



ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE

Spring Workshop



4 - Data



Spring Data

Spring Data JPA

Standard

JPA

Implementation

Hibernate

@Entity and other Annotations

- Associates a database table to a class
- Classname can be different from table name
- Be aware of Naming Strategies
 - Implicit
 - Physical
 - Spring



Entity

- @Entity
- @Table
- @Id
- @GeneratedValue
 - IDENTITY: auto increment MySQL, identity in SQL Server
 - SEQUENCE: Oracle
 - (TABLE)
- @Transient
- @Column
- BeanValidation



Repository

- Interface implemented by Spring Data
- Manages database specific actions
- Common Subtypes
 - (List)CrudRepository
 - (List)PagingAndSortingRepository
 - JpaRepository
- Customisable via find* methods



Customer as Entity

```
public class Customer {  
  
    private Long id;  
  
    private String firstname;  
    private String lastname;  
  
    private Boolean signedGdpr;  
}
```



ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE

Customer as Entity

`@Entity`

```
public class Customer {  
  
    private Long id;  
  
    private String firstname;  
    private String lastname;  
  
    private Boolean signedGdpr;  
}
```



ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE

Customer as Entity

`@Entity`

```
public class Customer {
```

`@Id`

```
@GeneratedValue(strategy = GenerationType.IDENTITY)
```

```
private Long id;
```

```
private String firstname;
```

```
private String lastname;
```

```
private Boolean signedGdpr;
```

```
}
```



ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE

CustomersRepository

```
package com.softarc.eternal.data;  
  
import com.softarc.eternal.domain.Customer;  
import java.util.List;  
import org.springframework.data.jpa.repository.JpaRepository;  
  
public interface CustomersRepository extends JpaRepository {}
```



CustomersRepository

```
package com.softarc.eternal.data;  
  
import com.softarc.eternal.domain.Customer;  
import java.util.List;  
import org.springframework.data.jpa.repository.JpaRepository;  
  
public interface CustomersRepository extends JpaRepository<Customer, Long> {}
```



CustomersRepository

```
package com.softarc.eternal.data;

import com.softarc.eternal.domain.Customer;
import java.util.List;
import org.springframework.data.jpa.repository.JpaRepository;

public interface CustomersRepository extends JpaRepository<Customer, Long> {
    List<Customer> findByFirstnameOrLastnameOrderByLastname(
        String firstname,
        String lastname
    );
}
```



Database Connection in application.yml for H2

```
spring:
  datasource:
    url: jdbc:h2:./dev-db
    driver-class-name: org.h2.Driver
    username: sa
    password: 'Sag ich dir nicht :)'
```

- H2 is embedded/file/server
- Possibility to use multiple databases
- Make use of profiles
- Credentials
 - plaintext
 - encrypted
 - OS env



Further useful configuration properties

```
spring:
  jpa:
    hibernate:
      ddl-auto: create-drop

logging:
  level:
    org.hibernate.orm.jdbc.bind: trace

spring:
  jpa:
    show-sql: true
```



Impact on Testing

- No on impact on Unit Tests (no DI)
- Override the datasource properties via `@SpringBootTest`
 - `jdbc:h2:mem:holidays-controller`
 - in-memory instance
- Mock the Repository via `@MockedBean` in Integration Tests



```
55 private void executeLoad(long timeout, int usersCount) {
```

```
56     showDebugInfo(timeout);
```

```
57     Load.setPages(URL, parsingTimeout);
```

```
58     Load.setTimeout(timeout);
```

```
59     List<Load> threads = new ArrayList<>();
```

```
60     for (int i = 0; i < usersCount; i++) {
```

```
61         threads.add(new Load(this.URL));
```

```
62     }
```

```
63     logger.info( s: usersCount +
```

```
64         " threads to process");
```

```
65     for (Load thread : threads)
```

```
66         thread.start();
```

```
67     }
```

```
68     logger.info( s: "All threads are started");
```

```
69     progressInfo(timeout);
```

```
70     System.out.print(".....DONE\nProcessing with data...\n");
```

```
71 }
```

```
72 private void executeAvailability(long timeout, int
```

```
73 }
```

```
74 private void
```

Demo

Lab Time



@DataJpaTest

- In-Memory Database is normally used
- Transactional with Rollback
- SQL commands and queries are printed



@DataJpaTest

```
@DataJpaTest(  
    properties = {  
        "spring.datasource.url=jdbc:h2:mem:test",  
        "spring.jpa.hibernate.ddl-auto=create-drop",  
    }  
)  
public class CustomersControllerIntegrationTest {  
  
    @Autowired  
    CustomersRepository repository;  
  
    @Test  
    void testRepo() {  
        var franz = CustomerMother.franz().build();  
        repository.save(franz);  
        Customer customer = repository.findAll().get(0);  
        assertThat(customer)  
            .usingRecursiveComparison()  
            .ignoringFields("id")  
            .isEqualTo(franz);  
    }  
}
```



Flyway

- Versioned Migration Tool
- Uses SQL
- Files in resources with convention V{n}__{name}.sql
- Liquibase (database agnostic)
- Careful with line ending (sql files are hashed)
- Rollback possible



Flyway / MySql Setup

1. Setup MySQL via Docker or in another way
2. Register datasource.url
 - a. `jdbc:mysql://localhost:3306/eternal`
3. Add dependencies
 - a. `implementation 'mysql:mysql-connector-java'`
`implementation 'org.flywaydb:flyway-core'`
`implementation 'org.flywaydb:flyway-mysql'`
4. Create init schema in `src/main/resources/db/main/V1__init.sql`
5. Restart Spring Boot



Relations

- Used quite often
 - @ManyToOne
 - @OneToMany
- Rarely
 - @OneToOne
 - @ManyToMany



Relations

@Entity

```
public class Country {
```

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

```
private Long id;
```

```
private String name;
```

@OneToMany(mappedBy = "country")

```
private List<Customer> customers;
```

```
}
```



ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE

Relations

@Entity

```
public class Customer {
```

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

```
private Long id;
```

```
private String firstname;
```

```
private String lastname;
```

```
private Boolean signedGdpr;
```

@ManyToOne

@JoinColumn(name = "country_id") // optional

```
private Country country;
```

```
}
```



ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE

```
55 private void executeLoad(long timeout, int usersCount) {
```

```
56     showDebugInfo(timeout);
```

```
57     Load.setPages(URL, parsingTimeout);
```

```
58     Load.setTimeout(timeout);
```

```
59     List<Load> threads = new ArrayList<>();
```

```
60     for (int i = 0; i < usersCount; i++) {
```

```
61         threads.add(new Load(this.URL));
```

```
62     }
```

```
63     logger.info( s: usersCount +
```

```
64         " threads to process");
```

```
65     for (Load thread : threads)
```

```
66         thread.start();
```

```
67     }
```

```
68     logger.info( s: "All threads are started");
```

```
69     progressInfo(timeout);
```

```
70     System.out.print(".....DONE\nProcessing with data...\n");
```

```
71 }
```

```
72
```

```
73
```

```
74
```

```
75
```

```
76
```

```
77
```

```
78
```

```
79
```

```
80
```

```
81
```

```
82
```

```
83
```

```
84
```

```
85
```

```
86
```

```
87
```

```
88
```

```
89
```

```
90
```

```
91
```

```
92
```

```
93
```

```
94
```

```
95
```

```
96
```

```
97
```

```
98
```

```
99
```

```
100
```

```
101
```

```
102
```

```
103
```

```
104
```

```
105
```

```
106
```

```
107
```

```
108
```

```
109
```

```
110
```

```
111
```

```
112
```

```
113
```

```
114
```

```
115
```

```
116
```

```
117
```

```
118
```

```
119
```

```
120
```

```
121
```

```
122
```

```
123
```

```
124
```

```
125
```

```
126
```

```
127
```

```
128
```

```
129
```

```
130
```

```
131
```

```
132
```

```
133
```

```
134
```

```
135
```

```
136
```

```
137
```

```
138
```

```
139
```

```
140
```

```
141
```

```
142
```

```
143
```

```
144
```

```
145
```

```
146
```

```
147
```

```
148
```

```
149
```

```
150
```

```
151
```

```
152
```

```
153
```

```
154
```

```
155
```

```
156
```

```
157
```

```
158
```

```
159
```

```
160
```

```
161
```

```
162
```

```
163
```

```
164
```

```
165
```

```
166
```

```
167
```

```
168
```

```
169
```

```
170
```

```
171
```

```
172
```

```
173
```

```
174
```

```
175
```

```
176
```

```
177
```

```
178
```

```
179
```

```
180
```

```
181
```

```
182
```

```
183
```

```
184
```

```
185
```

```
186
```

```
187
```

```
188
```

```
189
```

```
190
```

```
191
```

```
192
```

```
193
```

```
194
```

```
195
```

```
196
```

```
197
```

```
198
```

```
199
```

```
200
```

```
201
```

```
202
```

```
203
```

```
204
```

```
205
```

```
206
```

```
207
```

```
208
```

```
209
```

```
210
```

```
211
```

```
212
```

```
213
```

```
214
```

```
215
```

```
216
```

```
217
```

```
218
```

```
219
```

```
220
```

```
221
```

```
222
```

```
223
```

```
224
```

```
225
```

```
226
```

```
227
```

```
228
```

```
229
```

```
230
```

```
231
```

```
232
```

```
233
```

```
234
```

```
235
```

```
236
```

```
237
```

```
238
```

```
239
```

```
240
```

```
241
```

```
242
```

```
243
```

```
244
```

```
245
```

```
246
```

```
247
```

```
248
```

```
249
```

```
250
```

```
251
```

```
252
```

```
253
```

```
254
```

```
255
```

```
256
```

```
257
```

```
258
```

```
259
```

```
260
```

```
261
```

```
262
```

```
263
```

```
264
```

```
265
```

```
266
```

```
267
```

```
268
```

```
269
```

```
270
```

```
271
```

```
272
```

```
273
```

```
274
```

```
275
```

```
276
```

```
277
```

```
278
```

```
279
```

```
280
```

```
281
```

```
282
```

```
283
```

```
284
```

```
285
```

```
286
```

```
287
```

```
288
```

```
289
```

```
290
```

```
291
```

```
292
```

```
293
```

```
294
```

```
295
```

```
296
```

```
297
```

```
298
```

```
299
```

```
300
```

```
301
```

```
302
```

```
303
```

```
304
```

```
305
```

```
306
```

```
307
```

```
308
```

```
309
```

```
310
```

```
311
```

```
312
```

```
313
```

```
314
```

```
315
```

```
316
```

```
317
```

```
318
```

```
319
```

```
320
```

```
321
```

```
322
```

```
323
```

```
324
```

```
325
```

```
326
```

```
327
```

```
328
```

```
329
```

```
330
```

```
331
```

```
332
```

```
333
```

```
334
```

```
335
```

```
336
```

```
337
```

```
338
```

```
339
```

```
340
```

```
341
```

```
342
```

```
343
```

```
344
```

```
345
```

```
346
```

```
347
```

```
348
```

```
349
```

```
350
```

```
351
```

```
352
```

```
353
```

```
354
```

```
355
```

```
356
```

```
357
```

```
358
```

```
359
```

```
360
```

```
361
```

```
362
```

```
363
```

```
364
```

```
365
```

```
366
```

```
367
```

```
368
```

```
369
```

```
370
```

```
371
```

```
372
```

```
373
```

```
374
```

```
375
```

```
376
```

```
377
```

```
378
```

```
3
```


Lab Time

