Objective

The objective of this workshop is to write a simple address book. The information from the address book will be saved in a local filesystem as text file using Java IO package.

Setup

a. Create and clone a remote Git repository
b. Generate a SpringBoot application from Spring Initializr. Add DevTools, Spring Web and Thymeleaf as dependencies
c. Unpack the generated SpringBoot application in the Git repository

Workshop

Task 1

The application should accept a command line parameter call `--dataDir`. This options references a directory on your local computer eg.

```
     java myapp.jar --dataDir /opt/tmp/data
```

Refers to the directory `/opt/tmp/data`

If the directory does not exist, the application should create it.

If the `--dataDir` option is not specified, print an error message and stop.

See Javadocs on `java.nio.files.Paths` and `java.nio.files.Files`.

Task 2

Write a HTML form that collects the following information: name, email and phone number.

This information should be send to `/contact` resources with the HTTP `POST` method.

Task 3

The controller mapped to `/contact` process the data according to the following steps

a. Randomly generate an 8 character long hex string (eg `abcd1234`); this hex string will be used as the id for the data
b. Create a file with the above generated hex string in the directory specified by `--dataDir` option eg. you will create a file in `/opt/tmp/data/abcd1234`
c. Write the data into the file (`abcd1234`) as text (UTF-8) one field per line

Once the controller has completed the above steps, the controller should return the 'created' HTTP code with an appropriate message.

Important: All methods that manages the data directory should be handle by a single class and not in the controller. Call this class `Contacts`.

Hint. Write this class and do Task 5 first before continuing with Task 3.

Task 4

Write a controller to handle a `GET` to `/contact/<id>` where id is a 8 character long hex digit.

The controller will look into the data directory for a file with the corresponding <id> in the resource. Display the contents of the file in a HTML document.

If the <id> does not exist in the data directory, then return a not found status code with an appropriate message.

Task 5

Write test for `Contacts` class.

Note: we will not be deploying this application to Heroku because we are accessing the local file system.

Submission

When you have completed the workshop, commit and push your code to your Github repository.