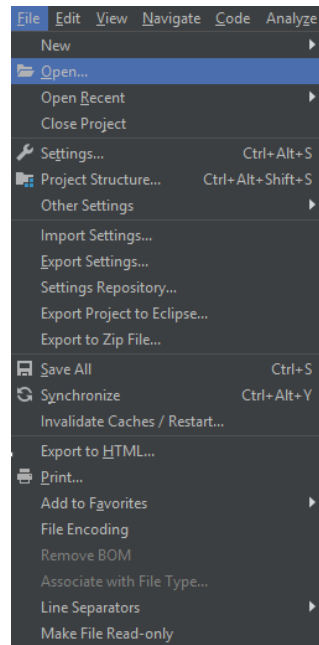


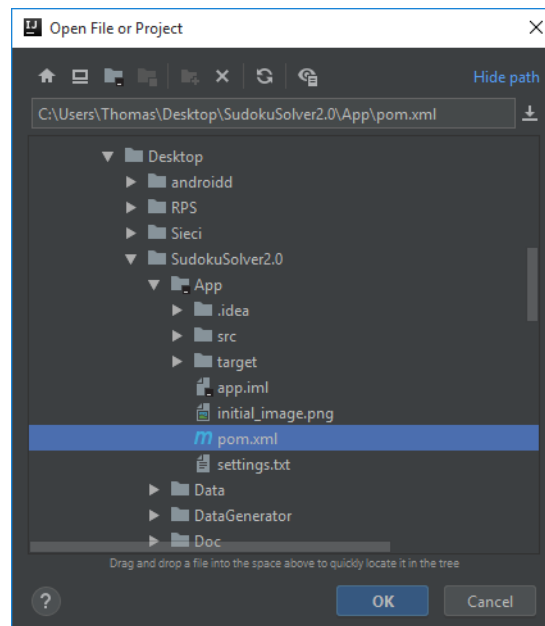
## Kompilacja aplikacji:

Używając IntelliJ w wersji sugerowanej 2018.1.6 przechodzimy kolejno tak jak na zdjęciach:

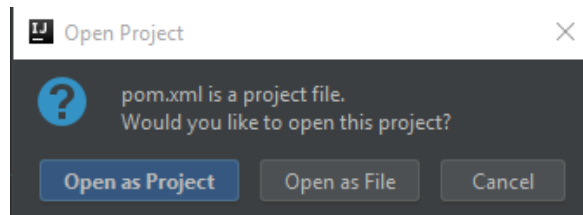
File->Open...



Otwiera się okienko wyboru ścieżki:

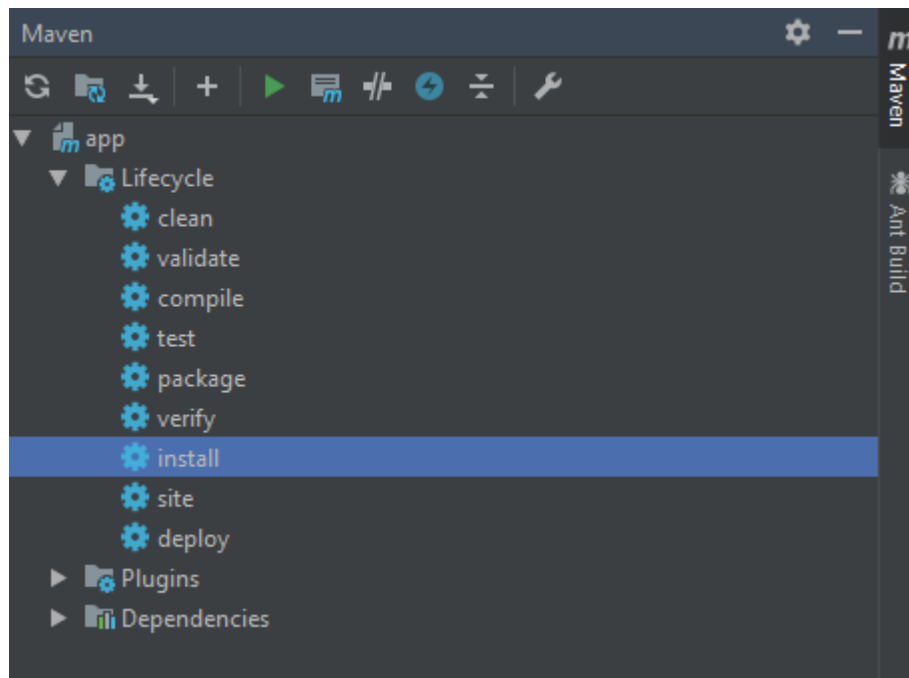


Następnie wybieramy plik pom.xml, który znajduje się w folderze o nazwie App i klikamy przycisk 'OK'.



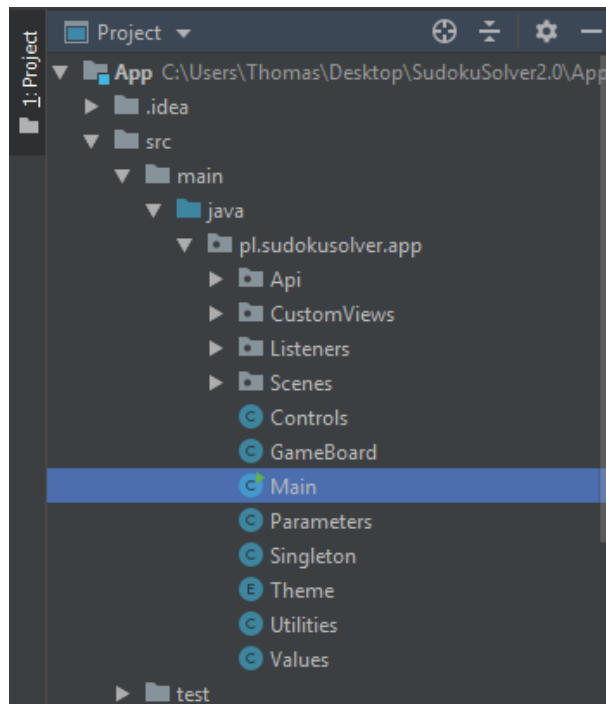
Później wybieramy 'Open as Project'.

Później trzeba otworzyć zakładkę Maven (domyślnie po prawej stronie w IDE), wybrać: app->Lifecycle->install i klikamy dwa razy uruchamiając proces instalacji



W projekcie otwieramy klasę 'Main' znajdującą się:

App->src->main->java->pl.sudokusolver.app->Main



Po otwarciu tej klasy należy kliknąć przycisk uruchomienia znajdujący się na lewo od 'public class Main extends Application {'.

```

1  package pl.sudokusolver.app;
2
3  import ...
7
8  public class Main extends Application {
9
10     @Override
11     public void start(Stage stage){
12         //check if java version is correct
13         if (!Utilities.isJavaValid()) {
14             new StageError(1);
15             return;
16         }
17         Utilities.readFile();
18         new StageMain();
19     }
20
21     public static void main(String[] args) { launch(args); }
24 }
  
```

Instalacja openCV

W przypadku systemu Windows należy pobrać: [https://sourceforge.net/projects/opencvlibrary/files/4.0.1/opencv-4.0.1-vc14\\_vc15.exe/download](https://sourceforge.net/projects/opencvlibrary/files/4.0.1/opencv-4.0.1-vc14_vc15.exe/download) następnie przy użyciu pobranego programu przeprowadzić instalację.

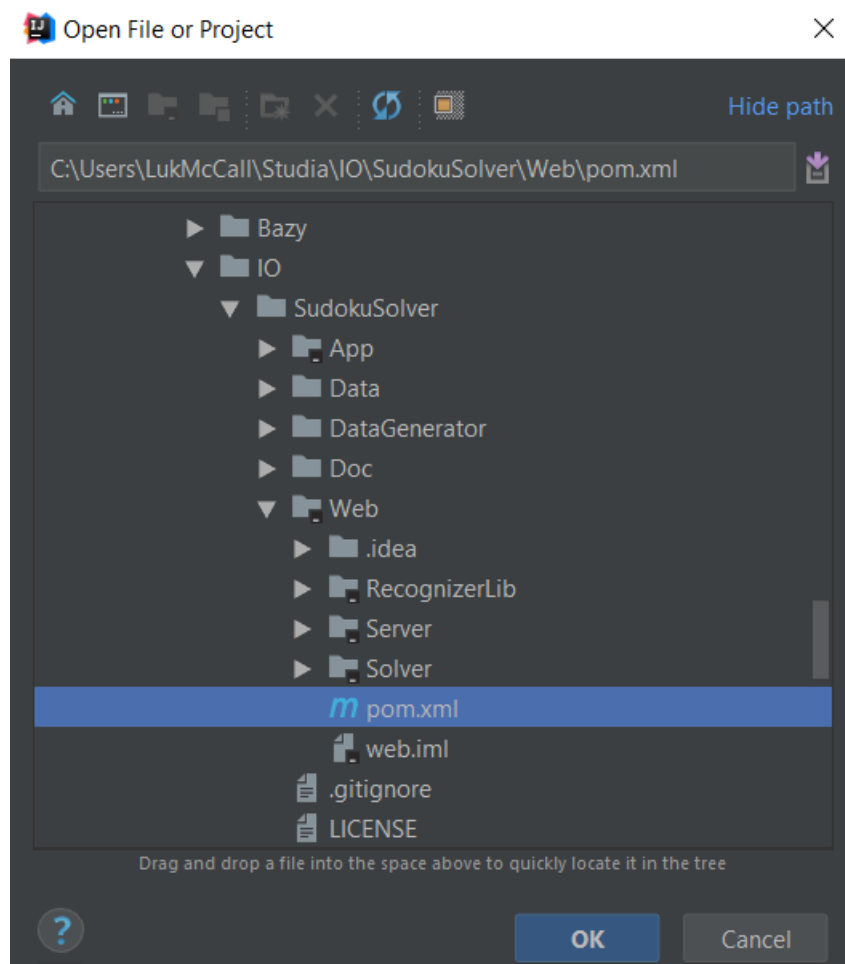
W przypadku innych systemów niż Windows należy skorzystać z oficjalnego poradnika znajdującego się pod [https://docs.opencv.org/4.0.1/d9/df8/tutorial\\_root.html](https://docs.opencv.org/4.0.1/d9/df8/tutorial_root.html).

**Podczas instalacji należy zapamiętać folder, w którym będzie znajdować się biblioteka.**

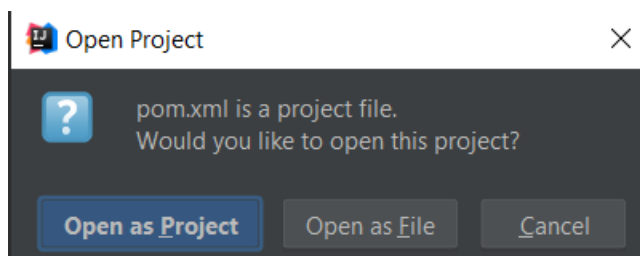
## Kompilacja Web:

Używając IntelliJ w wersji sugerowanej 2018.1.6 przechodzimy kolejno tak jak na zdjęciach:

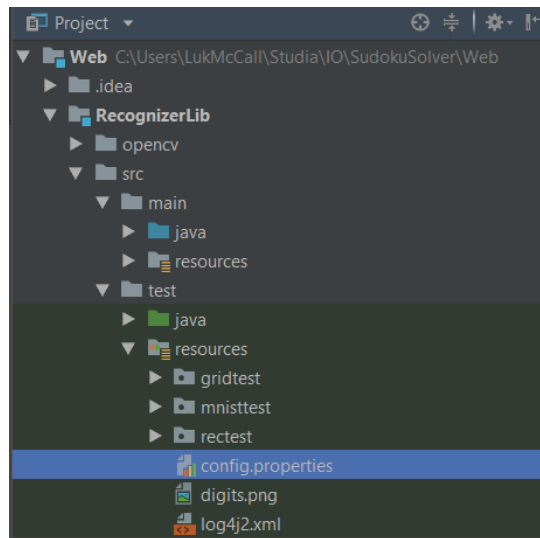
Nawigujemy do pliku *pom.xml* znajdującego się w *Web/pom.xml*



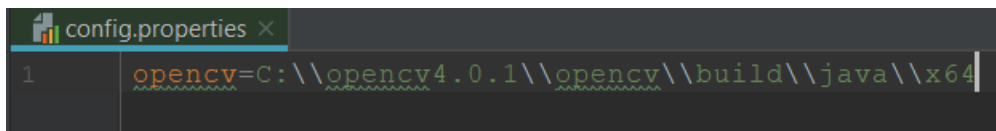
Następnie otwieramy ten plik jako projekt:



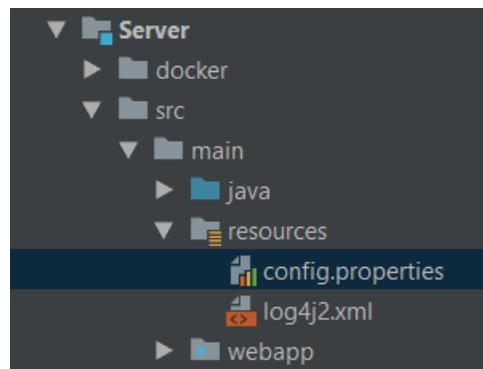
Otwieramy plik config.properties znajdujący się w *RecognizerLib/src/test/resources/*



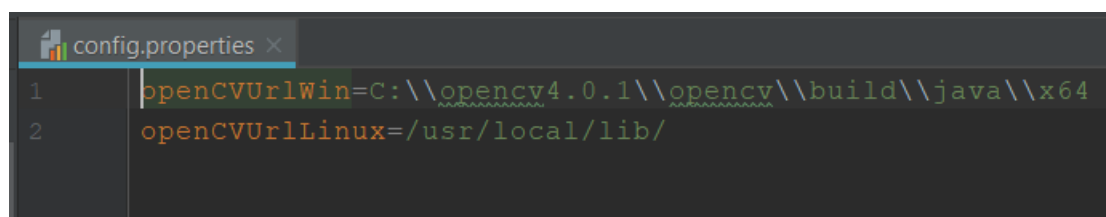
Zmieniamy w nim ścieżkę do opencv. Ustawiamy ją tak aby wskazywała na <folder opencv>/build/java/<wersja systemu>



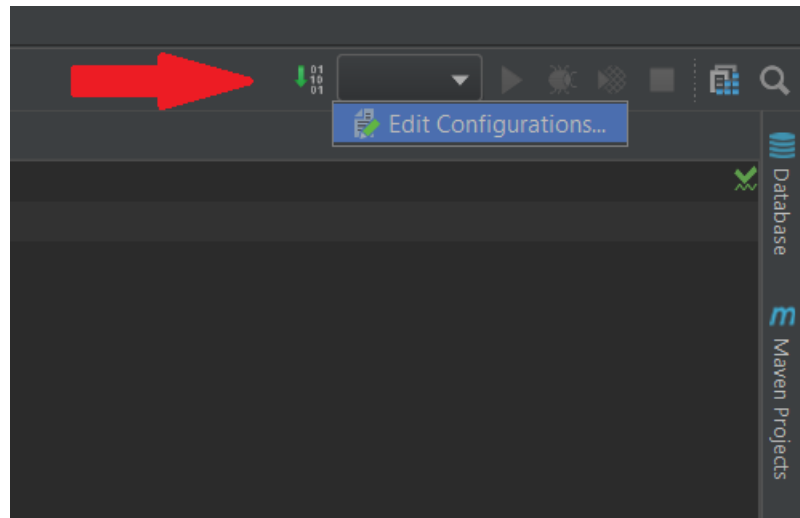
To samo robimy z plikiem config.properties znajdujący się w *Server/src/main/resources/*



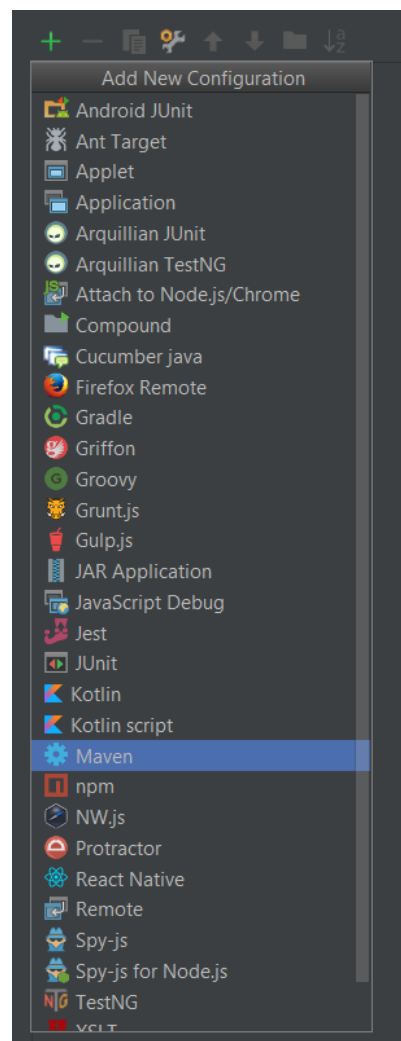
Edytujemy ten plik na tej samej zasadzie co wcześniej z tym że zmieniamy tylko ścieżkę dla aktualnie używanego systemu operacyjnego (zmiana openCVUtilLinux powoduje niemożność uruchomienia serwera w dockerze).



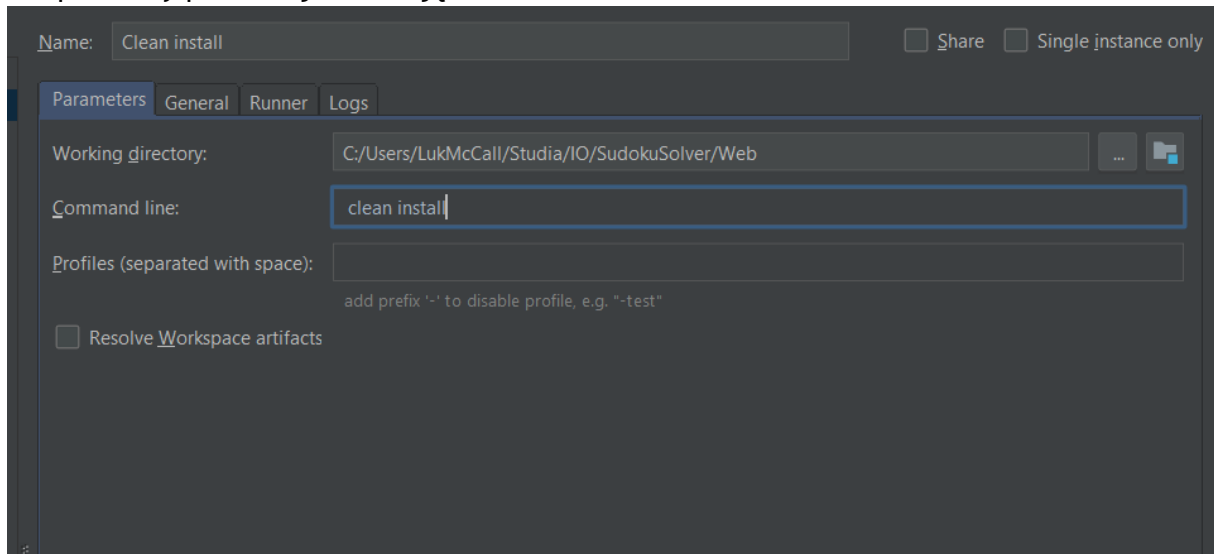
Następnie dodajemy komendę mavena do instalacji wszystkich modułów, aby to zrobić robimy to co na zdjęciach:



Dodajemy nową konfigurację (maven):

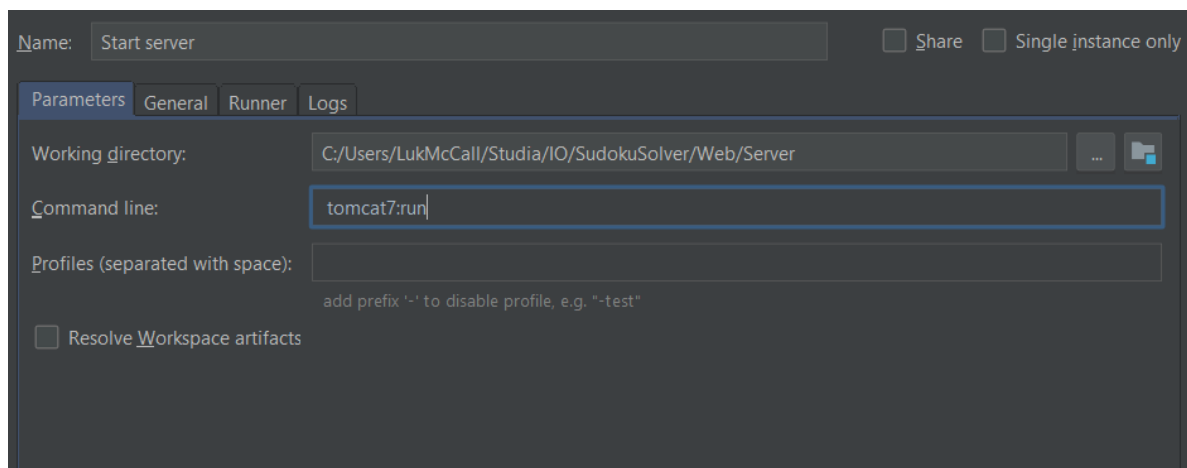


Uzupełniamy pola tak jak na zdjęciu:

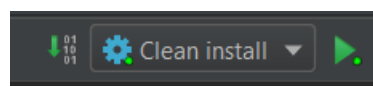


(Working directory ma wskazywać na folder Web)

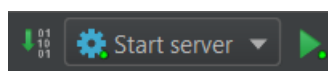
W ten sam sposób dodajemy drugą konfigurację:



Następnie uruchamiamy proces instalacji:



Po czym uruchamiamy serwer:



Serwer jest teraz dostępny pod adresem <http://localhost:8080/>. Można to sprawdzić wchodząc na ten adres przy użyciu przeglądarki.



### Zmiana adresu z którym łączy się aplikacja:

Domyślnie aplikacja klienta łączy się z zewnętrznym serwerem. Aby zmienić to zachowanie należy wejść w *App\src\main\java\pl\sudokusolver\app\Values.java* i w linii 8 należy zmienić

```
public static final String SERVER_URL = <tu nowy adres >;
```

Adres serwera uruchomionego w dockerze to *http://localhost:<wybrany port>/app/*

### Uruchomienie serwera przy pomocy dockera:

Po przeprowadzeniu instalacji serwera otwieramy wiersz poleceń i nawigujemy do folderu *Web\Server\docker\base*.

Następnie należy wykonać podaną komendę:

```
docker build --tag baseserver -f .\Dockerfile .
```

Proces ten może potrwać ponad godzinę.

Jeśli proces zakończy się powodzeniem należy wpisać podane komendy:

```
cd ..\server\
```

```
docker build --tag sudokuserver -f .\Dockerfile .
```

Serwer jest już gotowy do uruchomienia, aby to uczynić należy wykonać:

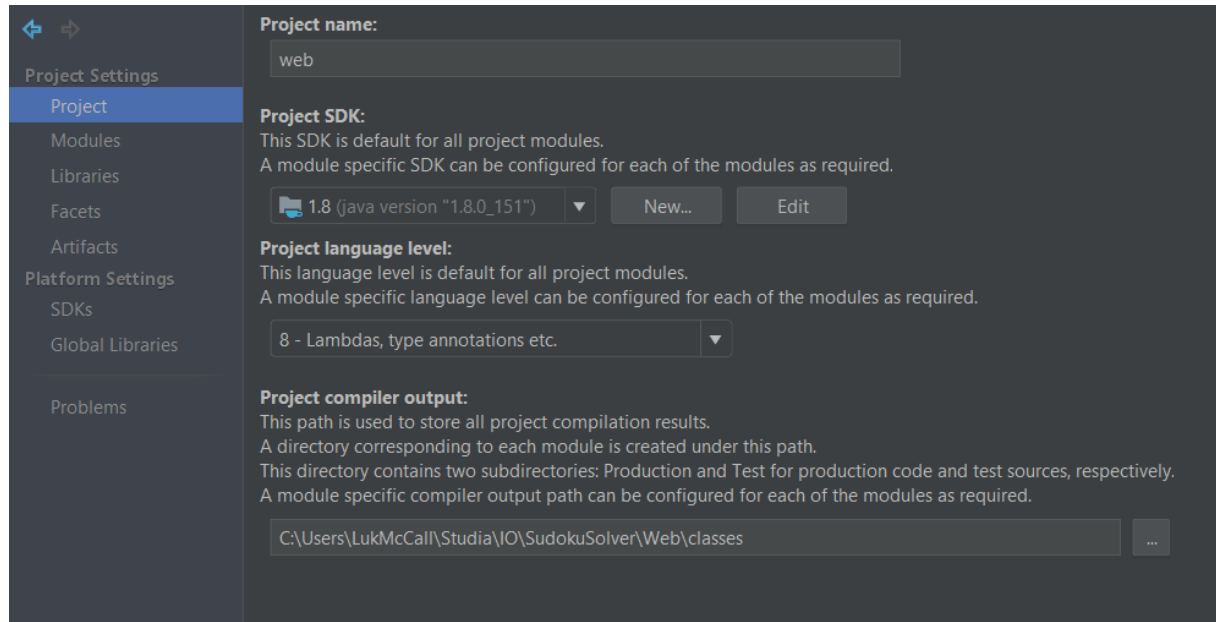
```
docker run --name server -it -p 8080:8080 sudokuserver
```

Po czym serwer zostanie uruchomiony w trybie interaktywnym.

Podczas budowania pojemnika o nazwie baseserver (pierwsza komenda) mogą wystąpić nieprzewidziane przez nas błędy. Wynika to z zależności tego obrazu od pewnej konfiguracji sieciowej. W celu ukończenia instalacji należy rozwiązać te problemy ręcznie, po czym jeszcze raz wykonać podaną komendę.

## Uwagi:

Projekt musi być kompilowany za pomocą javy sdk 1.8:



W przypadku błędnego ustawienia ścieżki do openCV podczas uruchomienia testów jednostkowych wystąpi błąd:

```
[ERROR] pl.sudokusolver.recognizerlib.data.MNISTReaderPrivateTest Time elapsed: 0.018 s <<< ERROR!  
java.lang.UnsatisfiedLinkError: Can't load library: C:\opencv4.0.1\opencv\build\x64\opencv_java401.dll
```

Podczas działania serwera nie możemy uruchamiać żadnej innej aplikacji, która nasłuchiwałaby na porcie 8080.