

ZIEL

Kunde & Memo-Liste anzeigen (ohne Add Memo Funktionalität).

Kundendaten

Bucher Michael
Bachtobelstrasse 21
8045 Zurich

Memos

		<button>Add Memo</button>
13.10.2017 11:26	Erstkontakt, Kunde unsicher, nachhaken	
13.10.2017 11:26	Chefsekretär, Hund heisst Mischu, keine Familie, Ansprechpartner für alles	

Abbildung 1 Kundendaten mit Memo

Wenn Sie sich die Datei «showCustomerList.html» ansehen, sehen Sie, dass wir dort einen Link definiert haben welcher die Anfrage an die URL «/user/showCustomer/{id}» sendet. «id» steht dabei für die Kundennummer («customerId») des anzuzeigenden Kunden.

Konsequenterweise brauchen wir in unserem Controller ein neues Mapping dass diese Anfrage der verantwortlichen Funktion zuweist.

Das ist soweit nichts Neues, wir sind der Annotation «@RequestMapping(...)» bereits begegnet. Neu ist, dass ein Teil der URL variabel ist und dass wir diesen variablen Teil brauchen um das passende Kundenobjekt zu finden.

«@RequestMapping» erlaubt es uns, einen Teil der URL als variabel zu definieren und «@PathVariable» ermöglicht es, diesen Teil automatisch in den Parameter einer Funktion zu übernehmen («inject»).

IMPLEMENTIEREN DES MVC-ABLAUFES

CONTROLLER

```
@RequestMapping("/user/showCustomer/{id}")
public String showCustomer(Model model, @PathVariable("id") long customerId) {
    // get the customer object from the customerService
    // store it as "customer" in the model
    // and return „showCustomer“ as the new view
}
```

{id} steht dabei als Platzhalter für einen Teil der URL, «@PathVariable(„id“») bedeutet, dass dieser Wert in die nachfolgende Variable «customerId» übernommen werden soll.

☺ Controller erledigt ☺

SERVICE

Die Funktion «getCustomer()» in der Klasse «CustomerServiceImpl» ist noch nicht fertig implementiert. Implementieren Sie diese, indem Sie die Funktionalität «customerRepository.findOne(...)» benutzen.

☺ Service erledigt ☺

VIEW

Code bereits erhalten. ☺ View erledigt ☺

TEST

Starten Sie den Server und benutzen Sie die neue Funktionalität. Was läuft und was nicht? Warum?

KUNDENINFORMATIONEN

Wie Sie gesehen haben werden keine Kundeninformationen (keine Liste von Memos) angezeigt. Das verwundert nicht weiter. Erstens haben wir noch keine Memos erstellt und zweitens fehlt noch die Verknüpfung von Customer-Objekten mit den zugehörigen Memo-Objekten.

Wir haben also folgende Dinge zu erledigen:

1. Implementieren der Klasse «MemoImpl»
2. Implementieren des Interface «MemoRepository»
3. Erstellen von Memo-Objekten in der Klasse «DataInitializer»
4. Implementieren der JPA-Verknüpfungen in den Klassen «CustomerImpl» und «MemoImpl»
5. Ergänzen der Klasse «CustomerServiceImpl»
6. Ergänzen der Funktion «run()» aus der Klasse «DataInitializer»

1-3 UND 5

Gehen Sie dazu analog vor, wie wir es für die Klasse Customer gemacht haben.

IMPLEMENTIEREN DER JPA-VERKNÜPFUNGEN IN DER KLASSE «CUSTOMERIMPL»

```
// Two directional mapping
// Each customer owns its memos
// Deleting a customer deletes its memos too.
@OneToMany(mappedBy="customer", fetch=FetchType.EAGER, cascade = CascadeType.REMOVE)
private List<MemoImpl> memos;
```

IMPLEMENTIEREN DER JPA-VERKNÜPFUNGEN IN DER KLASSE «MEMOIMPL»

```
// Two directional mapping,
// each memo belongs to exact one customer
@ManyToOne
private CustomerImpl customer;
```

ERGÄNZEN DER FUNKTION «RUN()» AUS DER KLASSE «DATAINITIALIZER»

Ergänzen Sie die Funktion «DataInitializer.run()» mit dem folgenden Code:

```
customerService.addMemoToCustomer(c0.getId(),
    "Erstkontakt positiv verlaufen");

customerService.addMemoToCustomer(c0.getId(),
    "Sitzung vereinbart");

customerService.addMemoToCustomer(c0.getId(),
    "Offerteneingang bestätigt");

customerService.addMemoToCustomer(c1.getId(),
    "Erstkontakt, Kunde unsicher, nachhaken");

customerService.addMemoToCustomer(c1.getId(),
    "Chefsekretärin, Hund heisst Mischu, keine Familie, Ansprechpartner für alles");
```

Wenn Sie alles richtiggemacht haben, sollten nun auch die Memos angezeigt werden.