

DIAMOND



PLATINUM



GOLD



Partner Communities

TABARA
DE TESTARETHE
BA
LOUNGEPM
ELITEPeople
of Design
Iasi

IASIJS

Scriu
Cod

.net IS

iOS
NSAgoraRGDA
ROMANIAN GAME DEVELOPERS ASSOCIATIONStartup
Weekend
IasiȘcoala
informală
de IT

AgileHub

Digital Product
Group Iasi

FinTech Camp

HANGARIADA
Festival de arte si gloriITstGirls in Tech
ROMANIAGIRLS
<code>
Romania

Media Partners

joobs

Iasi Airport
Your way to the skywink
public multimediaBellaria
HOTEL
****upc
business

Wellness Partners

fructe
in panere

kineticfit



Share the buzz!

#CodeCampplasi

Spring Data: Get rid of boilerplate DAOs

„Winners never quit & quitters never win.”

- Vince Lombardi

Cosmin Alexandru SPÎNU

Software Developer @ TSS-Yonder

CodeCamp, apr. 2016

meetAudience();

- ▶ What brings you here ?
- ▶ Who heard about Spring Data ?
 - ▶ x code campers
- ▶ Who worked with Spring Data ?
 - ▶ x code campers

why();

```
graph LR; A[Access Layer implementation with specific APIs] --> B[A lot of ways for implementing DAOs]; B --> C[Boilerplate]
```

Access Layer
implementation with
specific APIs

A lot of ways
for
implementing
DAOs

Boilerplate

searchSolution();

- ▶ Spring Data JPA ofc
- ▶ Easier way to develop apps using RDB or newer technologies
- ▶ JPA, MongoDB, Redis, Solr, REST, Cassandra, ElasticSearch and many others

printCurrentWay();

- ▶ Persistence mechanism from EJB
- ▶ JDBC API
- ▶ ORM - Hibernate
- ▶ JPA

printNextWay();

▶ Spring Framework

- ▶ Provides sophisticated support for traditional data access techs
- ▶ Simplified infrastructure setup and exception translation
- ▶ The transactional mechanism

printNextWay();

▶ Spring Data

- ▶ First release @ SpringOne 2010
- ▶ Provides a simple and consistent approach for working with data persistence
- ▶ Reduces the necessary effort for having the persistence layer in place

getRepository();

- ▶ *The goal of Spring Data repository abstraction is to significantly reduce the amount of boilerplate code required to implement data access layers for various persistence stores.*
 - This is what Spring officialy says.
- ▶ The central interface for repository abstraction is ... Repository 😊

getRepository();

Type hierarchy of 'org.springframework.data.repository.Repository':

- ▲ ⓘ Repository<T, ID extends Serializable> - org.springframework.data.repository
 - ▲ ⓘ CrudRepository<T, ID extends Serializable> - org.springframework.data.repository
 - ▲ ⓘ PagingAndSortingRepository<T, ID extends Serializable> - org.springframework.data.repository
 - ▲ ⓘ JpaRepository<T, ID extends Serializable> - org.springframework.data.jpa.repository
 - ▲ ⓘ SimpleJpaRepository<T, ID extends Serializable> - org.springframework.data.jpa.repository.support
 - ▲ ⓘ QueryDslJpaRepository<T, ID extends Serializable> - org.springframework.data.jpa.repository.support

getRepository();

► Configuration ?

```
<jpa:repositories base-package="ro.codecamp.repository" />
```

```
9  
0 @EnableJpaRepositories(basePackages= { "ro.codecamp.repository" })|  
1
```

getRepository();

```
/**
 * @NoRepositoryBean
 * public interface EntityRepository<T extends BaseEntity> extends JpaRepository<T, Long> {
 *
 * }
 */

/**
 * Repository (with Spring Data JPA) for {@link CodeCamper}.
 *
 * @author Cosmin Spinu (cosmin.spinu@tss-yonder.com)
 * @since 10 apr. 2016
 */
public interface CodeCamperRepository extends EntityRepository<CodeCamper> {
}
```

getRepository();

codeCamperRepository.
ation might have changed the

- count() : long - CrudRepository
- count(Example<S> example) : long - QueryByExampleExecutor
- delete(CodeCamper entity) : void - CrudRepository
- delete(Iterable<? extends CodeCamper> entities) : void - CrudRepository
- delete(Long id) : void - CrudRepository
- deleteAll() : void - CrudRepository
- deleteAllInBatch() : void - JpaRepository
- deleteInBatch(Iterable<CodeCamper> entities) : void - JpaRepository
- equals(Object obj) : boolean - Object
- exists(Example<S> example) : boolean - QueryByExampleExecutor
- exists(Long id) : boolean - CrudRepository
- findAll() : List<CodeCamper> - JpaRepository
- findAll(Example<S> example) : List<S> - JpaRepository
- findAll(Iterable<Long> ids) : List<CodeCamper> - JpaRepository
- findAll(Pageable pageable) : Page<CodeCamper> - PagingAndSortingRepository
- findAll(Sort sort) : List<CodeCamper> - JpaRepository
- findAll(Example<S> example, Pageable pageable) : Page<S> - QueryByExampleExecutor
- findAll(Example<S> example, Sort sort) : List<S> - JpaRepository
- findByEmail(String email) : CodeCamper - CodeCamperRepository
- findById(long id) : CodeCamper - EntityRepository
- findOne(Example<S> example) : S - QueryByExampleExecutor
- findOne(Long id) : CodeCamper - CrudRepository
- flush() : void - JpaRepository
- getByIsConfirmedTrue() : List<CodeCamper> - CodeCamperRepository
- getClass() : Class<?> - Object
- getOne(Long id) : CodeCamper - JpaRepository
- hashCode() : int - Object
- notify() : void - Object
- notifyAll() : void - Object
- save(Iterable<S> entities) : List<S> - JpaRepository
- save(S entity) : S - CrudRepository
- saveAndFlush(S entity) : S - JpaRepository
- toString() : String - Object
- wait() : void - Object
- wait(long timeout) : void - Object
- wait(long timeout, int nanos) : void - Object

om proposal table or click for focus

Press 'Ctrl+Space' to show Template Proposals

findBySomething();

► Query methods

- findBy, getBy, queryBy, removeBy, deleteBy, countBy
- findDistinctBy, findFirst3, findTop3
- findByFirstNameAndLastNameOrderByAgeDesc
- findByDocumentDocumentDate, findByDocument_DocumentDate

findBySomething();

► Keywords

- And, Or, Is, Equals, Between, LessThan, LessThanEqual, After/Before, IsNull, IsNotNull, Like, NotLike, StartingWith, Containing, OrderBy, Not, In, True, False, IgnoreCase, Regex etc.

► Return types

- Void, Primitives, Wrapper types, T, Iterator<T>, Collection>T>

queryAnnotation();

▶ @Query

- ▶ Supports both, JPQL and SQL queries
- ▶ It overrides the query method behaviour
- ▶ There is no naming convention

namedQuery();

▶ Named Queries

- ▶ Define them in properties files, annotation (@NamedQuery , @NamedNativeQuery) or orm.xml file
- ▶ CodeCamper.findByEmailNamed=„...” in a property file
- ▶ <named-query> or <named-native-query> in orm.xml
- ▶ There is no naming convention

sortAndPage();

- ▶ Keyword: OrderBy
- ▶ For @Query you can use ORDER BY in the query string
- ▶ Sort Class
- ▶ Extend PagingAndSortingRepository

customRepo();

- ▶ Create an interface CustomRepository
- ▶ Implement the interface needed methods in CustomRepositoryImpl and annotate it with @Repository
- ▶ In your main repository extend the CustomRepository

advanced();

- ▶ Supports also Criteria API in Custom repos
- ▶ Dynamic queries (Criteria API - the standard way in JPA)
- ▶ Supports advanced querying with QueryDSL
 - ▶ ofc instead of Criteria API ☺
- ▶ Caching is configurable as in any other Spring/Hibernate project

audit();

- ▶ @CreatedBy , @LastModifiedBy - who ?
- ▶ @CreatedDate, @LastModifiedDate - when ?
- ▶ Spring Security integration

demo();

<https://github.com/cosminspn/cc-apr-2016/>

<https://github.com/cosminspn/cc-apr-2016/tree/master/source/spring-data-example>

removeFromTheProject();

- ▶ When Spring Data does not bring value ?
 - ▶ A lot of JOIN queries
 - ▶ DB which do not use FKs
 - ▶ When even the CRUD methods should be overridden
 - ▶ May be other situations ...

hopeForTheBest();

- ▶ Who heard about Spring Data ?
 - ▶ x code campers
- ▶ Who worked with Spring Data ?
 - ▶ x code campers

thankYou();

THANK YOU !
QUESTIONS ?

COD^ECAMP



FEEDBACK



CODECAMP

Please fill the online evaluation form after event