31.8.–3.9.2015
in Nürnberg

Herbstcampus

Wissenstransfer
par excellence

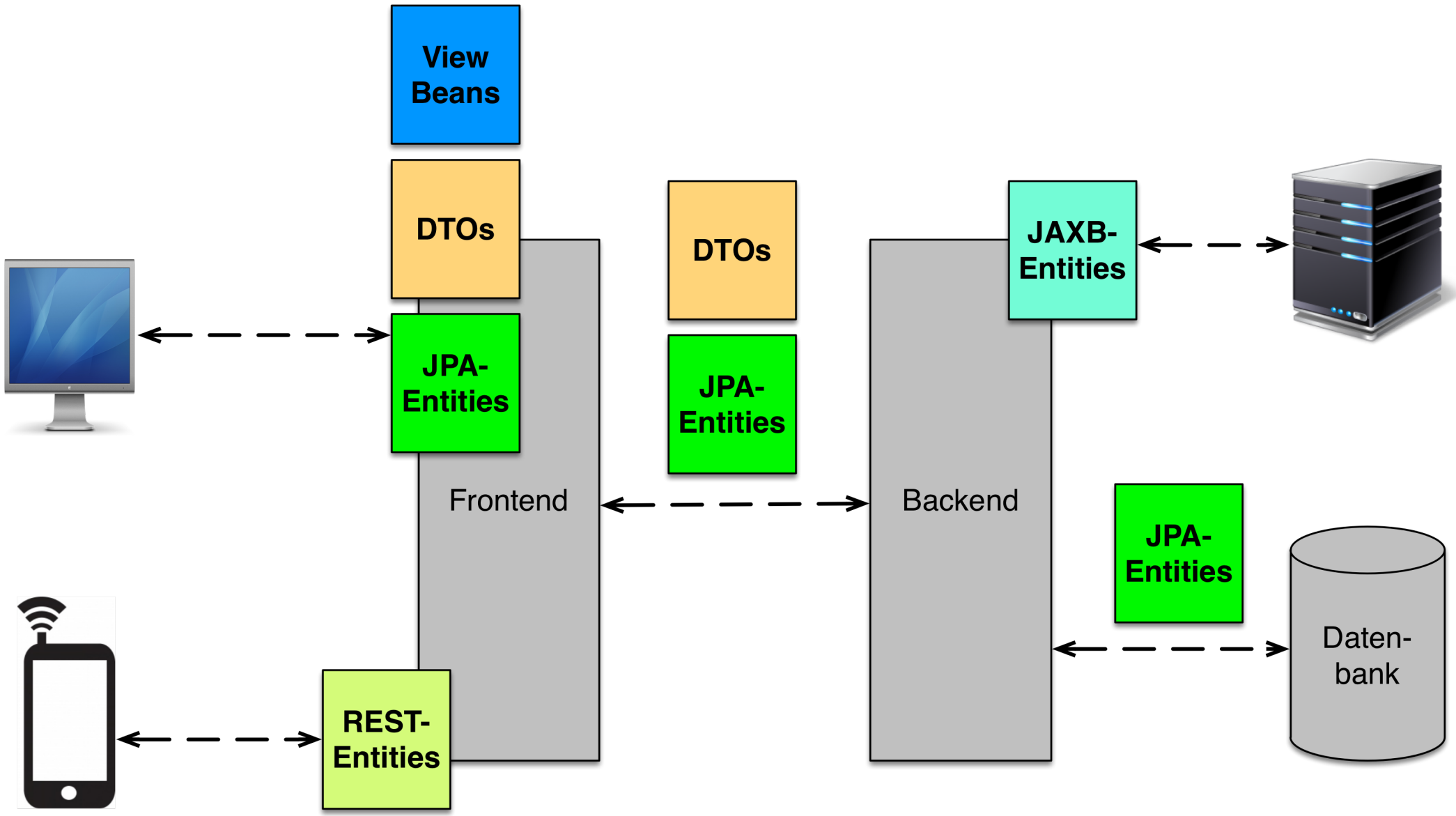
Bean-Mapping mit MapStruct

Ein Appetizer für Java-Entwickler

Thomas Much

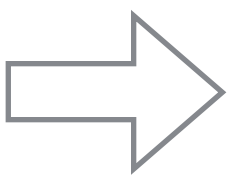
muchsoft.com

Beans, DTOs … in aktuellen Architekturen



Wie mappen wir folgende Beans?

```
@Entity
@Table(name = "KUNDEN")
public class Kunde {

  @Id
  Long id;

  long kundennummer;

  String name;

  @Enumerated
  KundenArt kundenart;

  … Getter und Setter! …
}
```



```
public class KundeDTO {

  long id;

  String kundennummer;

  String name;


  String kundenart;

  … Getter und Setter! …
}
```

MapStruct minimal

```
import org.mapstruct.Mapper;
import org.mapstruct.factory.Mappers;


@Mapper
public interface KundeMapper {

    KundeDTO kunde2KundeDto(Kunde kunde);


    KundeMapper INSTANCE =
        Mappers.getMapper(KundeMapper.class);
}
```

```
KundeDTO dto =
  KundeMapper.INSTANCE.kunde2KundeDto( kunde );
```

Implementierung?

- Wie erzeugen wir aus dem Interface die Mapping-Implementierung?

- *Speichern* z.B. in Eclipse.
- Genauer: Compiler-Aufruf (geht also auch bei javac)

- Annotations-Prozessor erledigt den Rest!

Generierter Code

```
@Generated(
    value = "org.mapstruct.ap.MappingProcessor",
    date = "2015-04-21T22:01:21+0200",
    comments = "version: 1.0.0.Beta4, compiler: Eclipse JDT (IDE) …"
)
public class KundeMapperImpl implements KundeMapper {

    @Override
    public KundeDTO kunde2KundeDto(Kunde kunde) {
        if ( kunde == null ) {
            return null;
        }

        KundeDTO kundeDTO = new KundeDTO();

        kundeDTO.setName( kunde.getName() );
        kundeDTO.setKundennummer( String.valueOf( kunde.getKundennummer() ) );
        if ( kunde.getKundenArt() != null ) {
            kundeDTO.setKundenArt( kunde.getKundenArt().toString() );
        }
        if ( kunde.getId() != null ) {
            kundeDTO.setId( kunde.getId() );
        }

        return kundeDTO;
    }
}
```

Typsicherheit

KundeMapper.java | KundeDTO.java | Kunde.java

MapStruct ▸ src/main/java ▸ mapper ▸ KundeMapper ▸ kunde2KundeDto(Kunde) : KundeDTO

```
1  package mapper;
2
3  import model.Kunde;
9
10 @Mapper
11 public interface KundeMapper {
12
13     KundeMapper INSTANCE = Mappers.getMapper(KundeMapper.class);
14
15     KundeDTO kunde2KundeDto(Kunde kunde);
16
17 }
```

Problems | Javadoc | Declaration

1 error, 1 warning, 0 others

Description	Resource	Path
Errors (1 item)		
Can't map property "model.KundenArt kundenArt" to "java.lang.Long kundenArt"....	KundeMapper.java	/MapStr
Warnings (1 item)		
Unmapped target property: "name".	KundeMapper.java	/MapStr

Flexibilität

- Nahezu alles konfigurierbar:

- Namen, Standard- und eigene Typkonvertierungen, Listen, Arrays, Objektgraphen, String-Formate bei Date …
- Komponentenmodell (CDI etc.)

Neugierig geworden?

http://mapstruct.org/

😀