Prerequisite

Test automation is written in Java and based on following technologies and frameworks. Please get familiar with them:

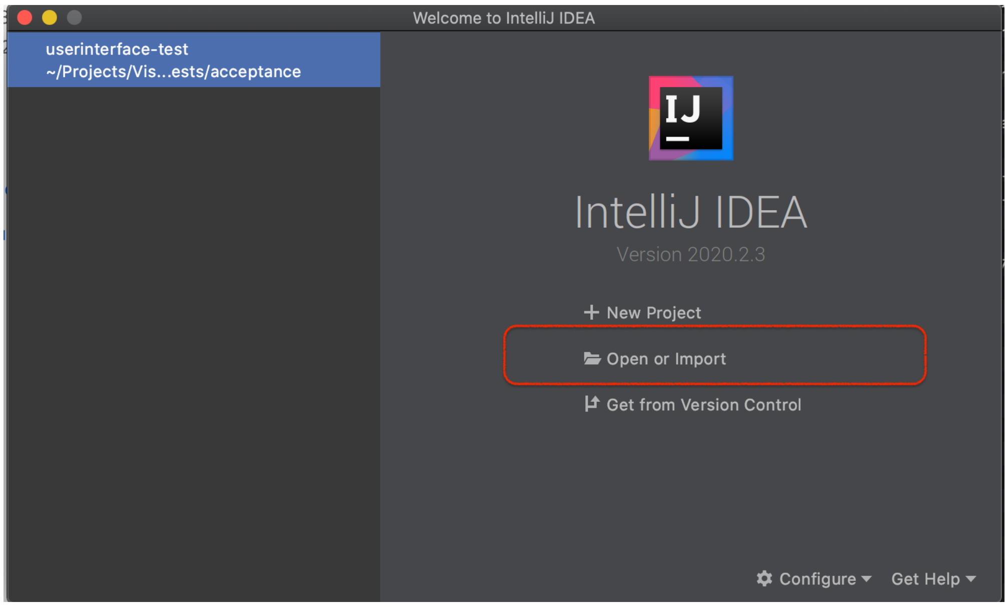
- Maven
- JUnit
- Selenium
- Selenide

Local test execution is usually using a local installed Chrome browser. In CI/CD Pipelines we are using SauceLabs multi-browser support. The tests are running in CI/CD side inside docker container.

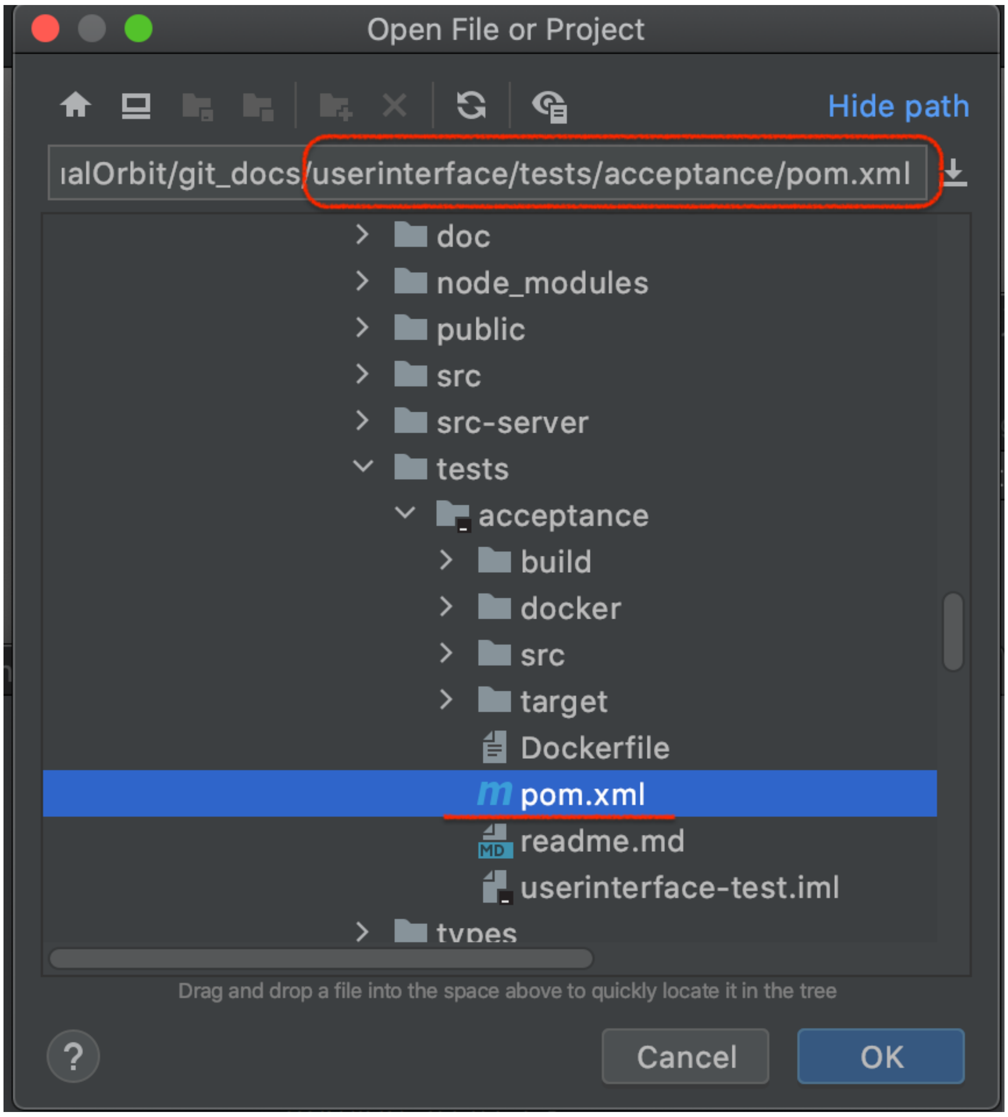
For best practices please check the Selenide project

Setup IDE

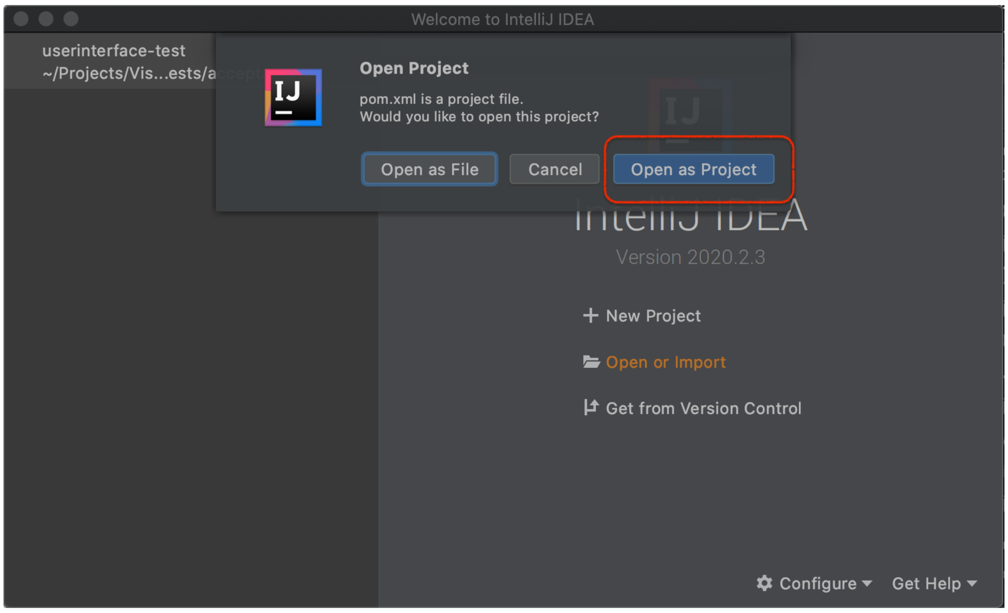
- Download and install Intellij IDEA Community Edition (recommended). You can as well use Eclipse or NetBeans or something else, but it is on your own.

- Import/open test project from userinterface as Maven project:
  Select 'Open or Import':
  

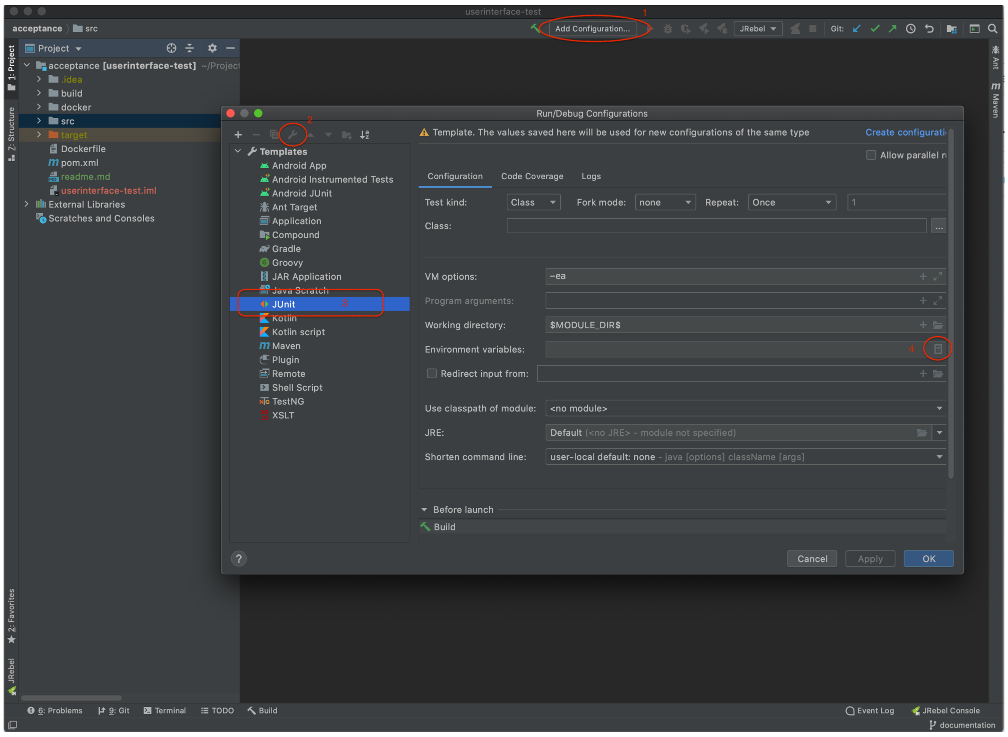
  Select pom.xml file inside the test automation project folder:



Open as Project:



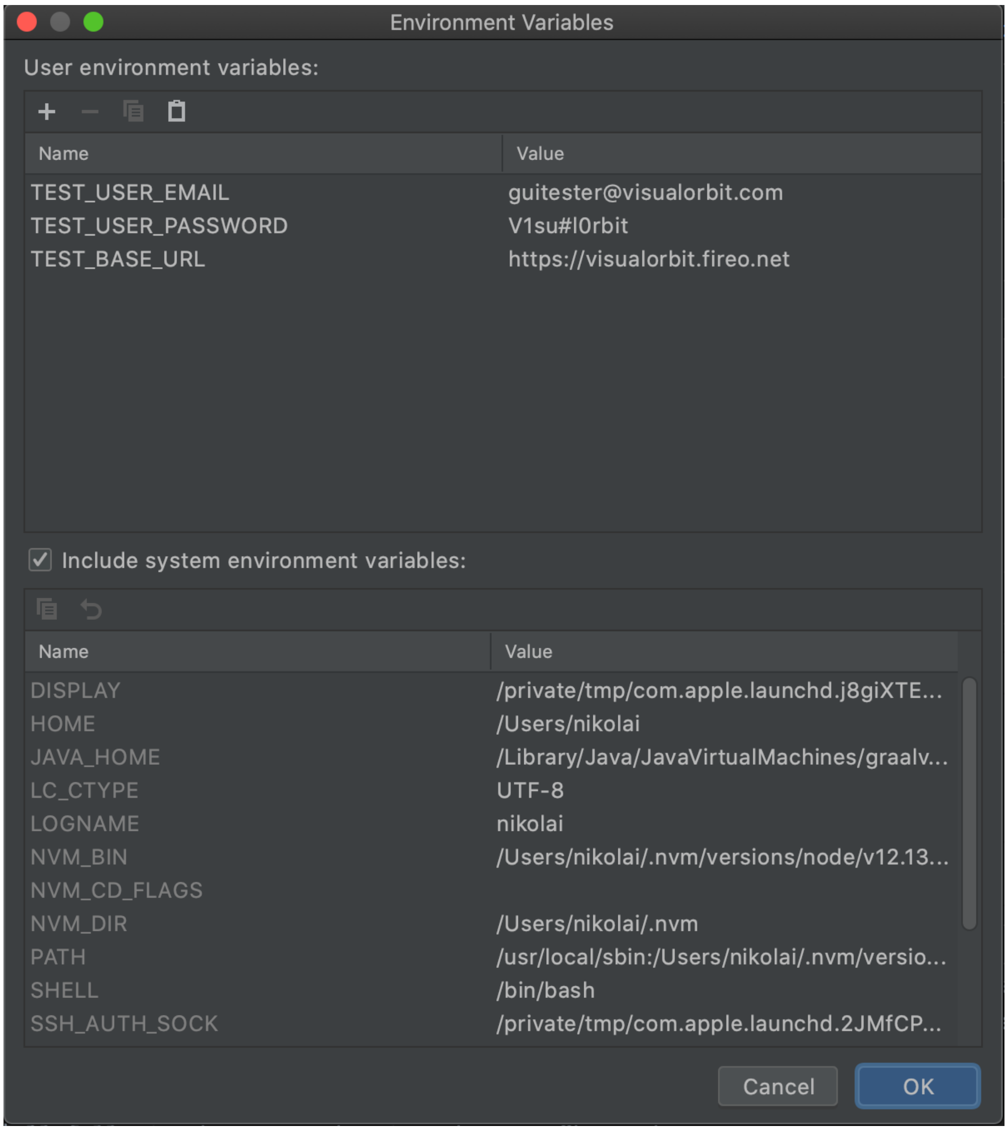
- Setup few environment variables in Run/Debug Configurations via templates. This will add defined environment variables to all tests later.
  Create new JUnit Run/Debug Template:



Template. The values saved here will be used for new configurations of the same type

Test kind: Class
Fork mode: none
Repeat: Once
Class:
VM options: -ea
Program arguments:
Working directory: $MODULE_DIR$
Environment variables:
Redirect input from:
Use classpath of module: <no module>
JRE: Default (<no JRE> - module not specified)
Shorten command line: user-local default: none - java [options] className [args]
Before launch
Build

Fill environment variables. Instead of test user you can use your personal account. Please do not share your password and do not commit it anywhere!



User environment variables:

Name	Value
TEST_USER_EMAIL	guitester@visualorbit.com
TEST_USER_PASSWORD	V1su#l0rbit
TEST_BASE_URL	https://visualorbit.fireo.net

☑ Include system environment variables:

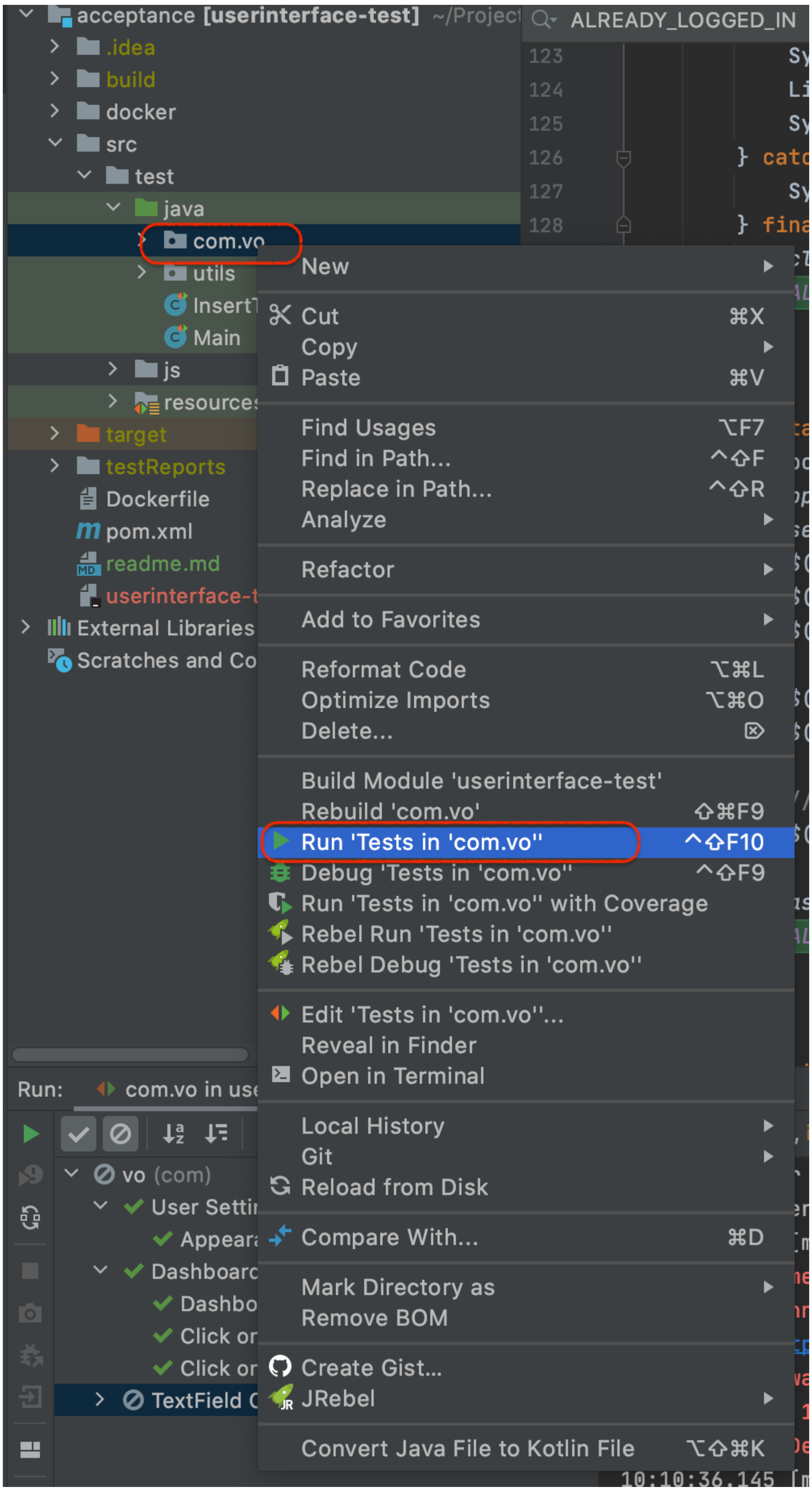
Name	Value
DISPLAY	/private/tmp/com.apple.launchd.j8giXTE...
HOME	/Users/nikolai
JAVA_HOME	/Library/Java/JavaVirtualMachines/graalv...
LC_CTYPE	UTF-8
LOGNAME	nikolai
NVM_BIN	/Users/nikolai/.nvm/versions/node/v12.13...
NVM_CD_FLAGS	
NVM_DIR	/Users/nikolai/.nvm
PATH	/usr/local/sbin:/Users/nikolai/.nvm/versio...
SHELL	/bin/bash
SSH_AUTH_SOCK	/private/tmp/com.apple.launchd.2JMfCP...

Example values:

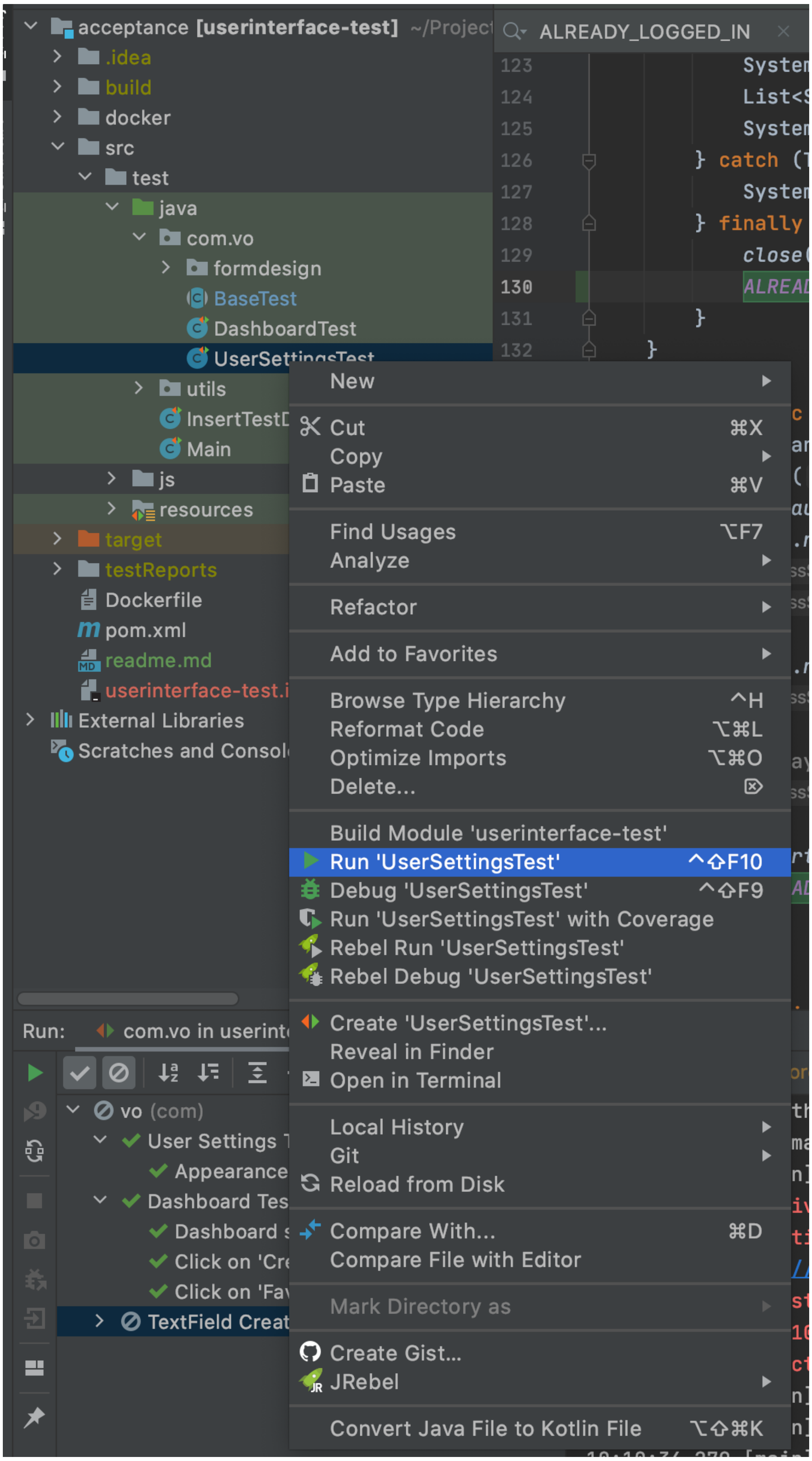
```
TEST_USER_EMAIL=guitester@visualorbit.com;
TEST_USER_PASSWORD=V1su#l0rbit;
TEST_BASE_URL=https://visualorbit.fireo.net
```

Running Tests in IDE

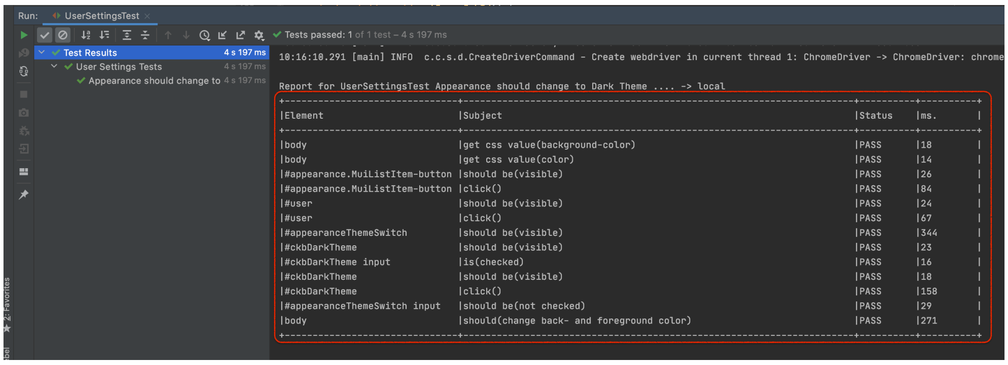
- Run all tests inside `vo.com` package:



- Run specific test by selecting the test class:



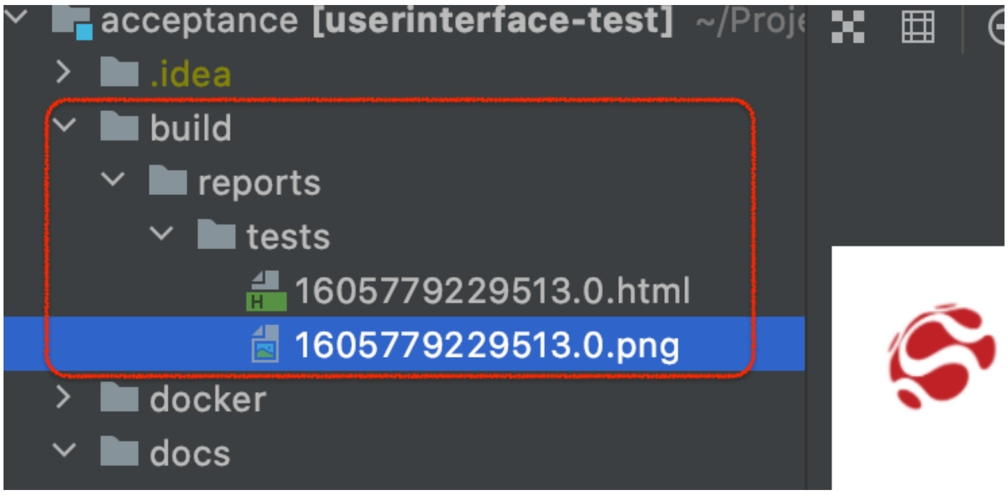
- Each test produces a report in a simple table ascii format and generates screenshots
  Table report:


Run: UserSettingsTest
Tests passed: 1 of 1 test – 4 s 197 ms
Test Results 4 s 197 ms
User Settings Tests 4 s 197 ms
Appearance should change to 4 s 197 ms

```
10:16:10.291 [main] INFO  c.c.s.d.CreateDriverCommand - Create webdriver in current thread 1: ChromeDriver -> ChromeDriver: chrome

Report for UserSettingsTest Appearance should change to Dark Theme .... -> local
+-------------------------------+----------------------------------------------------------------+----------+----------+
|Element                        |Subject                                                         |Status    |ms.       |
+-------------------------------+----------------------------------------------------------------+----------+----------+
|body                           |get css value(background-color)                                 |PASS      |18        |
|body                           |get css value(color)                                            |PASS      |14        |
|#appearance.MuiListItem-button |should be(visible)                                              |PASS      |26        |
|#appearance.MuiListItem-button |click()                                                         |PASS      |84        |
|#user                          |should be(visible)                                              |PASS      |24        |
|#user                          |click()                                                         |PASS      |67        |
|#appearanceThemeSwitch         |should be(visible)                                              |PASS      |344       |
|#ckbDarkTheme                  |should be(visible)                                              |PASS      |23        |
|#ckbDarkTheme input            |is(checked)                                                     |PASS      |16        |
|#ckbDarkTheme                  |should be(visible)                                              |PASS      |18        |
|#ckbDarkTheme                  |click()                                                         |PASS      |158       |
|#appearanceThemeSwitch input   |should be(not checked)                                          |PASS      |29        |
|body                           |should(change back- and foreground color)                       |PASS      |271       |
+-------------------------------+----------------------------------------------------------------+----------+----------+
```

Screenshots in `buld/reports/tests` folder



Best practices to write concise tests

1. Few lines as possible
2. Always chain element selectors with expectations and after that with actions. e.G. `$("#someId").shouldBe(visible).click()`
3. Always use following technics to select elements
    1. search for application related element ids. If not exists, ask developers to add it
    2. search for application related element css classes. If not exists, ask developers to add it
    3. if previous two points not possible (always check and follow up with developers first) use generated locators by chrome plugins (see later)
    4. as last fallback use xpath selectors. But they should be written in an relative manner and be safe against possible movements on the page.

Locating elements on the web page

Install following browser extensions to your Chrome browser:

- ChroPath - to easier lookup for CSS Selectors.
- XPath Helper - to test XPath expressions on te page.
- Selenium IDE - to make recordings and produce first approaches to skript the test.