



AKADEMIA GÓRNICZO-HUTNICZA  
IM. STANISŁAWA STASZICA W KRAKOWIE

INŻYNIERIA OPROGRAMOWANIA

PODRĘCZNIK

INSTALACJA, KONFIGURACJA, URUCHAMIANIE

---

## Wish List

---

*Autorzy:*

Łukasz GAJEWSKI

Grzegorz SIATKA

Patrycja WRONA

2016

# Spis treści

<b>1</b>	<b>Aplikacja</b>	<b>2</b>
1.1	Wymagania . . . . .	2
1.2	Repozytorium . . . . .	2
<b>2</b>	<b>Serwer</b>	<b>3</b>
2.1	Instalacja . . . . .	3
2.2	Uruchamianie . . . . .	3
2.3	Konfiguracja . . . . .	4
<b>3</b>	<b>Aplikacja mobilna - Android</b>	<b>5</b>
3.1	Instalacja i uruchamianie . . . . .	5
3.1.1	Gradle . . . . .	5
3.1.2	Android Studio . . . . .	5
3.2	Konfiguracja . . . . .	5
<b>4</b>	<b>Aplikacja webowa</b>	<b>7</b>
4.1	Instalacja . . . . .	7
4.2	Uruchamianie . . . . .	7
4.3	Konfiguracja . . . . .	7

# 1 Aplikacja

## 1.1 Wymagania

- JDK 1.8
- Gradle 2.3 lub nowszy
- Maven 3.0 lub nowszy
- Git 2.1.4 lub nowszy
- IntelliJ IDEA 2016.1
- NodeJS 4.4.5 w wersji LTS
- MongoDB 3.2.6 lub nowsze
- Android SDK

## 1.2 Repozytorium

Projekt składa się z 3 zasadniczych części (modułów):

- część serwerowa
- część mobilna, system Android
- część webowa

Wszystkie moduły zostały umieszczone w jednym, centralnym repozytorium na platformie GitHub:

|| <https://github.com/agh-ki-io/WishList>

Każdy z modułów jest umieszczony w osobnym katalogu, odpowiednim do jego nazwy:

- moduł serwerowy - 'server'
- moduł mobilny - 'android'
- moduł webowy - 'web'

Pozostałe katalogi dotyczą dokumentacji lub osobnych komponentów aplikacji. W celu uruchomienia oraz konfiguracji aplikacji nie jest konieczna ich zmiana lub osobna konfiguracja.

W celu pobrania całego repozytorium należy wykonać poniższe polecenie w terminalu:

```
|| git clone https://github.com/agh-ki-io/WishList.git
```

W aktualnie otwartym katalogu w terminalu pojawi się nowy o nazwie zgodnej z nazwą repozytorium.

## 2 Serwer

### 2.1 Instalacja

W celu uruchomienia części serwerowej należy otworzyć terminal w głównym katalogu pobranego repozytorium.

Projekt został skonfigurowany zgodnie z założeniami narzędzia do budowania Maven, dlatego wystarczy wykonać poniższą komendę, aby otrzymać świeżą, nowo zbudowaną aplikację, gotową do uruchomienia.

```
|| mvn clean install
```

Polecenie wyczyści wcześniej zbudowane moduły oraz przystąpi do budowania wszystkich zależnych od siebie podmodułów - krok po kroku, mając na uwadze zależności pomiędzy nimi.

Ponadto, polecenie 'install' wywołuje testy z każdego modułu osobno.

**Ważne!** Aby budowanie odbyło się prawidłowo należy posiadać działającą bazę MongoDB na porcie 27017.

Po kilkunastu sekundach otrzymujemy komunikat potwierdzający zbudowanie każdego z podmodułów, z rozpisanymi czasami budowania podanym w sekundach.

### 2.2 Uruchamianie

Upewnij się, że proces instalacji przeszedł prawidłowo. Następnie przejdź do katalogu 'server/target', gdzie znajdziesz zbudowaną aplikację, spakowaną do formatu ".jar". Aby uruchomić wpisz poniższą komendę w konsoli:

```
|| java -jar server-1.0-SNAPSHOT.jar
```

Proces uruchamiania buduje aplikację w oparciu o platformę Spring Boot. Po poprawnie wystartowanym serwerze, zainicjalizowaniu serwisów oraz servletów uruchamia się serwer na domyślnym porcie. Status, wraz z czasem startowania aplikacji dostajemy w konsoli:

```
|| Started Application in 16.413 seconds (JVM running for 19.086)
```

## 2.3 Konfiguracja

Całość konfiguracji modułu serwerowego opiera się na zapewnionym mechanizmie Properties. W katalogu z zasobami modułu 'serwer' znajdziemy pliki konfigurowalne.

Zgodnie z założeniami narzędzia do budowania Maven znajdują się one w katalogu 'server/src/main/resources'.

W katalogu znajdziemy następujące pliki, które można poddać konfiguracji:

- application.properties - możliwość zmiany portu, na którym działa serwer (domyślnie 80).

```
|| server.port=80
```

- email.properties - zmiana właściwości modułu mailowego, konfiguracja SMTP. Możliwość konfiguracji hostu, portu, używanego protokołu oraz loginu i hasła do konta mailowego.

```
|| smtp.host=smtp.gmail.com
|| smtp.port=465
|| smtp.protocol=smtps
|| smtp.username=register.in.wishlist
|| smtp.password=w1$hL15T
```

## 3 Aplikacja mobilna - Android

### 3.1 Instalacja i uruchamianie

W celu uruchomienia części serwerowej należy otworzyć terminal w katalogu 'android' pobranego repozytorium.

#### 3.1.1 Gradle

Projekt został skonfigurowany zgodnie z założeniami narzędzia do budowania Gradle przy współpracy z Gradle Wrapper działającym między platformowo, dlatego wystarczy wykonać poniższą komendę, aby otrzymać świeżą, nowo zbudowaną aplikację, gotową do uruchomienia.

Informacja na temat dostępnych zadań:

```
|| ./gradlew tasks
```

Zbudowanie aplikacji mobilnej:

```
|| ./gradlew build
```

Polecenie wyczyści wcześniej zbudowane moduły oraz przystąpi do budowania wszystkich zależnych od siebie podmodułów - krok po kroku, mając na uwadze zależności pomiędzy nimi.

Ponadto, polecenie 'build' tworzy plik instalacyjny ".apk", który można wgrać bezpośrednio na telefon dzięki komendzie "adb", dostarczonej przez Android SDK.

```
|| adb install android/app/build/outputs/apk/app-debug.apk
```

#### 3.1.2 Android Studio

Moduł 'android' jest także w pełni importowalny przez jedyne, wspierane przez producenta IDE - Android Studio. W przypadku korzystania z IDE należy uruchomić konfigurację o nazwie "app" z podpiętym telefonem w trybie debugowania. Aplikacja zbuduje się i zainstaluje automatycznie na urządzeniu.

### 3.2 Konfiguracja

Całość konfiguracji modułu serwerowego opiera się na zapewnionym mechanizmie pobierania 'property' dostarczonym przez producentów systemu Android. W katalogu z

zasobami modułu 'android/app' znajdziemy pliki konfigurowalne.

Zgodnie z założeniami narzędzia do budowania Maven+Android znajdują się one w katalogu 'android/app/src/main/res/values'.

W katalogu znajdziemy następujące pliki, które można poddać konfiguracji:

- colors.xml - zmiana kolorów layout'u, podana w kodzie heksadecymalnym.
- dims.xml - zmiana odstępów między komponentami w aplikacji
- strings.xml - zmiana domyślnego zestawu host:port, służącego do łączenia się z serwerem, możliwość zmiany wyświetlanych komunikatów w aplikacji

## 4 Aplikacja webowa

### 4.1 Instalacja

W celu uruchomienia części serwerowej należy otworzyć terminal w katalogu 'web' pobranego repozytorium.

Projekt został skonfigurowany, aby współpracować z narzędziami Node Package Manager oraz Bower, które służą instalowaniu zależności. Przed pierwszym uruchomieniem należy je pobrać za pomocą komend:

```
|| npm install -g bower  
|| npm install
```

Pierwsze polecenie pobierze plugin 'Bower' i zainstaluje go globalnie, a następne przeanalizuje wszystkie zależności zdefiniowane w plikach package.json oraz bower.json oraz umieści je w odpowiednich katalogach.

Z powodu, że JavaScript to język skryptowy, dlatego nie zetkniemy się w tym kroku z fazą kompilacji i budowania. Praktycznie aplikacja jest gotowa do postawienia na serwerze.

### 4.2 Uruchamianie

W celu przyspieszenia developmentu oraz uzyskania natychmiastowego rezultatu należy zainstalować dodatkowy moduł zwany 'Http Server' umożliwiający natychmiastowe włączenie serwera pod wskazanym portem. Uruchamiamy poniższe komendy w konsoli:

```
|| npm install -g http-server  
|| hs .
```

### 4.3 Konfiguracja

Konfiguracja aplikacji webowej sprowadza się do uzupełnienia informacji na temat pary host:port, na której postawiony jest serwer.

W pliku 'js/app.js' znajdziemy zmienną newBaseUrl, która została przypisana do domyślnego serwera:

```
|| newBaseUrl = "http://nat-1.d17.iisg.agh.edu.pl:60680/"
```