

Testaufgabe – Entwickler

Umzusetzen ist eine REST- Schnittstelle mittels JavaEE oder SpringBoot, welche folgende Anforderungen erfüllt:

- Über die Schnittstelle sollen Personen und ihre Lieblingsfarbe verwaltet werden.
- Diese Daten sollen aus einer CSV- Datei ausgelesen werden, diese soll nicht angepasst werden
- Über die Schnittstelle können alle Personen mit einer spezifischen Lieblingsfarbe ermittelt werden.

Beispiel für die CSV- Datei:

Müller, Hans, 67742 Lauterecken, 1

Petersen, Peter, 18439 Stralsund, 2

Johnson, Johnny, 88888 Ausgedacht, 3

Millenium, Milly, 77777 Auch ausgedacht, 4 Müller, Jonas, 32323 Hansstadt, 5 Fujitsu, Tastatur, 42342 Japan, 6 Andersson, Anders, 32132 Schweden - Bonus, 2 Bart, Bertram, 12313 Wasweißich, 1 Gerber, Gerda, 76535 Woanders, 3 Klaussen, Klaus, 43246 Hierach, 2

Die Zahlen der 4. Spalte entsprechen dabei folgenden Farben:

1 -> Blau

2 -> Grün

3 -> Lila

4 -> Rot

5 -> Zitronengelb

6 -> Türkis

7 -> Weiß

Das Rückgabeformat der Daten entspricht folgendem Format (Content-Type: application/json):

Die zu implementierende REST- Schnittstelle soll folgende Methoden umfassen:

GET /persons

Methode um alle in der CSV- vorhandenen User zu holen

Rückgabe- Beispiel:

```
[  
{  
  "id" : 1,  
  "vorname" : "Hans",
```

```
"nachname": "Müller",
"zipcode" : "67742",
"city" : "Lauterecken",
"color" : "Blau"
},
{
  "id" : 2,
  ...
}
]
```

GET /persons/{id}

Methode um die Person mit der gegebenen ID aus der CSV- Datei zu holen.

Da die Datei keine IDS beinhaltet ist die entsprechende Zeilennummer relevant (erste Zeile - Zeile 0).

Bsp: id 1 - Petersen, Peter, 18439 Stralsund, Blau Existiert keine Person mit dieser ID, so ist ein geeigneter Http- Statuscode zu verwenden.

Rückgabe- Beispiel:

```
{
  "id" : 1,
  "vorname" : "Hans",
  "nachname": "Müller",
  "zipcode" : "67742",
  "city" : "Lauterecken",
  "color" : "Blau"
}
```

GET /persons/color/{color}

Methode um alle Personen mit derselben Lieblingsfarbe zu holen. Die Farbe sollte entsprechend codiert werden.

Rückgabe- Beispiel:

```
[
{
  "id" : 1,
```

```
"vorname" : "Hans",  
"nachname": "Müller",  
"zipcode" : "67742",  
"city" : "Lauterecken",  
"color" : "Blau"
```

```
},
```

```
{
```

```
"id" : 2,
```

```
...
```

```
}
```

```
]
```

Aufgaben:

1. Auslesen der CSV- Datei und Zwischenspeichern in einer zum Json passenden Model- Klasse (selbst zu implementieren) Dies soll im Idealfall in einer Klasse geschehen welche den Zugriff auf die Datei wegekapselt, um diese später einfach durch eine Datenbank ersetzen zu können.

2. Implementieren der angegebenen REST- Schnittstelle Diese greift per Dependency Injection auf die Persistenz- Klasse zu.

3. Schreiben von passenden Unittests für die Schnittstelle (z.B. testGetUsersWithColor(), testGetAllUsers())

4. Bonus- Aufgaben:

- Umsetzung als Maven- Projekt

- Implementieren einer zusätzlichen Methode POST /persons welche einen neuen Datensatz zur CSV- Datei hinzufügt.

- JPA- Anbindung einer mysql- Datenbank zur persistierung der Daten