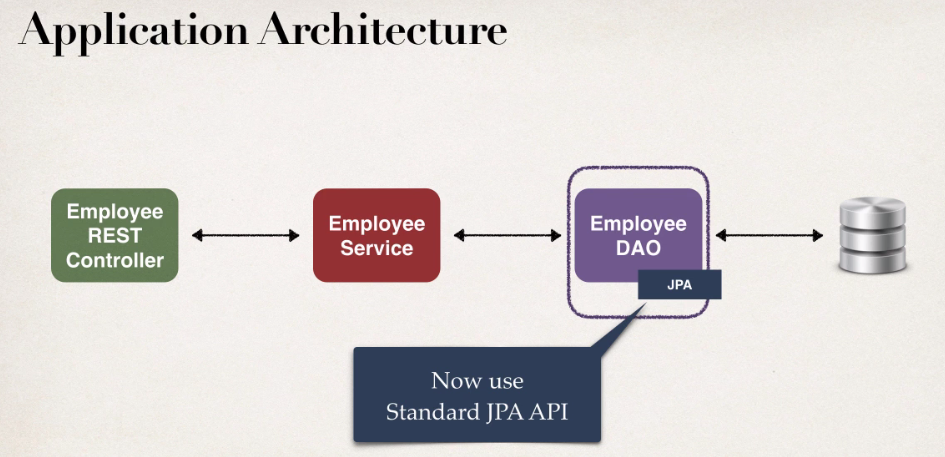
**75. Spring Boot - Build a REST CRUD API with JPA**

**Use EntityManager and standard JPA API**:



**The benefits of JPA**:

1. By using a standard API, we are not locking to vendor's implementation.
2. Maintain portable, flexible code.
3. Can switch vendor implementations.

**Standard JPA API**:

The JPA API methods are similar to Native Hibernate API.

JPA also supports a query language: JPQL (JPA Query Language)

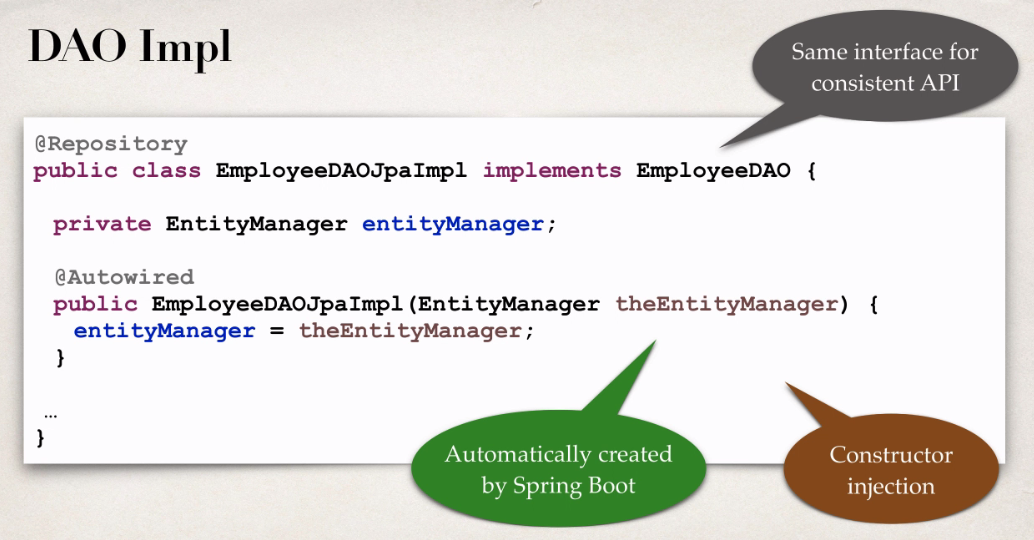
**Compare JPA to Native Hibernate Methods**:

|  |  |  |
| --- | --- | --- |
| **Action** | **Native Hibernate methods** | **JPA method** |
| Create/save new entity | session.save(...) | entityManager.persist(…) |
| Retrieve entity by id | session.get(…) / load(…) | entityManager.find(…) |
| Retrieve list of entities | session.createQuery(…) | entityManager.createQuery(…) |
| Save or update entity | session.saveOrUpdate(…) | entityManager.merge(…) |
| Delete entity | session.delete(…) | entityManager.remove(…) |

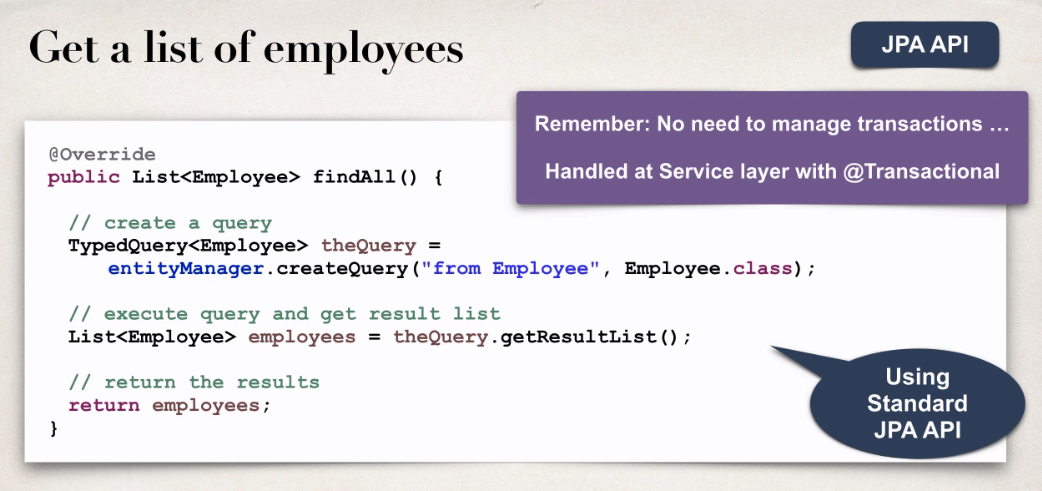
**Development Process**:

1. Set up database DEV environment.
2. Create Spring Boot project using Spring Initializer.
3. Get list of employees
4. Get single employee by ID
5. Add a new employee
6. Update an existing employee
7. Delete an existing employee

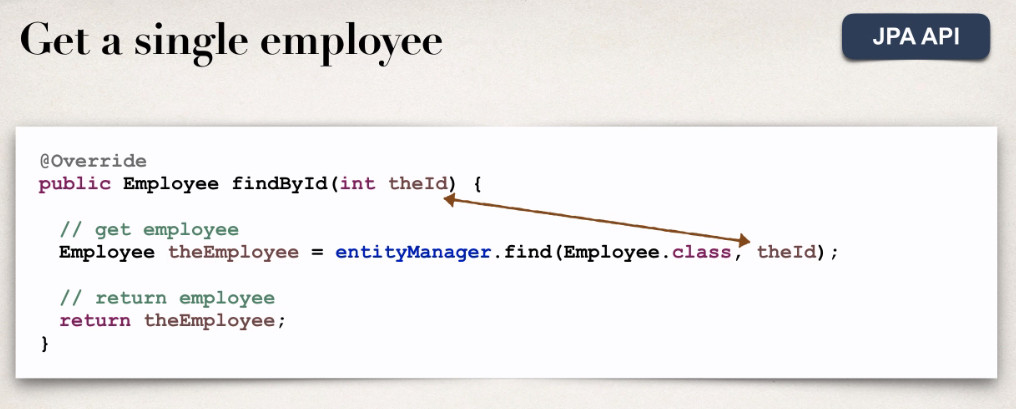
**DAO implementation**:



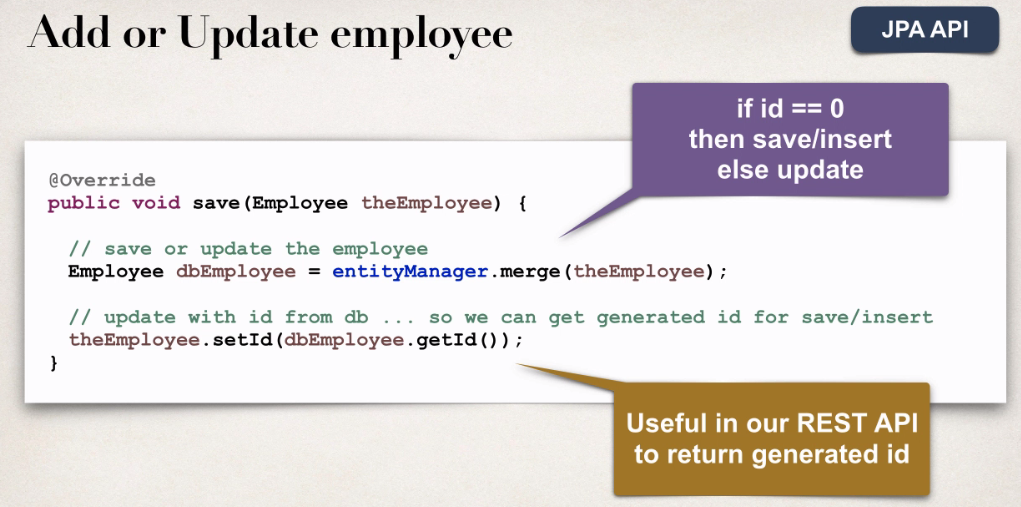
**Get a list of Employee**:



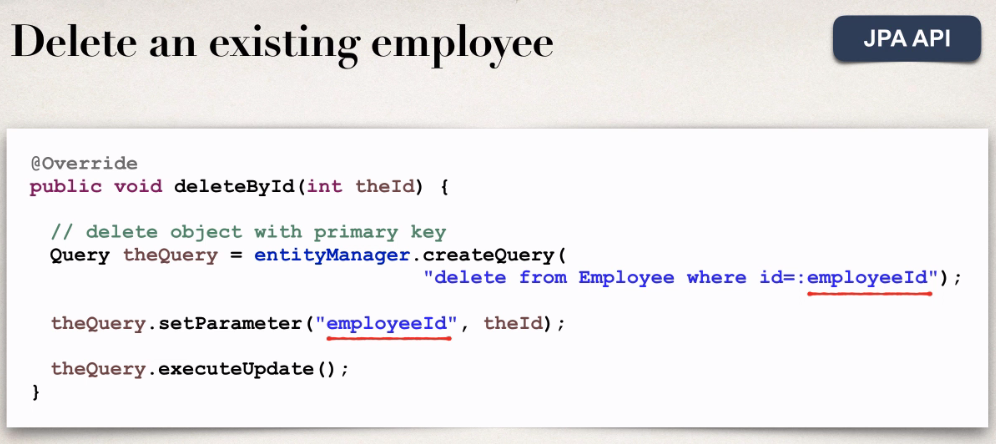
**Get a single employee**:



**Add or update employee**:



**Delete an existing employee**:



75. Spring Boot - Build a REST CRUD API with JPA