Adding a Transactional Repository

Antonio Goncalves
JAVA CHAMPION

@agoncal www.antoniogoncalves.org

Previous Module

Domain model

Book entity

Annotations

Deployment descriptor

- persistence.xml

Mapped to a table

Overview

Understanding repositories

Being transactional

Java Transaction API (JTA)

Basic CRUD operations

Using a query language

Java Persistence API (JPA)

What Is a Repository?

Between the domain and the data layer

Domain:
Book entity

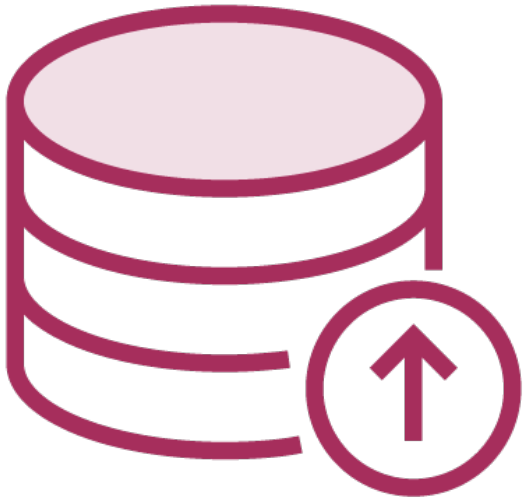
Data layer:
Relational database

Retrieve books objects from a relational database

Book Repository

```
public class BookRepository {

  public Book find(Long id)      { // ... }

  public Book create(Book book) { // ... }

  public void delete(Long id)    { // ... }
}
```

What Is the Entity Manager?



Performs database-related operations

Abstraction above JDBC

No SQL statements

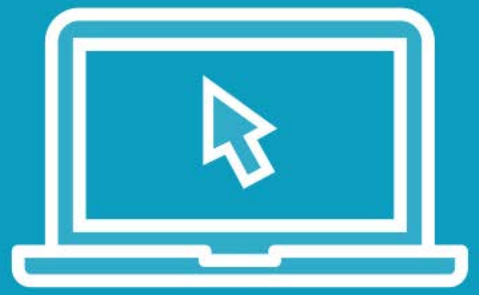
CRUD operations through methods

- (C)reate
- (R)ead
- (U)pdate
- (D)elete

Book Repository with JPA

```
public class BookRepository {

  @PersistenceContext(unitName = "bookStorePU")
  private EntityManager em;

  public Book find(Long id)     {
    return em.find(Book.class, id);
  }

  public Book create(Book book) {
    em.persist(book);
    return book;
  }

  public void delete(Long id)   {
              em.getReference(Book.class, id));
  }
```

Demo



Create a BookRepository

Implementing CRUD operations

- Find by ID
- Create
- Delete

Using the EntityManager

Querying the Book Entity

- Find by identifier
- Limiting
- Retrieve by criteria
- Inherent to relational databases
- Structured Query Language (SQL)

What Is Java Persistence Query Language?

- Searches against entities
- Independent of the underlying database
- Query language
- Entity or a collection of entities
- Syntax is object oriented
- Transforms a JPQL query into SQL

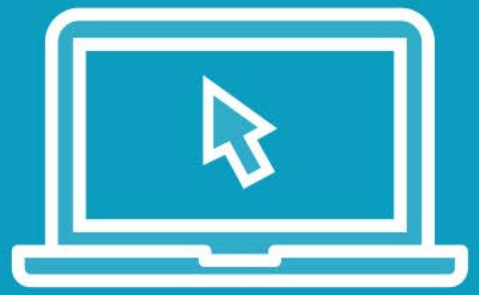
JPQL Syntax

```
SELECT    <select clause>
FROM      <from clause>
[WHERE    <where clause>]
[ORDER BY <order by clause>]
[GROUP BY <group by clause>]
[HAVING   <having clause>]

<function>    AVG, COUNT, MAX, MIN, SUM

<operators>   =, >, >=, <, <=, <>, [NOT] BETWEEN, [NOT] IN, [NOT] LIKE

<num exp.>    ABS, SQRT, MOD, SIZE, INDEX

<string exp.> CONCAT, SUBSTRING, TRIM, LOWER, UPPER, LENGTH, LOCATE

<date exp.>   CURRENT_DATE, CURRENT_TIME, CURRENT_TIMESTAMP
```

Book Repository with JPQL

```
public class BookRepository {

  @PersistenceContext(unitName = "bookStorePU")
  private EntityManager em;

  public List<Book> findAll() {
    TypedQuery<Book> query = em.createQuery(
                                        "SELECT b FROM Book b", Book.class);
    return query.getResultList();
  }

  public Long countAll() {
    TypedQuery<Long> query = em.createQuery(
            "SELECT COUNT(b) FROM Book b", Long.class);
    return query.getSingleResult();
  }
```

Demo



Implement JPQL queries

BookRepository

Select all the books ordered by title

Counting the number of books

What Is a Transaction?

Data is crucial

Data is kept in a consistent state

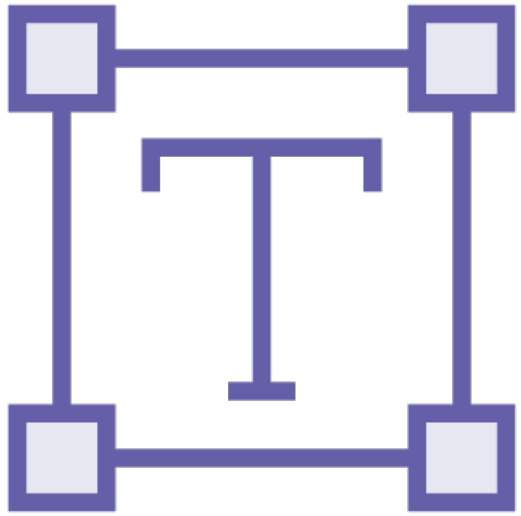
Logical group of operations

Performed as a single unit of work

Succeed or failed atomically

Java Transaction API

What Is Java Transaction API?

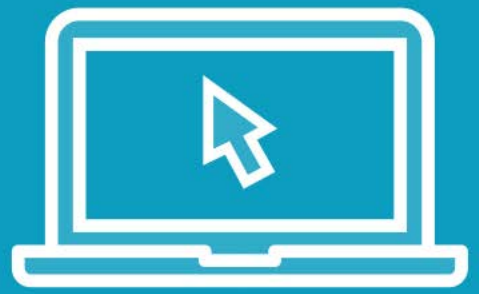


- Start, commit and rollback transactions
- Resource neutral
- Transaction demarcation
- Using annotations
- Transaction propagation

Book Repository with JTA

```
@Transactional(SUPPORTS)
public class BookRepository {

  @Transactional(SUPPORTS)
  public Book find(Long id) { ... }

  public List<Book> findAll() { ... }

  public Long countAll() { ... }

  @Transactional
  public Book create(Book book) { ... }

  @Transactional(REQUIRED)
  public void delete(Long id) { ... }
}
```

Demo



BookRepository

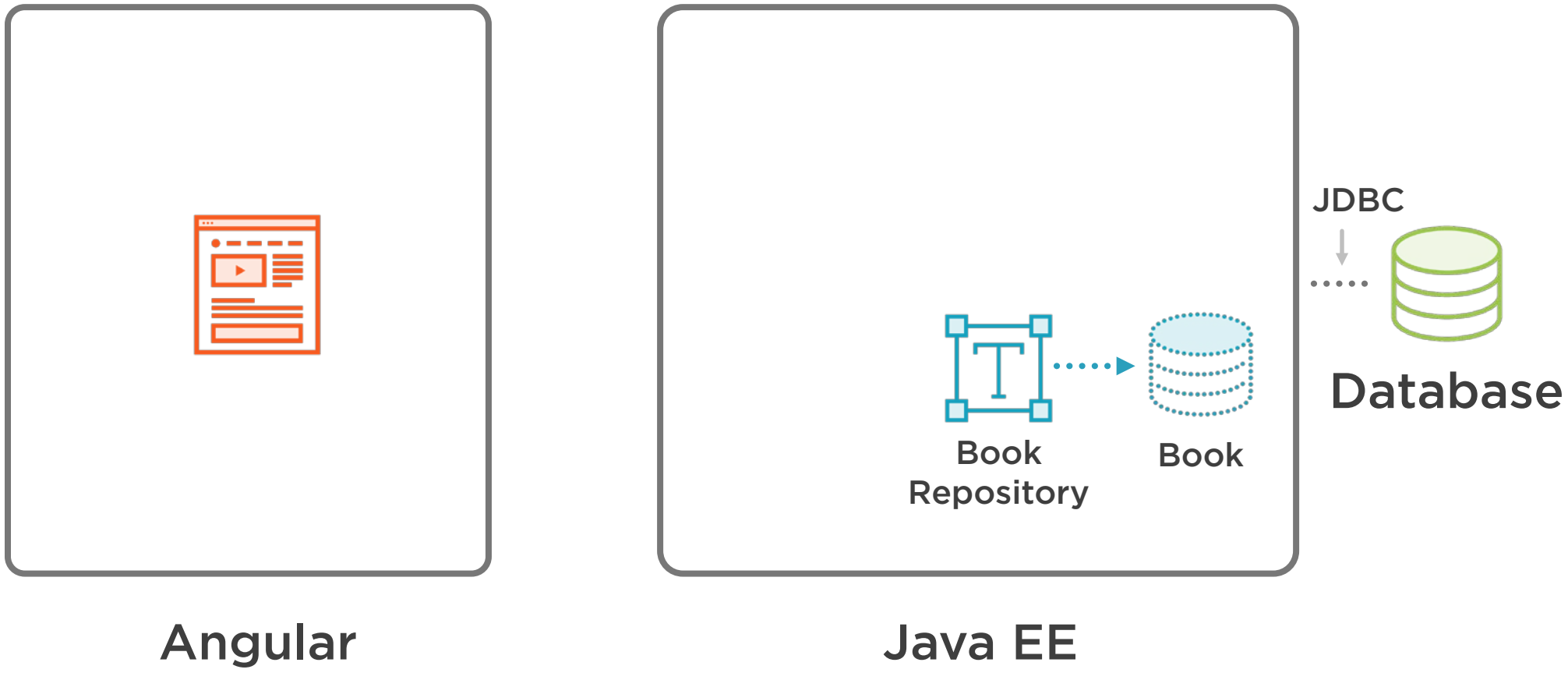
Implement transaction demarcation

@Transactional

REQUIRED always starts a new transaction

SUPPORTS inherit from caller transaction

Anatomy of the BookStore Application



Summary

- EntityManager
- CRUD operations
- In a single line of code
- Java Persistence Query Language
- Java Transaction API
- Transaction demarcation

Next Module

Test our code

Unit tests

Integration tests

Testing CRUD operations

Testing queries