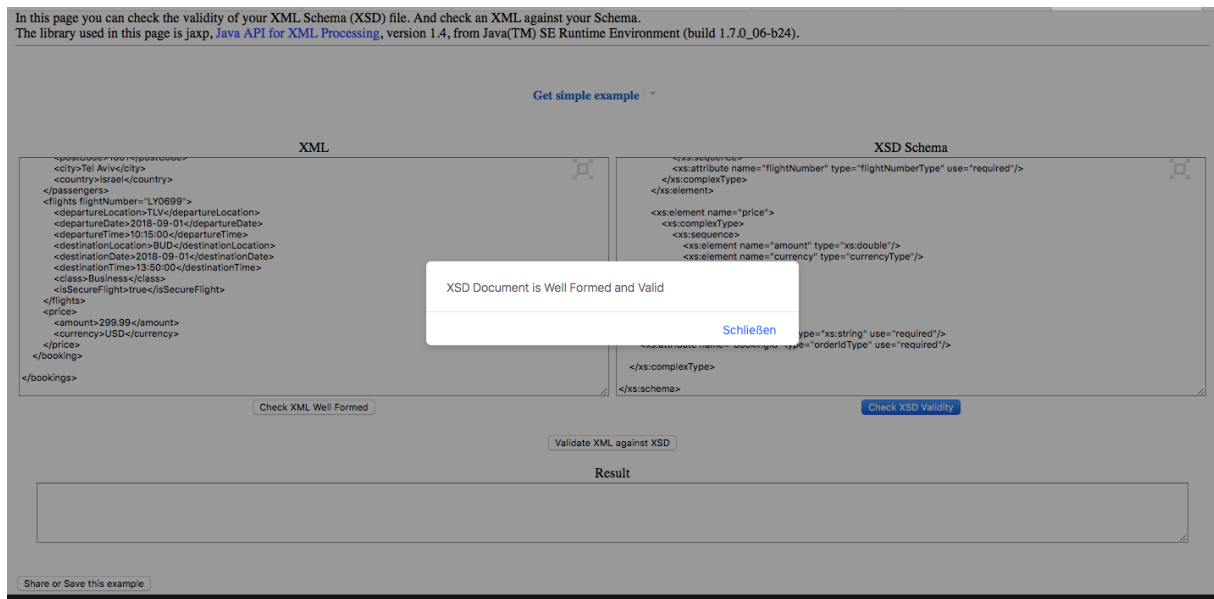
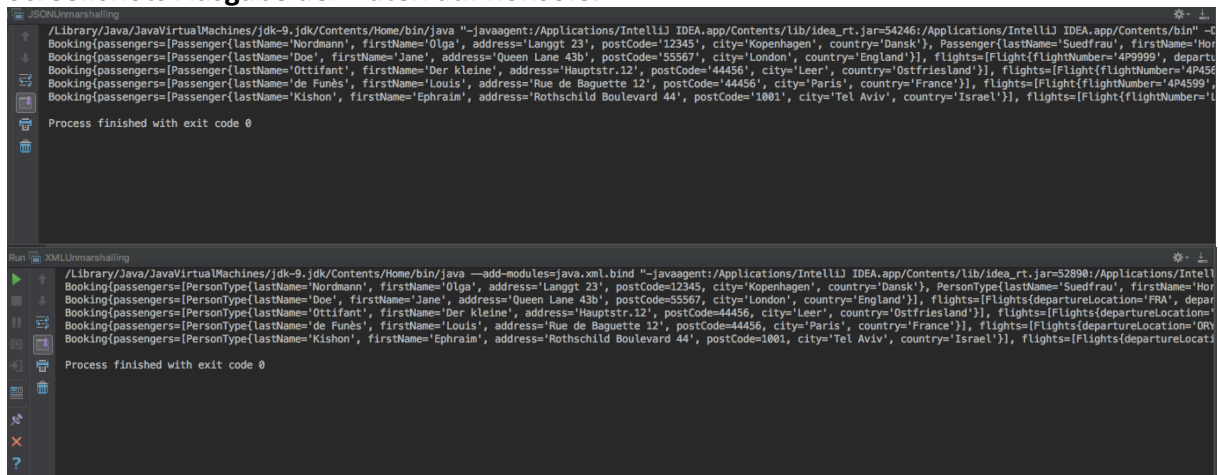


Screenshot Validierung:



Screenshots Ausgabe der Daten auf Konsole:



Fragen c:

1. Ja, generieren von Jackson Klassen umständlich (Eventuell fehlte der richtige Befehl).
2. So wie getan Generierung einer Jackson Klasse und anschließendes Unmarshalling.

Man könnte eventuell ein Attribut „Name“ einfügen und jedes Element so beschreiben.

```
{
  „flights“: [
    {
      „name“: „flight“
    }
  ]
}
```

}

Dieses Tool übernimmt dies z.T., aber evtl. nicht immer zuverlässig →
<https://timboudreau.com/blog/json/read>

D:

- Thread
- Socket
- OutputStream

Alle Klassen die nicht Serializable implementieren, sich nicht serialisierbar. Man muss das Interface Serializable implementieren um eine Klasse zu serialisieren.

Problem:

„To allow subtypes of non-serializable classes to be serialized, the subtype may assume responsibility for saving and restoring the state of the supertype's public, protected, and (if accessible) package fields.“

<https://docs.oracle.com/javase/8/docs/api/java/io/Serializable.html>

Lösung wäre im allgemeinen alle Zustände der Superklassen in einer serialisierbaren Klasse mit abzuspeichern / zu laden.

Ein weiteres Problem ist, das ein Thread nicht ohne weiteres gestoppt und auf einem anderen System wiederhergestellt werden kann, da sich die Systeme zu sehr unterscheiden (Speicheradressen etc.).