**Criteria Builder:**

Introduction to Criteria API

Building Queries with Criteria Builder

Dynamic Queries with Criteria API

Using Criteria Builder for Complex Queries

Best Practices for Criteria API

1. **Introduction to Criteria API:**

The Criteria API is part of JPA and allows you to **build type-safe queries** using Java objects.

This approach is beneficial for creating dynamic queries and handling complex query requirements programmatically.

To use the Criteria API, you need to create an instance of CriteriaBuilder and use it to construct your query.



**2. Dynamic Queries with Criteria API**

The Criteria API allows you to build dynamic queries based on various conditions.



**3. Using Criteria Builder for Complex Queries**

****

**4. Best Practices for Criteria API**

**Type Safety:** Use the Criteria API to leverage type safety and avoid runtime errors. **Reuse Predicates:** Reuse predicates to build complex queries incrementally.

**Avoid Overcomplicating:** Use the Criteria API for complex queries, but prefer JPQL for simpler queries to keep the code readable.

**Performance Considerations:** Optimize queries for performance by using appropriate indexes and avoiding unnecessary joins.

**Pagination and Sorting:**

Introduction to Pagination and Sorting

Implementing Pagination in JPA

Sorting Data with JPA

Custom Pagination and Sorting

Performance Considerations

**1.Introduction to Pagination and Sorting:**

Pagination and sorting are essential for handling large datasets efficiently. Pagination breaks down the dataset into manageable chunks, while sorting arranges the data in a specified order.

**2. Implementing Pagination in JPA**

To implement pagination, we use the **Pageable** interface provided by Spring Data JPA.

**Repository Interface**:

Extend JpaRepository and PagingAndSortingRepository:

A screenshot of a computer program

AI-generated content may be incorrect.

**Service Layer:** Implement pagination logic in the service layer:

A computer code with blue text

AI-generated content may be incorrect.

**Controller Layer:** Create an endpoint to handle pagination requests:

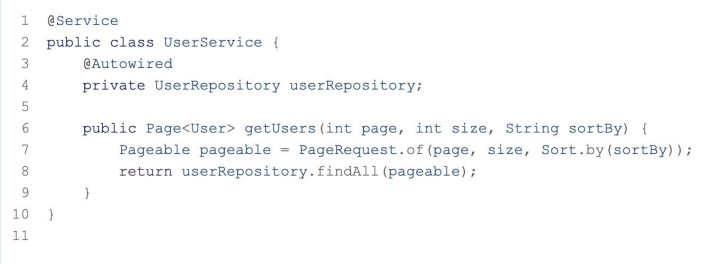
A computer code with text

AI-generated content may be incorrect.

**3. Sorting Data with JPA**

Sorting can be combined with pagination by specifying the sort criteria in the PageRequest.

**Service Layer:** Update the service layer to include sorting:



**Controller Layer:** Create an endpoint to handle pagination and sorting requests

A screenshot of a computer program

AI-generated content may be incorrect.

**4. Custom Pagination and Sorting**

**Service Layer:** Include sorting direction (ascending or descending)

A computer screen shot of a program code

AI-generated content may be incorrect.

**Controller Layer:** Create an endpoint to handle custom pagination and sorting requests

A computer code with text

AI-generated content may be incorrect.