

Spring AU '21 – Advanced Java – Afternoon Session

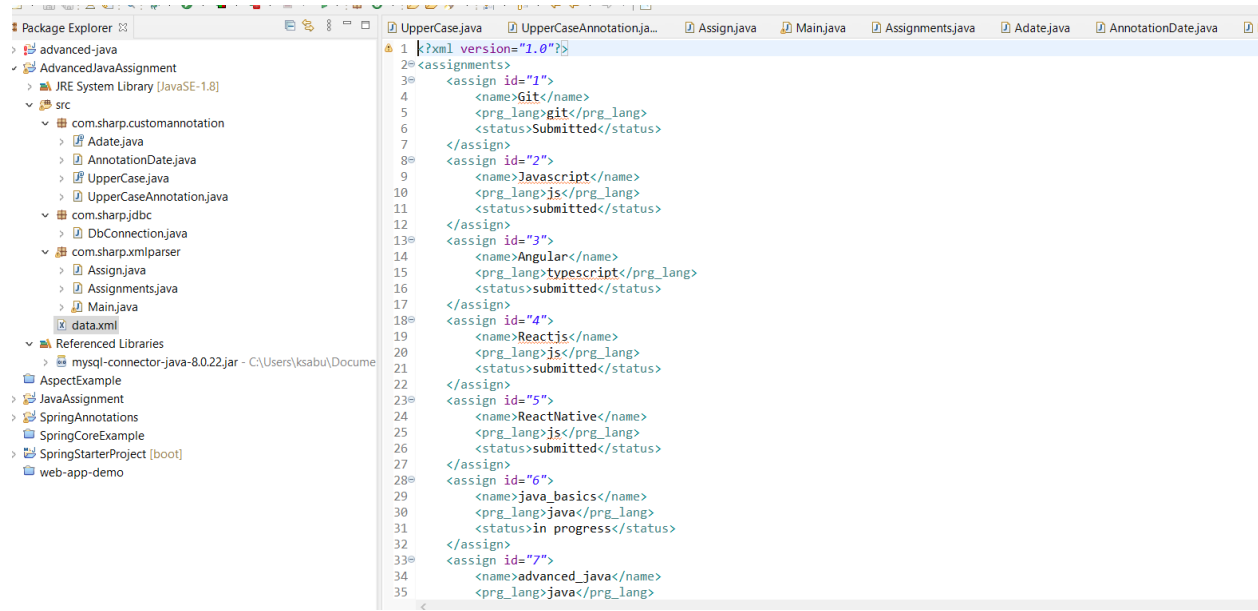
Name: Sheik Abudhahir K

Date: 18/01/2021

Source code is uploaded

1.Takes in assignment related information (name, prg language, status update) in XML file

Data are in small letter



The screenshot shows an IDE with a Package Explorer on the left and a code editor on the right. The Package Explorer shows a project named 'advanced-java' with a 'src' folder containing several Java files: 'Adate.java', 'AnnotationDate.java', 'UpperCase.java', 'UpperCaseAnnotation.java', 'DbConnection.java', 'Assign.java', 'Assignments.java', and 'Main.java'. There is also a 'data.xml' file. The code editor shows the content of 'data.xml', which is an XML document with a root element 'assignments' containing several 'assign' elements. Each 'assign' element has attributes 'id', 'name', 'prg_lang', and 'status'.

```
1 <?xml version="1.0"?>
2 <assignments>
3   <assign id="1">
4     <name>Git</name>
5     <prg_lang>git</prg_lang>
6     <status>Submitted</status>
7   </assign>
8   <assign id="2">
9     <name>Javascript</name>
10    <prg_lang>js</prg_lang>
11    <status>submitted</status>
12  </assign>
13  <assign id="3">
14    <name>Angular</name>
15    <prg_lang>typescript</prg_lang>
16    <status>submitted</status>
17  </assign>
18  <assign id="4">
19    <name>Reactjs</name>
20    <prg_lang>js</prg_lang>
21    <status>submitted</status>
22  </assign>
23  <assign id="5">
24    <name>ReactNative</name>
25    <prg_lang>js</prg_lang>
26    <status>submitted</status>
27  </assign>
28  <assign id="6">
29    <name>java_basics</name>
30    <prg_lang>java</prg_lang>
31    <status>in progress</status>
32  </assign>
33  <assign id="7">
34    <name>advanced_java</name>
35    <prg_lang>java</prg_lang>
```

2.Parses the xml (Preferably using JAXB)

```
public List<Assign> unMarshall() throws IllegalArgumentException, IllegalAccessException {
    List<Assign> students = new ArrayList<>();
    try {
        File file = new File("./src/data.xml");
        JAXBContext jaxbContext = JAXBContext.newInstance(Assignments.class);
        Unmarshaller unmarshaller = jaxbContext.createUnmarshaller();
        Assignments assignments = (Assignments) unmarshaller.unmarshal(file);
        System.out.println("Xml file data converted to java object");

        students = assignments.getAssign();

        for(Assign student:students) {
            UpperCaseAnnotation.toUpperCase(student);
            AnnotationDate.addDate(student);
        }

        System.out.println(students);

    } catch (JAXBException e) {
        e.printStackTrace();
    }

    return students;
}
```

Create a custom annotation to Capitalize names (name, language) in the assignment data and add date whenever you are processing a new record.

Here I use the custom annotation for capitalize the name and language, another one for adding date

```
1 package com.sharp.xmlparser;
2
3 import javax.xml.bind.annotation.XmlAttribute;
4 import javax.xml.bind.annotation.XmlElement;
5
6 import com.sharp.customannotation.Adate;
7 import com.sharp.customannotation.UpperCase;
8
9 public class Assign {
10     int id;
11
12     @UpperCase
13     String name;
14
15     @UpperCase
16     String prg_lang;
17
18     String status;
19
20     @Adate
21     String date;
22
23     public int getId() {
24         return id;
25     }
26
27     @XmlAttribute
28     public void setId(int id) {
29         this.id = id;
30     }
31 }
```

UpperCase.java UpperCaseAnnotation.java Assign.java Main.java Assignments.java Adate.java AnnotationI

```
1 package com.sharp.customannotation;
2
3 import java.lang.reflect.Field;
4
5 public class UpperCaseAnnotation {
6
7     public static void toUpperCase(Object object) throws IllegalArgumentException, IllegalAccessException
8     {
9         Class<?> cls = object.getClass();
10
11         for(Field field : cls.getDeclaredFields()) {
12             field.setAccessible(true);
13
14             if(field.isAnnotationPresent(UpperCase.class)) {
15                 UpperCase upperCase = field.getAnnotation(UpperCase.class);
16                 field.set(object, field.get(object).toString().toUpperCase());
17             }
18         }
19     }
20 }
21
22 }
```

```

UpperCase.java  UpperCaseAnnotation.java  Assign.java  Main.java  Assignments.java  Adate.java  AnnotationDate.java
1 package com.sharp.customannotation;
2
3 import java.lang.reflect.Field;
4 import java.time.LocalDate;
5
6 public class AnnotationDate {
7
8 public static void addDate(Object object) throws IllegalArgumentException, IllegalAccessException {
9
10     Class<?> cls = object.getClass();
11
12     for(Field field : cls.getDeclaredFields()) {
13         field.setAccessible(true);
14
15         if(field.isAnnotationPresent(Adate.class)) {
16             Adate date = field.getAnnotation(Adate.class);
17             field.set(object, LocalDate.now().toString());
18         }
19     }
20 }
21 }
22
23 }
24

```

3.Store details in MySQL DB using JDBC – stored data – code is added in repository

The screenshot shows the MySQL Workbench interface. On the left, the 'SCHEMAS' pane displays a tree view of the database structure. The 'assignment' schema is expanded, showing a table named 'assign' with columns: id, name, language, status, and date_of_submit. The 'Query 1' pane shows the SQL query: `select * from assign;`. The 'Result Grid' pane displays the query results in a table format.

	id	name	language	status	date_of_submit
1	1	GIT	GIT	Submitted	2021-01-21
2	2	JAVASCRIPT	JS	submitted	2021-01-21
3	3	ANGULAR	TYPESCRIPT	submitted	2021-01-21
4	4	REACTJS	JS	submitted	2021-01-21
5	5	REACTNATIVE	JS	submitted	2021-01-21
6	6	JAVA_BASICS	JAVA	in progress	2021-01-21
7	7	ADVANCED_JAVA	JAVA	in progress	2021-01-21
8	8	MAVEN	JAVA	in process	2021-01-21
9	9	SPRING	JAVA	in progress	2021-01-21
*	NULL	NULL	NULL	NULL	NULL

Final Output:

