# Hibernate Retake Exam – Instagraph Lite

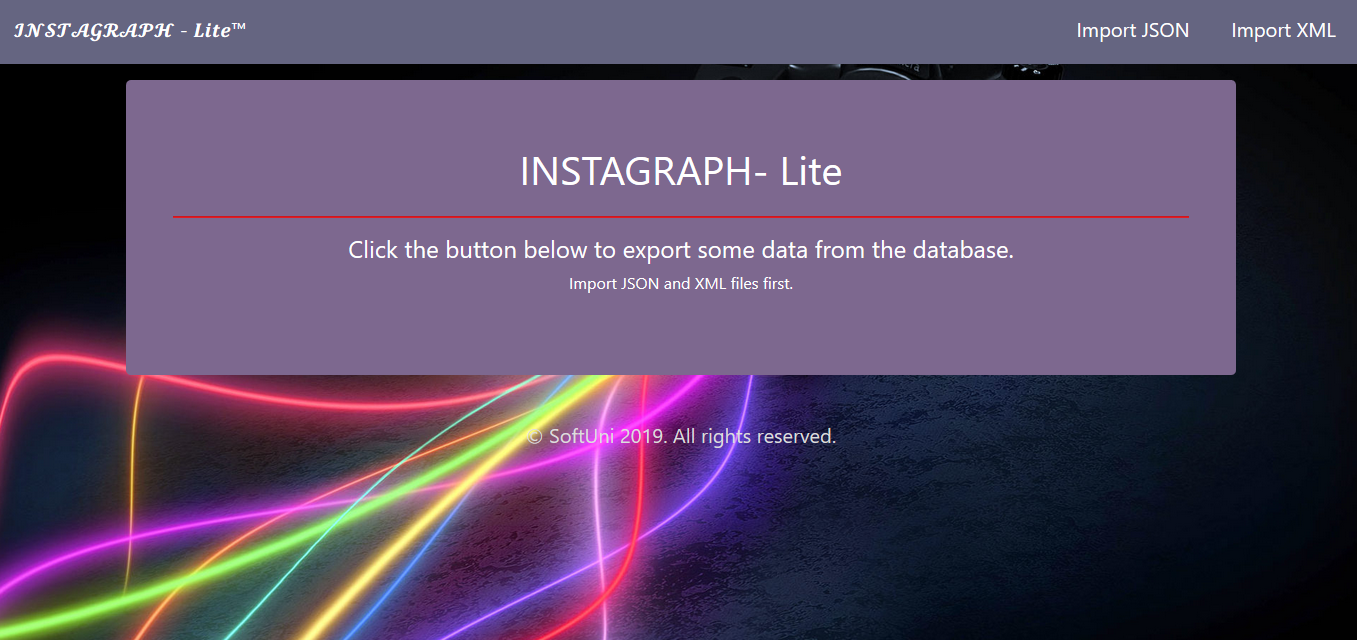
Some of you may know about Instagraph, some of you may not. For those who don’t know, Instagraph is the backup database of Instagram. And guess what, you were chosen to work on it. This time your task is to create a whole Hibernate & Spring Data application that will work with the database.

### Overview

The Instagraph database holds 3 entities. Pictures, Users and Posts . You will have to implement them in your application, so that they can be generated on the database. The Instagraph employers would also like for the application to support functionality for **importing** and **exporting** datain JSON and XML formats, so you’ll have to implement that too.

The technologies and ways which the Incorporation prefers that you use, is a **Code-First** **approach** application with **Hibernate** and **Spring Data**.

* Home page before importing anything:



### Data Models

You can see here what properties each model has:

#### Pictures

* id – an **Integer**.
* path – a **String**.
* size – a **Double**.

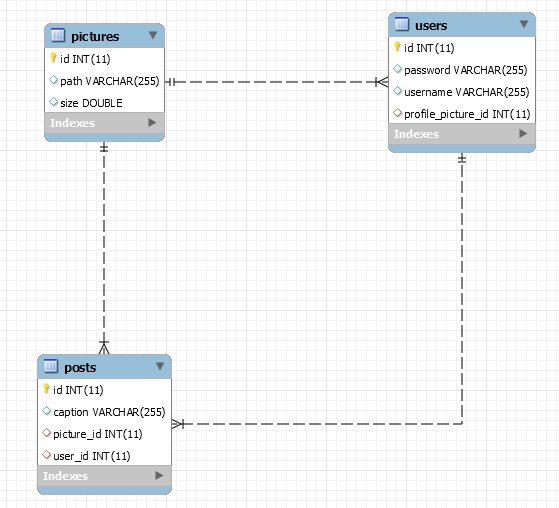
#### User

* id – an **Integer**.
* username – a **String**. Should be **UNIQUE**.
* password – a **String**.
* profilePicture – a Picture.
* posts – many Posts.

#### Post

* Id – an **integer**.
* Caption – a **string**.
* User – a User.
* Picture – a Picture.

**All data** is **REQUIRED**, unless it is explicitly said that it **null** is **allowed**.



### Importing Data

So here comes the **Importing** of **data** and the **populating** of the **database**. You have to **import** data from **JSON** and **XML** files.

Implement the needed **DTOs** for the imports.

Make sure all fields have been entered, otherwise the import **entity** data **should NOT be considered valid**.

You will also have to print a simple message indicating if the data has been imported successfully or there was an error.

In case of **SUCCESS** the message format is:

* **Picture**: "Picture with path:{path} and size:{size} successfully imported."
* **User**: "User {username} successfully imported.".
* **Post**: "**Post {caption} successfully imported.**".

In case of **ERROR** you always print "Error: Incorrect data."

#### Importing from JSON format

#### Pictures

##### Input

|  |
| --- |
| **pictures.json** |
| [  {  "path" : "src/folders/resources/images/profile/blocked/bmp/kjOJjKpKh4.bmp",  "size" : 32495.57  },  {  "path" : "src/folders/resources/images/post/timeline/png/27kLXVm22Q.png",  "size" : 44273.27  },  {  "path" : "src/folders/resources/images/profile/browsed/bmp/Q52q15Zefa.bmp"  },  {  "path" : "src/folders/resources/images/uploads/blocked/png/45Q92hqN5V.png",  "size" : 51087.72  },  {  "size" : 6638.98  },  . . .  ] |

##### Output

|  |
| --- |
| Successfully imported Picture src/folders/resources/images/profile/blocked/bmp/kjOJjKpKh4.bmp.  Successfully imported Picture src/folders/resources/images/post/timeline/png/27kLXVm22Q.png.  Error: Incorrect Data!  Successfully imported Picture src/folders/resources/images/uploads/blocked/png/45Q92hqN5V.png.  Error: Incorrect Data!  . . . |

#### Users

##### Input

|  |
| --- |
| **users.json** |
| [  {  "username" : "UnderSinduxrein",  "password" : "4l8nYGTKMW",  "profile\_picture" : "src/folders/resources/images/post/formed/digi/6YLvj97k03.digi"  },  {  "username" : "BlaAntigadsa",  "password" : ":Q5wjT4[e"  },  {  "password" : "El[MwhxY)J",  "profile\_picture" : "src/folders/resources/images/profile/blocked/jpg/pgfMG75k4e.jpg"  },  {  "username" : "ScoreImmagidefon",  "profile\_picture" : "src/folders/resources/images/uploads/browsed/vga/n5xg70JZDw.vga"  },  {  "username" : "BlaSinduxrein",  "password" : "wJyfcwg\*",  "profile\_picture" : "src/folders/resources/images/story/reformatted/img/hRI3TW31rC.img"  },  . . .  ] |

##### Output

|  |
| --- |
| Successfully imported User UnderSinduxrein.  Error: Invalid data.  Error: Invalid data.  Error: Invalid data.  Successfully imported User BlaSinduxrein.  . . . |

#### Importing from XML format

The other **table** must be populated with data in **XML** format.

#### Posts

##### Input

|  |
| --- |
| **posts.xml** |
| <?xml version="1.0" encoding="utf-8"?>  <posts>  <post>  <caption>#everything #top #ring #faith #insta #infinity #swag #sunglasses #suzanita #smiley #justdoit #the #sleepless #ocean</caption>  <user>ScoreAntigarein</user>  <picture>src/folders/resources/images/story/blocked/png/1S2el3wJ3v.png</picture>  </post>  <post>  <caption>Inv</caption>  <user>InvalidUser</user>  <picture>src/folders/resources/images/story/reformatted/img/hRI3TW31rC.img</picture>  </post>  <post>  <caption>#ring #infinity #justdoit #hope #dawn</caption>  <user>AryaDenotehow</user>  <picture>src/invalidPictures/invaliddd.img</picture>  </post> ...  </posts> |

##### Output

|  |
| --- |
| Successfully imported Post #everything #top #ring....  Error: Incorrect data.  Error: Incorrect data.  . . . |

### Exporting Data

## Data Export

Get ready to export the data you’ve imported in the previous task. Here you will have some pretty complex database querying. Export the data in the formats specified below.

#### Export users with their posts

**Export all users with their posts in following format, ordered by count of posts descending, then by user id**:

" **1. {username} – {postsCount} posts**

**Post Details:**

**-Caption: {caption}**

**-Picture: {pictureName}**

**. . .**

**2. {username} – {postsCount} posts**

**Post Details:**

**-Caption: {caption}**

**-Picture: {pictureName}**

**. . .** "



#### Export Pictures with size bigger than 30000

**Export** all the pictures with size bigger than 30000.

* **Order the result by size ascending.**

"**{PictureName} – {pictureSize}**

**. . .**"

### Example

