# [Automatically convert a parameter with Spring Data JPA](http://stackoverflow.com/questions/22965178/automatically-convert-a-parameter-with-spring-data-jpa)

<http://stackoverflow.com/questions/22965178/automatically-convert-a-parameter-with-spring-data-jpa>

You could use a [custom type in your entity with JPA 2.1 @Convert](http://stackoverflow.com/a/24194996/1064325), so JPA makes the conversion for you (I've tested it with Spring Data JPA also, and it worked transparently!), see:

Entity:

@Entity

class SomeEntity {

@Column(name = "my\_id\_column\_name")

@Convert(converter = MyIDConverter.class)

private ID itsID;

}

Converter:

public class MyIDConverter implements AttributeConverter<ID, String> {

@Override

public String convertToDatabaseColumn(ItStaticDataKey javaKey) {

// your implementation here

}

@Override

public ItStaticDataKey convertToEntityAttribute(final String databaseKey) {

// your implementation here

}

}

Notes:

1. Make sure you're using a JPA-2.1-compatible Spring Data JPA version. (I'm sure that Spring Data JPA 1.7.0 (and above) are (or at least should) be compatible).
2. Unfortunately, [it won't work if that field should also be annotated with @Id](http://stackoverflow.com/a/28391943/1064325).