## JPA Annotations and how to use them



Bryan Hansen

DIRECTOR OF SOFTWARE DEVELOPMENT

abh5k

## Annotaations

# Entity Annotations

- @Entity Declares Object
- @Table Table specifics
- @ld Primary Key
- @GeneratedValue Used with @Id

**IDENTITY - Column** 

**AUTO - Chooses from dialect** 

SEQUENCE - If database supports it

TABLE - Uses identity table

```
spring.jpa.generate-ddl=true
spring.jpa.hibernate.ddl-auto=create
```

spring.jpa.hibernate.naming.physical-strategy=
org.hibernate.boot.model.naming.PhysicalNamingStrategyStandardImpl

## Uppercase Names

PhysicalNamingStrategyStandardImpl

## Default Columns

#### **Override Defaults**

@Column - Override names or add info

columnDefinition scale

insertable table

length unique

name updatable

nullable

precision

How we use it

- @PersistenceContext Injects EntityManager
- @Service Location of business logic
- @Repository Database Integration
- @Transactional Beginning of Transaction

Join Types

Four join types

- @OneToOne
- @OneToMany
- @ManyToOne
- @ManyToMany

Join Types cont...

Various Configurations

Unidirectional

**Bidirectional** 

Cascade

## @OneToMany

**Most Common** 

Paired @ManyToOne

mappedBy

@OneToMany

A one-to-many relationship

Fetch Types

Lazy - DB query when property is called

Eager - DB query when object is created

Query q = em.createQuery("Select r from Registration");

JPQL

JPQL != SQL

**Centered around Objects** 

```
String jpql = "Select new
com.pluralsight.conference.model.RegistrationReport
(r.name, c.name, c.description)
from Registration r, Course c
where r.id = c.registration.id";
```

## Projection

Presents objects to the UI

Objects added using JPQL

Projection Objects can be JPA Entities

Constructor for the projection is needed

```
@NamedQueries({@NamedQuery(
   name=Regstration.FIND_REGISTRATION_REPORTS,
   query = Regstration.FIND_REGISTRATION_REPORTS_JPQL)})
```

### NamedQueries

Cleaner then adhoc JPQL

Not required, but focuses on the domain

Named parameters

## Summary

**Annotations** 

**Overriding Defaults** 

Service and Repository from Spring

Joins

FetchTypes

Projection

NamedQueries