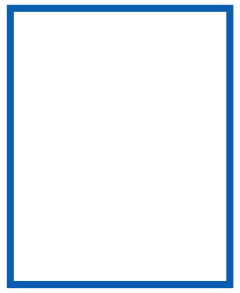
JPA 2.1 Within Java EE 7

Antonio Goncalves
www.antoniogoncalves.org
@agoncal

pluralsight
hardcore dev and IT training

Module Outline

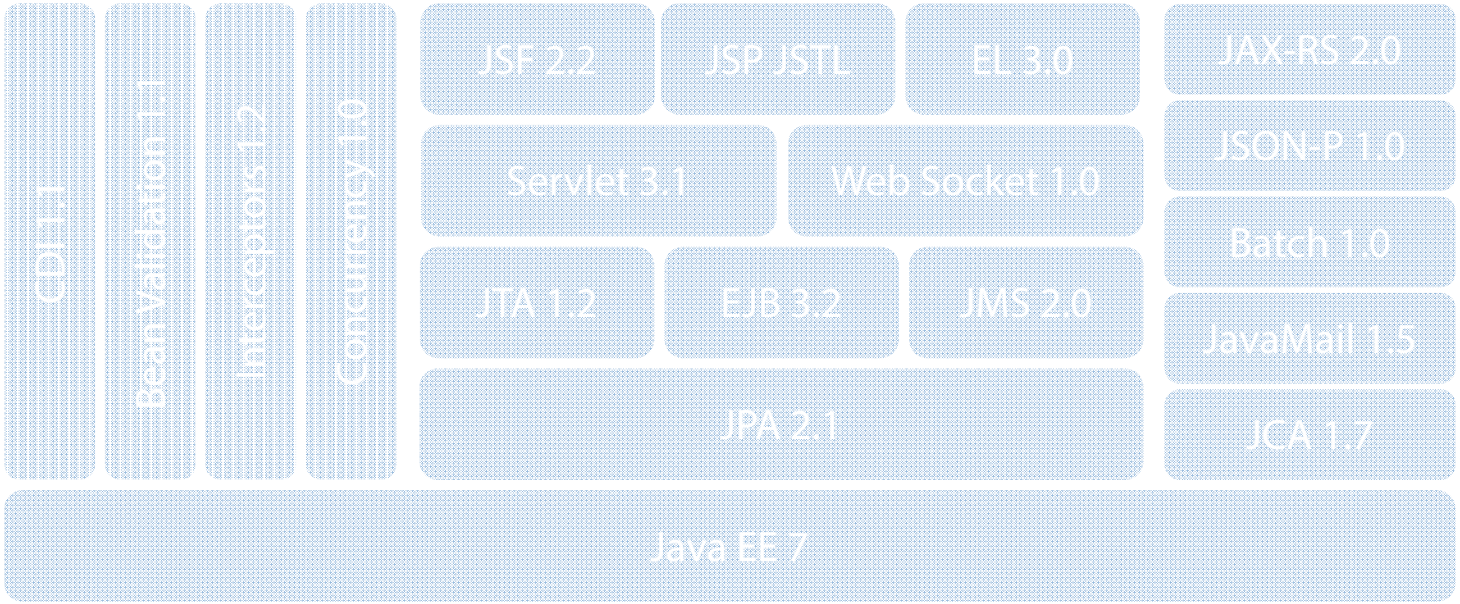
- **Java Enterprise Edition 7**
- **Services given by the Java EE container**
- **Java Persistence API integration**
  - Java EE and CDI
  - Bean Validation
  - Transactional components
  - External services (JAX-RS or JMS)
- **Summary**
- **References**



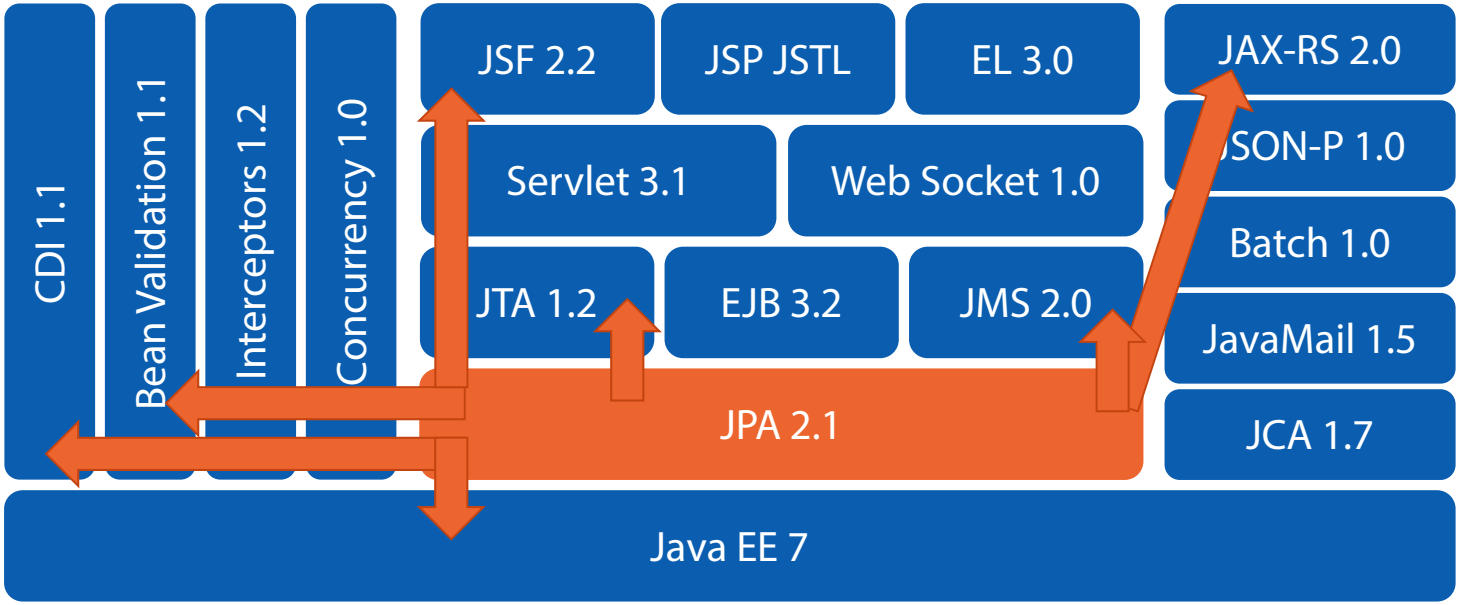
Quick Overview of Java EE 7

- **Java Enterprise Edition**
- **Umbrella specification**
  - JSR 342
  - Contains 32 specifications
  - Make them work together
- **Java EE 7 is a managed environment**
  - Container
  - Providers

Main Java EE 7 Specifications



JPA Interactions in Java EE 7



“

Demo

”

Integration With Java EE and CDI

- Java EE container controls life-cycle
- Bootstraps `EntityManager`
- Provides access to `EntityManager`
  - JNDI lookup
  - `@PersistenceContext`
  - `@Inject`
- Invokes `EntityManager.close()`

JPA in Java SE

```
public class BookService {




  public Book createBook(Book book) {
    EntityManagerFactory emf =
      Persistence.createEntityManagerFactory("myPU");
    EntityManager em = emf.createEntityManager();
    EntityTransaction tx = em.getTransaction();

    tx.begin();
    em.persist(book);
    tx.commit();
    em.close();
    emf.close();
    return book;
  }
}
```

Injection of EntityManager in Java EE

```
public class BookService {

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

  public Book createBook(Book book) {



    EntityTransaction tx = em.getTransaction();

    tx.begin();
    em.persist(book);
    tx.commit();


    return book;
  }
}
```

Injection of EntityManager in Java EE With CDI

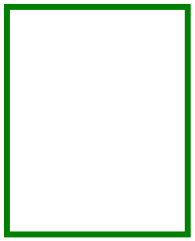
```
public class BookService {

  @Inject
  private EntityManager em;

  public Book createBook(Book book) {



    EntityTransaction tx = em.getTransaction();

    tx.begin();
    em.persist(book);
    tx.commit();


    return book;
  }
}
```

Injection of EntityManager in Java EE With CDI

```
public class BookService {

  @Inject
  private EntityManager em;

  public Book createBook(Book book) {



    EntityTransaction tx = em.getTransaction();

    tx.begin();
    em.persist(book);
    tx.commit();


    return book;
  }
}
```

Integration With Transactional Components

- **Consistent state**
- **Changes to the database succeed or fail atomically**
- **JPA `EntityTransaction`**
  - Resource local transaction
  - Explicitly invoke `commit` or `rollback`
  - Not propagated
- **JTA `UserTransaction`**
  - Resource neutral
  - Propagated
- **EntityManager cooperates with JTA transactions**



JPA EntityTransaction

```
public class BookService {

  @Inject
  private EntityManager em;

  public Book createBook(Book book) {

    EntityTransaction tx = em.getTransaction();

    tx.begin();
    em.persist(book);
    tx.commit();

    return book;
  }

}
```

JTA Transaction

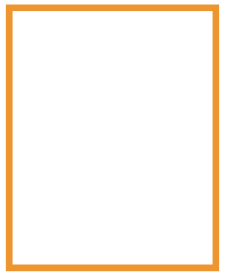
```
@Transactional
public class BookService {

  @Inject
  private EntityManager em;

  public Book createBook(Book book) {




    em.persist(book);


    return book;
  }

}
```

Starts a transaction

Commit or rollback the transaction

@Transactional

- **CDI interceptor binding**
- **Belongs to the JTA specification**
  - Java Transaction API
  - JSR 907
- **Class or method level**
- **Controls transaction boundaries**
- **On any Java EE managed bean**
  - CDI Bean
  - REST endoint
  - JSF backing bean
  - Servlet
  - EJB



Transactional CDI Bean

```
@Transactional

public class BookService {

  @Inject
  private EntityManager em;



  public Book createBook(Book book) {

    em.persist(book);

    return book;
  }

}
```

Transactional REST Endpoint

```
@Transactional
@Path("book")
public class BookService {

  @Inject
  private EntityManager em;

  @POST
  @Consumes(MediaType.APPLICATION_XML)
  public Book createBook(Book book) {

    em.persist(book);

    return book;
  }

}
```

Transactional JSF Backing Bean

```
@Transactional
@Named
public class BookService {

  @Inject
  private EntityManager em;



  public Book createBook(Book book) {

    em.persist(book);

    return book;
  }

}
```

Transactional Servlet

```
@Transactional
@WebServlet(urlPatterns={"/TxServlet"})
public class BookService extends HttpServlet {

  @Inject
  private EntityManager em;



  public Book createBook(Book book) {

    em.persist(book);

    return book;
  }

}
```

Transactional EJB

```
@Stateless
public class BookService {

  @Inject
  private EntityManager em;



  public Book createBook(Book book) {

    em.persist(book);

    return book;
  }

}
```

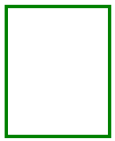
“

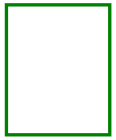
Demo

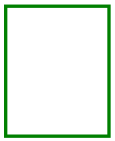
”

Bean Validation

- **Process, store, retrieve data**
- **Constrain our model**
  - Is the address valid?
  - Is the email well formed?
  - Is the customer's name null?
  - Is the birthday in the past?
- **Ensure data is valid**
- **The service will behave correctly**
- **Give feedback to the users**
- **Using annotations**







Bean Validation Constraints

```
public class Artist {

  private Long id;

  @NotNull @Size(min = 2, max = 50)
  private String firstName;

  @NotNull @Size(min = 4, max = 60)
  private String lastName;

  @Past
  private Date dateOfBirth;

}
```

Validating Constraints

```
public class ArtistService {

  @Inject
  private Validador validator;




  public void createArtist(Artist artist) {

    Set<ConstraintViolation<Artist>> violations;
    violations = validator.validate(artist);
    if (violations.size() > 0)
       throw new ConstraintViolationException(violations);
  }
}
```

Integration With Bean Validation

- Entities with Bean Validation annotations
- Validation automatically made
  - `@PrePersist`
  - `@PreUpdate`
  - `@PreRemove`
- Disable it in `persistence.xml`

Entity With Bean Validation Constraints

```
@Entity
public class Artist {

  @Id @GeneratedValue
  private Long id;

  @Column(name = "first_name", length = 50)
  @NotNull @Size(min = 2, max = 50)
  private String firstName;

  @Column(name = "last_name", length = 50)
  @NotNull @Size(min = 4, max = 60)
  private String lastName;

  @Temporal(TemporalType.DATE)
  @Past
  private Date dateOfBirth;

}
```

Validating an Entity

```
@Transactional
public class ArtistService {

  @Inject
  private Validador validator;

  @PersistenceContext
  private EntityManager em;

  public void createArtist(Artist artist) {

    Set<ConstraintViolation<Artist>> violations;
    violations = validator.validate(artist);
    if (violations.size() > 0)
       throw new ConstraintViolationException(violations);

    em.persist(artist);
  }
}
```

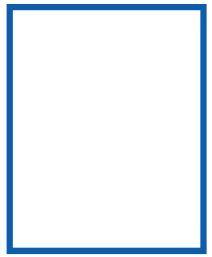
“

Demo

”

Integration With JAXB

- Java Architecture for XML Binding
- Binds Java to XML and vice versa
- Uses annotations
- `@XMLRootElement`
- `@XmlElement`, `@XmlAttribute`, `@XmlTransient`
- External services
  - SOAP Web Service
  - REST Web Service
  - JMS Messages
- Expose entities as XML



Entity With JAXB Annotations

```
@Entity
@XmlRootElement
public class Author {

  @Id @GeneratedValue
  @XmlAttribute
  private Long id;

  private String firstName;

  @Column(name = "last_name", length = 50)
  @XmlElement(name = "last-name”)
  private String lastName;

  @Transient
  @XmlTransient
  private Integer age;

}
```

Exposing an Entity as XML in REST Endpoint

```
@Transactional
@Path("book")
public class BookEndpoint {

  @Inject
  private EntityManager em;

  @GET
  @Produces(MediaType.APPLICATION_XML)
  public Book findBook(                    Long id) {
    return em.find(Book.class, id);
  }
}
```

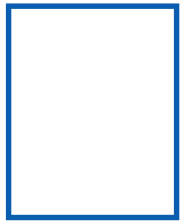
“

Demo

”

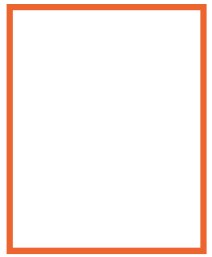
Summary

- Introduction
- Understanding Bean Validation
- Managing Elementary Entities with JPA
- Relationships and Inheritance
- Querying Entities
- Entity Lifecycle, Callbacks and Listeners
- Java Persistence API within Java EE

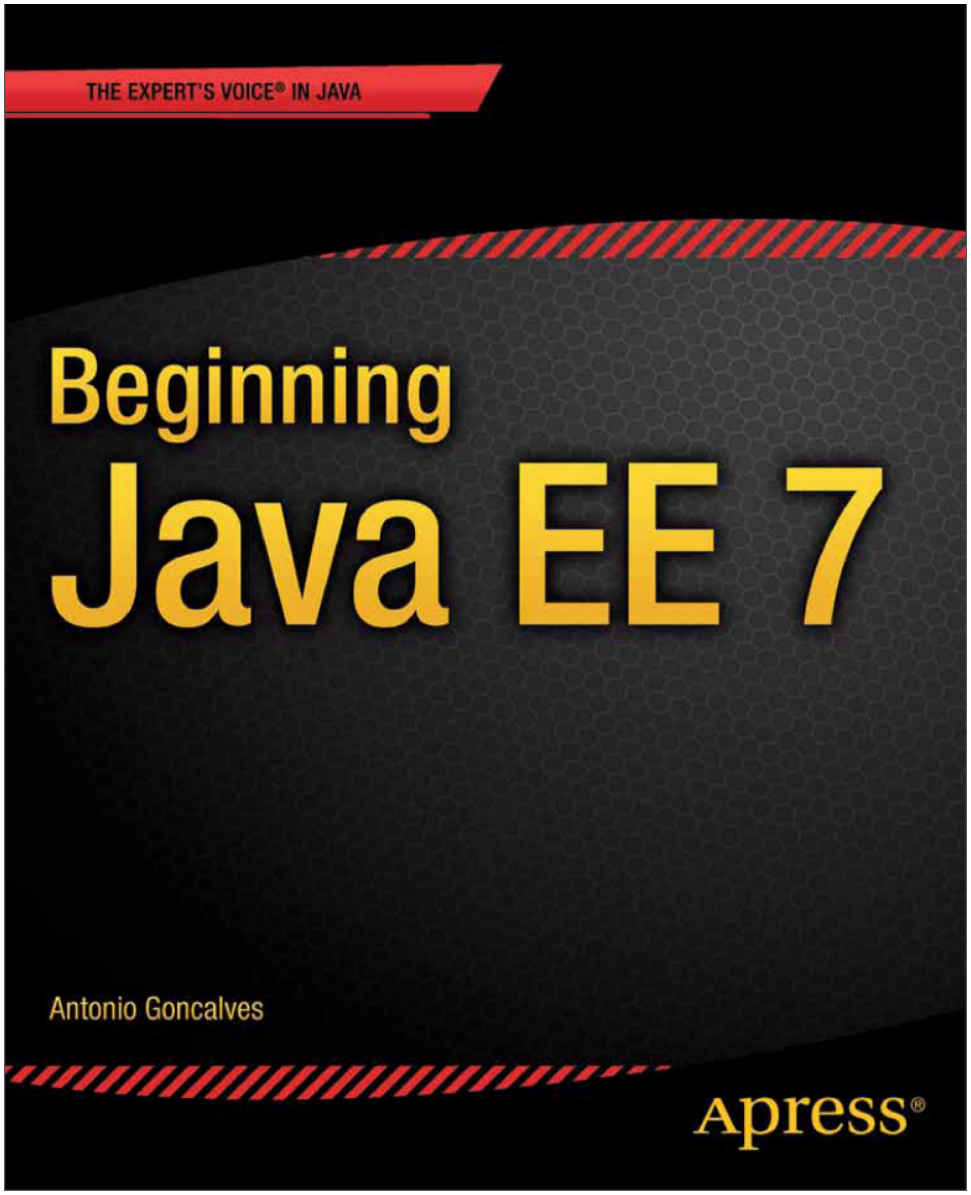


References

- **JPA 2.1 specification**
  - JSR 338
  - http://jcp.org/en/jsr/detail?id=338
- **Java EE 7 specification**
  - JSR 342
  - http://jcp.org/en/jsr/detail?id=342
- **Pluralsight courses**
  - Bean Validation



Java EE 7 Book



Java Persistence API 2.1

Map Java Objects to a Relational Database with JPA

Antonio Goncalves
www.antoniogoncalves.org
@agoncal

pluralsight
hardcore dev and IT training