**Serwer**

**Używane technologie**

Do budowania oraz zarządzania projektem korzystamy z maven wraz z kompilatorem javy w wersji 8. Projekt serwera został podzielony na 3 moduły.

Pokrótce opiszę technologie użyte przy ich tworzeniu:

* Solver – czysta java 8.
* RecofnizerLib
  + OpenCV 4.0.1 – biblioteka do rozpoznawania obrazu oraz do nauczania maszynowego.
  + Google guava 16.0.1 – zbiór pomocniczych funkcji i pojemników.
  + Tess4j 3.5.3 – wrapper do zewnętrznego ocra.
* Server
  + Spring MVC 5.1.6 – framework do tworzenia aplikacji webowych w javie.
  + Commons-fileupload 1.4 – dodatek do springa umożliwiający przetwarzanie zapytania http składającego się z wielu części (np. do odebrania zdjęcie).
  + Gson 2.8.5 – bilblioteka do parsowania jsona.

Serwer został przeniesiony w wirtualne środowisko przy pomocy dockera. Dzięki temu możemy w łatwy sposób instalować go oraz przenosić na różne platformy.

**Użyte algorymy**

Opisane w *Doc\Requirements*

**Użyte wzorce projektowe**

* Singleton
  + Tworzenie nowych obiektów ocrów jest bardzo kosztowane i czasochłonne temu też zdecydowaliśmy się opakowac je w ten wzorzec projektowy. Umożliwiło nam to swobodny dostęp do tych obiektów.
* Builder
  + Podczas każdego zapytania o ekstrakcjie sudoku tworzona jest nowa instancja BaseSudokuExtractora, który zarządza całym procesem ekstrakcji danych ze zdjęcia. Obiekt ten musi zostać stworzony na nowo, ponieważ korzysta on z danych przesłanych wraz z zapytaniem. Niestety konstrukcja tego obiektu jest skomplikowana dlatego też postanowiliśmy zastosować ten wzorze, aby ułatwić konstruowanie tak skomplikowanego obiektu oraz móc łatwiej zarządzać procesem tworzenia.
* Stategy
  + Najważniejsze składowe BaseSudokuExtractora są reprezentowane przez odpowiednie klasy. Dzięki takiemu rozwiązaniu możemy w łatwy sposób rozszerzać naszą RecognizerLib oraz w przyjemny sposób testować nasze oprogramowanie.