Taf- todoist

Test automation project (UI and API testing) for *todoist web app.*

Todoist is a web service and a collection of task management software.

By Nataliya Kleschenko https://github.com/NataliyaKleschenko
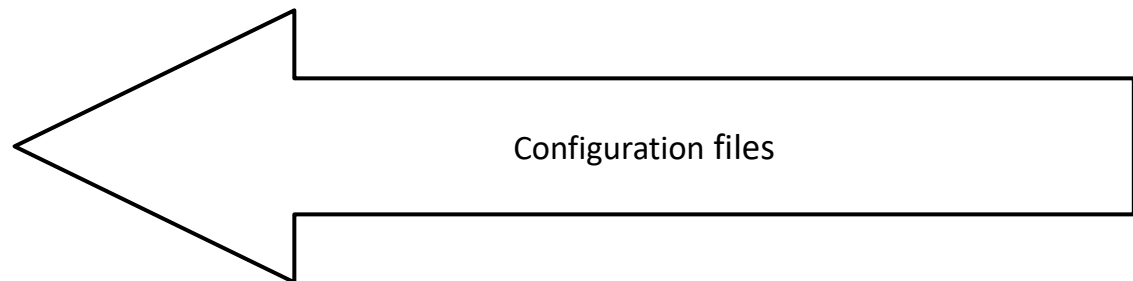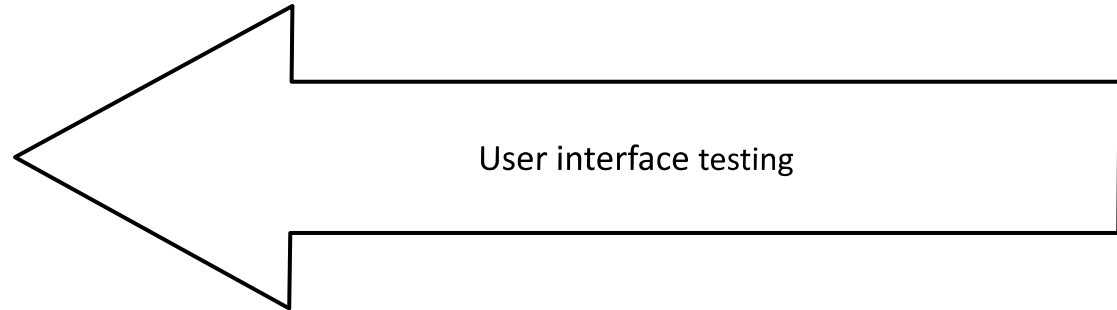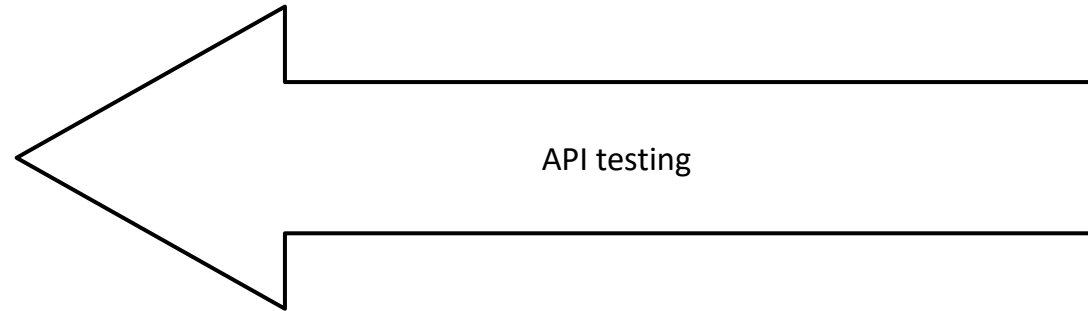
Used technology stack:

The project is written in JAVA, using Gradle as project builder. Selenium is used to test the user interface, Rest Assured is used to test the application programming interface. Allure framework is used for reporting.

Project structure:

```
∨ 📁 src
  › 📁 logs
  ∨ 📁 test
    ∨ 📁 java
      ∨ 📁 com
        ∨ 📁 todoist
          ∨ 📁 api
            › 📁 pojos
            › 📁 specifications
            › 📁 tests
          ∨ 📁 ui
            › 📁 driver
            › 📁 model
            › 📁 pages
            › 📁 steps
            › 📁 tests
            › 📁 util
    › 📁 resources
```

API testing

User interface testing

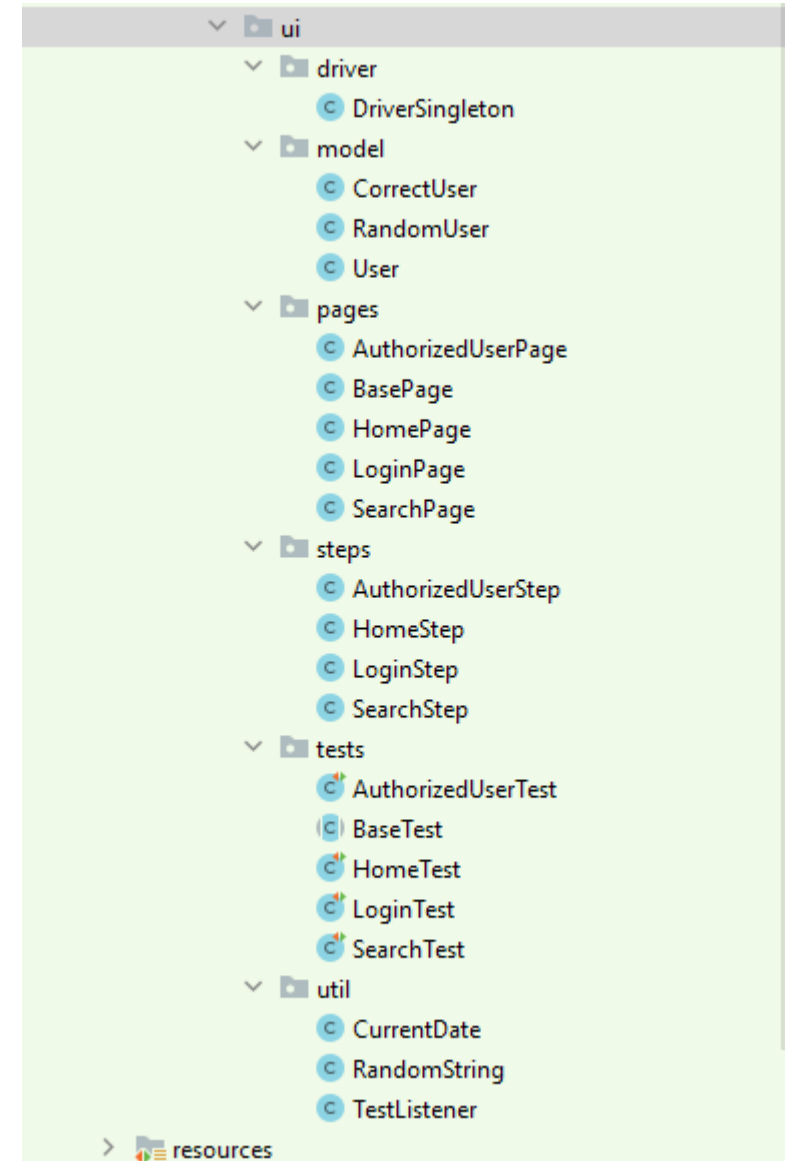Configuration files

UI testing:

To automate the actions of the web browser, I use Selenium WebDriver , having previously connected the library to the project by installing the dependency in the file build.gradle.
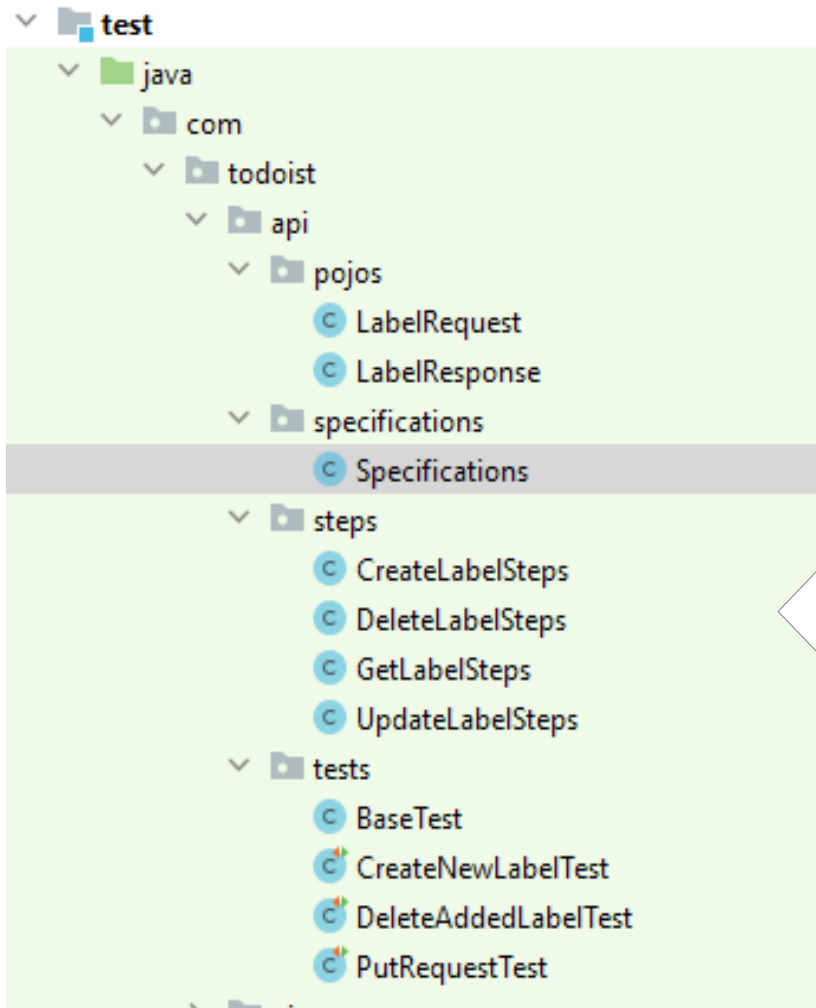
using the Driver Singleton pattern

using the Page object and Pagefactory desing pattern

test execution steps

running tests

```
∨  ui
   ∨  driver
        ©  DriverSingleton
   ∨  model
        ©  CorrectUser
        ©  RandomUser
        ©  User
   ∨  pages
        ©  AuthorizedUserPage
        ©  BasePage
        ©  HomePage
        ©  LoginPage
        ©  SearchPage
   ∨  steps
        ©  AuthorizedUserStep
        ©  HomeStep
        ©  LoginStep
        ©  SearchStep
   ∨  tests
        ©  AuthorizedUserTest
        ©  BaseTest
        ©  HomeTest
        ©  LoginTest
        ©  SearchTest
   ∨  util
        ©  CurrentDate
        ©  RandomString
        ©  TestListener
   >  resources
```

API testing


```
∨  📁 test
   ∨  📁 java
      ∨  📁 com
         ∨  📁 todoist
            ∨  📁 api
               ∨  📁 pojos
                     ⓒ LabelRequest
                     ⓒ LabelResponse
               ∨  📁 specifications
                     ⓒ Specifications
               ∨  📁 steps
                     ⓒ CreateLabelSteps
                     ⓒ DeleteLabelSteps
                     ⓒ GetLabelSteps
                     ⓒ UpdateLabelSteps
               ∨  📁 tests
                     ⓒ BaseTest
                     ⓒ CreateNewLabelTest
                     ⓒ DeleteAddedLabelTest
                     ⓒ PutRequestTest
```

API testing by sending requests and checking status codes, validating payloads with valid data, and checking error messages

Set up a continuous integration process in Jenkins

Reporting

completed test results report: