# Databases Frameworks Exam – Instagraph Advanced

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 4 entities. Pictures, Users, Posts and Comments. 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**.

### Data Models

You can see here what properties each model has:

#### Pictures

* Id – an **integer**.
* Path – a **string**.
* Size – a **decimal**.

#### User

* Id – an **integer**.
* Username – a **string**. Should be **UNIQUE**.
* Password – a **string**.
* Profile Picture – a Picture.
* Followers – many Users.
* Following – many Users.

#### Post

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

#### Comment

* Id – an **integer**.
* Content – a **string**.
* User – a User.
* Post – a Post.

**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**: “Successfully imported Picture {picturePath}.”.
* **User**: “Successfully imported User {username}.”.
* **Post**: “Successfully imported Post {postCaption}.”.
* **Comment**: “Successfully imported Comment {commentContent}.”.
* **User - Follower**: “Successfully imported Follower {followerUsername} to User {userUsername}.”

In case of **ERROR** you always print “Error: Invalid 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: Invalid data.  Successfully imported Picture src/folders/resources/images/uploads/blocked/png/45Q92hqN5V.png.  Error: Invalid 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.  . . . |

#### EmployeeCards

##### Input

|  |
| --- |
| **users\_followers.json** |
| [  {  "user" : "BlaImmagiIana",  "follower" : "ScoreSinduxIana"  },  {  "user" : "BlaSinduxrein",  "follower" : "RoundInspecindi"  },  {  "user" : "AryaNinehow",  "follower" : "DarkImmagidsa"  },  {  "user" : "BlaSinduxrein",  "follower" : "ZendArmyhow"  },  {  "user" : "ScoreSinduxIana",  "follower" : "RoundInspecindi"  },  . . .  ] |

##### Output

|  |
| --- |
| Successfully imported Follower ScoreSinduxIana to User BlaImmagiIana.  Successfully imported Follower RoundInspecindi to User BlaSinduxrein.  Successfully imported Follower DarkImmagidsa to User AryaNinehow.  Successfully imported Follower ZendArmyhow to User BlaSinduxrein.  Successfully imported Follower RoundInspecindi to User ScoreSinduxIana.  . . . |

#### Importing from XML format

The other **2 tables** must be populated with data in **XML** format.

#### Posts

##### Input

|  |
| --- |
| **posts.xml** |
| <?xml version="1.0" encoding="utf-8"?>  <posts>  <post>  <caption>#everything #fuck #ring #faith #insta #infinity #swag . . .</caption>  <user>ScoreAntigarein</user>  <picture>src/folders/resources/images/story/blocked/png/. . .</picture>  </post>  <post>  <caption>#cool #justdoit #sky #ocean #reason #feminist . . .</caption>  <user>HighAsmahow</user>  <picture>src/folders/resources/images/profile/blocked/. . .</picture>  </post>  ...  </posts> |
|  |

##### Output

|  |
| --- |
| Successfully imported Post #everything #fuck #ring....  Successfully imported Post #cool #justdoit #sky....  . . . |

#### Comments

##### Input

|  |
| --- |
| **comments.xml** |
| <?xml version="1.0" encoding="utf-8"?>  <comments>  <comment>  <content>What the fuck? Aaarghhhh! Why?</content>  <user>ScoreSinduxIana</user>  <post id="40" />  </comment>  <comment>  <content>Wow! Wow, epic!! How?</content>  <user>RoundAntigaBel</user>  <post id="22" />  </comment>  . . .  </comments> |

##### Output

|  |
| --- |
| Successfully imported Comment What the fuck? Aaarghhhh! Why?.  Successfully imported Comment Wow! Wow, epic!! How?  . . . |

### Exporting Data

So here comes the **Exporting** of **data** from the **database**. You have to **export** data from **JSON** and **XML** files.

#### Exporting to JSON format

First you will need to **write functionality** for exporting **filtered data** in **JSON** format.

#### Uncommented Posts

Export all posts which do **NOT** have **ANY comments**. Extract everything and **order** it by **id** in **ascending order**.

Extract the problem id, picture path as “**picture**” and user’s username as “**user**”.

##### Output

|  |
| --- |
| **uncommented-posts.json** |
| [  {  "id" : 1,  "picture" : "src/folders/resources/images/story/blocked/png/1S2el3wJ3v.png",  "user" : "ScoreAntigarein"  },  {  "id" : 4,  "picture" : "src/folders/resources/images/profile/blocked/jpeg/fB3x0NwaNY.jpeg",  "user" : "ScoreAntigarein"  },  {  "id" : 6,  "picture" : "src/folders/resources/images/story/blocked/png/1S2el3wJ3v.png",  "user" : "DarkImmagidsa"  },  . . .  ] |

#### Popular users

Extract all **users** which have **posts**, **commented** by **their followers**.

Extract the user’s username as “**user**”, and the user’s count of followers as “**followers**”.

Order the data by **id** in **descending order**.

##### Output

|  |
| --- |
| **popular-users.json** |
| [  {  "user" : "ZendArmyhow",  "followers" : 9  },  {  "user" : "HighAsmahow",  "followers" : 6  },  {  "user" : "ScoreAntigarein",  "followers" : 4  },  . . .  ] |

#### Exporting to XML format

Alright, we finished with the JSON, now let’s get it on with the XML.

#### Comments On Posts

Extract **all users**. For each **user**, **extract** the **post** which has the **most comments**.

Extract the user’s username as “**username**” and the count of comments on the top post as   
“**most-comments**”.

If a **person** has **no posts**, set his **comments** to **0**.

**Order** the **data** in **descending order** by **most-comments**.

##### Output

|  |
| --- |
| **comments-on-posts.xml** |
| <?xml version="1.0" encoding="utf-8"?>  <users>  <user>  <username>RoundAntigaBel</username>  <most-comments>4</most-comments>  </user>  <user>  <username>ScoreSinduxIana</username>  <most-comments>4</most-comments>  </user>  <user>  <username>ZendArmyhow</username>  <most-comments>3</most-comments>  </user>  . . .  </users> |

Well, that’s all folks. One more thing! The Instagraph Team would like from you to use the **best architecture** **possible**. Use all the knowledge you’ve gathered to create the most suitable **High Quality Structure** for the application.