JSP Mashalling

훈련명 2023 산대특 4차

이름 박연지

목차

INDEX

CHAPTER 01

직렬화 Serialization

객체 데이터를 일련의 byte stream으로 변환하는 작업

객체에 저장된 데이터를
스트림(Stream)에 쓰기 위해
연속적인(Serial)데이터로
변환하는 것을 의미합니다.

이는 데이터를 파일로 저장하거나
네트워크를 통해 데이터를 송/수신할 때
용이합니다.

-> 전송하기 편하게 데이터를 가공할 뿐 직렬화가 데이터를 송/수신한다는 것은 아닙니다.

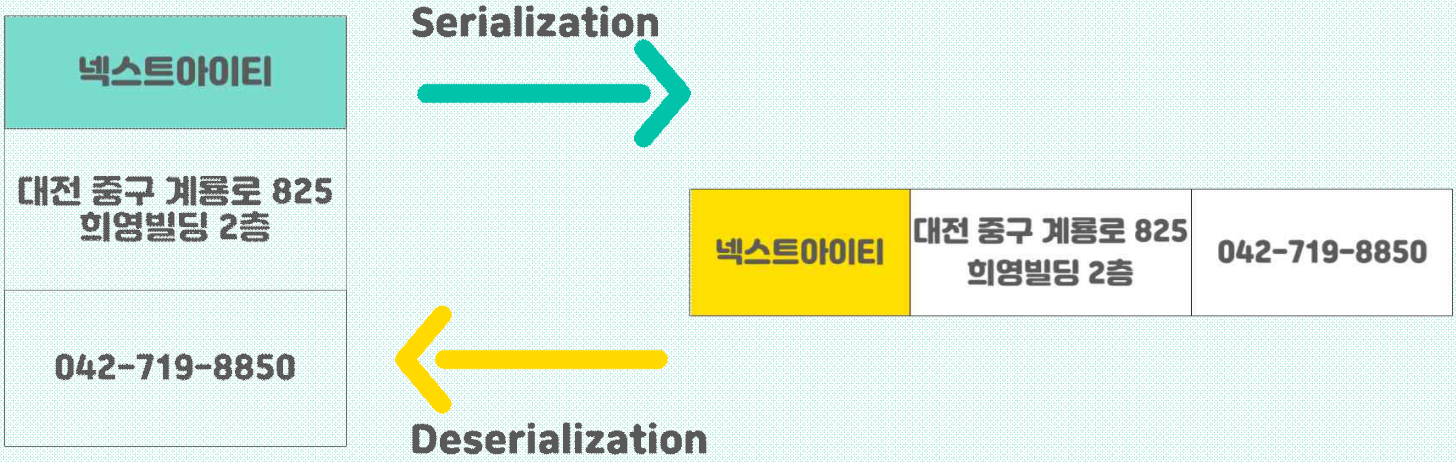
CHAPTER 02

직렬화 역직렬화

넥스트IT교육센터

CHAPTER 02

직렬화 역직렬화



CHAPTER 03

마샬링

서로 다른 컴퓨터 혹은 서로 다른 프로그램 간의
데이터를 이동시키고자 하는 경우에 사용합니다
직렬화는 byte stream으로 데이터를 가공하는 것이라면
마샬링은 데이터를 바이트의 덩어리로 만들어
스트림에 보낼 수 있는 형태로 바꾸는 변환 작업을 의미합니다

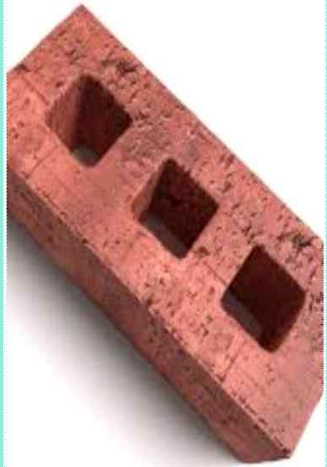
언마샬링

마샬링과 반대로 객체 스트림을 통해서 전달된 바이트 덩어리
를 원래의 객체로 복구하는 작업입니다
간단하게 말하자면 자바에서 언마샬링이란
마샬링과는 반대로 스트림이나 파일로부터
자바 객체를 만들어내는 작업을 일컫습니다.

01

갈마동

02



분해 (직렬화)
/ 이동

03

용두동

마샬링, 직렬화 의 차이점

마샬링

네트워크 전송에 특화

주로 분산 시스템이나 네트워크 프로그래밍에서 사용되어 바이트스트림으로 바이트 덩어리로 만들어진 데이터를 네트워크 전송을 위한 형태로 변환하여 전송한다.

직렬화

바이트스트림으로 데이터 변환

원본 데이터에서 바이트스트림으로 가공하는 그 과정 자체, 즉 객체에 포커싱되어있어 구조화된 데이터를 바이트 스트림 같은 형식으로 복사하는 것을 의미합니다.

CHAPTER 05

활용법 예제코드

예제코드를 보며
마샬링과 직렬화의 차이를
좀 더 느껴봅시다 !

```java
yeonji *
public class FileInput {

    yeonji *
    public static void main(String[] args) {
        String path = System.getProperty("user.dir");

        File context = new File( pathname: path + "\\src\\config\\context-db.properties");

        System.out.println(context.exists());
        StringBuffer sb = new StringBuffer();

        try (FileInputStream fis = new FileInputStream(context)) {
            byte[] bowl = new byte[3];
            while (true) {
                int cnt = fis.read(bowl);

                if (cnt == -1) {
                    break;
                }
                String text = new String(bowl);
                sb.append(text);
            }
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }
```

CHAPTER 05

직렬화

고급자바 수업코드

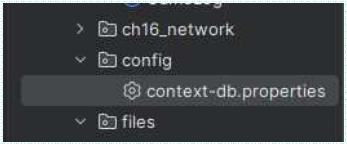
CHAPTER 05

직렬화

고급자바 수업코드

Run FileInput

```
C:\Users\admin\.jdks\corretto-17.0.9\bin\java.exe "-javaagent:C:\P
true
# context-db.properties
# Database Configuration
driver=oracle.jdbc.driver.OracleDriver
url=jdbc:oracle:thin:@nextit.or.kr:1521:xe
#url=jdbc:oracle:thin:@nextit.or.kr:1521:xe
id=std115
pw=oracle21c
maxConn=3n=
```



context-db.properties

```
1  # context-db.properties
2  # Database Configuration
3  driver=oracle.jdbc.driver.OracleDriver
4  url=jdbc:oracle:thin:@nextit.or.kr:1521:xe
5  #url=jdbc:oracle:thin:@nextit.or.kr:1521:xe
6  id=std115
7  pw=oracle21c
8  maxConn=3
```

CHAPTER 05

마샬링
- jaxb 라이브러리

1. **JAXB API**
javax.xml.bind » jaxb-api

JAXB provides an API and tools that automate the mapping between XML documents and Java objects.

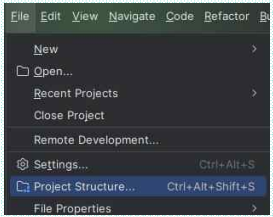
Last Release on Aug 30, 2018

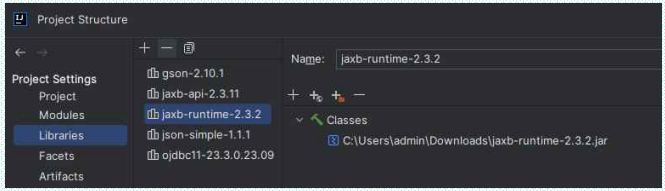
Relocated → jakarta.xml.bind » jakarta.xml.bind-api

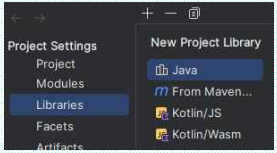
3. **JAXB Runtime**
org.glassfish.jaxb » jaxb-runtime

JAXB (JSR 222) Reference Implementation

Last Release on Mar 7, 2024







```java
package yeonji.marshalling;

import javax.xml.bind.annotation.XmlElement;
import javax.xml.bind.annotation.XmlRootElement;

3 usages  new *
@XmlRootElement
public class Person {
    3 usages
    private String name;
    3 usages
    private String team;
    1 usage  new *
    public Person(String name, String team) {
        this.name = name;
        this.team = team;
    }
    new *
    @XmlElement
    public String getName() {
        return name;
    }
    new *
    public void setName(String name) {
        this.name = name;
    }
    no usages  new *
    @XmlElement
    public String getTeam() {
        return team;
    }
    no usages  new *
    public void setTeam(String team) {
        this.team = team;
    }
}
```

CHAPTER 05

마샬링

java -> xml

```java
package yeonji.marshalling;

import javax.xml.bind.JAXBContext;
import javax.xml.bind.JAXBException;
import javax.xml.bind.Marshaller;
import java.io.StringWriter;

new *
public class Jaxb {
    new *
    public static void main(String[] args) {
        Person person = new Person( name: "김광현",  team: "SSG랜더스");

        try {
            // JAXBContext 생성
            JAXBContext context = JAXBContext.newInstance(Person.class);

            // Marshaller 생성
            Marshaller marshaller = context.createMarshaller();
            marshaller.setProperty(Marshaller.JAXB_FORMATTED_OUTPUT, true);

            // 객체를 XML로 마샬링
            StringWriter sw = new StringWriter();
            marshaller.marshal(person, sw);

            // XML 출력
            String xmlString = sw.toString();
            System.out.println(xmlString);

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

CHAPTER 05

마샬링

java -> xml

CHAPTER 05

마샬링

- jaxb 라이브러리

```xml
<?xml version="1.0" enconding="UTF-8" standalone="yes"?>
<person>
    <name>김광현</name>
    <team>SSG랜더스</team>
</person>
```

```
C:\Users\admin\.jdks\corretto-17.0.9\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2023.3.2\lib\idea_rt.jar=55386:C:\
Exception in thread "main" java.lang.NoClassDefFoundError Create breakpoint : javax/activation/DataSource
    at com.sun.xml.bind.v2.model.impl.RuntimeBuiltinLeafInfoImpl.<clinit>(RuntimeBuiltinLeafInfoImpl.java:474)
    at com.sun.xml.bind.v2.model.impl.RuntimeTypeInfoSetImpl.<init>(RuntimeTypeInfoSetImpl.java:63)
    at com.sun.xml.bind.v2.model.impl.RuntimeModelBuilder.createTypeInfoSet(RuntimeModelBuilder.java:128)
    at com.sun.xml.bind.v2.model.impl.RuntimeModelBuilder.createTypeInfoSet(RuntimeModelBuilder.java:84)
    at com.sun.xml.bind.v2.model.impl.ModelBuilder.<init>(ModelBuilder.java:162)
    at com.sun.xml.bind.v2.model.impl.RuntimeModelBuilder.<init>(RuntimeModelBuilder.java:92)
    at com.sun.xml.bind.v2.runtime.JAXBContextImpl.getTypeInfoSet(JAXBContextImpl.java:455)
    at com.sun.xml.bind.v2.runtime.JAXBContextImpl.<init>(JAXBContextImpl.java:303)
    at com.sun.xml.bind.v2.runtime.JAXBContextImpl.<init>(JAXBContextImpl.java:139)
    at com.sun.xml.bind.v2.runtime.JAXBContextImpl$JAXBContextBuilder.build(JAXBContextImpl.java:1156)
    at com.sun.xml.bind.v2.ContextFactory.createContext(ContextFactory.java:165) <4 internal lines>
    at javax.xml.bind.ContextFinder.newInstance(ContextFinder.java:297)
    at javax.xml.bind.ContextFinder.newInstance(ContextFinder.java:286)
    at javax.xml.bind.ContextFinder.find(ContextFinder.java:409)
    at javax.xml.bind.JAXBContext.newInstance(JAXBContext.java:721)
    at javax.xml.bind.JAXBContext.newInstance(JAXBContext.java:662)
    at yeonji.marshalling.Jaxb.main(Jaxb.java:14)
Caused by: java.lang.ClassNotFoundException Create breakpoint : javax.activation.DataSource <2 internal lines>
    at java.base/java.lang.ClassLoader.loadClass(ClassLoader.java:525)
    ... 21 more

Process finished with exit code 1
```

```
Run  Jaxb
C:\Users\admin\.jdks\corretto-17.0.9\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA
Exception in thread "main" java.lang.NoClassDefFoundError Create breakpoint : com/sun/istack/Pool
    at com.sun.xml.bind.v2.runtime.JAXBContextImpl$JAXBContextBuilder.build(JAXBContextImpl.java:1126)
    at com.sun.xml.bind.v2.ContextFactory.createContext(ContextFactory.java:135) <4 internal lines>
    at javax.xml.bind.ContextFinder.newInstance(ContextFinder.java:297)
    at javax.xml.bind.ContextFinder.newInstance(ContextFinder.java:286)
    at javax.xml.bind.ContextFinder.find(ContextFinder.java:409)
    at javax.xml.bind.JAXBContext.newInstance(JAXBContext.java:721)
    at javax.xml.bind.JAXBContext.newInstance(JAXBContext.java:662)
    at yeonji.marshalling.Jaxb.marshalTest(Jaxb.java:17)
    at yeonji.marshalling.Jaxb.main(Jaxb.java:8)
Caused by: java.lang.ClassNotFoundException Create breakpoint : com.sun.istack.Pool <2 internal lines>
    at java.base/java.lang.ClassLoader.loadClass(ClassLoader.java:525)
    ... 13 more

Process finished with exit code 1
```

CHAPTER 05

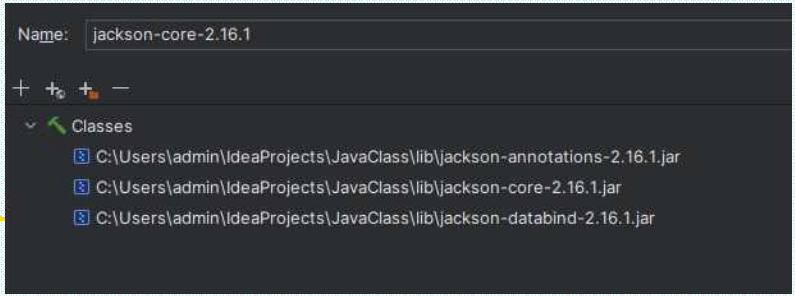
마샬링

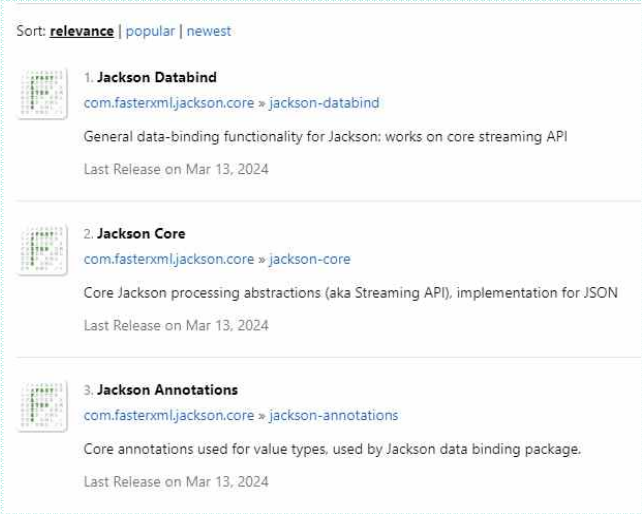
-jaxb 라이브러리

CHAPTER 05

마샬링

- Jackson 라이브러리





CHAPTER 05

마샬링

java -> json

json -> java

```java
package yeonji.marshalling;

import com.fasterxml.jackson.databind.ObjectMapper;

public class Person {
    4 usages
    private String player;
    4 usages
    private String team;

    // 기본 생성자
    no usages
    public Person() {}

    // 파라미터가 있는 생성자
    1 usage
    public Person(String player, String team) {...}
    // Getter 및 Setter
    no usages
    public String getPlayer() { return player; }
    no usages
    public void setPlayer(String player) { this.player = player; }
    no usages
    public String getTeam() { return team; }
    no usages
    public void setTeam(String team) { this.team = team; }
```

```java
    @Override
    public String toString() {
        return "Person{" +
                "player='" + player + '\'' +
                ", team='" + team + '\'' +
                '}';
    }
```

CHAPTER 05

마샬링

java -> json

json -> java

```java
public static void main(String[] args) {
    // 객체 생성
    Person person = new Person( player: "최지훈",  team: "SSG랜더스");

    // ObjectMapper 생성
    ObjectMapper objectMapper = new ObjectMapper();

    try {
        // 객체를 JSON 문자열로 마샬링 (직렬화)
        String jsonString = objectMapper.writeValueAsString(person);
        System.out.println("JSON으로 마샬링 결과 : " + jsonString);

        // JSON 문자열을 객체로 언마샬링 (역직렬화)
        Person deserializedPerson = objectMapper.readValue(jsonString, Person.class);
        System.out.println("언마샬링(역직렬화) 결과 : " + deserializedPerson);
    } catch (Exception e) {
        e.printStackTrace();
    }
}

}
```

```
C:\Users\admin\.jdks\corretto-17.0.9\bin\java.exe "-javaagent:C:\Progr
JSON으로 마샬링 결과 : {"player":"최지훈","team":"SSG랜더스"}
언마샬링(역직렬화) 결과 : Person{player='최지훈', team='SSG랜더스'}

Process finished with exit code 0
```

레퍼 런스

마샬링 > 직렬화라고 주장하는 글

- https://hyesun03.github.io/2019/09/08/marshalling-vs-serialization/
- https://tlatmsrud.tistory.com/117
- https://wookcode.tistory.com/193

마샬링 < 직렬화라고 주장하는 글

- https://weicomes.tistory.com/63
- https://medium.com/@nicegirl/marshalling-vs-serialized-252caf70ba9b

레퍼런스

jaxb 라이브러리 설치
- https://github.com/javaee/jaxb-v2

라이브러리, annotation 방식 코드 참고
- https://tychejin.tistory.com/135
- https://xens.tistory.com/entry/JAXB%EB%A5%BC-%EC%9D%B4%EC%9A%A9%ED%95%9C-XML-Marshalling-UnMarshallingFile-%EC%98%88%EC%A0%9C-1

...

jackson 라이브러리 설치
- https://au1802.tistory.com/57
- https://mvnrepository.com/search?q=Jackson

마샬링 코드 참고
- https://madplay.github.io/post/jaxb-marshal-unmarshal
- https://tychejin.tistory.com/134
- https://wookcode.tistory.com/193

언마샬링(JSON->Java Object) 코드 참고
- https://tychejin.tistory.com/135


THANK
YOU