mPloyee - Mitarbeiterdatenkonverter

Szenario

In einem fiktiven Unternehmen liegen die Mitarbeiterdaten ausschließlich im XML-Format vor. Die Geschäftsführung mag aber lieber MS-Office-Dokumente, und in diesem Fall am liebsten alles, was mit Excel erstellt wurde. Daher wurden Sie beauftragt, einen Konverter zu implementieren, der die XML-Daten zunächst einliest und anschließend in ein annehmbares Excel-Format exportiert. Dabei soll auch berücksichtigt werden, dass später die Mitarbeiterdaten auch in einer Datenbank exportiert werden sollen.

Technologien

JSE (7 oder 8), Maven, Git, eine Bibliothek zum XML-Import, eine Bibliothek zum MS Office Export, JUnit, Mockito

Aufgabe

Entwicklung einer Software, die genau diese Vorgaben erfüllt.

Umfang der Anwendung

- Import der rohen XML-Daten
- Mappen auf Klassen
- Erzeugen der Excel-Sheets
- Schreiben von Komponenten- und ggf. Integrationstests
- Persistierung in eine Datenbank ist **NICHT** Bestandteil dieser Aufgabe

Input

Das Format für die XML-Dateien ist im Grunde frei, könnte aber wie folgt aussehen (Pseudo XML):

```
<Employees>
  <Employee>
      <Personalnummer>1</Personalnummer>
      <Vorname>Hans</Vorname>
      <Nachname>Wurst</Nachname>
      <Abteilung>IT</Abteilung>
  </Employee>
  <Employee>
      <Personalnummer>2</Personalnummer>
      ...
  </Employee>
  ...
```

```
</Employees>
```

Wichtig Erzeugen sie dieses XML-Dokument für den Import selber. Dabei bitte folgende Hinweise beachten: Bitte eine Anzahl von mindestens 10-12 Mitarbeitern einlesen, wobei mindestens 3 verschiedene Abteilungen verwendet werden! Diese 3 Abteilungen könnten IT, Controlling und Sales sein.

Output

Die Daten der Mitarbeiter sollen in ein Excel-Dokument exportiert werden, welches aus zwei Sheets besteht. Im ersten Sheet sollen einfach in einer (formatierten) Tabelle die Mitarbeiter-Daten ausgegeben werden. Im zweiten Sheet soll für jede Abteilung die Anzahl der Mitarbeiter ausgegeben werden, das sähe in etwa aus wie folgt:

Abteilung	**Anzahl Mitarbeiter**
IT	12
Controlling	1
…	…

In beiden Tabellen soll der Text in den Header-Zeilen ***fett und kursiv*** dargestellt werden. Spätestens jetzt reicht ein banaler CSV-Export nicht mehr! Eine entsprechende Bibliothek existiert.

Sonstiges

- Services sind fachlich voneinander zu trennen
- Kann man in dieser Aufgabe eine Layer-Architektur sinnvoll verwenden?
- Ein einzelner Service ist mit JUnit und Mockito atomar zu vertesten
- Mehrere Services sind mit einem Integrationstest zu vertesten
- Es ist keine UI notwendig für diese Aufgabe. Es reicht, wenn mit Maven ein ausführbares Jar erzeugt werden kann.
- Wie oben beschrieben ist die Persistierung in eine Datenbank nicht Bestandteil dieser Aufgabe, es soll bei der Implementierung jedoch berücksichtigt werden, dass man das zukünftig so machen will und man jetzt bereits an entsprechende Entities denkt.
- Ich freue mich sehr über UML-Diagramme
- Git Repository: https://bitbucket.org/amonschau/mployee/ (ist auf public geschaltet, sollte also klappen. Wenn nicht, bitte melden)