

INTEGRANTES:

Nome: Lucas Sabino Vitachovic
RA:12522219636

Nome: Elias de Carvalho Santos
RA: 12522222372

Nome: Gustavo Arakaki
RA:12522222805

Nome: Richard Tundisi
RA:12522223359

Nome: Guilherme Guerreiro
RA:1252225522

Nome: Felipe Amorim
RA:12522223675

Nome: Marco Jesus
RA:12522232617

PROGRAMA MAIN:



MAIN

```
package br.com.cadastroFuncionarios;

import java.sql.Date;
import java.util.Arrays;
import java.util.Calendar;

import br.com.cadastroFuncionarios.repositories.CargoRepository;
import br.com.cadastroFuncionarios.repositories.DepartamentoRepository;
import br.com.cadastroFuncionarios.repositories.FuncionarioDepartamentoRepository;
import br.com.cadastroFuncionarios.repositories.FuncionarioRepository;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.boot.CommandLineRunner;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

import br.com.cadastroFuncionarios.domain.Cargo;
import br.com.cadastroFuncionarios.domain.Departamento;
import br.com.cadastroFuncionarios.domain.Funcionario;
import br.com.cadastroFuncionarios.domain.FuncionarioDepartamento;
import br.com.cadastroFuncionarios.domain.FuncionarioDepartamentoPK;

@SpringBootApplication
public class CadastroFuncionariosApplication implements CommandLineRunner{

    @Autowired
    private CargoRepository cargoRepository;

    @Autowired
    private DepartamentoRepository deptoRepository;

    @Autowired
    private FuncionarioRepository funcRepository;

    @Autowired
    private FuncionarioDepartamentoRepository funcDeptRepository;

    public static void main(String[] args) {
        SpringApplication.run(CadastroFuncionariosApplication.class, args);
    }

    @Override
    public void run(String... args) throws Exception {
        Cargo cargo1 = new Cargo(1, "Gerente");
        Cargo cargo2 = new Cargo(2, "Auxiliar");
        Cargo cargo3 = new Cargo(3, "Estagiario");

        cargoRepository.saveAll(Arrays.asList(cargo1, cargo2, cargo3));

        Departamento depto1 = new Departamento(null, "Financeiro");
    }
}
```



MAIN

```
Departamento depto1 = new Departamento(null, "Financeiro");
Departamento depto2 = new Departamento(null, "TI");
Departamento depto3 = new Departamento(null, "Juridico");
Departamento depto4 = new Departamento(null, "Contabilidade");

deptoRepository.saveAll(Arrays.asList( : depto1, : depto2, : depto3, : depto4));

Funcionario func1 = new Funcionario();
func1.setId(null);
func1.setName("TESTE");
func1.setAge(22);
func1.setBirthday(new Date( date: Calendar.getInstance().getTimeInMillis()));
func1.setDocument("11111111");
func1.setCargo(cargo1);

Funcionario func2 = new Funcionario();
func2.setId(null);
func2.setName("TESTE2");
func2.setAge(41);
func2.setBirthday(new Date( date: Calendar.getInstance().getTimeInMillis()));
func2.setDocument("22222222");
func2.setCargo(cargo2);

Funcionario func3 = new Funcionario();
func3.setId(null);
func3.setName("TESTE3");
func3.setAge(41);
func3.setBirthday(new Date( date: Calendar.getInstance().getTimeInMillis()));
func3.setDocument("33333333");
func3.setCargo(cargo3);

funcRepository.saveAll(Arrays.asList( : func1, : func2, : func3));

FuncionarioDepartamentoPK funcdeptopk1 = new FuncionarioDepartamentoPK();
funcdeptopk1.setFuncionario(func1);
funcdeptopk1.setDepartamento(depto1);
funcdeptopk1.setStartDate(new Date( date: Calendar.getInstance().getTimeInMillis()));

FuncionarioDepartamento funcdepto1 = new FuncionarioDepartamento();
funcdepto1.setId(funcdeptopk1);
```



MAIN

```
FuncionarioDepartamento funcdepto1 = new FuncionarioDepartamento();
funcdepto1.setId(funcdeptopk1);

Thread.sleep( millis: 1000 );

FuncionarioDepartamentoPK funcdeptopk2 = new FuncionarioDepartamentoPK();
funcdeptopk2.setFuncionario(func1);
funcdeptopk2.setDepartamento(depto2);
funcdeptopk2.setStartDate(new Date( date: Calendar.getInstance().getTimeInMillis() ));

FuncionarioDepartamento funcdepto2 = new FuncionarioDepartamento();
funcdepto2.setId(funcdeptopk2);

FuncionarioDepartamentoPK funcdeptopk3 = new FuncionarioDepartamentoPK();
funcdeptopk3.setFuncionario(func3);
funcdeptopk3.setDepartamento(depto1);
funcdeptopk3.setStartDate(new Date( date: Calendar.getInstance().getTimeInMillis() ));

FuncionarioDepartamento funcdepto3 = new FuncionarioDepartamento();
funcdepto3.setId(funcdeptopk3);

funcDeptoRepository.saveAll(Arrays.asList( a: funcdepto1, a: funcdepto2, a: funcdepto3 ));
```

CLASSES:

A Classe Cargo

```
package br.com.cadastroFuncionarios.domain;

import java.io.Serializable;
import java.util.ArrayList;
import java.util.List;

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.OneToMany;

import com.fasterxml.jackson.annotation.JsonBackReference;
import com.fasterxml.jackson.annotation.JsonIgnore;

@Entity
public class Cargo implements Serializable{

    /**
     *
     */
    private static final long serialVersionUID = 1L;

    @Id
    @GeneratedValue(strategy=GenerationType.IDENTITY)
    @Column(name="cargo_id")
    private Integer id;

    @Column(name="cargo_name", length=60)
    private String name;

    @JsonIgnore
    @OneToMany(mappedBy="cargo")
    private List<Funcionario> funcionarios = new ArrayList<>();

    public Cargo() {
        super();
        // TODO Auto-generated constructor stub
    }

    public Cargo(int id, String name) {
        super();
    }
}
```

A M Classe Cargo

```
public Cargo() {
    super();
    // TODO Auto-generated constructor stub
}

public Cargo(int id, String name) {
    super();
    this.id = id;
    this.name = name;
}

public Integer getId() {
    return id;
}

public void setId(Integer id) {
    this.id = id;
}

public String getName() {
    return name;
}

public void setName(String name) {
    this.name = name;
}

public List<Funcionario> getFuncionarios() {
    return funcionarios;
}

public void setFuncionarios(List<Funcionario> funcionarios) {
    this.funcionarios = funcionarios;
}
```

AM

Classe

Departamento

```
package br.com.cadastroFuncionarios.domain;

import com.fasterxml.jackson.annotation.JsonIgnore;
import java.io.Serializable;
import java.util.HashSet;
import java.util.Set;

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.OneToMany;

import com.fasterxml.jackson.annotation.JsonManagedReference;

@Entity
public class Departamento implements Serializable{

    /**
     *
     */
    private static final long serialVersionUID = 1L;

    @Id
    @GeneratedValue(strategy=GenerationType.IDENTITY)
    @Column(name="departamento_id")
    private Integer id;
    @Column(name="departamento_name", length=50)
    private String name;

    @JsonIgnore
    @OneToMany(mappedBy="id.departamento")
    private Set<FuncionarioDepartamento> funcionarios = new HashSet<>();

    public Integer getId() {
        return id;
    }
    public void setId(Integer id) {
        this.id = id;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public Set<FuncionarioDepartamento> getFuncionarios() {
        return funcionarios;
    }
}
```



Classe Departamento

```
}

public Set<FuncionarioDepartamento> getFuncionarios() {
    return funcionarios;
}

public void setFuncionarios(Set<FuncionarioDepartamento> funcionarios) {
    this.funcionarios = funcionarios;
}

public Departamento(Integer id, String name) {
    super();
    this.id = id;
    this.name = name;
}

public Departamento() {
    super();
    // TODO Auto-generated constructor stub
}
```



Classe Funcionario

```
import java.io.Serializable;
import java.util.ArrayList;
import java.util.Date;
import java.util.HashSet;
import java.util.List;
import java.util.Set;

import javax.persistence.Entity;
import javax.persistence.JoinColumn;
import javax.persistence.ManyToOne;
import javax.persistence.OneToMany;

import com.fasterxml.jackson.annotation.JsonBackReference;
import com.fasterxml.jackson.annotation.JsonManagedReference;

@Entity
public class Funcionario extends Pessoa implements Serializable {

    /**
     *
     */
    private static final long serialVersionUID = 1L;

    @JsonManagedReference
    @ManyToOne
    @JoinColumn(name="cargo_id")
    private Cargo cargo;

    @JsonBackReference
    @OneToMany(mappedBy="id.funcionario")
    private Set<FuncionarioDepartamento> departamentos = new HashSet<>();

    public Cargo getCargo() {
        return cargo;
    }

    public void setCargo(Cargo cargo) {
        this.cargo = cargo;
    }

    public List<Departamento> getDepartamentos() {
        List<Departamento> deptos = new ArrayList<>();
        for(FuncionarioDepartamento x : departamentos) {
            deptos.add(x.getDepartamento());
        }
        return deptos;
    }

    public void setDepartamentos(Set<FuncionarioDepartamento> departamentos) {
```



Classe Funcionario

```
        }
        return deptos;
    }

public void setDepartamentos(Set<FuncionarioDepartamento> departamentos) {
    this.departamentos = departamentos;
}

public Funcionario(Integer id, String name, int age, Date birthday, String document, Cargo cargo) {
    super();
    this.setId(id);
    this.setName(name);
    this.setAge(age);
    this.setBirthday(birthday);
    this.setDocument(document);
    this.cargo = cargo;
}

public Funcionario() {
    super();
    // TODO Auto-generated constructor stub
}
```



Classe Funcionario

Departamento

```
import java.io.Serializable;
import java.util.Date;

import javax.persistence.EmbeddedId;
import javax.persistence.Entity;

import com.fasterxml.jackson.annotation.JsonIgnore;

@Entity
public class FuncionarioDepartamento implements Serializable{

    /**
     *
     */
    private static final long serialVersionUID = 1L;

    @EmbeddedId
    private FuncionarioDepartamentoPK id = new FuncionarioDepartamentoPK();

    public FuncionarioDepartamento() {
        super();
        // TODO Auto-generated constructor stub
    }

    public FuncionarioDepartamento(Funcionario funcionario, Departamento departamento, Date startDate) {
        super();
        id.setFuncionario(funcionario);
        id.setDepartamento(departamento);
        id.setStartDate(startDate);
    }

    public FuncionarioDepartamentoPK getId() {
        return id;
    }

    public void setId(FuncionarioDepartamentoPK id) {
        this.id = id;
    }

    @JsonIgnore
    public Departamento getDepartamento() {
        return id.getDepartamento();
    }

    @JsonIgnore
    public Funcionario getFuncionario() {
        return id.getFuncionario();
    }
}
```

A M Classe Funcionario Departamento PK

```
import java.io.Serializable;
import java.util.Date;

import javax.persistence.Embeddable;
import javax.persistence.JoinColumn;
import javax.persistence.ManyToOne;

import com.fasterxml.jackson.annotation.JsonBackReference;
import com.fasterxml.jackson.annotation.JsonManagedReference;

@Embeddable
public class FuncionarioDepartamentoPK implements Serializable {

    /**
     *
     */
    private static final long serialVersionUID = 1L;

    @ManyToOne
    @JoinColumn(name="funcionario_id")
    @JsonManagedReference
    private Funcionario funcionario;

    @ManyToOne
    @JoinColumn(name="departamento_id")
    @JsonBackReference
    private Departamento departamento;

    private Date startDate;

    public Funcionario getFuncionario() {
        return funcionario;
    }

    public void setFuncionario(Funcionario funcionario) {
        this.funcionario = funcionario;
    }

    public Departamento getDepartamento() {
        return departamento;
    }

    public void setDepartamento(Departamento departamento) {
        this.departamento = departamento;
    }

    public Date getStartDate() {
        return startDate;
    }

    public void setStartDate(Date startDate) {
        this.startDate = startDate;
    }
}
```

A Classe Pessoa

```
import javax.persistence.*;
import java.util.Date;

@Entity
public class Pessoa {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Column(name="funcionario_id")
    private Integer id;
    @Column(name="funcionario_name", length=50)
    private String name;
    @Column(name="funcionario_age")
    private int age;
    @Column(name="funcionario_birthday")
    private Date birthday;
    @Column(name="funcionario_document", length=50)
    private String document;

    public Integer getId() {
        return id;
    }
    public void setId(Integer id) {
        this.id = id;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public int getAge() {
        return age;
    }
    public void setAge(int age) {
        this.age = age;
    }
    public Date getBirthday() {
        return birthday;
    }
    public void setBirthday(Date birthday) {
        this.birthday = birthday;
    }
    public String getDocument() {
        return document;
    }
    public void setDocument(String document) {
        this.document = document;
    }
}
```

CLASSE EXCEPTION

:

A Classe Error

```
package br.com.cadastroFuncionarios.resources.exception;

import java.io.Serializable;

public class Error implements Serializable{

    /**
     *
     */
    private static final long serialVersionUID = 1L;

    private Integer status;
    private String message;
    private Long timestamp;

    public Integer getStatus() {
        return status;
    }

    public void setStatus(Integer status) {
        this.status = status;
    }

    public String getMessage() {
        return message;
    }

    public void setMessage(String message) {
        this.message = message;
    }

    public Long getTimestamp() {
        return timestamp;
    }

    public void setTimestamp(Long timestamp) {
        this.timestamp = timestamp;
    }

    public Error(Integer status, String message, Long timestamp) {
        super();
        this.status = status;
        this.message = message;
        this.timestamp = timestamp;
    }

}
```

A
M

Classe ResourceException Handler

```
package br.com.cadastroFuncionarios.resources.exception;

import javax.servlet.http.HttpServletRequest;

import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.ControllerAdvice;
import org.springframework.web.bind.annotation.ExceptionHandler;

@ControllerAdvice
public class ResourceExceptionHandler {

    @ExceptionHandler(RuntimeException.class)
    public ResponseEntity<Error> raiseError(RuntimeException e, HttpServletRequest req) {
        Error err = new Error(HttpStatus.NOT_FOUND.value(), e.getMessage(), System.currentTimeMillis());
        return ResponseEntity.status(HttpStatus.NOT_FOUND).body(err);
    }
}
```

INTERFACE DE BANCO DE DADOS:



Cargo Repository

```
package br.com.cadastroFuncionarios.repositories;

import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;

import br.com.cadastroFuncionarios.domain.Cargo;

@Repository
public interface CargoRepository extends JpaRepository<Cargo, Integer>{}
```

WEBSITE BANCO DE DADOS:

localhost:8080/swagger-ui/index.html#/

Space Genesis price... Space Energy price... Bombcrypto Bomber Coin price... Market Luna Rush Ameaça de morte c... Ulife Aluno UAM TECH Week Experi...

 **Swagger** powered by SMARTBEAR

Select a definition default

Api Documentation 1.0

[Base URL: localhost:8080/]
<http://localhost:8080/v2/api-docs>

Api Documentation
Terms of service
Apache 2.0

basic-error-controller Basic Error Controller >

cargo-resource Cargo Resource >

departamento-resource Departamento Resource >

funcionario-resource Funcionario Resource >

Models >

para pesquisar     

28°C Nublado      

8080/swagger-ui/index.html#/

Space Genesis price... Space Energy price... Bombcrypto Bomber Coin price... Market Luna Rush Ameaça de morte c... Ulife Aluno UAM TECH Week Experi...

basic-error-controller Basic Error Controller

cargo-resource Cargo Resource

GET /cargos findAll

POST /cargos insert

GET /cargos/{id} find

PUT /cargos/{id} update

DELETE /cargos/{id} update

departamento-resource Departamento Resource

GET /departamentos findAll

POST /departamentos insert

GET /departamentos/{id} find

PUT /departamentos/{id} update

DELETE /departamentos/{id} update

funcionario-resource Funcionario Resource

GET /funcionarios findAll

POST /funcionarios insert

pesquisar

28°C Nublado

localhost:8080/swagger-ui/index.html#/

Space Genesis price... Space Energy price... Bombcrypto Bomber Coin price... Market Luna Rush Ameaça de morte c... Ulife Aluno UAM TECH Week Experienc...

GET /cargos/{id} update

PUT /cargos/{id} update

DELETE /cargos/{id} update

departamento-resource Departamento Resource

GET /departamentos findAll

POST /departamentos insert

GET /departamentos/{id} find

PUT /departamentos/{id} update

DELETE /departamentos/{id} update

funcionario-resource Funcionario Resource

GET /funcionarios findAll

POST /funcionarios insert

GET /funcionarios/{id} find

PUT /funcionarios/{id} update

DELETE /funcionarios/{id} update

Models >

para pesquisar

Cloud 28°C Nublado