### CONTACT MANAGEMENT APP

## Introduction

*Contact Management App* is an app which allows the user to create Contacts and make necessary amendments to the ones which were created.

The whole idea of the app was to write an application in Spring Boot which allows the user to test the REST-endpoints contained in the app in an easy and flexible manner.

It is also important to mention that the app supports four basic CRUD operations:

-CREATE

-READ

-UPDATE

-DELETE

I made use of POST-man to test the REST endpoints contained in the app. You may want to use another application to check if the application returns the relevant data.

Please feel free to explore this tutorial to get to know more about the application, how it’s structured and how it works.

## Contacts and what are they used for:

A Contact can be described as a set of user properties that is injected into POSTMAN and the relevant output is then returned.

After the Contact was created it can be amended, deleted, retrieved etc.

The Contact which the user creates within POST-man (or any REST-ful API application) must contain the following fields in the specified format:

**Contact (**

**firstName : String**

**lastName: String**

**Date of Birth: Array<Integer>**

**address: String**

**mobileNumber : Integer )**

That contact can then be declared as JSON.

In the code snippet below, I present an example JSON which contains all the properties which were mentioned above:

 {

        "firstName": "Janusz",

        "lastName": "Kowalski",

        "dateOfBirth": [

            1989,

            12,

            12

        ],

        "address": "Wiśniowa 2a",

        "mobileNumber": 111222333

    }

The JSON object can then be inserted into POSTMAN.

(or any other REST-ful platform used for building and testing API-s)

The below illustration shows the whole process:

A screenshot of a computer

Description automatically generated

## CRUD OPERATIONS

There are four possible CRUD operations which can be executed with the use of POSTMAN to expose the application functionality:

-CREATE

Executing the POST operation within POST-man looks like the following:

A screenshot of a computer

Description automatically generated

-READ

Executing the GET operation within POST-man looks like the following:

A screenshot of a computer

Description automatically generated

Example of retrieving many Contacts from POST-man using the GET operation:A screenshot of a computer

Description automatically generated

-UPDATE

Updating one Contact from within POST-man looks like the following:

A screenshot of a computer

Description automatically generated

-DELETE

Deleting one Contact within POST-man and then retrieving the List looks like the following:

-Deleting the Contact from the List

A computer screen shot of a computer

Description automatically generated

After deleting the Contact from the List, the List without the Contact that was deleted looks like the following:

A screenshot of a computer

Description automatically generated

-FILTER AND SEARCH

Additionally, I have also written the filtering functionality which allows the user to filter the data already present in the application by:

1.firstName

2.firstName and LastName

Below I present an example of filtering the Contacts by firstName:

A screenshot of a computer

Description automatically generated

Below I present an example of Filtering the Contacts by first Name and LastName:

A screenshot of a computer

Description automatically generated

## Additional Remarks:

The whole of the application was written in SpringBoot.

It makes use of SpringBoot project structure (most important layers include: repository, controller and service layer) to organize the whole of the app.

The technologies which were used in the whole of the application include:

Spring, Spring Boot, Maven, H2 Database, Junit.