### Prerequiste:

GIT should be installed and configured.

### Steps to clone the source:

1. Start command prompt
2. Set the location in command to clone the source (cd <source\_location>)
3. Execute the following command

git clone https://github.com/EbrosGaming/spring-petclinic-angular-and-rest-master.git

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Setup enviroment**

**Start the Rest Service**

**Steps:**

1. Start command prompt
2. Browse to source cloned location. (cd <source\_location>)
3. Browse to ..cd <source\_location>\spring-petclinic-rest-master\spring-petclinic-rest-master
4. Execute the following command:

maven spring-boot:run ## not working? try using mvn spring-boot:run

**Setup enviroment**

**Start the Angular Service**

Enter into the static-content directory (by running cd spring-petclinic-angular/static-content), run **one** of the following commands to run a simple web server on port 4200:

# Method 1 : using java

curl https://jcenter.bintray.com/com/athaydes/rawhttp/rawhttp-cli/1.0/rawhttp-cli-1.0-all.jar -o rawhttp.jar

java -jar ./rawhttp.jar serve . -p 4200

# Method 2 : using python3

python -m http.server 4200

# Method 3 : using python2

python -m SimpleHTTPServer 4200

# Method 4 : using ruby

ruby -run -ehttpd . -p4200

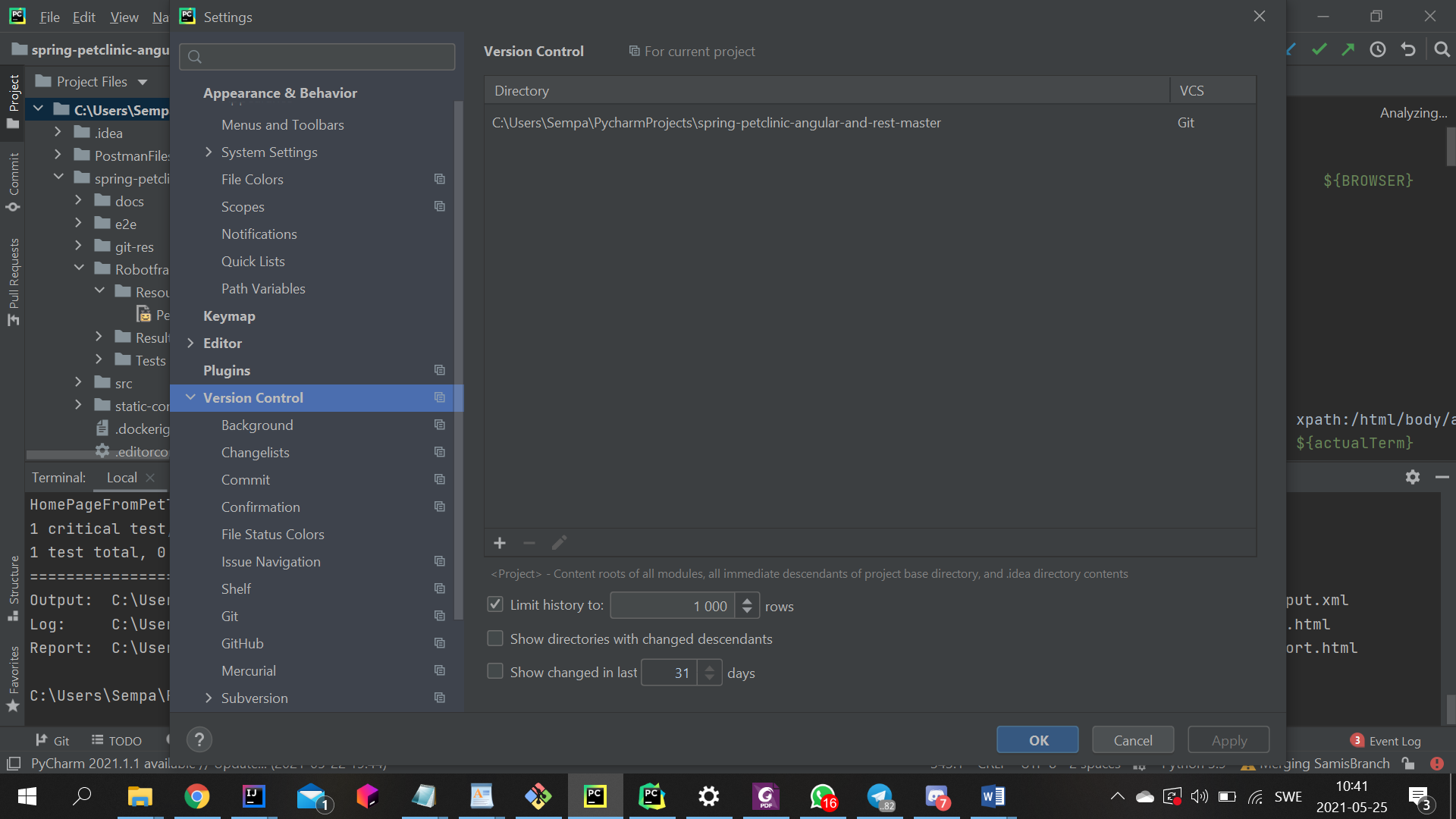
# Method 5 : using npm

npm install -g http-server

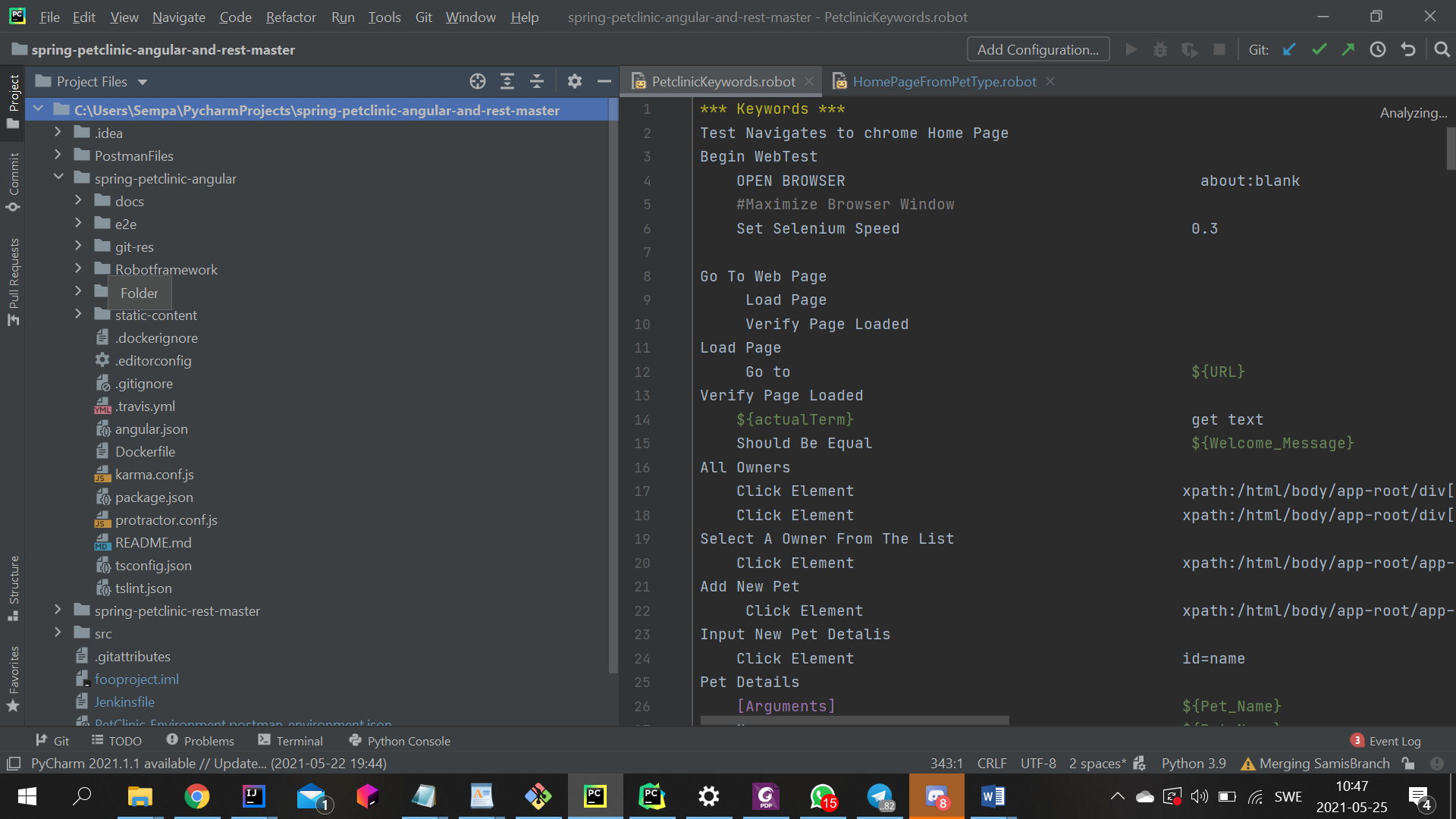
http-server -p 4200

Create task

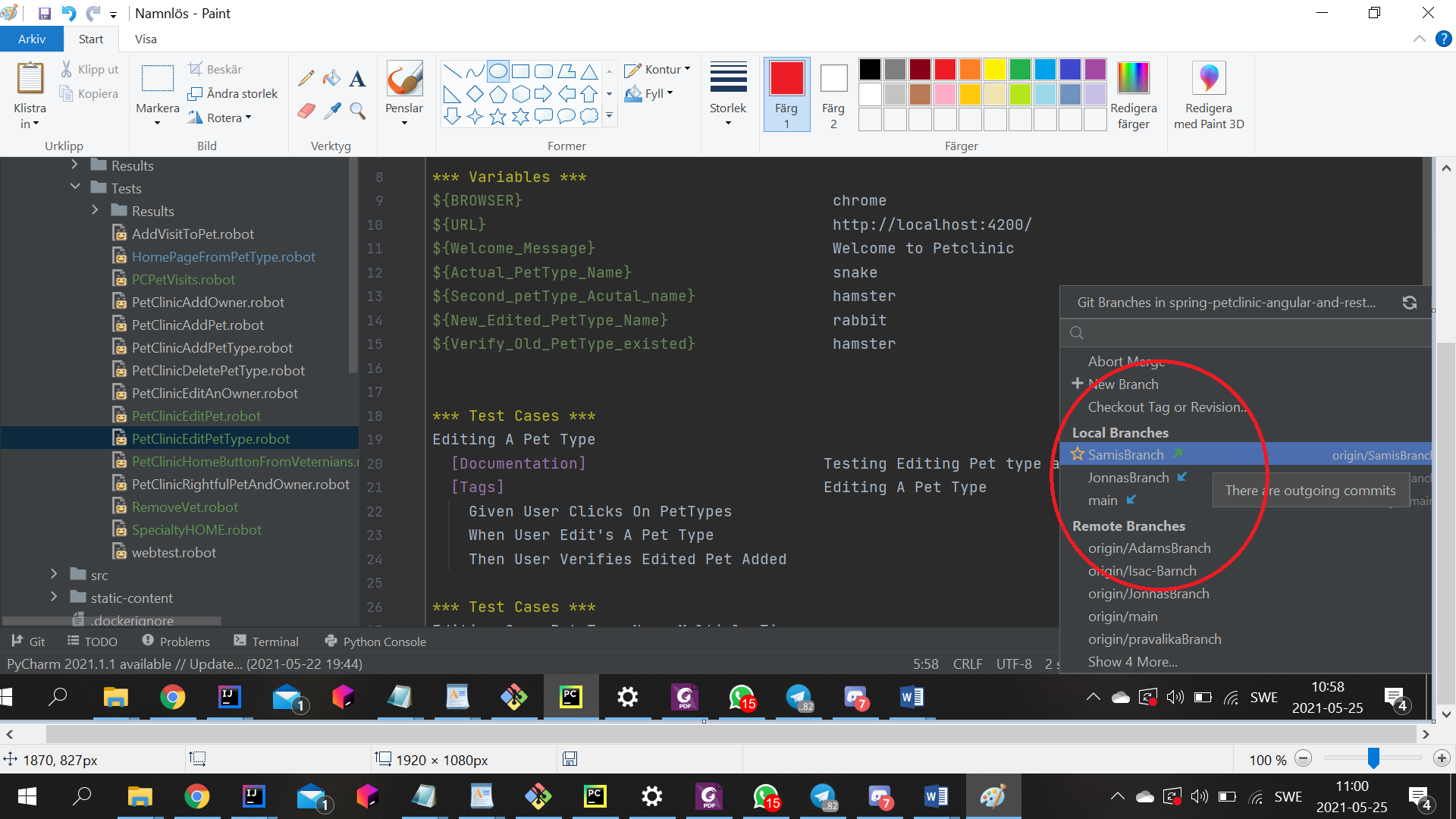
* Open PyCharm
* Add the git clone of <https://github.com/EbrosGaming/spring-petclinic-angular-and-rest-master.git> to the Version Control on Pycharm



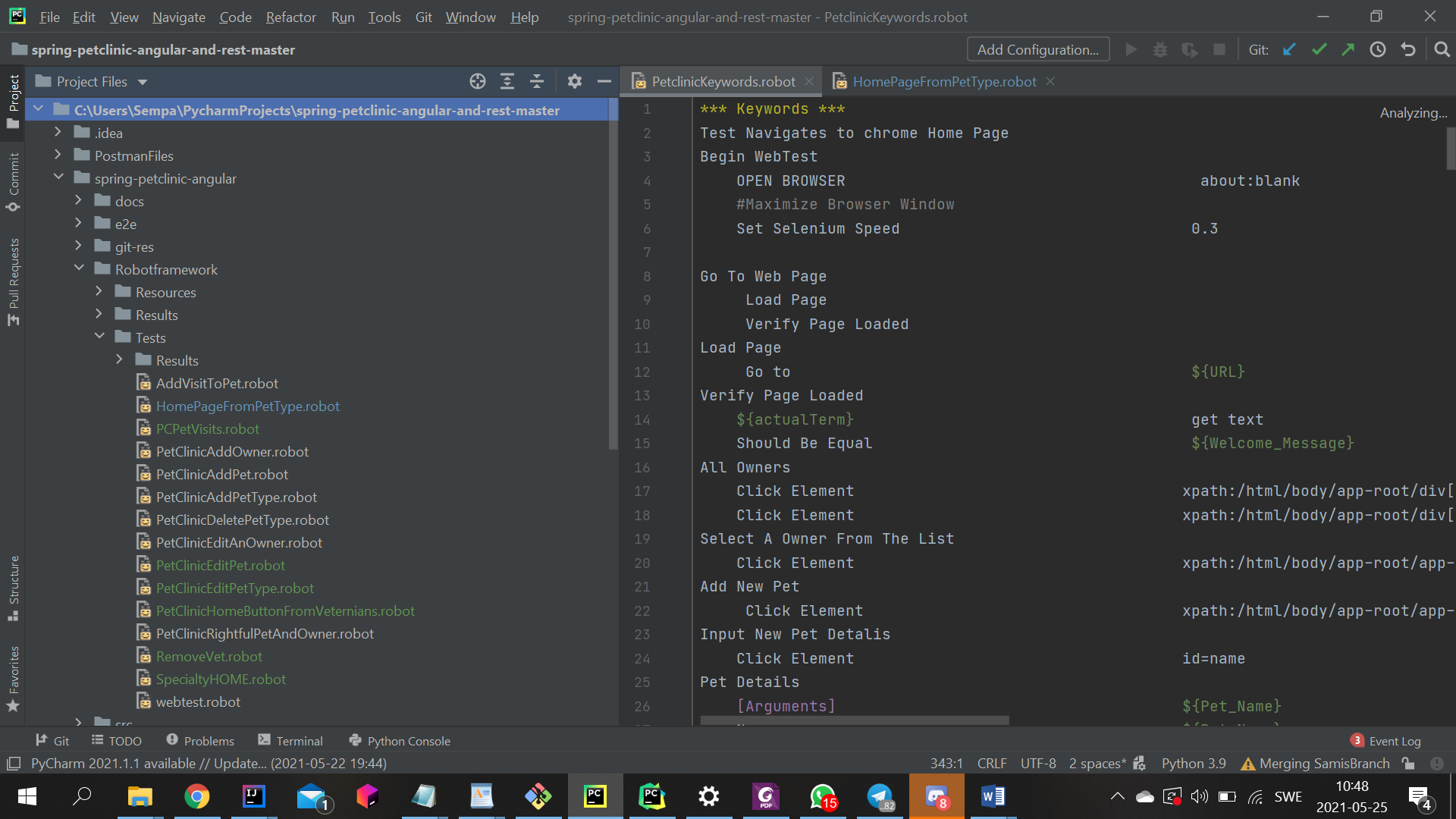
* Now the folder should be added to projects and look like this



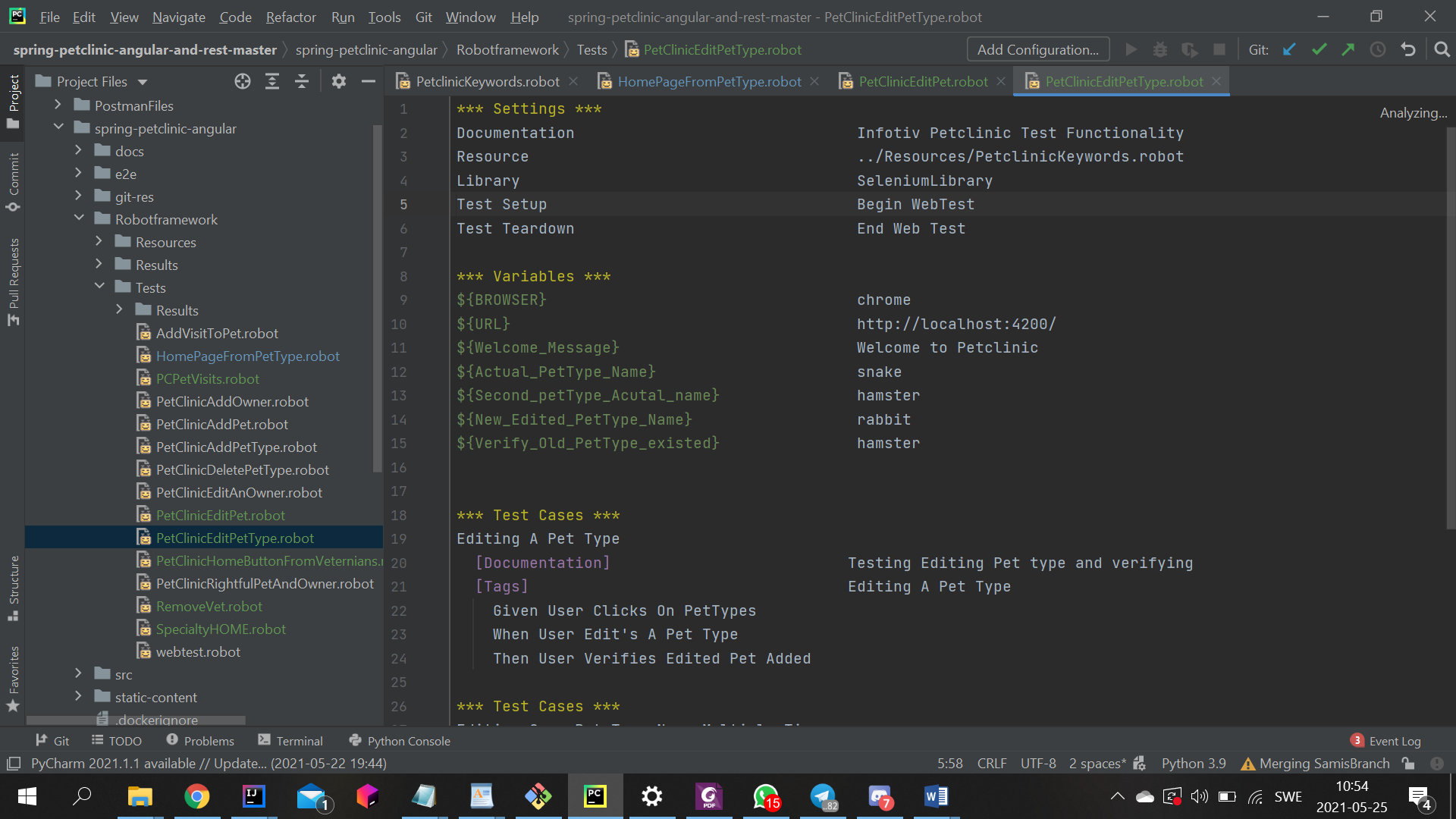
* Se till att du jobbar lokalt I din egna branch



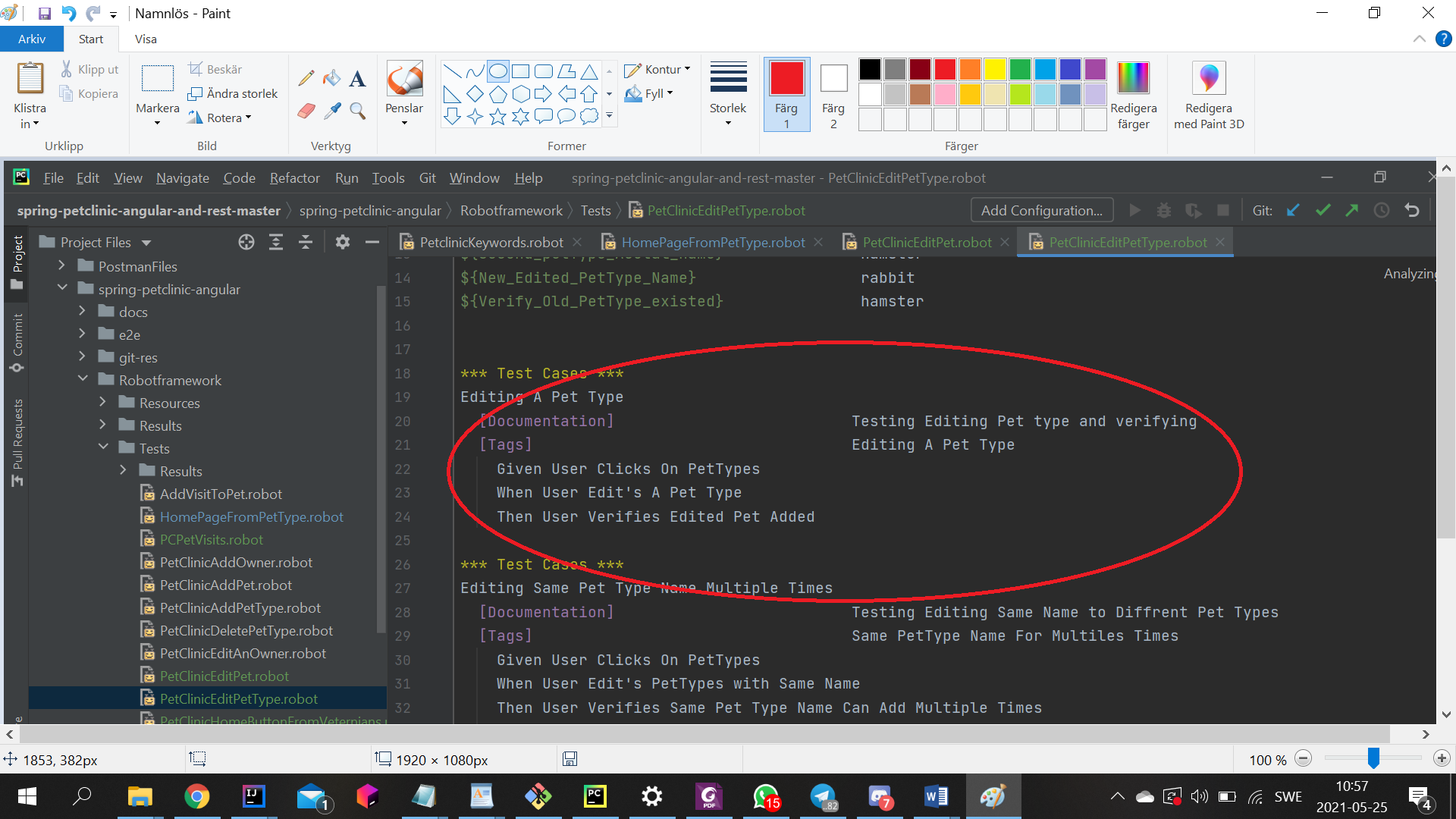
* Now you have access to the keywords file that the whole team uses.
* Add your task to the Tests folder by creating a new file into it



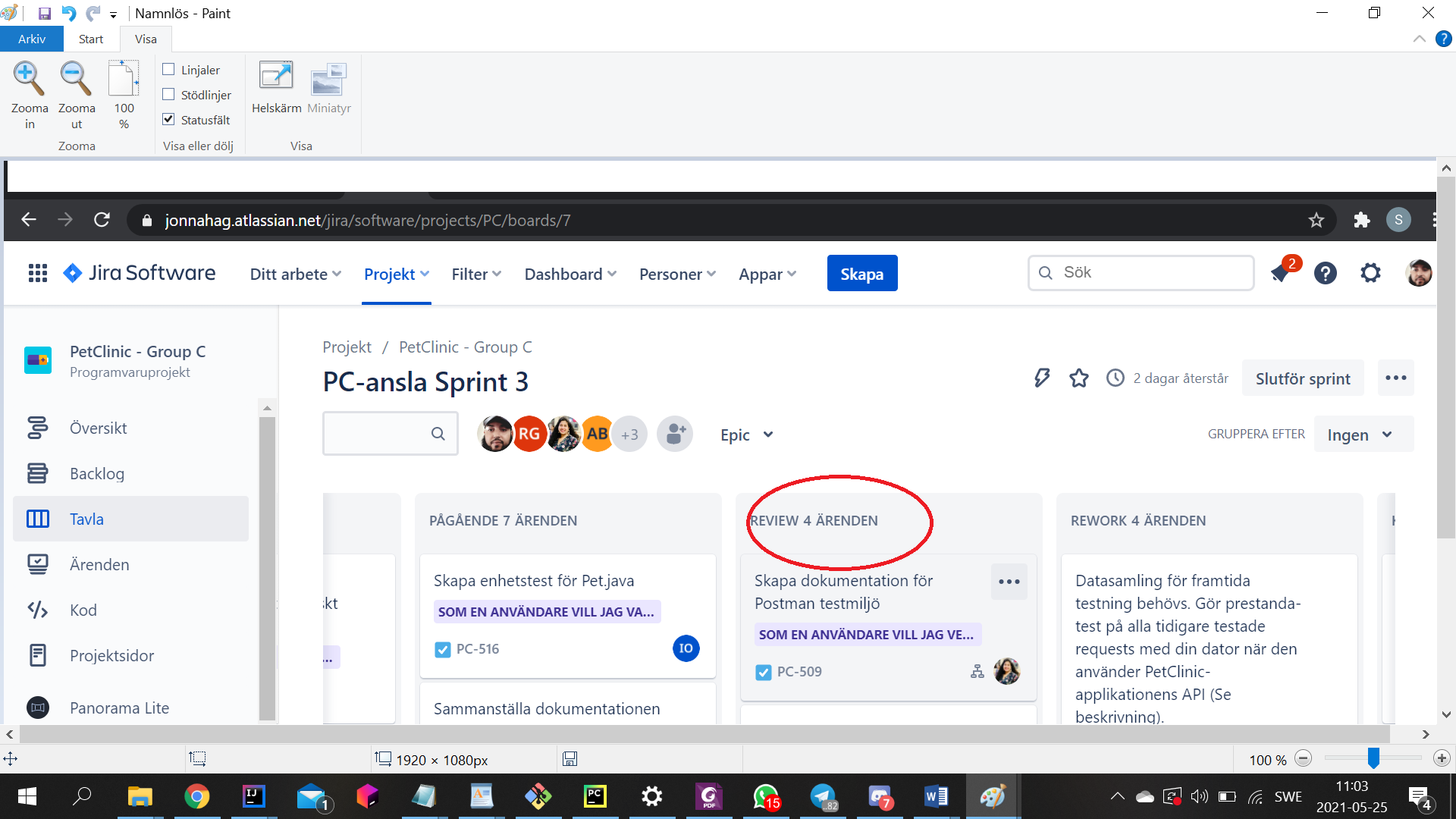
* Use the keywords from Resources
* In documentation describe your test
* Use variables and avoid hardcoding



* Describe the testcase in documentation
* Use Tags för to easier get to/understand wich test it is
* Use gherkin syntex when writing the testcase



* When you have run your test in the terminal and it passes
* Then comit and pusch it to your branch
* Put your task in review on jira so somebody can go through it



* If your test gets greenlight in the review then you push it to the main branch
* And continue with the next task ☺