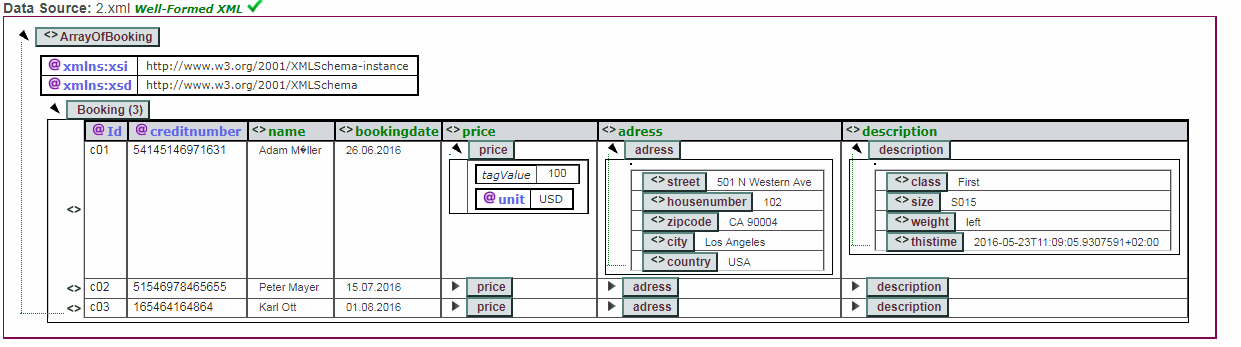
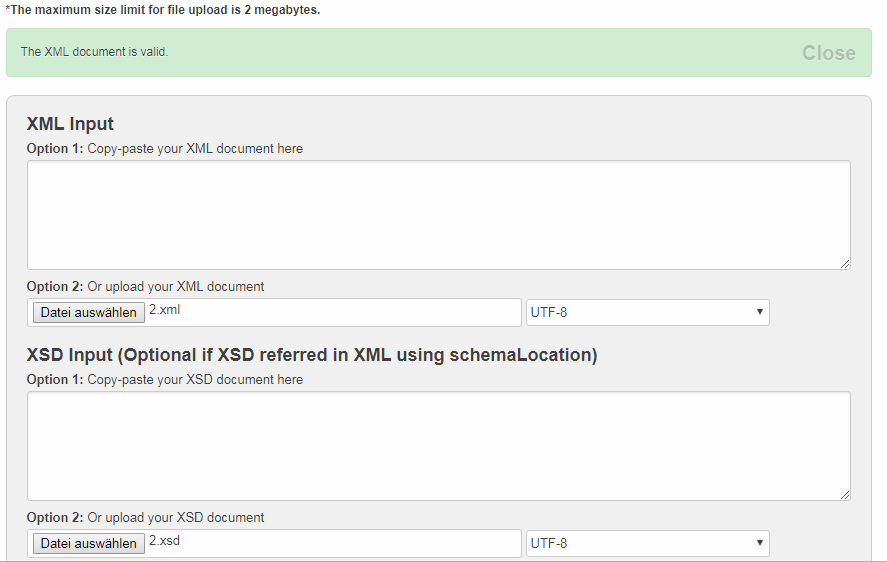
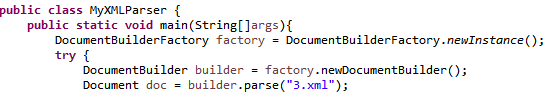
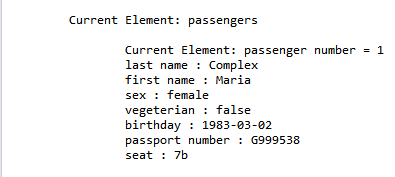
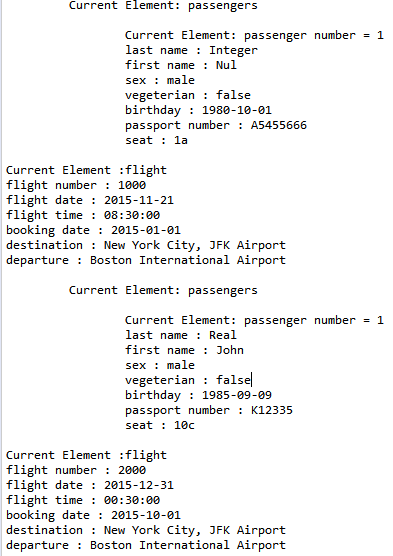
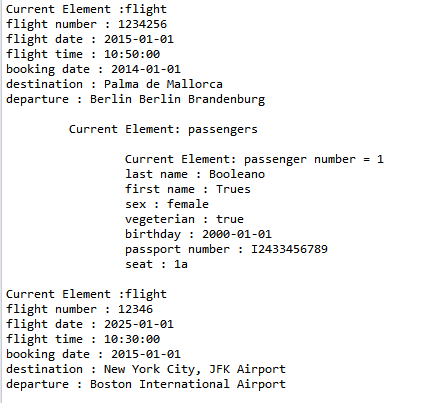
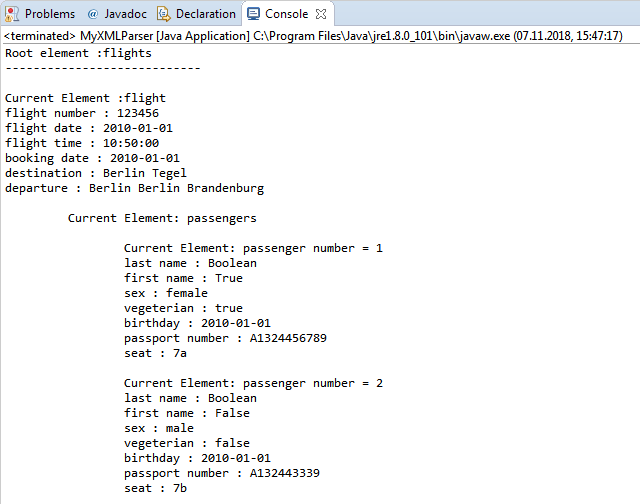
**1c)**



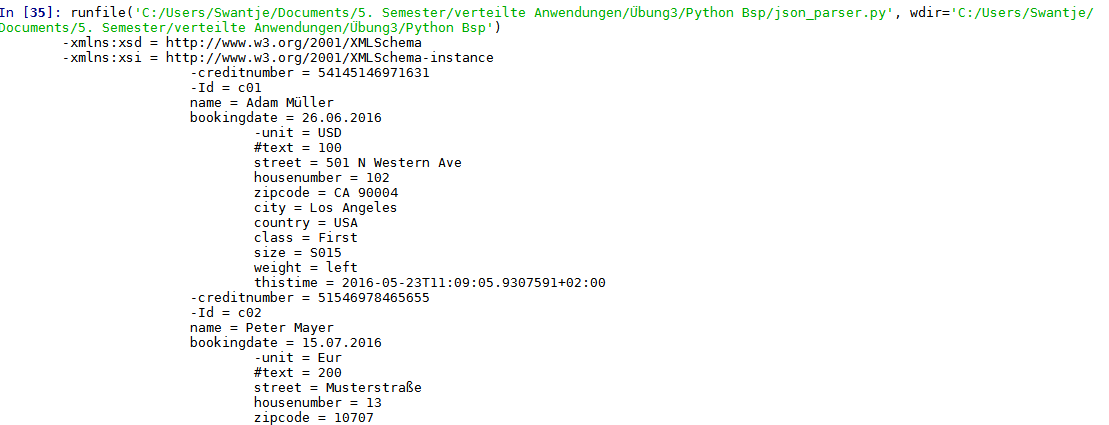
Wohlgeformt:

* am Beginn steht die Seite XML-Deklaration, die den Bezug zu XML herstellt.
* Es gibt mindestens ein Datenelement (im Beispiel sind es drei: <Dialog>...</Dialog>, <Adam>...</Adam> und <Eva>...</Eva>)
* Es gibt bei den Datenelementen ein äußerstes Element (auch als Dokument-Element bezeichnet), das alle anderen Datenelemente enthält (im Beispiel: <Dialog>...</Dialog>).

**2a)**



**2b)**



**2c)**

1: Ein Parser für XML ist meiner Meinung wesentlich komplizierter bzw. detaillierter aufgebaut, als ein JSON Parser, insbesondere, wenn es unterschiedlich viele ChildNodes gibt. So brauchte ich für den XML Parser doppelt so viele lines of Code als für den JSON Parser. Außerdem kommt man beim Schreiben einer XML Datei schnell mit den Tabs durcheinander.

2: Mit Hilfe von Mapping in Objekt umwandeln, z.B:   
 object = mapper.readValue( new File (”C:\\file.jason”), Object.class);  
 gibt Object als Array aus.

**2d)**

Json datetime (Python), Threads

Threads Problem: Zustand des Threads ändert sich ständig. Der Zustand muss so festgehalten werden, dass beim Deserialisieren auf einem anderen Rechner an diesem Zustand weiter gemacht werden kann.  
Lösung: Runnable Objekt mitgeben, das die Daten beinhaltet, Thread einfrieren (anhalten) und Zustand in Objekt mitgeben 🡪 woanders deserialisieren, neuen Thread starten

datetime Problem: wenn z.B. datetime.now gesetzt ist, dann ist es schwer diesen Zustand festzuhalten, da die momentane Uhrzeit zu dem Zeitpunkt eine andere ist, als wenn die Uhrzeit in 5 Minuten nochmal abgefragt wird.

Lösung: Datetime in einen String umwandeln, serialisieren und später wieder deserialisieren.