



## Spring Data

Spring Data JPA - Torna fácil a implementação de repositórios JPA

Spring Data KeyValue - Repositórios baseados em Map

Spring Data LDAP - Dá suporte à repositórios LDAP

Spring Data MongoDB - Suporte à repositórios MongoDB

Spring Data REST - Exporta repositórios como recursos RESTFull

# Configuração no Projeto

build.gradle

```
buildscript {
    ext {
        springBootVersion = '2.0.3.RELEASE'
    }
    repositories {
        mavenCentral()
    }
    dependencies {
        classpath("org.springframework.boot:spring-boot-gradle-plugin:${springBootVersion}")
    }
}

apply plugin: 'java'
apply plugin: 'eclipse'
apply plugin: 'org.springframework.boot'
apply plugin: 'io.spring.dependency-management'

group = 'br.senai.sp.informatica'
version = ''
sourceCompatibility = 1.8

repositories {
    mavenCentral()
}

dependencies {
    compile('org.springframework.boot:spring-boot-starter-thymeleaf')
    compile('org.springframework.boot:spring-boot-starter-web')
    compile('org.springframework.boot:spring-boot-starter-data-jpa')
    runtime('mysql:mysql-connector-java')
    compileOnly('org.projectlombok:lombok')
    testCompile('org.springframework.boot:spring-boot-starter-test')
}
```



```
import org.springframework.data.jpa.repository.JpaRepository;

public interface CadastroRepository extends JpaRepository<Cliente, Integer> {
}
```

```
@Service
public class CadastroService {
    @Autowired
    private CadastroRepository repo;

    public Cadastro getClientes() {
        return new Cadastro(repo.findAll(), null);
    }

    public void salvar(Cliente cliente) {
        repo.save(cliente);
    }

    public boolean removeCursos(Cadastro lista) {
        Arrays.stream(lista.getIds())
            .forEach(id -> repo.deleteById(id));

        return true;
    }

    public Cliente getCliente(int id) {
        return repo.findById(id)
            .orElse(null);
    }
}
```

# Os Métodos de acesso

Chave	Exemplo	Query JPQL
And	<code>findByLastNameAndFirstName</code>	<code>where x.lastname = ?1 and x.firstname = ?2</code>
Or	<code>findByLastNameOrFirstName</code>	<code>where x.lastname = ?1 or x.firstname = ?2</code>
Between	<code>findByStartDateBetween</code>	<code>where x.startDate between ?1 and ?2</code>
LessThan	<code>findByAgeLessThan</code>	<code>where x.age &lt; ?1</code>
GreaterThan	<code>findByAgeGreaterThan</code>	<code>where x.age &gt; ?1</code>
After	<code>findByStartDateAfter</code>	<code>where x.startDate &gt; ?1</code>
Before	<code>findByStartDateBefore</code>	<code>where x.startDate &lt; ?1</code>
IsNull	<code>findByAgeIsNull</code>	<code>where x.age is null</code>
IsNotNull , NotNull	<code>findByAge ( Is ) NotNull</code>	<code>where x.age is not null</code>
Like	<code>findByFirstNameLike</code>	<code>where x.firstname like ?1</code>
NotLike	<code>findByFirstNameNotLike</code>	<code>where x.firstname not like ?1</code>
StartingWith	<code>findByFirstNameStartingWith</code>	<code>where x.firstname like ?1</code> (parameter bound with appended <code>%</code> )
EndingWith	<code>findByFirstNameEndingWith</code>	<code>where x.firstname like ?1</code> (parameter bound with prepended <code>%</code> )
Containing	<code>findByFirstNameContaining</code>	<code>where x.firstname like ?1</code> (parameter bound wrapped in <code>%</code> )
OrderBy	<code>findByAgeOrderByLastNameDesc</code>	<code>where x.age = ?1 order by x.lastname desc</code>
Not	<code>findByLastNameNot</code>	<code>where x.lastname &lt;&gt; ?1</code>
In	<code>findByAgeIn</code> (Collection<Age> ages)	<code>where x.age in ?1</code>
NotIn	<code>findByAgeNotIn</code> (Collection<Age> ages)	<code>where x.age not in ?1</code>
True	<code>findByActiveTrue</code>	<code>where x.active = true</code>
False	<code>findByActiveFalse</code>	<code>where x.active = false</code>

```
public interface CursoRepository extends JpaRepository<Curso, Integer> {  
  
    @Query("SELECT c FROM Curso c WHERE LOWER(c.nome) LIKE LOWER(:nomeCurso)")  
    public Page<Curso> findCurso(@Param("nomeCurso") String nomeCurso);  
  
}
```

<https://docs.spring.io/spring-data/jpa/docs/current/reference/html/>

## Spring Data JPA - Reference Documentation

Oliver Gierke · Thomas Darimont · Christoph Strobl · Mark Paluch · Jay Bryant – Version 2.0.9.RELEASE, 2018-07-26

© 2008-2018 The original authors.



Copies of this document may be made for your own use and for distribution to others, provided that you do not charge any fee for such copies and further provided that each copy contains this Copyright Notice, whether distributed in print or electronically.

### Table of Contents

#### *Preface*

#### 1. Project Metadata

#### 2. New & Noteworthy

##### 2.1. What's New in Spring Data JPA 1.11

##### 2.2. What's New in Spring Data JPA 1.10

#### 3. Dependencies

##### 3.1. Dependency Management with Spring Boot

##### 3.2. Spring Framework