JAXB Examples -Annotation based -by Mr. RAGHU

pom.xml:-

```
<dependency>
		<groupId>javax.xml.bind</groupId>
		<artifactId>jaxb-api</artifactId>
		<version>2.3.1</version>
	</dependency>
	<dependency>
		<groupId>com.sun.xml.bind</groupId>
		<artifactId>jaxb-impl</artifactId>
		<version>2.2.11</version>
	</dependency>

	<dependency>
		<groupId>com.sun.xml.bind</groupId>
		<artifactId>jaxb-core</artifactId>
		<version>2.2.11</version>
</dependency>
```

1. Model class:-

```
package com.app;

import java.util.Map;
import java.util.Set;

import javax.xml.bind.annotation.XmlAccessType;
import javax.xml.bind.annotation.XmlAccessorType;
import javax.xml.bind.annotation.XmlAttribute;
import javax.xml.bind.annotation.XmlElement;
```

```
import javax.xml.bind.annotation.XmlElementWrapper;
import javax.xml.bind.annotation.XmlRootElement;

@XmlRootElement
@XmlAccessorType(XmlAccessType.FIELD)
public class Employee {
	//@XmlTransient
	@XmlAttribute
	private int empId;
	@XmlElement(name = "employee-name",required = true)
	private String empName;
	@XmlElement(name="employee-salary")
	private double empSal;
	@XmlElement(name = "address")
	private Address addr;
	@XmlElementWrapper(name = "projects")
	@XmlElement(name = "proj-name")
	private Set<String> projs;
	@XmlElementWrapper(name = "proj-version")
	private Map<String,Integer> versions;

	public Map<String, Integer> getVersions() {
		return versions;
	}
	public void setVersions(Map<String, Integer>
versions) {
		this.versions = versions;
	}
	public Set<String> getProjs() {
		return projs;
	}
	public void setProjs(Set<String> projs) {
		this.projs = projs;
	}
	public Address getAddr() {
		return addr;
	}
	public void setAddr(Address addr) {
		this.addr = addr;
	}
	public int getEmpId() {
		return empId;
```

```
	}
	public void setEmpId(int empId) {
		this.empId = empId;
	}
	public String getEmpName() {
		return empName;
	}
	public void setEmpName(String empName) {
		this.empName = empName;
	}
	public double getEmpSal() {
		return empSal;
	}
	public void setEmpSal(double empSal) {
		this.empSal = empSal;
	}
	@Override
	public String toString() {
		return "Employee [empId=" + empId + ", empName="
+ empName + ", empSal=" + empSal + ", addr=" + addr
				+ ", projs=" + projs + ", versions=" +
versions + "]";
	}

}
```

2. Test class:-

```
package com.app;

import java.io.File;
import java.util.HashMap;
import java.util.HashSet;
import java.util.Map;
import java.util.Set;

import javax.xml.bind.JAXBContext;
import javax.xml.bind.Marshaller;
import javax.xml.bind.Unmarshaller;
/**
 *
```

```
 * @author RAGHU
 *
 */
public class MarshalTest {

	public static void main(String[] args) {
		objectToXml();
		xmlToObject();
	}

	public static void xmlToObject() {
		try {
			JAXBContext
jb=JAXBContext.newInstance(Employee.class);
			Unmarshaller um=jb.createUnmarshaller();
			Employee emp=(Employee) um.unmarshal(new
File("d:/myxmls/emp.xml"));
			System.out.println(emp);
		} catch (Exception e) {
			e.printStackTrace();
		}
	}

	public static void objectToXml() {
		try {
			Address addr=new Address();
			addr.setAid(15);
			addr.setHno("4-44");
			addr.setLoc("HYD");
			Employee e=new Employee();
			e.setEmpId(10);
			e.setEmpName("A");
			e.setEmpSal(3.3);
			e.setAddr(addr);

			Set<String> projs=new HashSet<>();
			projs.add("HTC1");
			projs.add("HTC2");
			projs.add("HTC3");
			e.setProjs(projs);

			Map<String,Integer> map=new HashMap<>();
```

```
                map.put("V11", 1);
                map.put("AA", 2);
                map.put("BB", 3);
                e.setVersions(map);

                JAXBContext
context=JAXBContext.newInstance(Employee.class);
                Marshaller m=context.createMarshaller();
                m.marshal(e, new File("d:/myxmls/emp.xml"));
                System.out.println("done");
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
}
```

FB: https://www.facebook.com/groups/thejavatemple/

email: javabyraghu@gmail.com