

[JAVA] Serialization

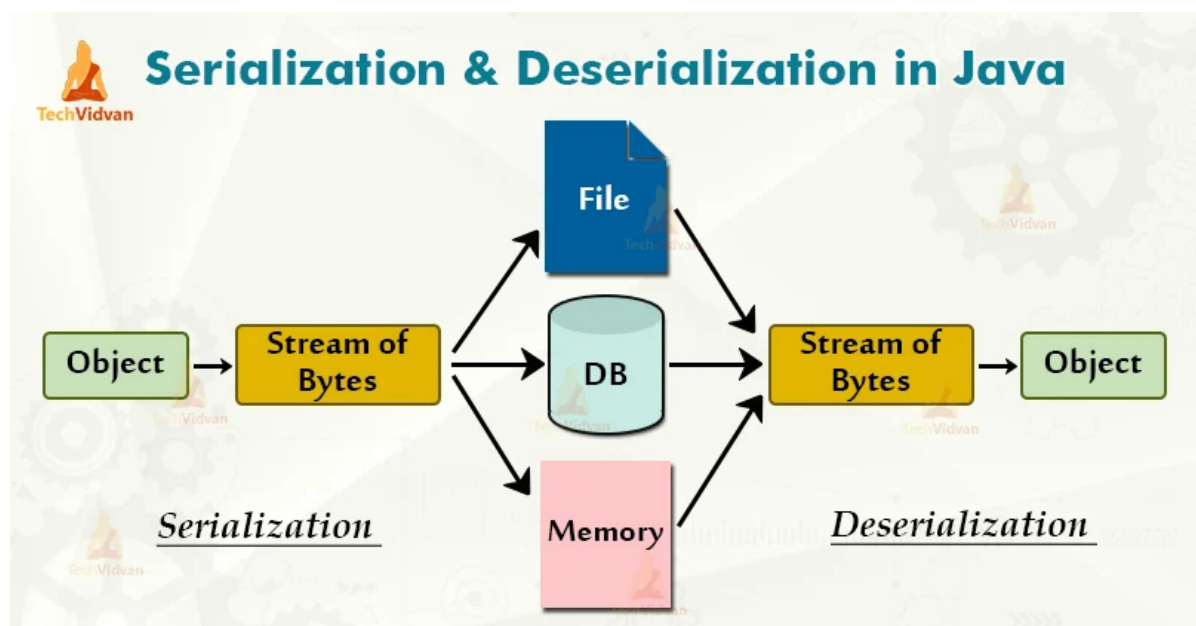
자바 시스템 내부에서 사용되는 객체 또는 데이터를 외부의 자바 시스템에서도 사용할 수 있도록 바이트(byte) 형태로 데이터 변환하는 기술

각자 PC의 OS마다 서로 다른 가상 메모리 주소 공간을 가지기 때문에 Reference Type의 데이터들은 인스턴스를 전달할 수 없다.

이 문제를 해결하기 위해서는 주소값이 아닌 Byte형태로 직렬화된 객체 데이터를 전달해야한다.

직렬화 된 데이터들은 모두 Primitive Type이 되고, 이는 파일 저장이나 네트워크 전송 시 파싱이 가능한 유의미한 데이터가 된다.

따라서, 전송 및 저장이 가능한 형태로 만들어주는 것이 **직렬화(Serialization)** 이다.



조건

자바에서는 간단히 `java.io.Serializable` 인터페이스 구현으로 직렬화 / 역직렬화가 가능하다.

- 직렬화 대상: Serializable 인터페이스를 상속받은 객체 or Primitive Type의 데이터

직렬화 구현

Serializable 인터페이스를 구현하면 SUID를 선언해야한다.

serialVersionUID는 Default 값으로 클래스의 기본 해쉬값을 가지고 있는데, 직렬화 된 데이터를 다른 클래스가 전달받아 역직렬화를 하게 되면 `java.io.InvalidClassException` 이 발생할 수 있다.

위 Exception이 발생하는 이유는 직렬화 하는 시스템과 역직렬화하는 시스템이 다른 경우에 발생하는 데, 각 시스템에서 사용하고 있는 모델의 버전 차이가 발생했을 경우에 생기는 문제이다.

ex) Class의 속성이 추가되어 멤버변수가 더 많을 때

이렇게 모델의 버전 간 호환성을 유지하기 위해서는 SUID를 정의해야 한다.

```
1 public class User implements Serializable{
2     private static final long serialVersionUID = 1L;
3
4     private String userId;
5     private String userName;
6 }
```

```
1 public static void main(String[] args){
2     User user = new User("id","name");
3     byte[] serializedUser;
4
5     try(ByteArrayOutputStream baos = new ByteArrayOutputStream()){
6         try(ObjectOutputStream oos = new ObjectOutputStream(baos)){
7             oos.writeObject(user);
8
9             serializedUser = baos.toByteArray();
10        }
11    }
12 }
```

위와 같이 `ObjectOutputStream` 클래스를 이용해 직렬화를 진행하면 byte로 변환된 값을 얻을 수 있다.

역 직렬화 구현

```
1 public static void main(String[] args){
2     byte[] serializedUser;
3
4     try(ByteArrayInputStream bais = new
5     ByteArrayInputStream(serializedUser)){
6         try(ObjectInputStream ois = new ObjectInputStream(bais)){
7
8             object objectUser = ois.readObject();
9             User user = (User) objectUser;
10        }
11    }
12 }
```

`ObjectInputStream` 클래스로 byte값을 다시 객체로 만들 수 있게 된다.

직렬화 데이터는 타입, 클래스 메타정보를 포함하므로 사이즈가 크다.

트래픽에 따라 비용 증가 문제가 발생할 수 있기 때문에

- 표 형태의 다량의 데이터 -> CSV
- 구조적인 데이터 -> XML, JSON

위와 같은 포맷으로 직렬화 하는 것이 좋다.

JSON 포맷으로 직렬화를 하기 위해서는 Jackson이나 Gson라이브러리를 사용하면 된다.