Generic DAOs with Hades

Oliver Gierke - Synyx GmbH & Co. KG

„Simple things should be simple, complex things should be possible.“
Alan Kay

SYNYX OpenSource Solutions

Synyx GmbH & Co. KG
Karlstraße 68
76137 Karlsruhe

phone +49(0)721 66 24 866
info@synyx.de

Agenda

- Database access with JPA / Spring
- GenericDao
- Finder methods
- Spring namespace configuration
- Base classes for domain objects
- Auditing

Synyx GmbH & Co. KG
Karlstraße 68
76137 Karlsruhe

phone +49(0)721 66 24 866
info@synyx.de

How to implement data access with JPA / Spring?

State of the art ORM

SYNYX
OpenSource Solutions

Synyx GmbH & Co. KG
Karlstraße 68
76137 Karlsruhe

phone +49(0)721 66 24 866
info@synyx.de

Database access with JPA / Spring

```
public class EntityDao {

   @PersistenceContext
   private EntityManager em;

   @PersistenceUnit
   private EntityManagerFactory emf;

   public void bar() {

      MyEntity x = em.find(1, MyEntity.class);
   }
}
```


Synyx GmbH & Co. KG
Karlstraße 68
76137 Karlsruhe

phone +49(0)721 66 24 866
info@synyx.de

Code Samples

Synyx GmbH & Co. KG
Karlstraße 68
76137 Karlsruhe

phone +49(0)721 66 24 866
info@synyx.de

Database access with JPA / Spring

- Issues
  - Generalize?
  - Paging? / Sorting?
  - Query by example?
  - Executing finder methods?
  - Auditing?

Synyx GmbH & Co. KG
Karlstraße 68
76137 Karlsruhe

phone +49(0)721 66 24 866
info@synyx.de

GenericDao

Implementing DRY and KISS

SYNYX
OpenSource Solutions

Synyx GmbH & Co. KG
Karlstraße 68
76137 Karlsruhe

phone +49(0)721 66 24 866
info@synyx.de

CRUD Methods

```
interface GenericDao
   <T extends Persistable<PK>, PK> {

   void save(T entity);

   void delete(T entity);

   T readByPrimaryKey(PK primaryKey);

   Page<T> readAll(Pageable pageable,
      Sort s);

   ...
}
```


Synyx GmbH & Co. KG
Karlstraße 68
76137 Karlsruhe

phone +49(0)721 66 24 866
info@synyx.de

GenericDao

- Based on plain JPA
  - No criteria API (**`readByExample`**)
  - Every vendor supported
- Usage
  - Declare interface for strong typing
  - Create instance with **`GenericDaoFactoryBean`**
    - AOP Proxy

Synyx GmbH & Co. KG
Karlstraße 68
76137 Karlsruhe

phone +49(0)721 66 24 866
info@synyx.de

Code Samples

Synyx GmbH & Co. KG
Karlstraße 68
76137 Karlsruhe

phone +49(0)721 66 24 866
info@synyx.de

GenericDao

- Issues
  - Generalize?
  - Paging? / Sorting?
  - Query by example?
  - Executing finder methods?
  - Auditing?

Synyx GmbH & Co. KG
Karlstraße 68
76137 Karlsruhe

phone +49(0)721 66 24 866
info@synyx.de

GenericDao

- Issues
  - ✓ Generalize
  - ✓ Paging / Sorting
  - Query by example?
  - Executing finder methods?
  - Auditing?

Synyx GmbH & Co. KG
Karlstraße 68
76137 Karlsruhe

phone +49(0)721 66 24 866
info@synyx.de

Extended Generic Dao

```
// Generics omitted ;)

ExtendedGenericDao extends GenericDao {

   List<T> readByExample(T... examples);
   ...
}
```


Synyx GmbH & Co. KG
Karlstraße 68
76137 Karlsruhe

phone +49(0)721 66 24 866
info@synyx.de

Extended Generic Dao

- Uses vendor specific API (e.g. Hibernate)
- Provides features not available with plain JPA
  - **`readByExample`**
- Usage
  - DAO interface has to extend **`ExtendedGenericDao`**
  - use provider specific **`Generic${provider}JpaDao`**

Synyx GmbH & Co. KG
Karlstraße 68
76137 Karlsruhe

phone +49(0)721 66 24 866
info@synyx.de

Code Samples

Synyx GmbH & Co. KG
Karlstraße 68
76137 Karlsruhe

phone +49(0)721 66 24 866
info@synyx.de

ExtendedGenericDao

- Issues
  - ✓ Generalize
  - ✓ Paging / Sorting
  - Query by example?
  - Executing finder methods?
  - Auditing?

Synyx GmbH & Co. KG
Karlstraße 68
76137 Karlsruhe

phone +49(0)721 66 24 866
info@synyx.de

ExtendedGenericDao

- Issues
  - ✓ Generalize
  - ✓ Paging / Sorting
  - ✓ Query by example
  - Executing finder methods?
  - Auditing?

Synyx GmbH & Co. KG
Karlstraße 68
76137 Karlsruhe

phone +49(0)721 66 24 866
info@synyx.de

Finder methods

Synyx GmbH & Co. KG
Karlstraße 68
76137 Karlsruhe

phone +49(0)721 66 24 866
info@synyx.de

Finder methods

- Methods to find entities based on certain criterias
  - **`UserDao -> findByLastname(String lastname)`**
- Usage
  - Declare typed interface with finder methods
  - Use **`GenericDaoFactoryBean`** to create instance

Synyx GmbH & Co. KG
Karlstraße 68
76137 Karlsruhe

phone +49(0)721 66 24 866
info@synyx.de

Finder methods

- How to get from the method to the query?
  - **`@NamedQuery`**
  - Extract query from method name
  - Configurable strategy
    - default: **`CREATE_IF_NOT_FOUND`**

Synyx GmbH & Co. KG
Karlstraße 68
76137 Karlsruhe

phone +49(0)721 66 24 866
info@synyx.de

Finder methods

```
UserDao extends GenericDao<User, Long> {

  List<User> findByUsername(String username);
}
```


Synyx GmbH & Co. KG
Karlstraße 68
76137 Karlsruhe

phone +49(0)721 66 24 866
info@synyx.de

Finder methods

```
UserDao extends GenericDao<User, Long> {

  List<User> findByUsername(String username);
}
```

```
from User u where u.username = ?
```


Synyx GmbH & Co. KG
Karlstraße 68
76137 Karlsruhe

phone +49(0)721 66 24 866
info@synyx.de

Finder methods

```
UserDao extends GenericDao<User, Long> {

  List<User> findByUsername(String username);
}
```

```
from User u where u.username = ?
```

```
@NamedQuery(name="User.findByUsername",
  query="from User u where u.username = ?")
```


Synyx GmbH & Co. KG
Karlstraße 68
76137 Karlsruhe

phone +49(0)721 66 24 866
info@synyx.de

GenericDaoFactoryBean

- Issues
  - ✓ Generalize
  - ✓ Paging / Sorting
  - ✓ Query by example
  - Executing finder methods?
  - Auditing?

Synyx GmbH & Co. KG
Karlstraße 68
76137 Karlsruhe

phone +49(0)721 66 24 866
info@synyx.de

GenericDaoFactoryBean

- Issues
  - ✓ Generalize
  - ✓ Paging / Sorting
  - ✓ Query by example
  - ✓ Executing finder methods
  - Auditing?

Synyx GmbH & Co. KG
Karlstraße 68
76137 Karlsruhe

phone +49(0)721 66 24 866
info@synyx.de

Spring namespace configuration

Let XML speak and hide complexity

Synyx GmbH & Co. KG
Karlstraße 68
76137 Karlsruhe

phone +49(0)721 66 24 866
info@synyx.de

Spring namespace configuration

```
<hades:dao-config
  entity-package-name="com.acme.domain"
  dao-package-name="com.acme.dao" />
```


Synyx GmbH & Co. KG
Karlstraße 68
76137 Karlsruhe

phone +49(0)721 66 24 866
info@synyx.de

Spring namespace configuration

```
<hades:dao-config
  entity-package-name="com.acme.domain"
  dao-package-name="com.acme.dao" />

// Defaults
dao-base-class = GenericJpaDao

dao-name-postfix = "Dao" (for bean name)
dao-impl-postfix = „DaoImpl" (custom impls)

finder-lookup-strategy = „create-if-not-found"
finderPrefix = „findBy"
```


Synyx GmbH & Co. KG
Karlstraße 68
76137 Karlsruhe

phone +49(0)721 66 24 866
info@synyx.de

Spring namespace configuration

- Registers **`GenericDaoFactoryBean`**s
- Registers necessary **`*PostProcessors`**
- Auto configuration possible by classpath scanning
- Convention over configuration

Synyx GmbH & Co. KG
Karlstraße 68
76137 Karlsruhe

phone +49(0)721 66 24 866
info@synyx.de

Domain objects

„Place the project's primary focus on the domain “

Eric Evans - Domain Driven Design

Synyx GmbH & Co. KG
Karlstraße 68
76137 Karlsruhe

phone +49(0)721 66 24 866
info@synyx.de

Domain objects

- Only dependency - **`Persistable<PK>`**
- Usually common base class for domain classes
  - **`AbstractPersistable<PK>`**
    - defines id property as well as **`isNew()`**
  - **`AbstractAuditable<U>`**
    - see next slides

Synyx GmbH & Co. KG
Karlstraße 68
76137 Karlsruhe

phone +49(0)721 66 24 866
info@synyx.de

Auditing

Synyx GmbH & Co. KG
Karlstraße 68
76137 Karlsruhe

phone +49(0)721 66 24 866
info@synyx.de

Auditing

- Keeping track of who changed what when
- Usage
  - Implement at least **`AbstractAuditable<U>`**
  - Enable **`AuditingAdvice`**
  - Provide auditor by implementing **`AuditorAware`**
    - optional

Synyx GmbH & Co. KG
Karlstraße 68
76137 Karlsruhe

phone +49(0)721 66 24 866
info@synyx.de

Auditing

```
interface Auditable extends Persistable {

   // Getters omitted

   setCreatedBy(T who);
   setCreatedDate(Date date);

   setLastModifiedBy(T who);
   setLastModifiedDate(Date date);
}
```


Synyx GmbH & Co. KG
Karlstraße 68
76137 Karlsruhe

phone +49(0)721 66 24 866
info@synyx.de

AuditingAdvice

```
@Aspect
public class AuditingAdvice {

   @Before(„execution(* GenericDao+.save*(..))
     && args(auditable)“)
   public void touch(Auditable auditable) {
      ...
   }
}
```


Synyx GmbH & Co. KG
Karlstraße 68
76137 Karlsruhe

phone +49(0)721 66 24 866
info@synyx.de

Meta stuff

Give me the high level picture!

Synyx GmbH & Co. KG
Karlstraße 68
76137 Karlsruhe

phone +49(0)721 66 24 866
info@synyx.de

Overview

- Implementation of CRUD Operations
- Executing finder methods
  - `@NamedQuery`
  - Dynamic query creation
- Base classes for domain objects
- Auditing
  - Setting creation and modification date and user
- Configuration via Spring namespace

Synyx GmbH & Co. KG
Karlstraße 68
76137 Karlsruhe

phone +49(0)721 66 24 866
info@synyx.de

Q & A

Thanks for your attention!

SYNYX
OpenSource Solutions

Synyx GmbH & Co. KG
Karlstraße 68
76137 Karlsruhe

phone +49(0)721 66 24 866
info@synyx.de

Backups

- Apache 2.0 licence
- Hades project home
- Don‘t repeat the DAO
- Spring & DAO - Eberhard Wolff, JavaMagazin 11/2007

Synyx GmbH & Co. KG
Karlstraße 68
76137 Karlsruhe

phone +49(0)721 66 24 866
info@synyx.de