Criteria Builder is a programmatic way of creating queries in Java. It provides a type-safe and fluent API to build dynamic queries that are easy to read and maintain. Criteria Builder is part of the Java Persistence API (JPA) and is commonly used with Hibernate ORM.

The Criteria Builder API provides a set of classes and methods to create queries programmatically. Some of the benefits of using Criteria Builder are:

1. Type safety: Criteria Builder uses Java classes and objects to create queries, which makes it type-safe and eliminates the need for writing SQL strings.
2. Compile-time checking: Since Criteria Builder queries are built using Java classes and objects, errors can be caught at compile-time rather than run-time.
3. Dynamic queries: Criteria Builder queries can be built dynamically based on runtime parameters, which makes them very flexible.

Here's an example of how to use Criteria Builder to create a query:

1. Create an instance of the CriteriaBuilder interface

```java
CriteriaBuilder cb = entityManager.getCriteriaBuilder();
```

2. Create a CriteriaQuery object and set the root entity

```python
CriteriaQuery<Employee> query = cb.createQuery(Employee.class); Root<Employee> employee = query.from(Employee.class);
```

3. Add conditions to the query using the CriteriaBuilder methods

```csharp
query.select(employee).where(cb.equal(employee.get("name"), "John Doe"));
```

4. Execute the query using the EntityManager object

```scss
List<Employee> employees = entityManager.createQuery(query).getResultList();
```

In the above example, we create a CriteriaQuery object and set the root entity to the Employee class. We then add a condition to the query using the equal method of the CriteriaBuilder interface to retrieve all employees with the name "John Doe". Finally, we execute the query using the EntityManager object and retrieve the results as a List.

I hope this helps you understand how to use Criteria Builder to create queries in Java. Let me know if you have any more questions!