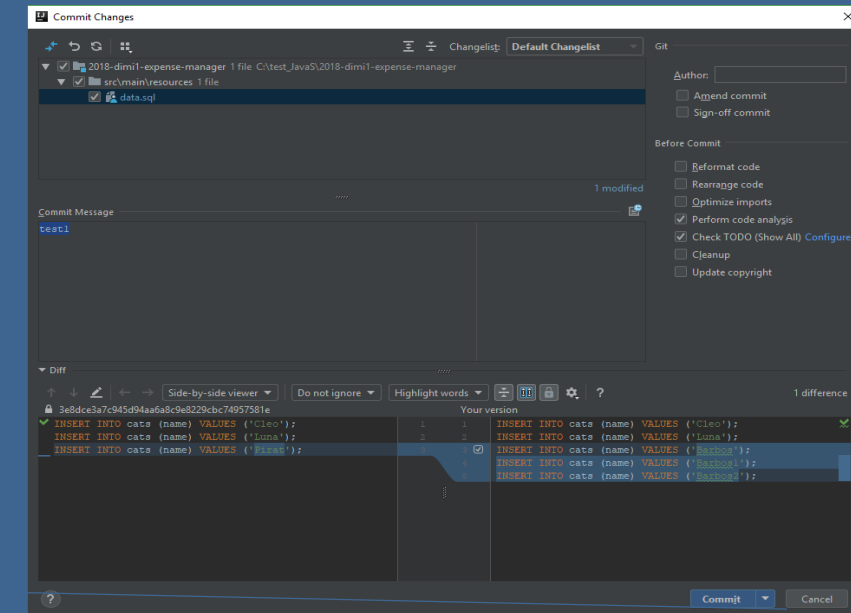
IntelliJ Idea (настройка)



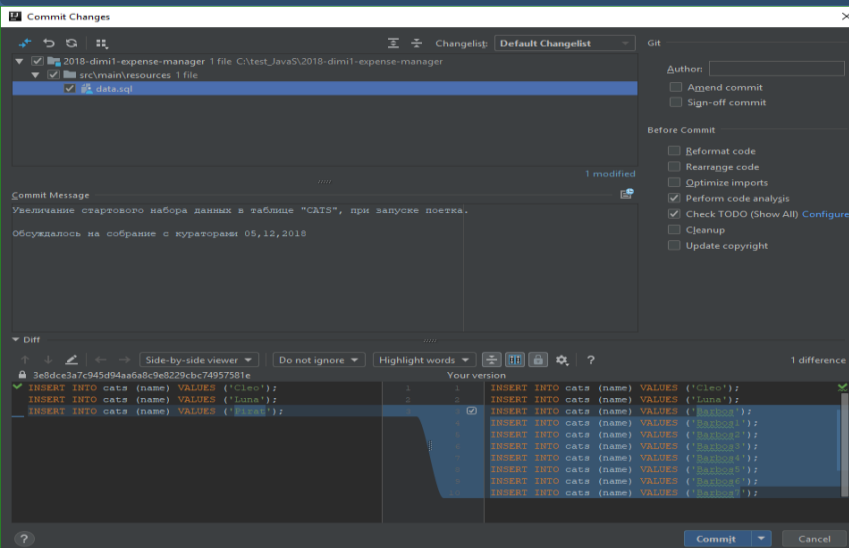


Во время коммита принято оставлять комментарии

Пример – не правильного комментария



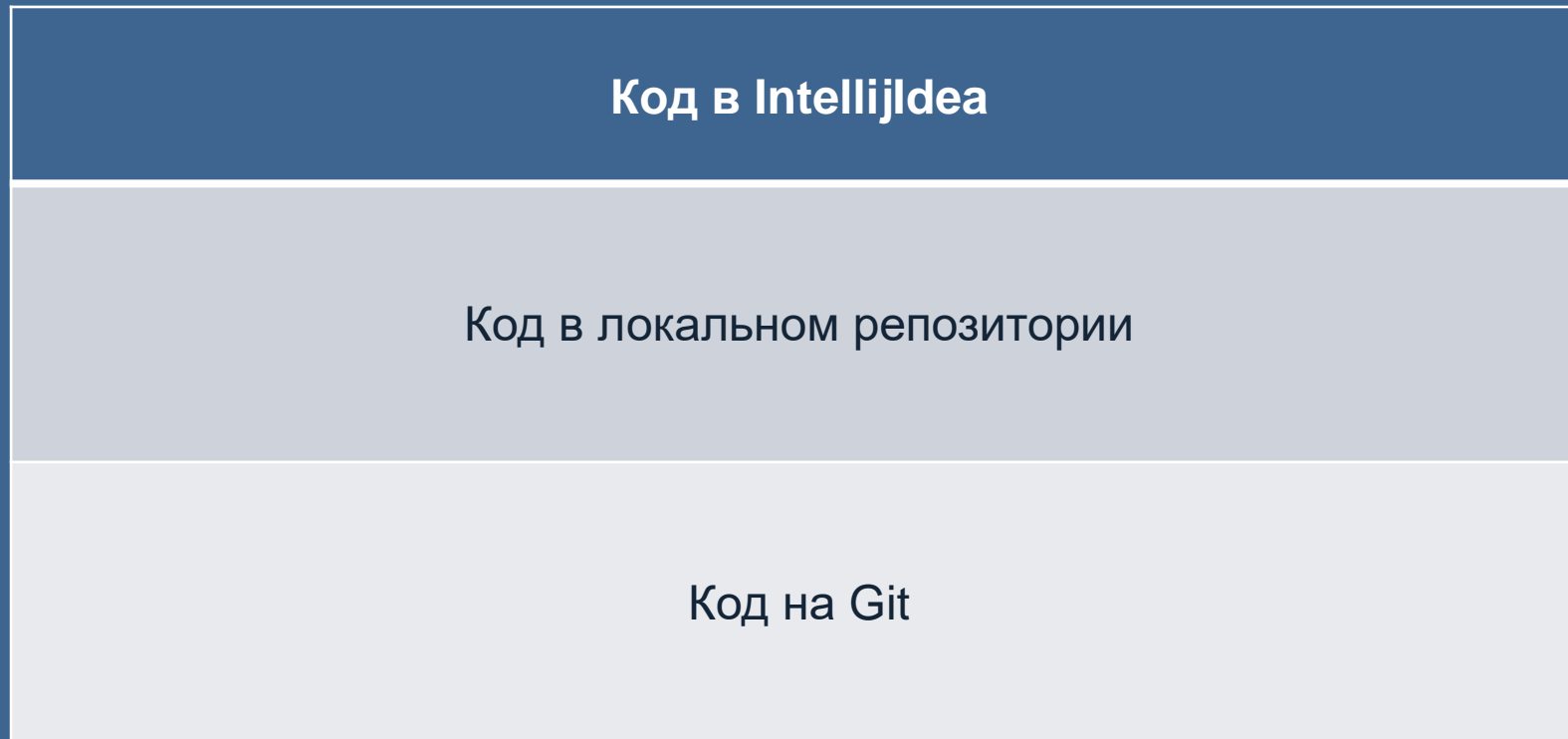
Комментарии должны
ясно отображать суть и
причину изменений!



Пример – правильного комментария




код->локальный/удаленный- >репозитории.



Проблемы / трудности – при выполнении вышеописанного

- Касперский может мешать настроить соединение с Git (решение – выключить его на некоторое время).

- После выкачивания проекта он не будет запускаться – для этого с правой стороны нужно нажать на Maven Projects и нажать на кнопку обновить  далее он сам подключится и произведет необходимые настройки.

После этого многие из строчек будут помечены красным цветом

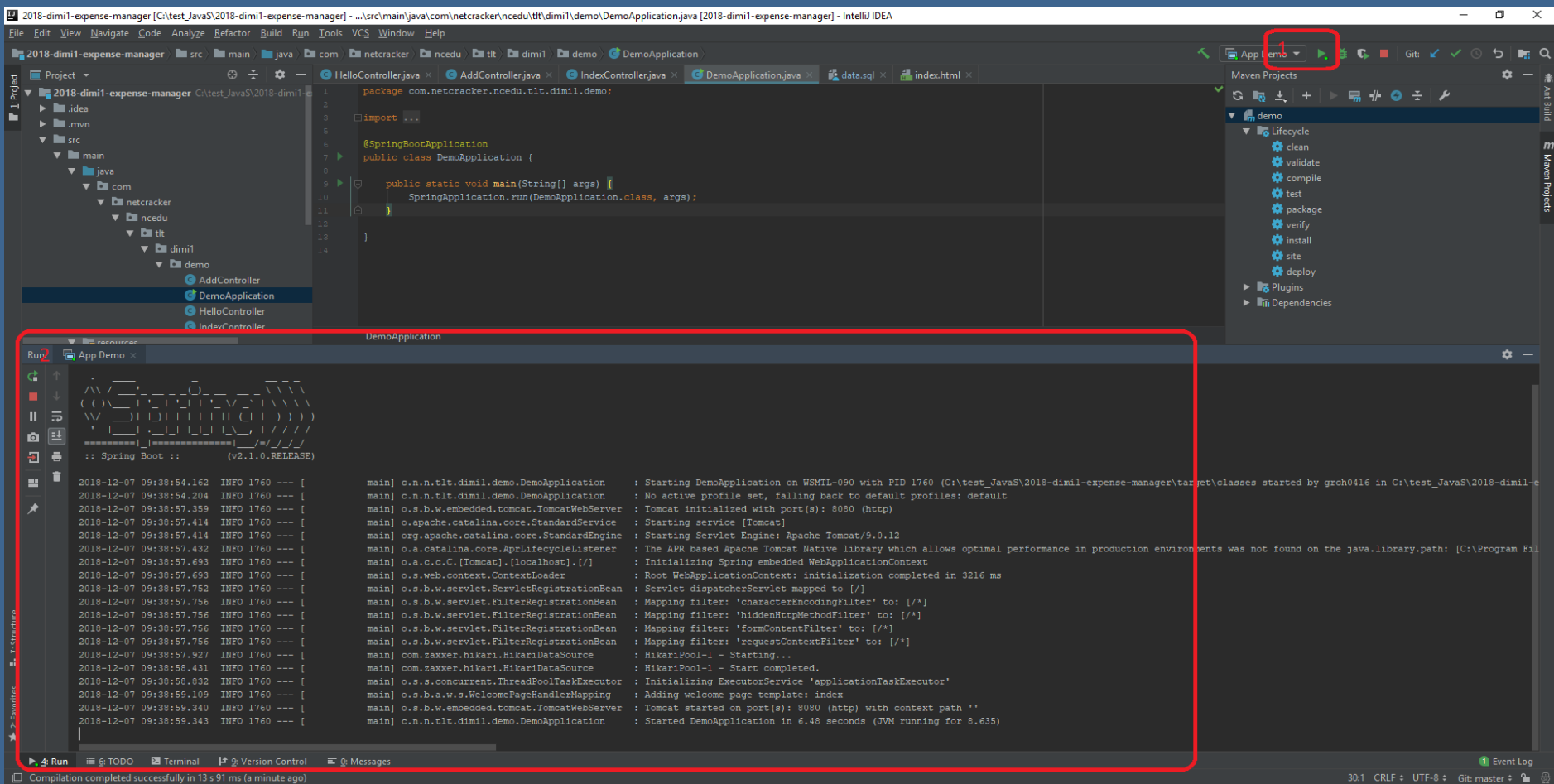
```
@RequestMapping("/")
String home(ModelMap model) {
    model.addAttribute("cats", jdbcTemplate.queryForList("sql:
    return "index";
}
```

нажмите на любую из них и нажмите Alt+пробел – будет предложено подключить JDK. Эти действия приведут проект в актуальное состояние и можно будет его запустить нажав на зеленый треугольник

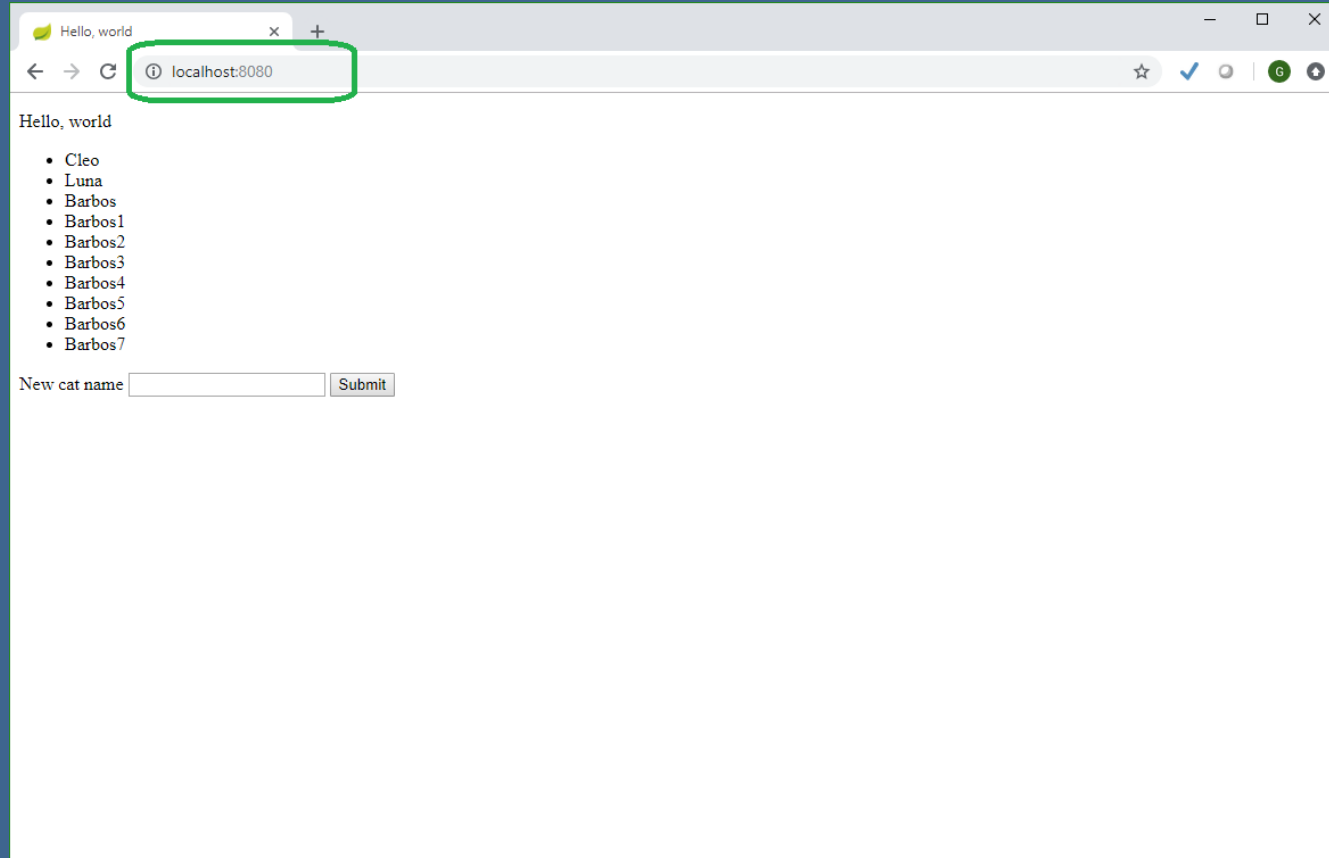


Далее если зайти на localhost:8080 через браузер, то увидите запущенное приложение в действие.

Эти действия приведут проект в актуальное состояние и можно будет его запустить нажав на зеленый треугольник (метка 1) с низу в логах отобразится процесс запуска (метка 2)



После запуска программы наберите в адресной строке любого установленного у вас веб-браузера localhost:8080



Программа запущена.

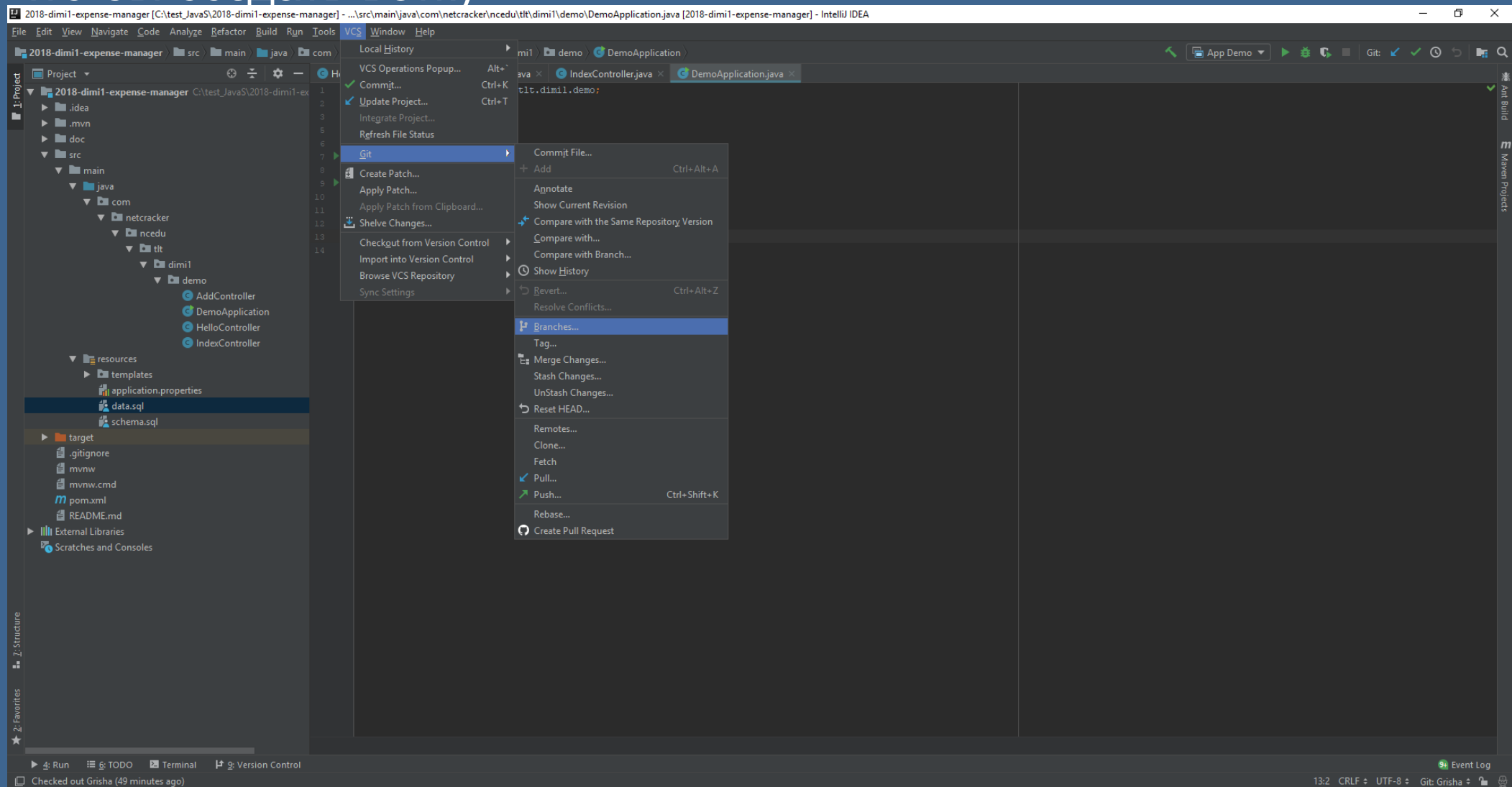
Что такое branch и в чем его смысл.

Как правило у каждого проекта имеется основная ветка {master} все наработки попадают туда только после полной проверки работоспособности и исключения конфликтов между модулями и функционалом.

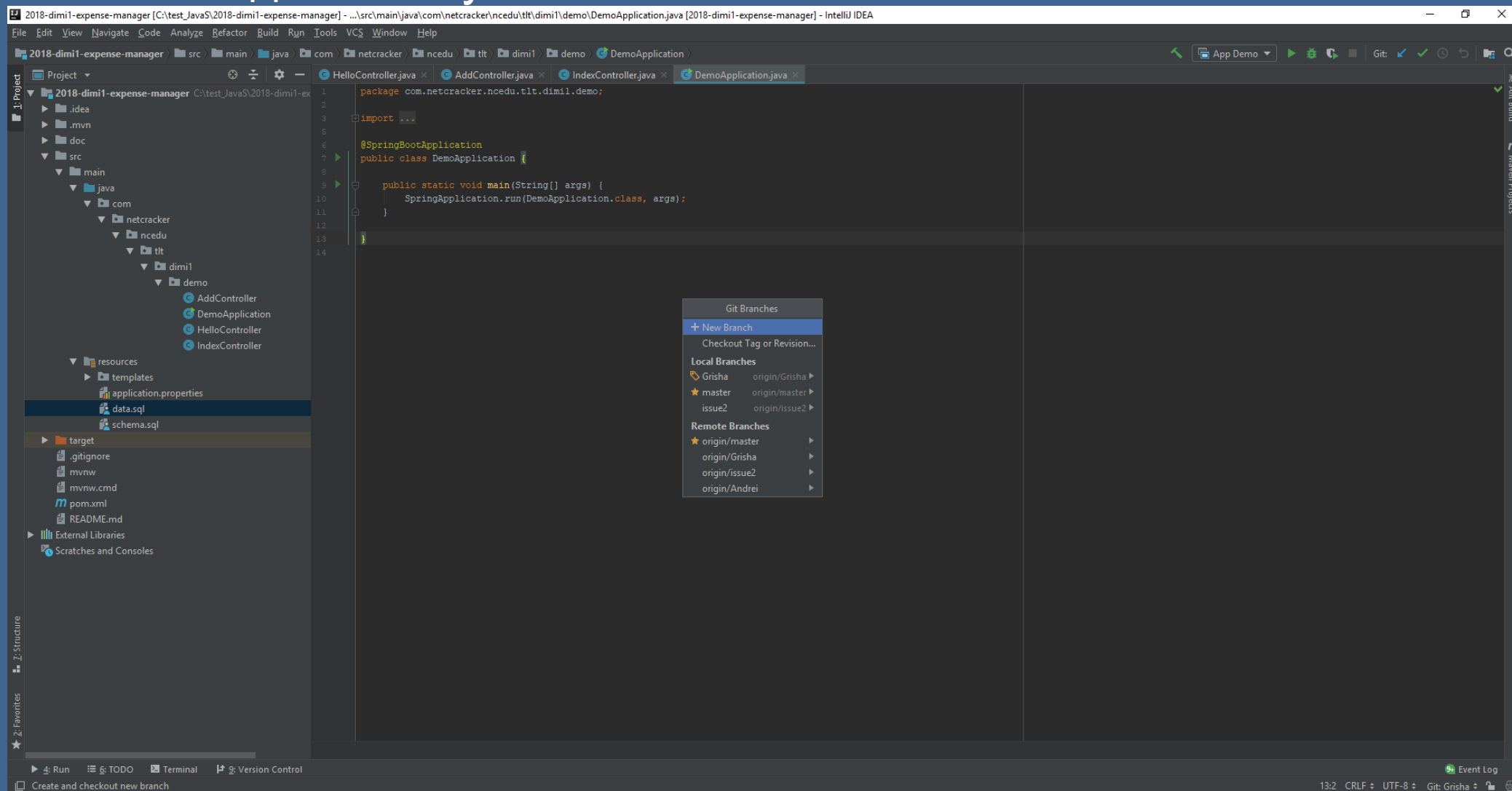
Отдельный разработчик либо группа что бы не поломать основную ветку, делает свою ветку на основе ветки {master}, но называет её например {master_copy} и производит в ней какие либо работы (например разрабатывает новый функционал).

Смысл отведения ветки в том что она будет содержать в себе копию ветки {master}, но все изменения сделанные в ней не отразятся в основной ветке, пока не будут проверены и смёржены в основную ветку.

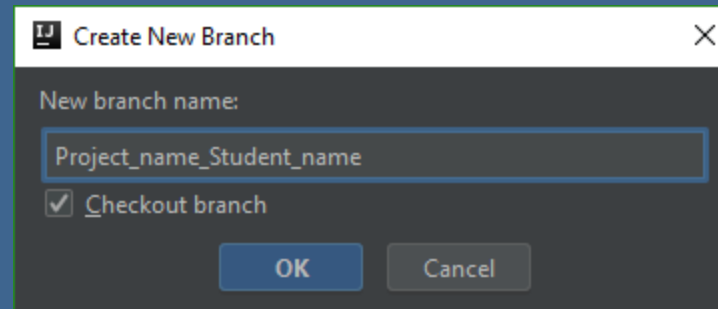
Что бы создать ветку



Что бы создать ветку



Что бы создать ветку



A screenshot of a 'Create New Branch' dialog box. The dialog has a title bar with a small icon and the text 'Create New Branch' and a close button. The main area contains the text 'New branch name:' followed by a text input field containing 'Project_name_Student_name'. Below the input field is a checked checkbox labeled 'Checkout branch'. At the bottom are 'OK' and 'Cancel' buttons.

Create New Branch

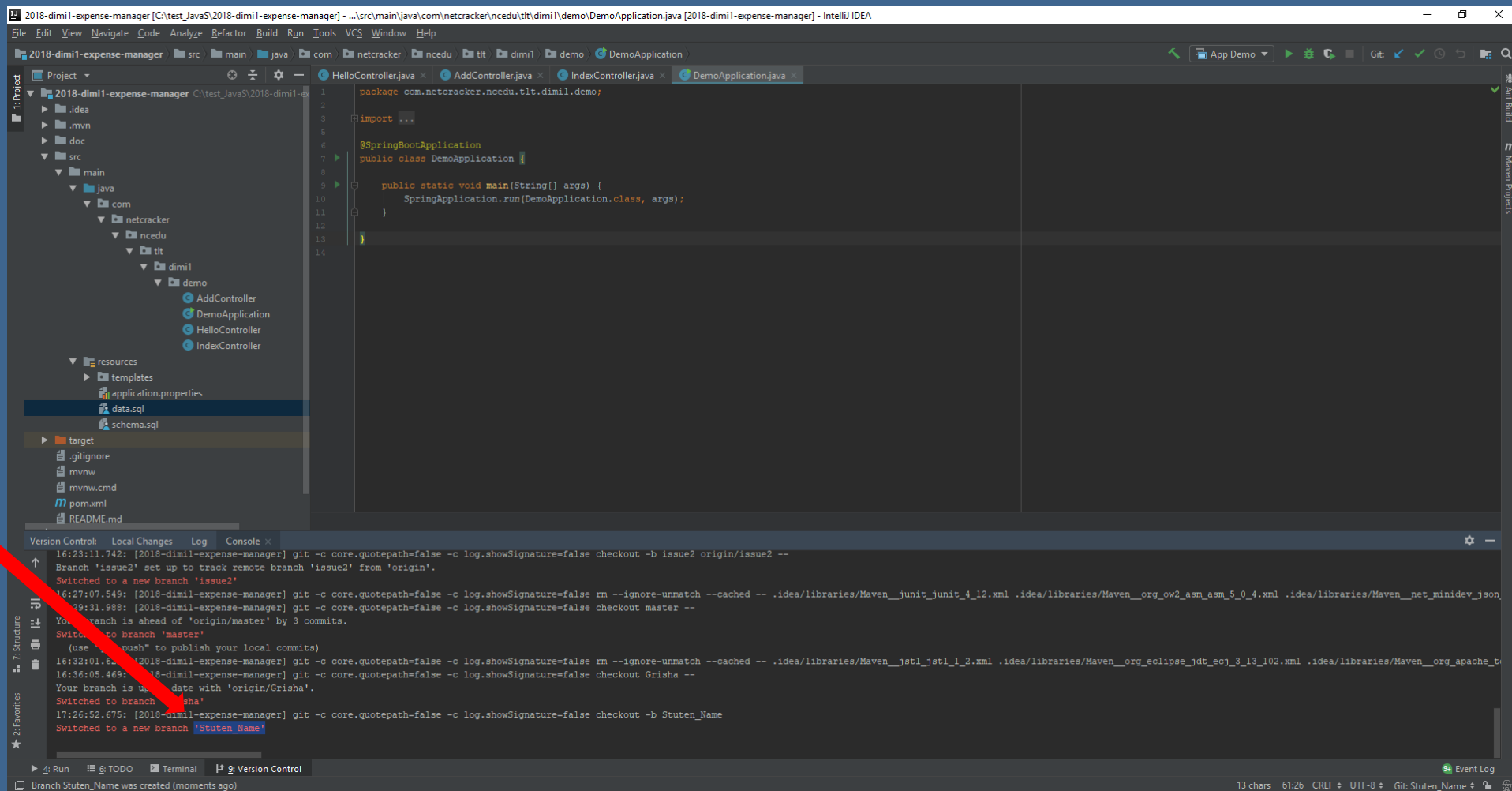
New branch name:

Project_name_Student_name

☒ Checkout branch

OK Cancel

Видим что ветка
создалась, но
она пока не
будет
отображаться
в GitHub а
только на
локальном
репозитории



2018-dimil-expense-manager [C:\test_JavaS\2018-dimil-expense-manager] - ...src\main\java\com\netcracker\ncedu\slt\dimil\demo\DemoApplication.java [2018-dimil-expense-manager] - IntelliJ IDEA

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

2018-dimil-expense-manager > src > main > java > com > netcracker > ncedu > slt > dimil > demo > DemoApplication

Project

- 2018-dimil-expense-manager
 - .idea
 - .mvn
 - doc
 - src
 - main
 - java
 - com
 - netcracker
 - ncedu
 - slt
 - dimil
 - demo
 - AddController
 - DemoApplication
 - HelloController
 - IndexController
 - resources
 - templates
 - application.properties
 - data.sql
 - schema.sql
 - target
 - .gitignore
 - mvnw
 - mvnw.cmd
 - pom.xml
 - README.md

```

1 package com.netcracker.ncedu.slt.dimil.demo;
2
3 import ...
4
5
6 @SpringBootApplication
7 public class DemoApplication {
8
9     public static void main(String[] args) {
10         SpringApplication.run(DemoApplication.class, args);
11     }
12
13 }
14

```

Git Branches

- + New Branch
- Checkout Tag or Revision...
- Local Branches
 - Stuten_Name
 - master origin/master
 - Grisha origin/Grisha
 - issue2 origin/issue2
- Remote Branches
 - origin/master
 - origin/Grisha
 - origin/issue2
 - origin/Andrei

Version Control: Local Changes Log Console

```

16:23:11.742: [2018-dimil-expense-manager] git -c core.quotepath=false -c log.showSignature=false checkout -b issue2 origin/issue2 --
Branch 'issue2' set up to track remote branch 'issue2' from 'origin'.
Switched to a new branch 'issue2'
16:27:07.549: [2018-dimil-expense-manager] git -c core.quotepath=false -c log.showSignature=false rm --ignore-unmatch --cached -- .idea/libraries/Maven__junit_junit_4_12.xml .idea/libraries/Maven__org_ow2_asm_asm_5_0_4.xml .idea/libraries/Maven__net_minidev_json
16:29:31.988: [2018-dimil-expense-manager] git -c core.quotepath=false -c log.showSignature=false checkout master --
Your branch is ahead of 'origin/master' by 3 commits.
Switched to branch 'master'
(use "git push" to publish your local commits)
16:32:01.621: [2018-dimil-expense-manager] git -c core.quotepath=false -c log.showSignature=false rm --ignore-unmatch --cached -- .idea/libraries/Maven__jstl_jstl_1_2.xml .idea/libraries/Maven__org_eclipse_jdt_ecj_3_13_102.xml .idea/libraries/Maven__org_apache_t
16:36:05.469: [2018-dimil-expense-manager] git -c core.quotepath=false -c log.showSignature=false checkout Grisha --
Your branch is up to date with 'origin/Grisha'.
Switched to branch 'Grisha'
17:26:52.675: [2018-dimil-expense-manager] git -c core.quotepath=false -c log.showSignature=false checkout -b Stuten_Name
Switched to a new branch 'Stuten_Name'

```

4: Run | TODO | Terminal | 9: Version Control

Branch Stuten_Name was created (4 minutes ago)

Event Log

14:1 CRLF UTF-8 Git: Stuten_Name

Переключение между ветками

The screenshot shows the IntelliJ IDEA IDE interface. The top toolbar includes buttons for running the application (a green play button), debugging (a green bug icon), and other development tools. The main editor displays the `DemoApplication.java` file, which contains the following code:

```
1 package com.netcracker.ncedu.tlt.dimi1.demo;
2
3 import ...
4
5 @SpringBootApplication
6 public class DemoApplication {
7
8     public static void main(String[] args) {
9         SpringApplication.run(DemoApplication.class, args);
10    }
11
12 }
13
14
```

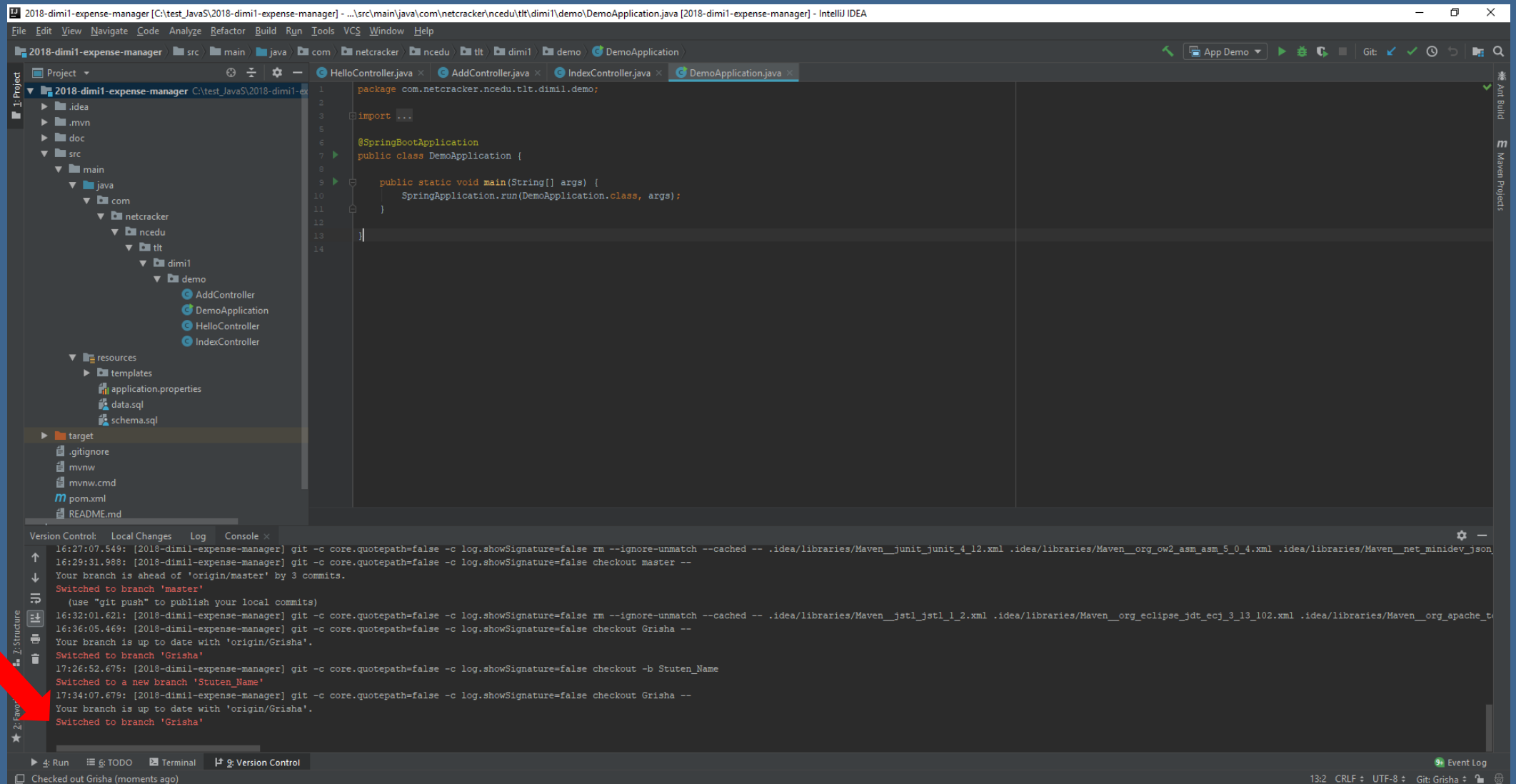
A context menu titled "Git Branches" is open over the code editor. It lists the following options:

- + New Branch
- Checkout Tag or Revision...
- Local Branches
 - Stuten_Name
- ★ master origin/master
- ★ Grisha origin/Grisha
- issue2 origin/issue2
- Remote Branches
 - ★ origin/master
 - origin/Grisha
 - origin/issue2
 - origin/Andrei
- Checkout
- Checkout As...
- Compare With...
- Rebase Current onto Selected
- Checkout with Rebase
- Merge into Current
- Rename...
- Delete

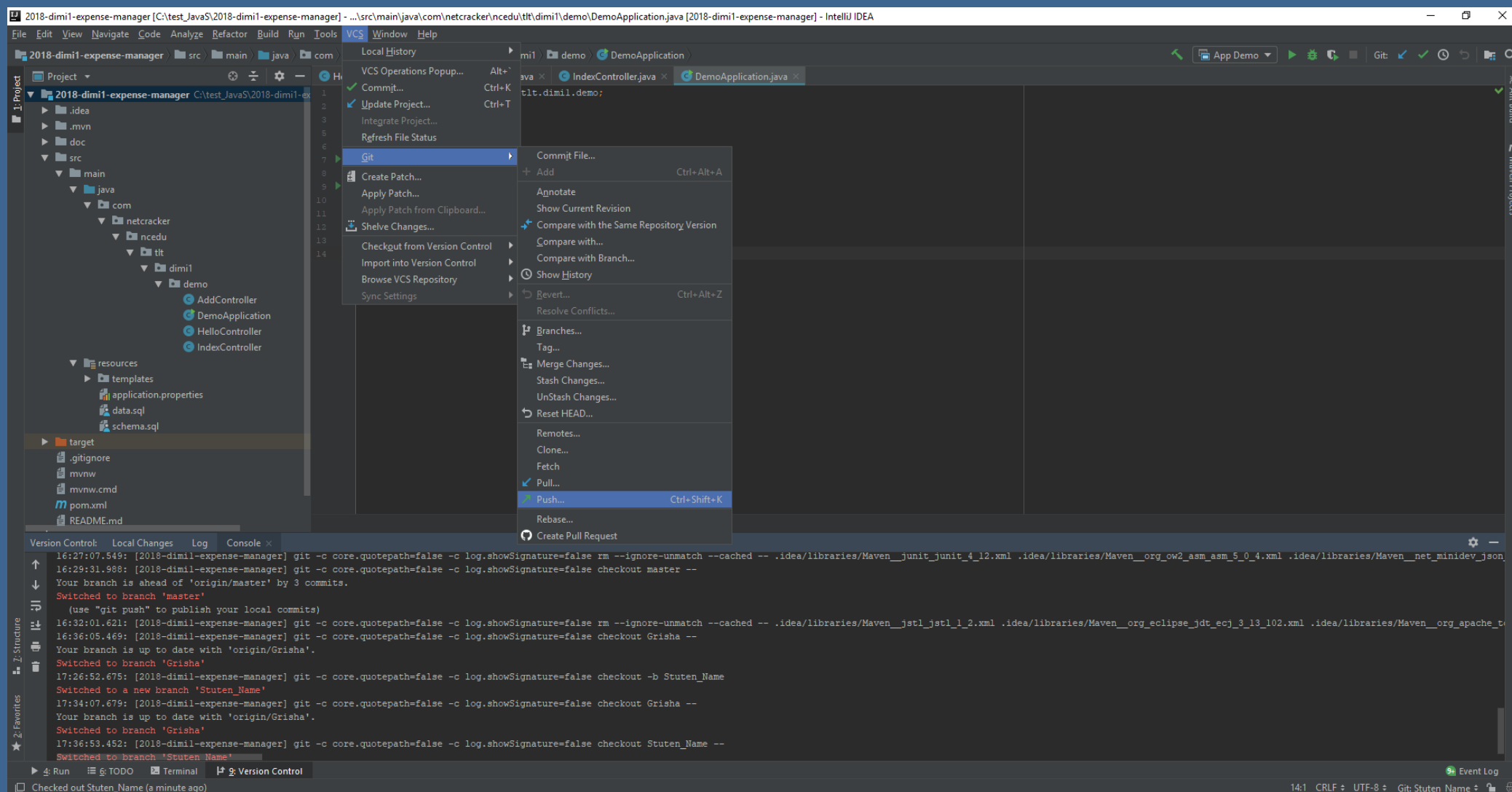
The bottom panel shows the "Version Control" tab, which displays the following log entries:

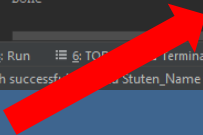
```
16:23:11.742: [2018-dimil-expense-manager] git -c core.quotePath=false -c log.showSignature=false checkout -b issue2 origin/issue2 --
Branch 'issue2' set up to track remote branch 'issue2' from 'origin'.
Switched to a new branch 'issue2'
16:27:07.549: [2018-dimil-expense-manager] git -c core.quotePath=false -c log.showSignature=false rm --ignore-unmatch --cached -- .idea/libraries/Maven__junit_junit_4_12.xml .idea/libraries/Maven__org_ow2_asm_asm_5_0_4.xml .idea/libraries/Maven__net_minidev_json
16:29:31.988: [2018-dimil-expense-manager] git -c core.quotePath=false -c log.showSignature=false checkout master --
Your branch is ahead of 'origin/master' by 3 commits.
Switched to branch 'master'
(use "git push" to publish your local commits)
16:32:01.621: [2018-dimil-expense-manager] git -c core.quotePath=false -c log.showSignature=false rm --ignore-unmatch --cached -- .idea/libraries/Maven__jstl_jstl_1_2.xml .idea/libraries/Maven__org_eclipse_jdt_ecj_3_13_102.xml .idea/libraries/Maven__org_apache_t
16:36:05.469: [2018-dimil-expense-manager] git -c core.quotePath=false -c log.showSignature=false checkout Grisha --
Your branch is up to date with 'origin/Grisha'.
Switched to branch 'Grisha'
17:26:52.675: [2018-dimil-expense-manager] git -c core.quotePath=false -c log.showSignature=false checkout -b Stuten_Name
Switched to a new branch 'Stuten_Name'
```

Видим, что мы
переключились
на другую
ветку



Для того чтобы новая ветка отобразилась в репозитории GitHub, необходимо её отправить на сервер командой Push





Заходим на GitHub, проверяем. Теперь она видна/доступна всем участникам команды.

