

# AVRO - DESERIALIZATION USING PARSERS

[http://www.tutorialspoint.com/avro/deserialization\\_using\\_parsers.htm](http://www.tutorialspoint.com/avro/deserialization_using_parsers.htm)

Copyright © tutorialspoint.com

As described earlier, one can read an Avro schema into a program either by generating a class corresponding to the schema or by using the parsers library. This chapter describes how to read the schema by using parser library and Deserialize the data using Avro.

## Deserialization Using Parsers Library

In our last example, the serialized data was stored in the file **mydata.txt**. We shall now see how to deserialize it and read it using Avro. The procedure is as follows –

### Step 1

First of all, read the schema from the file. To do so, use **Schema.Parser** class. This class provides methods to parse the schema in different formats.

Instantiate the **Schema.Parser** class by passing the file path where the schema is stored.

```
Schema schema = new Schema.Parser().parse(new File("/path/to/emp.avsc"));
```

### Step 2

Create an object of **DatumReader** interface using **SpecificDatumReader** class.

```
DatumReader<emp>empDatumReader = new SpecificDatumReader<emp>(emp.class);
```

### Step 3

Instantiate **DataFileReader** class. This class reads serialized data from a file. It requires the **DatumReader** object, and path of the file where the serialized data exists, as a parameters to the constructor.

```
DataFileReader<GenericRecord> dataFileReader = new DataFileReader<GenericRecord>(new File("/path/to/mydata.txt"), datumReader);
```

### Step 4

Print the deserialized data, using the methods of **DataFileReader**.

- The **hasNext** method returns a boolean if there are any elements in the Reader.
- The **next** method of **DataFileReader** returns the data in the Reader.

```
while(dataFileReader.hasNext()){  
    em=dataFileReader.next(em);  
    System.out.println(em);  
}
```

## Example - Deserialization Using Parsers Library

The following complete program shows how to deserialize the serialized data using Parsers library –

```
public class Deserialize {  
    public static void main(String args[]) throws Exception{  
  
        //Instantiating the Schema.Parser class.  
        Schema schema = new Schema.Parser().parse(new
```

```

File("/home/Hadoop/Avro/schema/emp.avsc"));
    DatumReader<GenericRecord> datumReader = new
GenericDatumReader<GenericRecord>(schema);
    DataFileReader<GenericRecord> dataFileReader = new
DataFileReader<GenericRecord>(new
File("/home/Hadoop/Avro_Work/without_code_gen/mydata.txt"), datumReader);
    GenericRecord emp = null;

    while (dataFileReader.hasNext()) {
        emp = dataFileReader.next(emp);
        System.out.println(emp);
    }
    System.out.println("hello");
}
}
}

```

Browse into the directory where the generated code is placed. In this case, it is at **home/Hadoop/Avro\_work/without\_code\_gen**.

```
$ cd home/Hadoop/Avro_work/without_code_gen/
```

Now copy and save the above program in the file named **DeSerialize.java**. Compile and execute it as shown below –

```
$ javac Deserialize.java
$ java Deserialize
```

## Output

```

{"name": "ramu", "id": 1, "salary": 30000, "age": 25, "address": "chennai"}
{"name": "rahman", "id": 2, "salary": 35000, "age": 30, "address": "Delhi"}

```

Loading [Mathjax]/jax/output/HTML-CSS/fonts/TeX/fontdata.js