

## JPA mapping the path variable

If you need to map a path variable in a JPA repository method, you can achieve this using the `@Param` annotation along with a query method. Here's how you can do it:

Let's say you have a `MyEntity` entity with a `name` field, and you want to find entities based on a path variable `name`. Your repository interface might look like this:

```
```java
import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.data.jpa.repository.Query;
import org.springframework.data.repository.query.Param;

public interface MyEntityRepository extends JpaRepository<MyEntity, Long> {

    @Query("SELECT e FROM MyEntity e WHERE e.name = :name")
    MyEntity findByName(@Param("name") String name);
}
```
```

In this example:

- `MyEntity` is the entity class you're working with.
- `name` is the field in `MyEntity` that you want to query by.
- `findByName` is a custom query method defined in the repository interface.
- `@Query` annotation is used to specify the JPQL query.
- `:name` is the parameter placeholder in the JPQL query.
- `@Param("name")` annotation is used to map the path variable to the parameter in the query method.

Now, you can use the `findByName` method in your code to query entities based on the `name` field. For example:

```
```java
MyEntity myEntity = myEntityRepository.findByName("exampleName");
```
```

Make sure to replace `MyEntity`, `name`, and `exampleName` with your actual entity class, field name, and the value you want to query by.