

XStream

1. ¿Qué es XStream?

XStream es una biblioteca de Java que permite convertir objetos en XML y JSON y viceversa. Es útil para la persistencia de datos, configuración de aplicaciones y transmisión de objetos en redes.

Características principales

- Fácil de usar, con una API simple.
- No requiere modificar las clases (no necesita implementar `Serializable`).
- Soporta alias para simplificar la salida XML.
- Permite personalización mediante convertidores (`Converters`).
- Compatible con múltiples formatos de entrada y salida (XML y JSON).

2. Instalación

Para usar XStream, se debe agregar la dependencia en **Maven**:

```
<dependency>
  <groupId>com.thoughtworks.xstream</groupId>
  <artifactId>xstream</artifactId>
  <version>1.4.20</version>
</dependency>
```

O descargar la librería desde XStream.

3. Uso Básico

3.1. Serialización (Objeto → XML)

```
import com.thoughtworks.xstream.XStream;

public class Main {
    public static void main(String[] args) {
        XStream xstream = new XStream();
        Persona persona = new Persona("Juan", 30);
        String xml = xstream.toXML(persona);
        System.out.println(xml);
    }
}

class Persona {
```

```
private String nombre;
private int edad;

public Persona(String nombre, int edad) {`
    `this.nombre = nombre;`
    `this.edad = edad;`
}
}
```

Salida XML:

```
<com.ejemplo.Persona>
  <nombre>Juan</nombre>
  <edad>30</edad>
</com.ejemplo.Persona>
```

3.2. Deserialización (XML → Objeto)

```
Persona personaDeserializada = (Persona) xstream.fromXML(xml);
System.out.println(personaDeserializada.getNombre());
```

4. Personalización

4.1. Alias (Renombrar etiquetas XML)

```
xstream.alias("persona", Persona.class);
```

Generará:

```
<persona>
  <nombre>Juan</nombre>
  <edad>30</edad>
</persona>
```

4.2. Omitir campos

```
xstream.omitField(Persona.class, "edad");
```

Faltan atributos

5. Conversión a JSON

5.1. Habilitar JSON en XStream

```
import com.thoughtworks.xstream.io.json.JsonHierarchicalStreamDriver;
XStream xstream = new XStream(new JsonHierarchicalStreamDriver());
xstream.alias("persona", Persona.class);
String json = xstream.toXML(persona);
System.out.println(json);
```

Salida JSON:

```
{
  "persona": {
    "nombre": "Juan",
    "edad": 30
  }
}
```

6. Seguridad en XStream

Desde versiones recientes, XStream **requiere permisos explícitos** para evitar ataques de deserialización maliciosa.

6.1. Permitir clases específicas

```
xstream.allowTypes(new Class[]{Persona.class});
```

O bien:

```
xstream.allowTypesByWildcard(new String[]{"com.ejemplo.*"});
```