

spring boot

<http://localhost:8080/>

C:\Program Files\Apache Software Foundation\Tomcat 8.5\webapps

=====  
=====

JSTL

<https://www.codejava.net/java-ee/jstl/jstl-format-tag-formatdate>

=====  
=====

TROCAR O JAVA JRE

botão direito no projeto > Build path > configure build path.

dai você vera uma aba library , nessa abra vc vai selecionar

a versão do jre e clicar em editar no canto direito da janela .

dai você escolhe a versão

=====  
=====

INSTALAR O PACOTE DE DESCKTOP NO ECLIPSE

clicar no menu Help -> Eclipse Marketplace

search -- find: WindowBuilder

instalar

<http://download.eclipse.org/windowbuilder/WB/integration/4.8/>

=====

\*\*\*\*\*

## PUBLICAR PROJETOS JAVA

\*\*\*\*\*

### Gerando o JAR ou WAR pelo Eclipse

#### JAR

- \* - Clique com o botão direito em cima do nome do seu projeto e selecione a opção Export.
- \* - Na tela Export (como mostra a figura abaixo), selecione a opção "JAR file" e aperte o botão "Next".
- \* - Na opção "JAR file:", selecione o local que você deseja salvar o arquivo JAR.

#### WAR

botão direito do mouse sobre o projeto clicar na opção de exportar escolher Web -> WAR file

Next

Web project ja vem preenchido

Destination - escolher o caminho onde vai gerar o war

Na aba Servers vai ter um tomcat clique duplo nele

abrindo assim a aba tomcat v...

=====

## SERVIDORES

### APACHE TOMCAT WEB

<https://tomcat.apache.org/download-90.cgi>

download e instalar

na pesquisa do windows

abrir o painel de Serviços (services)

Apache

se estiver parado entao deve iniciar novamente

abrir a pasta

C:\Program Files\Apache Software Foundation\Tomcat 9.0\bin

Tomcat9w.exe - botao direito e clicar em abrir

colocar pra abrir automaticamente...

copiar um atalho pra desktop

Fazer um teste

http://localhost:8080

Criar Testes

C:\Program Files\Apache Software Foundation\Tomcat 9.0\webapps

talvez precise dar permissao

botao direito na pasta Tomcat -> propriedade

aba Segurança

clicar no nome do grupo e depois dar a permissao de modificar

ler executar etc...

Criar uma pasta exemplo app1

criar um arquivo Index.jsp

```
<%= "Hello World!" %>
```

http://localhost:8080/app1/Index.jsp

=====

=====

TOMCAT - ESTRUTURA PUBLICAÇÃO

alterar login e senha do Apache Tomcat

C:\Program Files\Apache Software Foundation\Tomcat 9.0\conf\tomcat-users.xml

```
<?xml version="1.0" encoding="utf-8"?>
<tomcat-users>
<role rolename="tomcat"/>
<role rolename="role1"/>
<role rolename="admin"/>
<role rolename="manager"/>
<role rolename="manager-gui"/>
<user username="USUARIO" password="SENHA" roles="manager-
script,tomcat,role1,admin,manager,manager-gui"/>
</tomcat-users>
```

abrir o administrador do Apache  
tem botoes de administrador onde pode fazer upload  
escolher manage app  
ele automaticamente vai pegar os war e vai gerar  
o arquivos desta forma exibindo os dados

\*\*\*\*\*

\*\*\*\*\*

exemplo de

web.xml

localizar

C:\Program Files\Apache Software Foundation\Tomcat 8.5\webapps\Exemplojsp

aqui fica o index.jsp

C:\Program Files\Apache Software Foundation\Tomcat 8.5\webapps\Exemplojsp\WEB-INF

aqui fica o web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"
```

```
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
    http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd"
version="3.1"
metadata-complete="true">
```

```
<display-name>Exemplo 01 html</display-name>
```

```
<servlet>
```

```
    <servlet-name>WelcomeServlet</servlet-name>
```

```
    <servlet-class>pct.WelcomeServlet</servlet-class>
```

```
    <load-on-startup>5</load-on-startup>
```

```
</servlet>
```

```
<servlet-mapping>
```

```
    <servlet-name>WelcomeServlet</servlet-name>
```

```
    <url-pattern>/paginaServlet</url-pattern>
```

```
</servlet-mapping>
```

```
<session-config>
```

```
    <session-timeout>30</session-timeout>
```

```
</session-config>
```

```
<welcome-file-list>
```

```
    <welcome-file>index.html</welcome-file>
```

```
    <welcome-file>index.htm</welcome-file>
```

```
    <welcome-file>index.xhtml</welcome-file>
```

```
    <welcome-file>index.jsp</welcome-file>
```

```
    <welcome-file>default.html</welcome-file>
```

```
    <welcome-file>default.htm</welcome-file>
```

```
    <welcome-file>default.jsp</welcome-file>
```

```
</welcome-file-list>
```

</web-app>

\*\*\*\*\*

\*\*\*\*\*

=====  
=====  
=====  
=====

SPRING TOOL SUIT -- SPRING BOOT

=====  
SPRING BOOT COM JSP  
=====

<https://www.codejava.net/java-ee/jstl/introduction-to-jstl>

<https://start.spring.io/>

new -> Spring Boot -> Spring Starter Project

Maven

war

... preencher os outros

next

escolher somente Spring Web

Finish

\*\*\*\*\*

CRIAR OS PACOTES

observacao sempre usar o primeiro pacote como referencia

olha a sequencia abaixo

com.api.springboot06

com.api.springboot06.controller

com.api.springboot06.email

com.api.springboot06.modal

com.api.springboot06.repository

com.api.springboot06.service

\*\*\*\*\*

pom.xml

```
<dependency>
  <groupId>org.apache.tomcat</groupId>
  <artifactId>tomcat-jasper</artifactId>
  <version>9.0.24</version>
</dependency>
```

a versao é de acordo com o que esta na biblioteca

Maven Dependencies

procure o arquivo

tomcat-embed-core-....

no browser procurar

mvnrepository

<https://mvnrepository.com/>

procurar - Tomcat Jasper

clicar na opção Tomcat Jasper

<https://mvnrepository.com/artifact/org.apache.tomcat/tomcat-jasper>

pegar a versao que foi visto no Maven Dependencies

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
  https://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>2.1.8.RELEASE</version>
    <relativePath/> <!-- lookup parent from repository -->
  </parent>
  <groupId>com.api</groupId>
  <artifactId>springboot06</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <packaging>war</packaging>
  <name>springboot06</name>
  <description>Exemplo sistema rest 06</description>
  <properties>
    <java.version>1.8</java.version>
  </properties>
  <dependencies>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-web</artifactId>
    </dependency>
  </dependencies>
</project>
```



```
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-tomcat</artifactId>
    <scope>provided</scope>
</dependency>
```

```
<dependency>
    <groupId>org.apache.tomcat</groupId>
    <artifactId>tomcat-jasper</artifactId>
    <version>9.0.24</version>
</dependency>
```

```
<dependency>
    <groupId>org.hibernate</groupId>
    <artifactId>hibernate-entitymanager</artifactId>
</dependency>
```

```
<!-- JSTL for JSP -->
<dependency>
    <groupId>javax.servlet</groupId>
    <artifactId>jstl</artifactId>
</dependency>
```

```
<!-- Need this to compile JSP -->
<dependency>
    <groupId>org.apache.tomcat.embed</groupId>
    <artifactId>tomcat-embed-jasper</artifactId>
    <scope>provided</scope>
</dependency>
```

```
<!-- Need this to compile JSP, -->
```

```
<!-- JpaRepository -->
```

```
<dependency>
```

```
<groupId>org.springframework.boot</groupId>
```

```
<artifactId>spring-boot-starter-data-rest</artifactId>
```

```
</dependency>
```

```
<dependency>
```

```
<groupId>org.springframework.boot</groupId>
```

```
<artifactId>spring-boot-starter-data-jpa</artifactId>
```

```
</dependency>
```

```
<dependency>
```

```
<groupId>com.h2database</groupId>
```

```
<artifactId>h2</artifactId>
```

```
</dependency>
```

```
<!-- fim JpaRepository -->
```

```
<!-- MYSQL -->
```

```
<dependency>
```

```
<groupId>mysql</groupId>
```

```
<artifactId>mysql-connector-java</artifactId>
```

```
<scope>compile</scope>
```

```
</dependency>
```

```
<!-- FIM MYSQL -->
```

```
<dependency>
```

```
<groupId>com.sun.mail</groupId>
```

```
<artifactId>javax.mail</artifactId>
```

```
</dependency>
```

```
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-test</artifactId>
            <scope>test</scope>
        </dependency>
    </dependencies>

    <build>
        <plugins>
            <plugin>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-maven-plugin</artifactId>
            </plugin>
        </plugins>
    </build>

</project>
```

\*\*\*\*\*

APPLICATION.PROPERTIES

#spring.resources.static-locations=classpath:/custom/

#PORTA ONDE VAI SER EXECUTADO O NOSSO TOMCAT

server.port = 8082

#INFORMAÇÕES PARA CONEXÃO COM O BANCO DE DADOS MYSQL

spring.datasource.url=jdbc:mysql://179.188.16.6:3306/sindsmart2

spring.datasource.username=sindsmart2

spring.datasource.password=odavi240175

spring.datasource.driver-class-name=com.mysql.jdbc.Driver

```
#PARA CRIAR A TABELA NO BANCO BASTA COLOCAR update cria as tabelas

#create

spring.jpa.hibernate.ddl-auto=update

# LazyInitializationException = spring.jpa.open-in-view=false

spring.jpa.open-in-view=false

spring.jpa.generate-ddl=true

spring.jpa.show-sql=true

spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL5Dialect

spring.jpa.properties.hibernate.format_sql=true
```

```
#CAMINHO DOS ARQUIVOS DE EXIBIÇÕES jsp
```

```
#spring.mvc.view.prefix=/WEB-INF/jsp/
```

```
#spring.mvc.view.suffix=.jsp
```

```
#=====
```

```
# = Thymeleaf configurations
```

```
#=====
```

```
#PARA USAR O JSP TEM DE MANTER ESTAS LINHAS ABAIXO COMENTADAS
```

```
#spring.thymeleaf.check-template-location=true
```

```
#spring.thymeleaf.prefix=classpath:/templates/
```

```
#spring.thymeleaf.suffix=.html
```

```
#spring.thymeleaf.content-type=text/html
```

```
#spring.thymeleaf.cache=false
```

```
#server.contextPath=/
```

```
#spring.thymeleaf.check-template-location=true
```

```
#spring.thymeleaf.prefix=classpath:/templates/
```

```
#spring.thymeleaf.suffix=.html
```

```
#spring.thymeleaf.mode=LEGACYHTML5
```

```
#spring.thymeleaf.encoding=UTF-8
```

```
#spring.thymeleaf.content-type=text/html
```

```
#spring.thymeleaf.cache=false
```

```
#=====
```

```
#For detailed logging during development
```

```
logging.level.org.springframework=TRACE
```

```
logging.level.com=TRACE
```

```
*****
```

```
SPRINGBOOT06APPLICATION.JAVA
```

```
package com.api.springboot06;
```

```
import org.springframework.boot.SpringApplication;
```

```
import org.springframework.boot.autoconfigure.EnableAutoConfiguration;
```

```
import org.springframework.context.annotation.ComponentScan;
```

```
import org.springframework.context.annotation.Configuration;
```

```
// @SpringBootApplication
```

```
@Configuration
```

```
@ComponentScan(basePackages = {"com.api.springboot06.controller"})
```

```
//@ComponentScan ({ "com.api.controller", "com.api.service" })
```

```
//@ComponentScan(basePackageClasses={TBJEmpresa.class})
```

```
//@EnableJpaRepositories(basePackages = "com.api.repository")
```

```
//@ComponentScan ({"com.api.springboot06", "com.api.controller",  
"com.api.service", "com.api.repository" })
```

```
@EnableAutoConfiguration
```

```
public class Springboot06Application {
```

```
public static void main(String[] args) {  
    SpringApplication.run(Springboot06Application.class, args);  
}  
  
}
```

\*\*\*\*\*

na pasta static criar pasta  
css, js, imagens, download, documentos  
criar os arquivos

layout.css

```
.hello-title{  
    color: darkgreen;  
}
```

indexjs.js

```
//load pagina  
/*  
window.addEventListener("load", function(){  
    alert("teste java script");  
});
```

```
(function(){  
    console.log("Hello World!");
```

```
})();
```

```
*/
```

```
window.addEventListener("load", function(){
```

```
    //...
```

```
});
```

```
*****
```

```
*****
```

NO CAMINHO SRC -> MAN -> WEBAPP

deve criar as paginas jsp

```
*****
```

```
*****
```

INDEX.JSP

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
```

```
    pageEncoding="ISO-8859-1"%>
```

```
<!DOCTYPE html>
```

```
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
```

```
<html >
```

```
<head>
```

```
<meta charset="ISO-8859-1">
```

```
<title>Projeto Base teste</title>
```

```
<link href="/css/layout.css" rel="stylesheet">
```

```
<link rel="stylesheet" href="/css/bootstrap.min.css">
```

```
<script src="/js/jquery.mim.js"></script>
```

```
<script src="/js/bootstrap.min.js"></script>
```

```
<style>
```

```
/* Set height of the grid so .sidenav can be 100% (adjust if needed) */
```

```
.row.content {height: 1500px}
```

```
/* Set gray background color and 100% height */
```

```
.sidenav {
```

```
background-color: #f1f1f1;
```

```
height: 100%;
```

```
}
```

```
/* Set black background color, white text and some padding */
```

```
footer {
```

```
background-color: #555;
```

```
color: white;
```

```
padding: 15px;
```

```
}
```

```
/* On small screens, set height to 'auto' for sidenav and grid */
```

```
@media screen and (max-width: 767px) {
```

```
.sidenav {
```

```
height: auto;
```

```
padding: 15px;
```

```
}
```

```
.row.content {height: auto;}
```

```
}
```

```
.responsive {
```

```
width: 100%;
```

```
height: 100%;
```



```
    min-height: 300px;
    overflow-y: hidden;
    overflow-x: visible;
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<div class="container-fluid">
```

```
<div class="row content">
```

```
<div style="height:100px;" >
```

```
<label>${sExibirTexto} </label>
```

```
<jsp:include page="menu_incluir.jsp" />
```

```
</div>
```

```
<div class="col-sm-3 sidenav">
```

```
<h4>Vanderlei - Blog</h4>
```

```
<ul class="nav nav-pills nav-stacked">
```

```
<li class="active"><a href="${getURLControle}/home" target="iframeprinc"
>Home</a></li>
```

```
<li><a href="${getURLControle}/TBJEmpresaGrid" target="iframeprinc">Empresa</a></li>
```

```
<li><a href="#section3" target="iframeprinc" >Filial</a></li>
```

```
<li><a href="#section3" target="iframeprinc">Usuario</a></li>
```

```
</ul><br>
```

```
<div class="input-group">
```

```
  <input type="text" class="form-control" placeholder="Search Blog..">
```

```
  <span class="input-group-btn">
```

```
    <button class="btn btn-default" type="button">
```

```
      <span class="glyphicon glyphicon-search"></span>
```

```
    </button>
```

```
  </span>
```

```
</div>
```

```
</div>
```

```
<div class="col-sm-9">
```

```
  <iframe id="iframeprinc" name="iframeprinc" class='embed-responsive-item'  
  src="{getURLControle}/home.jsp" frameborder="0" width="100%" height="1200"></iframe>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
<footer class="container-fluid">
```

```
  <p>Footer Text</p>
```

```
</footer>
```

```
<c:import url="menu_incluir.jsp"></c:import>
```

```
  <h2 style="display:none;" class="hello-title">Hello ${name}!</h2>
```

```
  <script src="/js/indexjs.js"></script>
```

</body>

</html>

\*\*\*\*\*

HOME.JSP

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
```

```
    pageEncoding="ISO-8859-1"%>
```

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<meta charset="ISO-8859-1">
```

```
<title>Insert title here</title>
```

```
<link rel="stylesheet" href="/css/bootstrap.min.css">
```

```
<script src="/js/jquery.mim.js"></script>
```

```
<script src="/js/bootstrap.min.js"></script>
```

```
<style>
```

```
    /* Set height of the grid so .sidenav can be 100% (adjust if needed) */
```

```
    .row.content {height: 1500px}
```

```
    /* Set gray background color and 100% height */
```

```
    .sidenav {
```

```
        background-color: #f1f1f1;
```

```
        height: 100%;
```

```
    }
```

```
    /* Set black background color, white text and some padding */
```

```
    footer {
```

```
        background-color: #555;
```

```
color: white;
padding: 15px;
}

/* On small screens, set height to 'auto' for sidenav and grid */
@media screen and (max-width: 767px) {
  .sidenav {
    height: auto;
    padding: 15px;
  }
  .row.content {height: auto;}
}
</style>
<script>
    $(document).ready(function() {
        alert("teste java script");
    });

</script>
</head>
<body>
    <div class="context" >
        <h2 class="hello-title">Nome: ${namevand}</h2>
        
    </div>
</body>
</html>
```

\*\*\*\*\*

TBJEMPRESAGRID.JSP

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<%@ page import="com.api.springboot06.Comandosuteis,java.text.SimpleDateFormat" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<%@ taglib uri="http://java.sun.com/jstl/fmt_rt" prefix="fmt"%>
```

```
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
<link rel="stylesheet" href="/css/bootstrap.min.css">
<script src="/js/jquery.mim.js"></script>
<script src="/js/bootstrap.min.js"></script>
<script src="//ajax.googleapis.com/ajax/libs/jquery/1.8.3/jquery.min.js"></script>
<script type="text/javascript">
```

```
$(document).ready(function() {

    //alert("vanderlei teste jquery alert");
    //location.reload();

    /*
        var form = $('#form1');
        form.submit(function () {
            return true;
        });
    */
```

```
});
```

```
</script>
```

```
</head>
```

```
<body>
```

```
<div class="context" >
```

```
<h2 class="hello-title">Listagem Empresa</h2>
```

```
<p>
```

```
<a title="Novo" href="TBJEmpresaForm?id=0" class="btn btn btn-default  
command-news"><i class="glyphicon glyphicon-new-window"></i></a>
```

```
<a title="Atualizar" href="TBJEmpresaGrid" class="btn btn btn-default  
command-news"><i class="glyphicon glyphicon-refresh"></i></a>
```

```
</p>
```

```
<br>
```

```
<form id="form1" name="form1" action="TBJEmpresaGrid" Class="form-horizontal"  
method="GET" >
```

```
<table id="tbdados" name="tbdados" class="table" border=1 >
```

```
<tr>
```

```
<th>Codigo </th>
```

```
<th>Nome </th>
```

```
<th>Dadata Cadastro</th>
```

```
<th>Valor</th>
```

```
<th>Ativo</th>
```

```
<th>Controle</th>
```

```
</tr>
```

```
<c:forEach var="linha" items="${Listadados}" >
```

```
<tr>
```

```
<td> ${linha.getIdtbjempresa()}</td>
```

```
<td> ${linha.getNome()}</td>
```

```
<td>
```

```

                                <fmt:formatDate
value="{linha.getDatacadastro()}" var="dataAlteracaoFormatada" pattern="dd/MM/yyyy" />
                                ${dataAlteracaoFormatada}
                                </td>
                                <td> ${linha.getValor()}</td>
                                <td> ${linha.getAtivo()}</td>
                                <td>
                                <a title="Atualizar"
href="TBJEmpresaForm?id=${linha.getIdtbjempresa()}" >clique aqui</a>
                                </td>
                                </tr>
                                </c:forEach>
                                </table>
                                </form>
                                </div>

```

```
<script type="text/javascript">
```

```
//para acertar depois que salva ele nao consegue alimentar a tabela novamente
```

```
$(window).bind("load", function() {
```

```
    var tabela = document.getElementById('tbdados');
```

```
    var linhas = tabela.getElementsByTagName('tr');
```

```
    if(linhas.length <= 1){
```

```
        window.location.replace("TBJEmpresaGrid");    }
```

```
    });
```

```
</script>
```

```
</body>
```

```
</html>
```

```
*****
```

TBJEMPRESAFORM.JSP

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%@ taglib uri="http://java.sun.com/jstl/fmt_rt" prefix="fmt"%>
<%@ page import="java.io.*,java.util.*,java.sql.*"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/sql" prefix="sql"%>

<html xmlns="http://www.w3.org/1999/xhtml"
    xmlns:th="http://www.thymeleaf.org">
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
<link rel="stylesheet" href="/css/bootstrap.min.css">
<script src="/js/jquery.mim.js"></script>
<script src="/js/bootstrap.min.js"></script>

</head>
<body>
    <div class="context" >

        <div class="form-horizontal">
<h4>Dados da Empresa</h4>
<hr />
<form action="TBJEmpresaSalvar" Class="form-horizontal" method="POST" >
    <div class="col-lg-6">

        <div class="form-group">
            <Label class = "control-label col-md-2">Codigo</Label>
```



```

        <div class="col-md-2">
            <input type="text" id="txtidtbjempresa" name="txtidtbjempresa" class =
"form-control input-default col-md-2" readonly value="{dadosEmpr.getIdtbjempresa()}" />
        </div>
    </div>
    <div class="form-group">
        <Label class = "control-label col-md-2">Nome</Label>
        <div class="col-md-2">
            <input type="text" id="txtnome" name="txtnome" class = "form-control input-
default" required="" value="{dadosEmpr.getNome()}" />
        </div>
    </div>
    <div class="form-group">
        <Label class = "control-label col-md-2">datacadastro</Label>
        <div class="col-md-2">
            <fmt:formatDate value="{dadosEmpr.getDatacadastro()}"
var="dataAlteracaoFormatada" pattern="yyyy-dd-MM" />
            <input type="date" id="txtdatacadastro" name="txtdatacadastro" class="form-
control input-default" data-date-format="DD MMMM YYYY"
value="{dataAlteracaoFormatada}" />
        </div>
    </div>
    <div class="form-group">
        <Label class = "control-label col-md-2">valor</Label>
        <div class="col-md-2">
            <input type="number" id="txtvalor" name="txtvalor" class = "form-control
input-default" pattern="([0-9]{1,3}).([0-9]{1,3})" value="{dadosEmpr.getValor()}" >
        </div>
    </div>
    <div class="form-group ">
        <Label class = "control-label col-md-2">ativo</Label>
        <div class="col-md-2">
            <input type="text" id="txtativo" name="txtativo" class = "form-control input-
default" value="{dadosEmpr.getAtivo()}" />
        </div>
    </div>

```

```
        </div>
    </div>

</div>

<div class="form-group">
    <div class="col-md-offset-2 col-md-10">
        <input type="submit" value="Salvar" class="btn btn-default" />
    </div>
</div>
</form>

</div>

<div>
    <a title="Voltar" href="#" class="btn btn-default news"><i class="glyphicon glyphicon-
backward"></i></a>
</div>

</div>
</body>
</html>
```

\*\*\*\*\*

MENU\_INCLUIR.JSP

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
```

```
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
```

```
<html>
```

```
<head>
```

```
<meta charset="ISO-8859-1">
```

```
<title>Insert title here</title>
```

```
<link rel="stylesheet" href="/css/bootstrap.min.css">
```

```
<script src="/js/jquery.mim.js"></script>
```

```
<script src="/js/bootstrap.min.js"></script>
```

```
<script src="//ajax.googleapis.com/ajax/libs/jquery/1.8.3/jquery.min.js"></script>
```

```
</head>
```

```
<body>
```

```
<h2 class="hello-title"> Pagina Incluir dentro de outra pagina</h2>
```

```
</body>
```

```
</html>
```

```
*****
```

```
UPLOAD.JSP
```

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
```

```
pageEncoding="ISO-8859-1"%>
```

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<meta charset="ISO-8859-1">
```

```
<title>Insert title here</title>
```

```
</head>
<body>
<h1>Spring Boot file upload example</h1>

<form method="POST" action="/upload" enctype="multipart/form-data">
  <input type="file" name="file" /><br/><br/>
  <input type="submit" value="Submit" />
</form>
</body>
</html>
```

\*\*\*\*\*

UPLOADSATATUS.JSP

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
  pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html lang="en" xmlns:th="http://www.thymeleaf.org">
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<h1>Spring Boot - Upload Status</h1>

<div th:if="${message}">
  <h2 th:text="${message}"/>
</div>
</body>
```

</html>

\*\*\*\*\*

\*\*\*\*\*

PACOTES

\*\*\*\*\*

\*\*\*\*\*

+++++

com.api.springboot06

+++++

DataConfiguration.java

```
package com.api.springboot06;
```

```
import javax.sql.DataSource;
```

```
import org.springframework.context.annotation.Bean;
```

```
import org.springframework.context.annotation.Configuration;
```

```
import org.springframework.jdbc.datasource.DriverManagerDataSource;
```

```
import org.springframework.orm.jpa.JpaVendorAdapter;
```

```
import org.springframework.orm.jpa.vendor.Database;
```

```
import org.springframework.orm.jpa.vendor.HibernateJpaVendorAdapter;
```

```
@Configuration
```

```
public class DataConfiguration {
```

```
    @Bean
```

```
    public DataSource dataSource(){
```

```
        DriverManagerDataSource dataSource = new DriverManagerDataSource();
```

```
        dataSource.setDriverClassName("com.mysql.jdbc.Driver");
```

```
        dataSource.setUrl("jdbc:mysql://179.188.16.6:3306/sindsmart2");
```

```
dataSource.setUsername("sindsmart2");  
dataSource.setPassword("odavi240175");  
return dataSource;  
}
```

```
@Bean
```

```
public JpaVendorAdapter jpaVendorAdapter(){  
    HibernateJpaVendorAdapter adapter = new HibernateJpaVendorAdapter();  
    adapter.setDatabase(Database.MYSQL);  
    adapter.setShowSql(true);  
    adapter.setGenerateDdl(true);  
    adapter.setDatabasePlatform("org.hibernate.dialect.MySQLDialect");  
    adapter.setPrepareConnection(true);  
    return adapter;  
}
```

```
}
```

```
/*
```

```
Connection conn = dataSource.getConnection();  
Statement stmt = conn.createStatement();  
ResultSet rs = stmt.executeQuery("SELECT ID FROM USERS");  
Statement stmt = con.createStatement();  
  
ResultSet rs = stmt.executeQuery("SELECT a, b, c FROM Table");  
  
while (rs.next()) {
```

```
int x = rs.getInt("a");  
String s = rs.getString("b");  
float f = rs.getFloat("c");  
}
```

```
rs.close();  
stmt.close();  
conn.close();
```

usando o DataSource

```
Connection conn = null;  
Statement stmt = null;  
ResultSet rs = null;
```

```
try {
```

```
    // Get connection from DataSource  
    conn = getMySQLDataSource().getConnection();  
    stmt = conn.createStatement();  
    rs = stmt.executeQuery("SELECT version()");
```

```
    if (rs.next()) {  
        System.out.println("Database Version : " + rs.getString(1));  
    }  
}
```

```
} catch (SQLException e) {  
    e.printStackTrace();  
} finally {
```

```
    try {  
        if (rs != null) {
```

```

        rs.close();
    }
    if (stmt != null) {
        stmt.close();
    }
    if (conn != null) {
        conn.close();
    }
} catch (Exception e) {
    e.printStackTrace();
}
}

*/

+++++

Comandosuteis.java

package com.api.springboot06;

import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Date;

public class Comandosuteis {

    /**
     * Formata a data no formato Brasil 99/99/9999
     * @param pData tipo date

```



```

* @return tipo date
*/
public static Date Data_FormatarDataBRA(Date pData) {
    Date dataretono = pData;

    try {
        dataretono = new
SimpleDateFormat("dd/MM/yyyy").parse(pData.toString());
    } catch (ParseException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }

    return dataretono;
}

/**
 * Formata a data no formato Brasil 99/99/9999
 * @param pData tipo date
 * @return tipo date
 */
public static Date Data_FormatarDataBRA(String pData) {
    Date dataretono = null;
    try {
        dataretono = new
SimpleDateFormat("dd/MM/yyyy").parse(pData.toString());
    } catch (ParseException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}

```

```

        return dataretono;
    }

    /**
     * Formata a data no formato USA 9999-99-99
     * @param pData tipo date
     * @return tipo date
     */
    public static Date Data_FormatarDataUSA(Date pData) {
        Date dataretono = pData;

        try {
            dataretono = new SimpleDateFormat("yyyy-MM-
dd").parse(pData.toString());
        } catch (ParseException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }

        return dataretono;
    }

    /**
     * Formata a data no formato USA 9999-99-99
     * @param pData tipo date
     * @return tipo date
     */
    public static Date Data_FormatarDataUSA(String pData) {
        Date dataretono = null;

```

```
    try {
        dataretono = new SimpleDateFormat("yyyy-MM-dd").parse(pData);
    } catch (ParseException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    return dataretono;
}
```

```
/**
 * Criar data atual
 * @return retorna a data atual
 */
public static Date Data_DataAtual() {
    java.util.Date date = Calendar.getInstance().getTime();
    java.util.Date sqlDate = new java.util.Date(date.getTime());
    return sqlDate;
}
```

```
public static Calendar Data_DateToCalendar(Date date, boolean setTimeToZero){
    Calendar calendario = Calendar.getInstance();
    calendario.setTime(date);
    if(setTimeToZero){
        calendario.set(Calendar.HOUR_OF_DAY, 0);
        calendario.set(Calendar.MINUTE, 0);
        calendario.set(Calendar.SECOND, 0);
        calendario.set(Calendar.MILLISECOND, 0);
    }
}
```

```

        return calendario;
    }

    public static int Data_DiferencaDatas(String dataInicial, String dataFinal){

        Date data1 = new Date(), data2 = new Date();

        Calendar c1 = Calendar.getInstance();

        //Pega a primeira data
        c1.set(Integer.parseInt(dataInicial.substring(0, 4)),
Integer.parseInt(dataInicial.substring(4, 6)), Integer.parseInt(dataInicial.substring(6, 8)));
        data1.setTime(c1.getTimeInMillis());

        //Pega a segunda data
        c1.set(Integer.parseInt(dataFinal.substring(0, 4)),
Integer.parseInt(dataFinal.substring(4, 6)), Integer.parseInt(dataFinal.substring(6, 8)));
        data2.setTime(c1.getTimeInMillis());

        return (int) ((data2.getTime() - data1.getTime()) /1000/60/60/24);
    }

    public static Date Data_adicionarDiasUteis(Date data, Integer qtdeDiasAcrescentados)
    {

        Calendar dataInicial = Calendar.getInstance();
        dataInicial.setTime(data);
        while(qtdeDiasAcrescentados > 0){
            dataInicial.add(Calendar.DAY_OF_MONTH, 1);
            int diaDaSemana = dataInicial.get(Calendar.DAY_OF_WEEK);
            if (diaDaSemana != Calendar.SATURDAY && diaDaSemana !=
Calendar.SUNDAY) {

```

```
        --qtdeDiasAcrescentados;
    }
}
return dataInicial.getTime();
}
```

```
public static String completeToLeft(String value, char c, int size) {
    String result = value;

    while (result.length() < size) {
        result = c + result;
    }

    return result;
}
```

```
//completeToLeft("12938", '0', 15);
```

```
public static int Data_getIdade(Date data) {
    Calendar cData = Calendar.getInstance();
    Calendar cHoje= Calendar.getInstance();
    cData.setTime(data);
    cData.set(Calendar.YEAR, cHoje.get(Calendar.YEAR));
    int idade = cData.after(cHoje) ? -1 : 0;
    cData.setTime(data);
    idade += cHoje.get(Calendar.YEAR) - cData.get(Calendar.YEAR);
    return idade;
}
```

```

public static int Data_calculIdade(java.util.Date dataNasc) {

    Calendar dataNascimento = Calendar.getInstance();
    dataNascimento.setTime(dataNasc);
    Calendar hoje = Calendar.getInstance();

    int idade = hoje.get(Calendar.YEAR) - dataNascimento.get(Calendar.YEAR);

    if (hoje.get(Calendar.MONTH) < dataNascimento.get(Calendar.MONTH)) {
        idade--;
    }
    else
    {
        if (hoje.get(Calendar.MONTH) == dataNascimento.get(Calendar.MONTH) &&
hoje.get(Calendar.DAY_OF_MONTH) < dataNascimento.get(Calendar.DAY_OF_MONTH)) {
            idade--;
        }
    }

    return idade;
}

}

/*

public static int idade(final LocalDate aniversario) {
    final LocalDate dataAtual = LocalDate.now();

```

```
final Period periodo = Period.between(aniversario, dataAtual);  
return periodo.getYears();  
}
```

```
    // Cria um Objeto LocalDate com a data atual.  
    LocalDate hoje = LocalDate.now();  
  
    // Cria um Objeto LocalDate com a data 26/09/2020.  
    LocalDate outraData = LocalDate.of(2020, Month.SEPTEMBER, 26);  
    // Calcula a diferença de dias entre as duas datas  
    long diferencaEmDias = ChronoUnit.DAYS.between(hoje, outraData);  
    // Calcula a diferença de meses entre as duas datas  
    long diferencaEmMes = ChronoUnit.MONTHS.between(hoje, outraData);  
    // Calcula a diferença de anos entre as duas datas  
    long diferencaEmAnos = ChronoUnit.YEARS.between(hoje, outraData);
```

```
import java.time.LocalDate;  
import java.time.Period;  
  
public class DiferencaEntreDatasEmMesesEDias {  
    public static void main(String[] args) {  
  
        LocalDate inicio = LocalDate.of(2011, 7, 28);  
        LocalDate fim = LocalDate.of(2014, 3, 18);  
        Period periodo = Period.between(inicio, fim);
```

```

int anos = periodo.getYears();
int meses = periodo.getMonths();
int dias = periodo.getDays();

    System.out.println("A diferença é de " + anos + " anos, " + meses + " meses e " + dias + "
dias.");
}
}

```

```

Calendar c2 = Calendar.getInstance();
    c2.set(2011, 11, 25);

```

```

    feriados.add(c1.getTime());
    feriados.add(c2.getTime());

```

```

int daysToAdd = 15; // quantidade de dias a adicionar

```

```

// Inicializa data

```

```

Calendar cal = Calendar.getInstance();

```

```

        for (int i = 0; i < daysToAdd; i++) {
            cal.add(Calendar.DATE, 1);
            if (cal.get(Calendar.DAY_OF_WEEK) == Calendar.SATURDAY
                || cal.get(Calendar.DAY_OF_WEEK) ==
Calendar.SUNDAY
                || feriados.contains(cal.getTime())) {

                daysToAdd++;
            }
        }

```



```
        }  
    }  
  
    System.out.println(cal.getTime());
```

```
*/
```

```
+++++
```

```
+++++
```

```
com.api.springboot06.controller
```

```
+++++
```

```
HomeController.java
```

```
package com.api.springboot06.controller;
```

```
import java.text.ParseException;
```

```
import java.text.SimpleDateFormat;
```

```
import java.util.Date;
```

```
import javax.servlet.http.HttpServletRequest;
```

```
import org.springframework.stereotype.Controller;
```

```
import org.springframework.ui.Model;
```

```
import org.springframework.web.bind.annotation.RequestMapping;
```

```
import org.springframework.web.bind.annotation.RequestMethod;
```

```
import com.api.springboot06.Comandosuteis;
```

```
@Controller
```

```
public class HomeController {
```

```
    @RequestMapping(value={"/", "/", "/index"})
```

```
    public String Index(Model model, HttpServletRequest request) {
```

```
        //PREENCHENDO VARIAVEIS DA ESTAD OU DIREITA COM UMA QUANTIDADE
```

```
        /*
```

```
        <dependency>
```

```
        <groupId>org.apache.commons</groupId>
```

```
        <artifactId>commons-lang3</artifactId>
```

```
        <version>3.8</version>
```

```
</dependency>
```

```
        */
```

```
        //String numero = "123";
```

```
        //System.out.println(StringUtils.leftPad(numero, 15, "0"));
```

```
        //System.out.println(StringUtils.rightPad(numero, 15, "0"));
```

```
        //System.out.println(StringUtils.leftPad("123", 15, "0"));
```

```
        //String sExibirTexto = String.format("%10s", "1234").replace(' ', '0');
```

```
        //model.addAttribute("sExibirTexto", sExibirTexto);
```

```
        SimpleDateFormat dateFormat3 = new SimpleDateFormat("dd/MM/yyyy");
```

```

Date date1 = null;

    try {
        date1 = dateFormat3.parse("24/01/1975");
    } catch (ParseException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }

//Date date2 = dateFormat3.parse("15/10/2007");

    int ildade = Comandosuteis.Data_getIdade(date1);

    model.addAttribute("sExibirTexto",ildade);

//PREENCHENDO VARIAVEIS DA ESTAD OU DIREITA COM UMA QUANTIDADE

    model.addAttribute("getURLControle",request.getRequestURL().toString());
    model.addAttribute("name","Vanderlei novo teste");
    return "index.jsp";
}

@RequestMapping(value="/home", method= RequestMethod.GET)
public String home(Model model ) {

    model.addAttribute("namevand","vanderlei soares");
    return "home.jsp";

}

```

```
}
```

```
+++++
```

```
TBJEmpresaController.java
```

```
package com.api.springboot06.controller;
```

```
import java.util.ArrayList;
```

```
import java.util.List;
```

```
import javax.servlet.http.HttpServletRequest;
```

```
import org.springframework.beans.factory.annotation.Autowired;
```

```
import org.springframework.stereotype.Controller;
```

```
import org.springframework.ui.Model;
```

```
import org.springframework.web.bind.annotation.RequestMapping;
```

```
import org.springframework.web.bind.annotation.RequestMethod;
```

```
import org.springframework.web.bind.annotation.RequestParam;
```

```
import com.api.springboot06.Comandosuteis;
```

```
import com.api.springboot06.modal.TBJEmpresa;
```

```
import com.api.springboot06.service.TBJEmpresaService;
```

```
@Controller
```

```
public class TBJEmpresaController {
```

```
    @Autowired
```

```
    private TBJEmpresaService service;
```

```
    @RequestMapping(value="/TBJEmpresaGrid", method= RequestMethod.GET)
```

```
    public String TBJEmpresaGrid(Model model ) {
```

```

        List<TBJEmpresa> Listadados = new ArrayList<TBJEmpresa>();
        Listadados = (List<TBJEmpresa>) this.service.ListaTodos();
        //Listadados = (List<TBJEmpresa>) this.service.BuscarSqlNativo("VANDERLEI
SOARES");

        model.addAttribute("Listadados", Listadados);
        return "TBJEmpresaGrid.jsp";

    }

```

```

@RequestMapping(value="/TBJEmpresaForm", method= RequestMethod.GET)
public String TBJEmpresaForm(@RequestParam("id") int pid, Model model ) {

    TBJEmpresa dados = new TBJEmpresa();
    dados.setIdtbjempresa(0);
    dados.setNome("");

    dados.setDatacadastro(Comandosuteis.Data_DataAtual());
    dados.setValor(00.00);
    dados.setAtivo("SIM");

    if(pid > 0) {
        dados = this.service.BuscarById(pid);
    }

    //Date date1 = Comandosuteis.Data_FormatarDataBRA(dados.getDatacadastro());
    //dados.setDatacadastro(date1);

    //model.addAttribute("dataexib", date1);
    model.addAttribute("dadosEmpr", dados);

    return "TBJEmpresaForm.jsp";
}

```

```

    }

    /*
    na classe model tem que colocar
    @NotNull sobre cada campo que nao pode ser null

    public String TBJEmpresaSalvar(Model model, BindingResult result, RedirectAttributes
attribute){
    if (result.hasErrors()) {
                                model.addAttribute("ErroMensagem", "Os campos devem ser
preenchidos.");
    } else {

    }

    */

    @RequestMapping(value="/TBJEmpresaSalvar", method= RequestMethod.POST)
    public String TBJEmpresaSalvar(Model model, HttpServletRequest request) {

        TBJEmpresa dados = new TBJEmpresa();

        dados.setIdtbjempresa(Integer.parseInt(request.getParameter("txtidtbjempresa")));
        dados.setNome(request.getParameter("txtnome"));

        dados.setDatacadastro(Comandosuteis.Data_FormatarDataUSA(request.getParameter("txtdat
acadastro")));

        String sValor = request.getParameter("txtvalor");

```

```
        if(sValor.indexOf(",") > 0) {
            sValor=sValor.replace(".", "");
            sValor=sValor.replace(",", ".");
        }
        dados.setValor(Double.parseDouble(sValor));

        dados.setAtivo(request.getParameter("txtativo"));

        System.out.println("vanderlei teste "+dados.getValor());

        String smensagem = this.service.Salvar(dados);

        //String sUrl = request.getRequestURL().toString().replace("TBJEmpresaSalvar", "");
        return "redirect:/TBJEmpresaGrid.jsp";

    }

}
```

\*\*\*\*\*

EmpresaRestController.java

```
package com.api.springboot06.controller;
```

```
import java.util.ArrayList;
```

```
import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.DeleteMapping;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.PutMapping;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;
import com.api.springboot06.modal.TBJEmpresa;
import com.api.springboot06.service.TBJEmpresaService;
```

```
@RestController
```

```
@RequestMapping("/empresarest")
```

```
public class EmpresaRestController {
```

```
    @Autowired
```

```
    private TBJEmpresaService service;
```

```
    // @RequestMapping(value = "/pessoa", method = RequestMethod.GET)
```

```
    @GetMapping
```

```
    public List<TBJEmpresa> Get() {
```

```
        List<TBJEmpresa> Listadados = new ArrayList<TBJEmpresa>();
```

```
        Listadados = (List<TBJEmpresa>) this.service.ListaTodos();
```

```
        return Listadados;
```

```
    }
```

```
    @PostMapping
```

```
    public String Post(TBJEmpresa dados) {
```



```
        String smensagem = service.Salvar(dados);  
    return smensagem;  
}
```

```
    @PutMapping  
    public String Put(TBJEmpresa dados) {  
        String smensagem = service.Salvar(dados);  
        return smensagem;  
    }
```

```
    @DeleteMapping  
    public String Delete(Integer pcodigo) {  
        String smensagem = service.Deletar(pcodigo);  
        return smensagem;  
    }
```

```
    //@RequestMapping(value = "/pessoa", method = RequestMethod.POST)
```

```
    //@RequestMapping(value = "/pessoa/{id}", method = RequestMethod.PUT)
```

```
    //@RequestMapping(value = "/pessoa/{id}", method = RequestMethod.DELETE)
```

```
}
```

\*\*\*\*\*

ImageController.java

```
package com.api.springboot06.controller;
```

```
import java.io.File;
```

```
import java.io.IOException;
```

```
import java.nio.file.Files;
```

```
import java.nio.file.Path;
```

```
import java.nio.file.Paths;
```

```
import javax.activation.FileTypeMap;
```

```
import javax.servlet.ServletContext;
```

```
import javax.servlet.http.HttpServletRequest;
```

```
import org.springframework.beans.factory.annotation.Autowired;
```

```
import org.springframework.http.MediaType;
```

```
import org.springframework.http.ResponseEntity;
```

```
import org.springframework.web.bind.annotation.GetMapping;
```

```
import org.springframework.web.bind.annotation.PostMapping;
```

```
import org.springframework.web.bind.annotation.RequestParam;
```

```
import org.springframework.web.bind.annotation.RestController;
```

```
import org.springframework.web.multipart.MultipartFile;
```

```
import org.springframework.web.servlet.mvc.support.RedirectAttributes;
```

```
@RestController
```

```
//@RequestMapping("/empresarest")
```

```
public class ImageController {
```

```
@Autowired
```

```
ServletContext context ;
```

```
@GetMapping("enderecoreal")
```

```
public String Get(HttpServletRequest request) {  
    ServletContext context = request.getSession().getServletContext();  
    String targetPath = context.getRealPath("/static/imagens");  
    targetPath = targetPath.replace("webapp", "resources");  
  
    return targetPath;  
}
```

```
@GetMapping("downloadimage")
```

```
public ResponseEntity<byte[]> getDownloadImage(HttpServletRequest request)  
throws IOException{  
    //src/main/resources/static/  
    //F:\Exemplo\Eclipse\springboot06\src\main\webapp\static\imagens  
    //F:\Exemplo\Eclipse\springboot06\src\main\resources\static\imagens  
    //File img = new File("/imagens/Logo03.png");  
    ServletContext context = request.getSession().getServletContext();  
    String targetPath = context.getRealPath("/static/imagens");  
    targetPath = targetPath.replace("webapp", "resources");  
  
    File img = new File(targetPath+"\\Logo03.png");  
    if(img.exists()) {  
        System.out.println("Vanderlei encontrou o arquivo");  
    }  
  
    return  
    ResponseEntity.ok().contentType(MediaType.valueOf(FileTypeMap.getDefaultFileTypeMap().getContentType(img))).body(Files.readAllBytes(img.toPath()));  
}
```

```
@GetMapping("downloadpdf")
```

```

    public ResponseEntity<byte[]> getDownloadPdf(HttpServletRequest request) throws
IOException{

        ServletContext context = request.getSession().getServletContext();

        String targetPath = context.getRealPath("/static/documentos");

        targetPath = targetPath.replace("webapp", "resources");

        File file = new File(targetPath+"/vandsouares.pdf");

        return ResponseEntity.ok()

            .header("Content-Disposition", "attachment; filename=" +file.getName())

            .contentType(MediaType.valueOf(FileTypeMap.getDefaultFileTypeMap().getContentType(file))
            )

            .body(Files.readAllBytes(file.toPath()));

    }

```

@PostMapping("/upload") // //new annotation since 4.3 imagem

```

public String singleFileUpload(@RequestParam("file") MultipartFile file,RedirectAttributes
redirectAttributes,HttpServletRequest request) {

```

```

    if (file.isEmpty()) {

        redirectAttributes.addFlashAttribute("message", "Please select a file to upload");

        return "redirect:uploadStatus";

    }

```

```

try {

    // Get the file and save it somewhere

    byte[] bytes = file.getBytes();

    String targetPath = context.getRealPath("/static/imagens");

        targetPath = targetPath.replace("webapp", "resources");

    Path path = Paths.get(targetPath + file.getOriginalFilename());

    Files.write(path, bytes);

```

```

        redirectAttributes.addFlashAttribute("message",
            "Foi enviado o arquivos '" + file.getOriginalFilename() + "'");

    } catch (IOException e) {
        e.printStackTrace();
    }

    //criar uma pagina para exibir a imagem ou mensagem de que deu certo ou nao
    return "redirect:/uploadStatus";
}

    @GetMapping("/uploadStatus")
    public String uploadStatus() {
        return "uploadStatus";
    }

    /*
    @PostMapping("/uploadFile")
    public UploadFileResponse uploadFile(@RequestParam("file") MultipartFile file) {
        String fileName = fileStorageService.storeFile(file);

        String fileDownloadUri = ServletUriComponentsBuilder.fromCurrentContextPath()
            .path("/downloadFile/")
            .path(fileName)
            .toUriString();

        return new UploadFileResponse(fileName, fileDownloadUri, file.getContentType(),
file.getSize());
    }

```

```

    @PostMapping("/uploadMultipleFiles")
    public List<UploadFileResponse> uploadMultipleFiles(@RequestParam("files")
MultipartFile[] files) {
        return Arrays.asList(files)
            .stream()
            .map(file -> uploadFile(file))
            .collect(Collectors.toList());
    }

    @GetMapping("/downloadFile/{fileName:.+}")
    public ResponseEntity<Resource> downloadFile(@PathVariable String fileName,
HttpServletRequest request) {
        // Load file as Resource
        Resource resource = fileStorageService.loadFileAsResource(fileName);

        // Try to determine file's content type
        String contentType = null;
        try {
            contentType =
request.getServletContext().getMimeType(resource.getFile().getAbsolutePath());
        } catch (IOException ex) {
            logger.info("Could not determine file type.");
        }

        // Fallback to the default content type if type could not be determined
        if(contentType == null) {
            contentType = "application/octet-stream";
        }

        return ResponseEntity.ok()
            .contentType(MediaType.parseMediaType(contentType))

```

```
        .header(HttpHeaders.CONTENT_DISPOSITION, "attachment; filename=\"" +
resource.getFilename() + "\"")
        .body(resource);
    }

    */
```

```
}
```

```
*****
```

```
*****
```

```
com.api.springboot06.email
```

```
*****
```

```
*****
```

```
SendMail.java
```

```
package com.api.springboot06.email;
```

```
import java.util.Properties;
```

```
import javax.mail.Message;
```

```
import javax.mail.Session;
```

```
import javax.mail.Transport;
```

```
import javax.mail.internet.InternetAddress;
```

```
import javax.mail.internet.MimeMessage;
```

```
import javax.mail.Authenticator;
```

```
import javax.mail.PasswordAuthentication;
```

```
public class SendMail {

    /*
    //https://jar-download.com/artifacts/com.sun.mail/javax.mail/1.6.1/source-code

        <dependency>
            <groupId>com.sun.mail</groupId>
            <artifactId>javax.mail</artifactId>
        </dependency>

    */

    private String mailSMTPServer;
        private String mailSMTPServerPort;

    /*
    * quando instanciar um Objeto ja sera atribuido o servidor SMTP do GMAIL
    * e a porta usada por ele
    */
    public SendMail () { //Para o GMAIL
        mailSMTPServer = "smtp.gmail.com";
        mailSMTPServerPort = "465";
    }

    /*
    * caso queira mudar o servidor e a porta, so enviar para o contrutor
    * os valor como string
    */
}
```



```

    public SendMail (String mailSMTPServer, String mailSMTPServerPort) { //Para outro
Servidor

        this.mailSMTPServer = mailSMTPServer;

        this.mailSMTPServerPort = mailSMTPServerPort;

    }

    public void sendMail(String from, String to, String subject, String message) {

        Properties props = new Properties();

        // quem estiver utilizando um SERVIDOR PROXY descomente essa parte e atribua as
propriedades do SERVIDOR PROXY utilizado

        /*
        props.setProperty("proxySet", "true");
        props.setProperty("socksProxyHost", "192.168.155.1"); // IP do Servidor Proxy
        props.setProperty("socksProxyPort", "1080"); // Porta do servidor Proxy
        */

        props.put("mail.transport.protocol", "smtp"); //define protocolo de envio
como SMTP

        props.put("mail.smtp.starttls.enable", "true");

        props.put("mail.smtp.host", mailSMTPServer); //server SMTP do GMAIL

        props.put("mail.smtp.auth", "true"); //ativa autenticao

        props.put("mail.smtp.user", from); //usuario ou seja, a conta que esta
enviando o email (tem que ser do GMAIL)

        props.put("mail.debug", "true");

        props.put("mail.smtp.port", mailSMTPServerPort); //porta

        props.put("mail.smtp.socketFactory.port", mailSMTPServerPort); //mesma
porta para o socket

        props.put("mail.smtp.socketFactory.class", "javax.net.ssl.SSLSocketFactory");

        props.put("mail.smtp.socketFactory.fallback", "false");

        //Cria um autenticador que sera usado a seguir

```

```
SimpleAuth auth = null;

auth = new SimpleAuth ("websalt@websalt.com.br", "websalt240175");

//Session - objeto que ira realizar a conexão com o servidor
/*Como há necessidade de autenticação é criada uma autenticação que
* é responsável por solicitar e retornar o usuário e senha para
* autenticação */
Session session = Session.getDefaultInstance(props, auth);
session.setDebug(true); //Habilita o LOG das ações executadas durante o envio
do email

//Objeto que contém a mensagem
Message msg = new MimeMessage(session);

try {
    //Setando o destinatário
    msg.setRecipient(Message.RecipientType.TO, new
InternetAddress(to));
    //Setando a origem do email
    msg.setFrom(new InternetAddress(from));
    //Setando o assunto
    msg.setSubject(subject);
    //Setando o conteúdo/corpo do email
    msg.setContent(message, "text/plain");

} catch (Exception e) {
    System.out.println(">> Erro: Completar Mensagem");
    e.printStackTrace();
}

//Objeto encarregado de enviar os dados para o email
Transport tr;
```

```

try {
    tr = session.getTransport("smtp"); //define smtp para transporte
    /*
    * 1 - define o servidor smtp
    * 2 - seu nome de usuario do gmail
    * 3 - sua senha do gmail
    */
    tr.connect(mailSMTPServer, "websalt@websalt.com.br",
"websalt240175");

    msg.saveChanges(); // don't forget this
    //envio da mensagem
    tr.sendMessage(msg, msg.getAllRecipients());
    tr.close();
} catch (Exception e) {
    // TODO Auto-generated catch block
    System.out.println(">> Erro: Envio Mensagem");
    e.printStackTrace();
}

}
}
}

```

```

class SimpleAuth extends Authenticator {

```

```

    public String username = null;

```

```

    public String password = null;

```

```
public SimpleAuth(String user, String pwd) {
    username = user;
    password = pwd;
}

protected PasswordAuthentication getPasswordAuthentication() {
    return new PasswordAuthentication (username,password);
}

}

/*

    EXEMPLO DE USO DA CLASS DE ENVIO DE EMAIL

    SendMail mail = new SendMail();

    try {
        mail.sendMail("websalt@websalt.com.br", "sovanderlei@hotmail.com", "TESTE
JAVAMAIL", "TESTE");
    } catch (Exception e) {
        e.printStackTrace();
    }

*/
```

\*\*\*\*\*

\*\*\*\*\*

com.api.springboot06.modal

\*\*\*\*\*

\*\*\*\*\*

TBJEmpresa.java

```
package com.api.springboot06.modal;
```

```
import java.io.Serializable;
```

```
import java.util.Date;
```

```
import javax.persistence.Column;
```

```
import javax.persistence.Entity;
```

```
import javax.persistence.GeneratedValue;
```

```
import javax.persistence.Id;
```

```
import javax.persistence.Table;
```

```
@Entity
```

```
@Table(name="TBJEmpresa")
```

```
public class TBJEmpresa implements Serializable {
```

```
    /**
```

```
     *
```

```
    */
```

```
    private static final long serialVersionUID = 1L;
```

```
    @Id @GeneratedValue
```

```
    @Column(name="idtbjempresa")
```

```
    private int idtbjempresa;
```

```
@Column(name="nome")
@NotEmpty(message="O nome nao pode esta vazio")
private String nome;
```

```
@Column(name="datacadastro", nullable=false)
//@Temporal(TemporalType.DATE)
//@Temporal(TemporalType.TIMESTAMP)
private Date datacadastro;
```

```
@Column(name="valor")
private Double valor;
```

```
@Column(name="ativo")
private String ativo;
```

```
public static long getSerialVersionUID() {
    return serialVersionUID;
}
```

```
public int getIdtbjempresa() {
    return idtbjempresa;
}
```

```
public void setIdtbjempresa(int idtbjempresa) {
    this.idtbjempresa = idtbjempresa;
}
```

```
public String getNome() {
    return nome;
}
```

```
}
```

```
public void setNome(String nome) {
```

```
    this.nome = nome;
```

```
}
```

```
public Date getDatacadastro() {
```

```
    return datacadastro;
```

```
}
```

```
public void setDatacadastro(Date datacadastro) {
```

```
    this.datacadastro = datacadastro;
```

```
}
```

```
public Double getValor() {
```

```
    return valor;
```

```
}
```

```
public void setValor(Double valor) {
```

```
    this.valor = valor;
```

```
}
```

```
public String getAtivo() {
```

```
    return ativo;
```

```
}
```

```
public void setAtivo(String ativo) {
```

```
    this.ativo = ativo;
```

```
}
```

```
}
```

```
*****
```

```
*****
```

```
com.api.springboot06.repository
```

```
*****
```

```
*****
```

```
TBJEmpresaRepository.java
```

```
package com.api.springboot06.repository;
```

```
import java.util.List;
```

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

```
import org.springframework.data.repository.CrudRepository;
```

```
import org.springframework.data.repository.query.Param;
```

```
import org.springframework.stereotype.Repository;
```

```
import com.api.springboot06.modal.TBJEmpresa;
```

```
@Repository
```

```
public interface TBJEmpresaRepository extends CrudRepository<TBJEmpresa, Integer> {
```



```
        @Query("select s from TBJEmpresa s")
        List<TBJEmpresa> findByNaoNativeQuery();

        @Query(value = "SELECT * FROM tbjempresa where nome = (:nome) ", nativeQuery =
true)
        List<TBJEmpresa> findByNativeQuery(@Param("nome") String nome);

    }
}
```

```
*****
```

```
*****
```

```
com.api.springboot06.service
```

```
*****
```

```
*****
```

```
TBJEmpresaService.java
```

```
package com.api.springboot06.service;
```

```
import java.util.ArrayList;
```

```
import java.util.Calendar;
```

```
import java.util.List;
```

```
import java.util.Optional;
```

```
import org.springframework.stereotype.Service;
```

```
import com.api.springboot06.modal.TBJEmpresa;
import com.api.springboot06.repository.TBJEmpresaRepository;

@Service
public interface TBJEmpresaService extends TBJEmpresaRepository {
```

```
    public default List<TBJEmpresa> ListaTodos(){
        List<TBJEmpresa> ListaRetorno = new ArrayList<TBJEmpresa>();
        ListaRetorno = (List<TBJEmpresa>) this.findAll();
        return ListaRetorno;
    }
```

```
    public default String Salvar(TBJEmpresa pdados) {
        try {

            this.save(pdados);

            return "Não foi possível gravar os dados.";

        }catch(Exception e) {

            return e.getMessage() ;
        }
    }
```

```
    public default String Deletar(int codigo){
```

```

Optional<TBJEmpresa> dadosPesc = this.findByld(codigo);

try {

    TBJEmpresa dados = new TBJEmpresa();
    if (dadosPesc.isPresent()) {
        dados = dadosPesc.get();
    }

    this.delete(dados);

    return "Registro excluido com sucesso!";

} catch (Exception e) {
    return e.getMessage();
}
}

```

```

public default TBJEmpresa BuscarByld(int codigo){

    TBJEmpresa dadosretorno = new TBJEmpresa();
    dadosretorno.setldtbjempresa(0);
    dadosretorno.setNome("");

    java.util.Date date = Calendar.getInstance().getTime();
    java.sql.Date sqlDate = new java.sql.Date(date.getTime());
    dadosretorno.setDatacadastro(sqlDate);
    dadosretorno.setValor(00.00);
    dadosretorno.setAtivo("SIM");

```

```

Optional<TBJEmpresa> dadosPesc = this.findByld(codigo);

```

```
try {  
  
    if (dadosPesc.isPresent()) {  
        dadosretorno = dadosPesc.get();  
    }  
  
    return dadosretorno;  
  
}catch(Exception e) {  
    return dadosretorno;  
}  
  
}
```

```
public default List<TBJEmpresa> BuscarSqlNativo(String pNome){  
  
    List<TBJEmpresa> Lista = new ArrayList<TBJEmpresa>();  
  
    Lista = findByNativeQuery(pNome);  
  
    return Lista;  
  
}
```

}

\*\*\*\*\*

=====  
=====  
=====  
=====

SPRING TOOL SUIT -- SPRING BOOT

=====

SPRING BOOT COM HTML JSF

=====

<https://www.thymeleaf.org/doc/tutorials/3.0/usingthymeleaf.html>

<https://start.spring.io/>

web starter, devtools, thymeleaf

descompactar e importar usando a opção no menu

usar o maven

\*\*\*\*\*

CRIAR OS PACOTES

observacao sempre usar o primeiro pacote como referencia

olha a sequencia abaixo

com.api.springboot06  
com.api.springboot06.controller  
com.api.springboot06.email  
com.api.springboot06.modal  
com.api.springboot06.repository  
com.api.springboot06.service

\*\*\*\*\*

pom.xml

```
<dependency>  
  <groupId>org.apache.tomcat</groupId>  
  <artifactId>tomcat-jasper</artifactId>  
  <version>9.0.24</version>  
</dependency>
```

a versao é de acordo com o que esta na biblioteca

Maven Dependencies

procure o arquivo

tomcat-embed-core-....

no browser procurar

mvnrepository

<https://mvnrepository.com/>

procurar - Tomcat Jasper

clicar na opção Tomcat Jasper

<https://mvnrepository.com/artifact/org.apache.tomcat/tomcat-jasper>

pegar a versao que foi visto no Maven Dependencies

<?xml version="1.0" encoding="UTF-8"?>

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
https://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <parent>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-parent</artifactId>
        <version>2.1.8.RELEASE</version>
        <relativePath/> <!-- lookup parent from repository -->
    </parent>
    <groupId>com.api</groupId>
    <artifactId>exemplosphtml</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <packaging>war</packaging>
    <name>exemplosphtml</name>
    <description>Exemplo html </description>

    <properties>
        <java.version>1.8</java.version>
    </properties>

    <dependencies>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-thymeleaf</artifactId>
        </dependency>

        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-web</artifactId>
        </dependency>
    </dependencies>
</project>
```

```
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-devtools</artifactId>
    <scope>runtime</scope>
    <optional>true</optional>
</dependency>
```

```
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-tomcat</artifactId>
    <scope>provided</scope>
</dependency>
```

```
<dependency>
    <groupId>org.apache.tomcat</groupId>
    <artifactId>tomcat-jasper</artifactId>
    <version>9.0.24</version>
</dependency>
```

```
<dependency>
    <groupId>org.hibernate</groupId>
    <artifactId>hibernate-entitymanager</artifactId>
</dependency>
```

```
<!-- JSTL for JSP -->
```

```
<dependency>
    <groupId>javax.servlet</groupId>
    <artifactId>jstl</artifactId>
</dependency>
```



```
<!-- Need this to compile JSP -->
<dependency>
    <groupId>org.apache.tomcat.embed</groupId>
    <artifactId>tomcat-embed-jasper</artifactId>
    <scope>provided</scope>
</dependency>
```

```
<!-- Need this to compile JSP, -->
```

```
<!-- JpaRepository -->
```

```
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-data-rest</artifactId>
</dependency>
```

```
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-data-jpa</artifactId>
</dependency>
```

```
<dependency>
    <groupId>com.h2database</groupId>
    <artifactId>h2</artifactId>
</dependency>
```

```
<!-- fim JpaRepository -->
```

```
<!-- MYSQL -->
```

```
<dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
    <scope>compile</scope>
```

```
        </dependency>
<!-- FIM MYSQL -->

<!-- Template Engine Thymeleaf -->
<dependency>
    <groupId>nz.net.ultraq.thymeleaf</groupId>
    <artifactId>thymeleaf-layout-dialect</artifactId>
</dependency>

<dependency>
    <groupId>org.thymeleaf</groupId>
    <artifactId>thymeleaf</artifactId>
</dependency>

<dependency>
    <groupId>org.thymeleaf</groupId>
    <artifactId>thymeleaf-spring4</artifactId>
    <version> ${thymeleaf.version}</version>
</dependency>

    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-test</artifactId>
        <scope>test</scope>
    </dependency>
</dependencies>

<build>
    <plugins>
        <plugin>
```

```
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-maven-plugin</artifactId>
            </plugin>
        </plugins>
    </build>

</project>
```

\*\*\*\*\*

APPLICATION.PROPERTIES

#spring.resources.static-locations=classpath:/custom/

#PORTA ONDE VAI SER EXECUTADO O NOSSO TOMCAT

server.port = 8082

#INFORMAÇÕES PARA CONEXÃO COM O BANCO DE DADOS MYSQL

spring.datasource.url=jdbc:mysql://179.188.16.6:3306/sindsmart2

spring.datasource.username=sindsmart2

spring.datasource.password=odavi240175

spring.datasource.driver-class-name=com.mysql.jdbc.Driver

#PARA CRIAR A TABELA NO BANCO BASTA COLOCAR update cria as tabelas

#create

spring.jpa.hibernate.ddl-auto=update

# LazyInitializationException = spring.jpa.open-in-view=false

spring.jpa.open-in-view=false

spring.jpa.generate-ddl=true

spring.jpa.show-sql=true

spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL5Dialect

spring.jpa.properties.hibernate.format\_sql=true

```
#=====
```

```
# = Thymeleaf configurations
```

```
#=====
```

```
#PARA USAR O JSP TEM DE MANTER ESTAS LINHAS ABAIXO COMENTADAS
```

```
spring.thymeleaf.check-template-location=true
```

```
spring.thymeleaf.prefix=classpath:/templates/
```

```
spring.thymeleaf.suffix=.html
```

```
spring.thymeleaf.servlet.content-type=text/html
```

```
spring.thymeleaf.cache=false
```

```
server.servlet.context-path=/
```

```
#=====
```

```
#For detailed logging during development
```

```
logging.level.org.springframework=TRACE
```

```
logging.level.com=TRACE
```

```
*****
```

```
na pasta static criar pasta
```

```
css, js, imagens, download, documentos
```

```
criar os arquivos
```

```
layout.css
```

```
.hello-title{
```

```
    color: darkgreen;
```

```
}
```

indexjs.js

```
//load pagina
```

```
/*
```

```
window.addEventListener("load", function(){
```

```
    alert("teste java script");
```

```
});
```

```
(function(){
```

```
    console.log("Hello World!");
```

```
})();
```

```
*/
```

```
window.addEventListener("load", function(){
```

```
    //...
```

```
});
```

```
*****
```

```
SPRINGBOOT06APPLICATION.JAVA
```

```
package com.api.exemploshtml;
```

```
import org.springframework.boot.SpringApplication;
```

```
import org.springframework.boot.autoconfigure.SpringBootApplication;
```

```
@SpringBootApplication
```

```
public class ExemploshtmlApplication {
```

```
    public static void main(String[] args) {
```

```
        SpringApplication.run(ExemplosphtmlApplication.class, args);
    }
}
```

```
*****
```

```
package com.api.exemplosphtml;
```

```
import javax.sql.DataSource;
```

```
import org.springframework.context.annotation.Bean;
```

```
import org.springframework.context.annotation.Configuration;
```

```
import org.springframework.jdbc.datasource.DriverManagerDataSource;
```

```
import org.springframework.orm.jpa.JpaVendorAdapter;
```

```
import org.springframework.orm.jpa.vendor.Database;
```

```
import org.springframework.orm.jpa.vendor.HibernateJpaVendorAdapter;
```

```
@Configuration
```

```
public class DataConfiguration {
```

```
    @Bean
```

```
    public DataSource dataSource(){
```

```
        DriverManagerDataSource dataSource = new DriverManagerDataSource();
```

```
        dataSource.setDriverClassName("com.mysql.jdbc.Driver");
```

```
        dataSource.setUrl("jdbc:mysql://179.188.16.6:3306/sindsmart2");
```

```
        dataSource.setUsername("sindsmart2");
```

```
        dataSource.setPassword("odavi240175");
```

```
        return dataSource;
```

```
    }
```

```
@Bean
```

```
public JpaVendorAdapter jpaVendorAdapter(){
```

```
    HibernateJpaVendorAdapter adapter = new HibernateJpaVendorAdapter();
```

```
    adapter.setDatabase(Database.MYSQL);
```

```
    adapter.setShowSql(true);
```

```
    adapter.setGenerateDdl(true);
```

```
    adapter.setDatabasePlatform("org.hibernate.dialect.MySQLDialect");
```

```
    adapter.setPrepareConnection(true);
```

```
    return adapter;
```

```
}
```

```
}
```

```
/*
```

```
<dependency>
```

```
    <groupId>com.oracle</groupId>
```

```
    <artifactId>ora-jdbc</artifactId>
```

```
    <version>7</version>
```

```
</dependency>
```

```
https://www.logicbig.com/tutorials/spring-framework/spring-data-access-with-jdbc/connect-oracle.html
```

```
*/
```

```
*****
```

```
package com.api.exemploshtml;
```

```
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Date;

public class Comandosuteis {

    /**
     * Formata a data no formato Brasil 99/99/9999
     * @param pData tipo date
     * @return tipo date
     */
    public static Date Data_FormatarDataBRA(Date pData) {
        Date dataretono = pData;

        try {
            dataretono = new
SimpleDateFormat("dd/MM/yyyy").parse(pData.toString());
        } catch (ParseException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }

        return dataretono;
    }
}
```



```

/**
 * Formata a data no formato Brasil 99/99/9999
 * @param pData tipo date
 * @return tipo date
 */
public static Date Data_FormatarDataBRA(String pData) {
    Date dataretono = null;
    try {
        dataretono = new
SimpleDateFormat("dd/MM/yyyy").parse(pData.toString());
    } catch (ParseException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    return dataretono;
}

/**
 * Formata a data no formato USA 9999-99-99
 * @param pData tipo date
 * @return tipo date
 */
public static Date Data_FormatarDataUSA(Date pData) {
    Date dataretono = pData;

    try {
        dataretono = new SimpleDateFormat("yyyy-MM-
dd").parse(pData.toString());
    } catch (ParseException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}

```

```

        return dataretono;

    }

    /**
     * Formata a data no formato USA 9999-99-99
     * @param pData tipo date
     * @return tipo date
     */
    public static Date Data_FormatarDataUSA(String pData) {
        Date dataretono = null;
        try {
            dataretono = new SimpleDateFormat("yyyy-MM-dd").parse(pData);
        } catch (ParseException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
        return dataretono;
    }

    /**
     * Formata a data no formato USA 9999-99-99 00:00
     * @param pData tipo date
     * @return tipo date
     */
    public static Date Data_FormatarDataUSAIncluirTime(String pData) {
        Date dataretono = null;
        try {
            //2019-01-01 00:00:00

```

```

        pData = pData+" 00:00:00";
        dataretono = new SimpleDateFormat("yyyy-MM-dd").parse(pData);
    } catch (ParseException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    return dataretono;
}

```

```

public static Date Data_FormatarDataUSAIncluirTime(Date pData) {
    Date dataretono = pData;

    try {
        String sData = pData.toString()+" 00:00:00";
        dataretono = new SimpleDateFormat("yyyy-MM-dd").parse(sData);
    } catch (ParseException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }

    return dataretono;
}

```

```

/**
 * Criar data atual com horas
 * @return retorna a data atual
 */
public static Date Data_DataTimeAtual() {
    java.util.Date date = Calendar.getInstance().getTime();
}

```

```

        java.util.Date sqlDate = new java.util.Date(date.getTime());

        return sqlDate;
    }

    /**
     * Criar data atual
     * @return retorna a data atual
     */
    public static Date Data_DataAtual() {
        Date dataHoraAtual = new Date();

        //String data = new
SimpleDateFormat("dd/MM/yyyy").format(dataHoraAtual);

        //String hora = new SimpleDateFormat("HH:mm:ss").format(dataHoraAtual);

        Date date1 = new Date();

        try {
            date1 = new
SimpleDateFormat("dd/MM/yyyy").parse(dataHoraAtual.toString());
        } catch (ParseException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }

        return date1;
    }

    public static String Data_DataAtualString() {
        String sDataRetorno = "";

        Date dataHoraAtual = new Date();

        sDataRetorno = new
SimpleDateFormat("dd/MM/yyyy").format(dataHoraAtual);

```

```

        return sDataRetorno;

    }

    public static Calendar Data_DateToCalendar(Date date, boolean setTimeToZero){
        Calendar calendario = Calendar.getInstance();
        calendario.setTime(date);
        if(setTimeToZero){
            calendario.set(Calendar.HOUR_OF_DAY, 0);
            calendario.set(Calendar.MINUTE, 0);
            calendario.set(Calendar.SECOND, 0);
            calendario.set(Calendar.MILLISECOND, 0);
        }

        return calendario;
    }

    public static int Data_DiferencaDatas(String dataInicial, String dataFinal){

        Date data1 = new Date(), data2 = new Date();

        Calendar c1 = Calendar.getInstance();

        //Pega a primeira data
        c1.set(Integer.parseInt(dataInicial.substring(0, 4)),
Integer.parseInt(dataInicial.substring(4, 6)), Integer.parseInt(dataInicial.substring(6, 8)));
        data1.setTime(c1.getTimeInMillis());
    }

```

```

        //Pega a segunda data
        c1.set(Integer.parseInt(dataFinal.substring(0, 4)),
Integer.parseInt(dataFinal.substring(4, 6)), Integer.parseInt(dataFinal.substring(6, 8)));

        data2.setTime(c1.getTimeInMillis());

        return (int) ((data2.getTime() - data1.getTime()) /1000/60/60/24);
    }

    public static Date Data_adicionarDiasUteis(Date data, Integer qtdeDiasAcrescentados)
    {

        Calendar dataInicial = Calendar.getInstance();
        dataInicial.setTime(data);
        while(qtdeDiasAcrescentados > 0){

            dataInicial.add(Calendar.DAY_OF_MONTH, 1);

            int diaDaSemana = dataInicial.get(Calendar.DAY_OF_WEEK);
            if (diaDaSemana != Calendar.SATURDAY && diaDaSemana !=
Calendar.SUNDAY) {

                --qtdeDiasAcrescentados;

            }

        }

        return dataInicial.getTime();
    }

    public static String completeToLeft(String value, char c, int size) {

        String result = value;

        while (result.length() < size) {

            result = c + result;

        }

        return result;
    }

```

```
}
```

```
//completeToLeft("12938", '0', 15);
```

```
public static int Data_getIdade(Date data) {  
    Calendar cData = Calendar.getInstance();  
    Calendar cHoje= Calendar.getInstance();  
    cData.setTime(data);  
    cData.set(Calendar.YEAR, cHoje.get(Calendar.YEAR));  
    int idade = cData.after(cHoje) ? -1 : 0;  
    cData.setTime(data);  
    idade += cHoje.get(Calendar.YEAR) - cData.get(Calendar.YEAR);  
    return idade;  
}
```

```
}
```

```
public static int Data_calculaidade(java.util.Date dataNasc) {  
  
    Calendar dataNascimento = Calendar.getInstance();  
    dataNascimento.setTime(dataNasc);  
    Calendar hoje = Calendar.getInstance();  
  
    int idade = hoje.get(Calendar.YEAR) - dataNascimento.get(Calendar.YEAR);  
  
    if (hoje.get(Calendar.MONTH) < dataNascimento.get(Calendar.MONTH)) {  
        idade--;  
    }  
    else  
    {
```

```
        if (hoje.get(Calendar.MONTH) == dataNascimento.get(Calendar.MONTH) &&
hoje.get(Calendar.DAY_OF_MONTH) < dataNascimento.get(Calendar.DAY_OF_MONTH)) {
            idade--;
        }
    }

    return idade;
}
```

```
}
```

```
/*
```

```
public static int idade(final LocalDate aniversario) {
    final LocalDate dataAtual = LocalDate.now();
    final Period periodo = Period.between(aniversario, dataAtual);
    return periodo.getYears();
}
```



```
        // Cria um Objeto LocalDate com a data atual.
LocalDate hoje = LocalDate.now();

// Cria um Objeto LocalDate com a data 26/09/2020.
LocalDate outraData = LocalDate.of(2020, Month.SEPTEMBER, 26);

    // Calcula a diferença de dias entre as duas datas
long diferencaEmDias = ChronoUnit.DAYS.between(hoje, outraData);

// Calcula a diferença de meses entre as duas datas
long diferencaEmMes = ChronoUnit.MONTHS.between(hoje, outraData);

// Calcula a diferença de anos entre as duas datas
long diferencaEmAnos = ChronoUnit.YEARS.between(hoje, outraData);

import java.time.LocalDate;
import java.time.Period;

public class DiferencaEntreDatasEmMesesEDias {
    public static void main(String[] args) {

        LocalDate inicio = LocalDate.of(2011, 7, 28);
        LocalDate fim = LocalDate.of(2014, 3, 18);
        Period periodo = Period.between(inicio, fim);

        int anos = periodo.getYears();
        int meses = periodo.getMonths();
        int dias = periodo.getDays();
    }
}
```

```
        System.out.println("A diferença é de " + anos + " anos, " + meses + " meses e " + dias + "
dias.");
    }
}
```

```
Calendar c2 = Calendar.getInstance();
```

```
    c2.set(2011, 11, 25);
```

```
    feriados.add(c1.getTime());
```

```
    feriados.add(c2.getTime());
```

```
int daysToAdd = 15; // quantidade de dias a adicionar
```

```
// Inicializa data
```

```
Calendar cal = Calendar.getInstance();
```

```
        for (int i = 0; i < daysToAdd; i++) {
            cal.add(Calendar.DATE, 1);
            if (cal.get(Calendar.DAY_OF_WEEK) == Calendar.SATURDAY
                || cal.get(Calendar.DAY_OF_WEEK) ==
Calendar.SUNDAY
                || feriados.contains(cal.getTime())) {
                daysToAdd--;
            }
        }

        System.out.println(cal.getTime());
```

```
*/
```

```
*****
```

```
package com.api.exemplosphtml.controller;
```

```
import org.springframework.stereotype.Controller;
```

```
import org.springframework.ui.Model;
```

```
import org.springframework.web.bind.annotation.RequestMapping;
```

```
import com.api.exemplosphtml.Comandosuteis;
```

```
@Controller
```

```
public class HomeController {
```

```
    //@RequestMapping(value="/home", method= RequestMethod.GET)
```

```
    @RequestMapping(value = {"", "/", "index", "Layout"})
```

```
    public String home(Model model ) {
```

```
        model.addAttribute("Idtexibir",Comandosuteis.Data_DataAtualString() );
```

```
        return "Index";
```

```
    }
```

```
}
```

\*\*\*\*\*

```
package com.api.exemplosphtml.controller;
```

```
import java.util.ArrayList;
```

```
import java.util.List;
```

```
import org.springframework.beans.factory.annotation.Autowired;
```

```
import org.springframework.stereotype.Controller;
```

```
import org.springframework.ui.Model;
```

```
import org.springframework.web.bind.annotation.RequestMapping;
```

```
import org.springframework.web.bind.annotation.RequestMethod;
```

```
import org.springframework.web.bind.annotation.RequestParam;
```

```
import com.api.exemplosphtml.Comandosuteis;
```

```
import com.api.exemplosphtml.modal.TBJEmpresa;
```

```
import com.api.exemplosphtml.service.TBJEmpresaService;
```

```
@Controller
```

```
public class TBJEmpresaController {
```

```
    @Autowired
```

```
    private TBJEmpresaService service;
```

```
    @RequestMapping(value="/TBJEmpresaGrid",method=RequestMethod.GET)
```

```
    public String TBJEmpresaGrid(Model model) {
```

```
        List<TBJEmpresa> Listadados = new ArrayList<TBJEmpresa>();
```

```
        Listadados = (List<TBJEmpresa>) this.service.ListaTodos();
```

```
        model.addAttribute("Listadados", Listadados);
```

```

        return "TBJEmpresaGrid";
    }

    @RequestMapping(value="/TBJEmpresaForm",method=RequestMethod.GET)
    public String TBJEmpresaForm(@RequestParam("id") int pid, Model model) {

        TBJEmpresa dados = new TBJEmpresa();
        dados.setIdtbjempresa(0);
        dados.setNome("");

        dados.setDatacadastro(Comandosuteis.Data_DataAtual());
        dados.setValor(00.00);
        dados.setAtivo("SIM");

        if(pid > 0) {
            dados = this.service.BuscarById(pid);
        }

        model.addAttribute("TBJEmpresa", dados);
        return "TBJEmpresaForm";
    }

    //@RequestMapping(value="/TBJEmpresaForm",method=RequestMethod.POST)
    @RequestMapping( value="/TBJEmpresaForm", method=RequestMethod.POST,
params={"Salvar"} )
    public String TBJEmpresaSalvar(Model model, TBJEmpresa pdados) {

```

```

        TBJEmpresa dados = new TBJEmpresa();

        dados.setIdtbjempresa(pdados.getIdtbjempresa());
        dados.setNome(pdados.getNome());

        dados.setDatacadastro(Comandosuteis.Data_FormatarDataUSA(pdados.getDatacadastro()));

        //dados.setDatacadastro(Comandosuteis.Data_FormatarDataBRA(pdados.getDatacadastro()));

        String sValor = pdados.getValor().toString();
        if(sValor.indexOf(",") > 0) {
            sValor=sValor.replace(".", "");
            sValor=sValor.replace(",", ".");
        }
        dados.setValor(Double.parseDouble(sValor));

        dados.setAtivo(pdados.getAtivo());

        System.out.println("vanderlei teste "+dados.getValor());

        String smensagem = this.service.Salvar(dados);

        //return "TBJEmpresaGrid";
        return "redirect:TBJEmpresaGrid";

    }

    @RequestMapping( value="/TBJEmpresaForm", method=RequestMethod.POST,
params={"Deletar"} )

```

```
public String TBJEmpresaDeletar(Model model, TBJEmpresa pdados) {  
  
    if(pdados.getIdtbjempresa() > 0) {  
        this.service.Deletar(pdados.getIdtbjempresa());  
    }  
  
    //return "TBJEmpresaGrid";  
    return "redirect:TBJEmpresaGrid";  
  
    }  
  
}
```

\*\*\*\*\*

```
package com.api.exemplosphtml.controller;
```

```
import java.util.ArrayList;
```

```
import java.util.List;
```

```
import org.springframework.beans.factory.annotation.Autowired;
```

```
import org.springframework.web.bind.annotation.DeleteMapping;
```

```
import org.springframework.web.bind.annotation.GetMapping;
```

```
import org.springframework.web.bind.annotation.PostMapping;
```

```
import org.springframework.web.bind.annotation.PutMapping;
```

```
import org.springframework.web.bind.annotation.RequestMapping;
```

```
import org.springframework.web.bind.annotation.RestController;
```

```
import com.api.exemplosphtml.modal.TBJEmpresa;
```

```
import com.api.exemplosphtml.service.TBJEmpresaService;

@RestController
@RequestMapping("/empresarest")
public class EmpresaRestController {

    @Autowired
    private TBJEmpresaService service;

    // @RequestMapping(value = "/pessoa", method = RequestMethod.GET)
    @GetMapping
    public List<TBJEmpresa> Get() {
        List<TBJEmpresa> Listadados = new ArrayList<TBJEmpresa>();
        Listadados = (List<TBJEmpresa>) this.service.ListaTodos();
        return Listadados;
    }

    @PostMapping
    public String Post(TBJEmpresa dados) {
        String smensagem = service.Salvar(dados);
        return smensagem;
    }

    @PutMapping
    public String Put(TBJEmpresa dados) {
        String smensagem = service.Salvar(dados);
        return smensagem;
    }
}
```



```

        @DeleteMapping
public String Delete(Integer pcodigo) {
            String smensagem = service.Deletar(pcodigo);
return smensagem;
}

//@RequestMapping(value = "/pessoa", method = RequestMethod.POST)
//@RequestMapping(value = "/pessoa/{id}", method = RequestMethod.PUT)
//@RequestMapping(value = "/pessoa/{id}", method = RequestMethod.DELETE)

}

*****

package com.api.exemplosphtml.modal;

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

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.Id;
import javax.persistence.Table;
import javax.persistence.Temporal;

```

```
import javax.persistence.TemporalType;

import org.springframework.format.annotation.DateTimeFormat;

@Entity
@Table(name="TBJEmpresa")
public class TBJEmpresa implements Serializable {

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

    @Id @GeneratedValue
    @Column(name="idtbjempresa")
    private int idtbjempresa;

    @Column(name="nome")
    @NotEmpty(message="O nome nao pode esta vazio")
    private String nome;

    @Column(name="datacadastro", nullable=false)
    @Temporal(TemporalType.DATE)
    @DateTimeFormat(pattern="YYYY-MM-dd")
    //@Temporal(TemporalType.DATE)
    //@Temporal(TemporalType.TIMESTAMP)
    private Date datacadastro;

    @Column(name="valor")
    private Double valor;
```

```
@Column(name="ativo")
private String ativo;

public static long getSerialVersionUID() {
    return serialVersionUID;
}

public int getIdtbjempresa() {
    return idtbjempresa;
}

public void setIdtbjempresa(int idtbjempresa) {
    this.idtbjempresa = idtbjempresa;
}

public String getNome() {
    return nome;
}

public void setNome(String nome) {
    this.nome = nome;
}

public Date getDatacadastro() {
    return datacadastro;
}

public void setDatacadastro(Date datacadastro) {
    this.datacadastro = datacadastro;
}
```

```
}

public Double getValor() {
    return valor;
}

public void setValor(Double valor) {
    this.valor = valor;
}

public String getAtivo() {
    return ativo;
}

public void setAtivo(String ativo) {
    this.ativo = ativo;
}
```

```
}
```

```
*****
```

```
package com.api.exemplosphtml.repository;
```

```
import java.util.List;
```

```
import org.springframework.data.jpa.repository.Query;
import org.springframework.data.repository.CrudRepository;
import org.springframework.data.repository.query.Param;
import org.springframework.stereotype.Repository;

import com.api.exemplosphtml.modal.TBJEmpresa;
```

```
@Repository
```

```
public interface TBJEmpresaRepository extends CrudRepository<TBJEmpresa, Integer> {
```

```
    @Query("select s from TBJEmpresa s")
```

```
    List<TBJEmpresa> findByNaoNativeQuery();
```

```
    @Query(value = "SELECT * FROM tbjempresa where nome = (:nome) ", nativeQuery =
true)
```

```
    List<TBJEmpresa> findByNativeQuery(@Param("nome") String nome);
```

```
}
```

```
*****
```

```
package com.api.exemplosphtml.service;
```

```
import java.util.ArrayList;

import java.util.Calendar;

import java.util.List;

import java.util.Optional;

import org.springframework.stereotype.Service;

import com.api.exemplosphtml.modal.TBJEmpresa;

import com.api.exemplosphtml.repository.TBJEmpresaRepository;

@Service

public interface TBJEmpresaService extends TBJEmpresaRepository {

    public default List<TBJEmpresa> ListaTodos(){

        List<TBJEmpresa> ListaRetorno = new ArrayList<TBJEmpresa>();

        ListaRetorno = (List<TBJEmpresa>) this.findAll();

        return ListaRetorno;

    }

    public default String Salvar(TBJEmpresa pdados) {

        try {

            this.save(pdados);

            return "Não foi possível gravar os dados.";

        }catch(Exception e) {

            return e.getMessage();

        }

    }

}
```

```

    }

}

public default String Deletar(int codigo){

    Optional<TBJEmpresa> dadosPesc = this.findByld(codigo);

    try {

        TBJEmpresa dados = new TBJEmpresa();
        if (dadosPesc.isPresent()) {
            dados = dadosPesc.get();
        }

        this.delete(dados);

        return "Registro excluido com sucesso!";

    }catch(Exception e) {
        return e.getMessage();
    }
}

```

```

public default TBJEmpresa BuscarByld(int codigo){

    TBJEmpresa dadosretorno = new TBJEmpresa();
    dadosretorno.setIdtbjempresa(0);
    dadosretorno.setNome("");
}

```

```
java.util.Date date = Calendar.getInstance().getTime();
java.sql.Date sqlDate = new java.sql.Date(date.getTime());
dadosretorno.setDataCadastro(sqlDate);
dadosretorno.setValor(00.00);
dadosretorno.setAtivo("SIM");
```

```
Optional<TBJEmpresa> dadosPesc = this.findByld(codigo);
```

```
try {
```

```
    if (dadosPesc.isPresent()) {
        dadosretorno = dadosPesc.get();
    }
```

```
    return dadosretorno;
```

```
}catch(Exception e) {
    return dadosretorno;
}
```

```
}
```

```
public default List<TBJEmpresa> BuscarSqlNativo(String pNome){
```

```
    List<TBJEmpresa> Lista = new ArrayList<TBJEmpresa>();
```



```
        Lista = findByNativeQuery(pNome);

        return Lista;

    }

}

*****

<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org"
      xmlns:layout="http://www.ultraq.net.nz/web/thymeleaf/layout"
      layout:decorate="Layout">
<head>
    <title>Content page 1</title>
    <link href="#" th:href="@{../static/css/layout.css}" rel="stylesheet"/>
    <link href="#" th:href="@{../static/css/bootstrap.min.css}" rel="stylesheet"/>
    <script src="#" th:src="@{../static/js/jquery.mim.js}" ></script>
    <script src="#" th:src="@{../static/js/bootstrap.min.js}" ></script>

</head>
<body>
    <section layout:fragment="content">

        <div class="container">
            <h3>SEJA BEM VINDO</h3>
```

```
<p></p>
<label th:text="{ldtexibir}" > </label><br>
<div th:text="{ldtexibir}" > </div>
</div>
```

```
</section>
```

```
<footer>
```

```
<p layout:fragment="custom-footer">Rodape da pagina index</p>
```

```
</footer>
```

```
</body>
```

```
</html>
```

```
*****
```

```
<!DOCTYPE html>
```

```
<html xmlns:th="http://www.thymeleaf.org"
```

```
xmlns:layout="http://www.ultraq.net.nz/web/thymeleaf/layout">
```

```
<head>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
```

```
<meta name="description" content="Descricao"/>
```

```
<meta name="author" content="Vanderlei Soares de Oliveira"/>
```

```
<title>Layout pagina</title>
```

```
<link href="#" th:href="@{/css/layout.css}" rel="stylesheet"/>
```

```
<link href="#" th:href="@{/css/bootstrap.min.css}" rel="stylesheet"/>
```

```
<script src="#" th:src="@{/js/jquery.mim.js}" ></script>
```

```
<script src="#" th:src="@{/js/bootstrap.min.js}" ></script>
```

```
<link rel="stylesheet"
```

```
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/css/bootstrap.min.css">
```

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
```

```
<script
```

```
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/js/bootstrap.min.js"></script>
```

```
<script>
    $(document).ready(function() {
        //alert("teste java script");
    });
</script>
```

```
</head>
```

```
<body>
```

```
<header>
```

```
<nav class="navbar navbar-inverse">
    <div class="container-fluid">
        <div class="navbar-header">
            <a class="navbar-brand" href="#"></a>
```

```
</div>
```

```
<ul class="nav navbar-nav">
```

```
<li class="active"><a href="/index">Principal</a></li>
```

```
<li><a th:href="@{/TBJEmpresaGrid}">Empresa</a></li>
```

```
<li><a href="/TBJEmpresaGrid" >Empresa 2</a></li>
```

```
<li><a href="#">Page 3</a></li>
```

```
</ul>
```

```
</div>
```

```
</nav>
```

```
</header>
```

```

<div class="container">
  <section layout:fragment="content">
    <p>Conteudo das paginas</p>
  </section>
  <footer>
    <p layout:fragment="custom-footer">Exibir Rodape das paginas</p>
    <hr>

    <div class="navbar navbar-inverse navbar-fixed-bottom">
  <div class="container">
    <div class="navbar-collapse collapse" id="footer-body">
      <ul class="nav navbar-nav">
        <li><a href="#">Sobre nós</a></li>
        <li><a href="#">Contato</a></li>
        <li><a href="#">Mapa</a></li>
        <li><a href="#">menu 01</a></li>
        <li><a href="#">menu 02</a></li>
        <li><a href="#">menu 01 & amp; menu 01</a></li>
        <li><a href="#"> <label th:text="{ldtexitibir}" > </label> </a></li>
      </ul>
    </div>
    <div class="navbar-header">
      <button type="button" class="navbar-toggle" data-toggle="collapse" data-
target="#footer-body">
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
      </button>
      <ul class="footer-bar-btns visible-xs">
        <li><a href="#" class="btn" title="History"><i class="glyphicon glyphicon-
globe"></i></a></li>

```

```
        <li><a href="#" class="btn" title="Favourites"><i class="glyphicon glyphicon-
briefcase"></i></a></li>

        <li><a href="#" class="btn" title="Subscriptions"><i class="glyphicon glyphicon-
bell"></i></a></li>

    </ul>

</div>
```

```
</div>
</div>
```

```
</footer>
</div>
```

```
</body>
</html>
```

\*\*\*\*\*

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org"
      xmlns:layout="http://www.ultraq.net.nz/web/thymeleaf/layout"
      layout:decorate="Layout">
<head>
<title>Empresa grid</title>
<link href="#" th:href="@{../static/css/layout.css}" rel="stylesheet"/>
<link href="#" th:href="@{../static/css/bootstrap.min.css}" rel="stylesheet"/>
<script src="#" th:src="@{../static/js/jquery.mim.js}" ></script>
<script src="#" th:src="@{../static/js/bootstrap.min.js}" ></script>
```

```

</head>
<body>
  <section layout:fragment="content">

  <div class="context" >
    <h2 class="hello-title">Listagem Empresa</h2>
    <p>
      <a title="Novo" href="TBJEmpresaForm?id=0" class="btn btn btn-default
command-news"><i class="glyphicon glyphicon-new-window"></i></a>
      <a title="Atualizar" href="TBJEmpresaGrid" class="btn btn btn-default
command-news"><i class="glyphicon glyphicon-refresh"></i></a>
    </p>
    <br>
    <table class="table" border=1 >
      <thead>
        <tr>
          <th>Codigo </th>
          <th>Nome </th>
          <th>Dadata Cadastro</th>
          <th>Valor</th>
          <th>Ativo</th>
          <th>Controle</th>
        </tr>
      </thead>
      <tbody>
        <tr th:if="{Listadados.empty}">
          <td colspan="2"> Não foi encontrado os dados </td>
        </tr>
        <tr th:each="item : {Listadados}">
          <td><span th:text="{item.getIdtjempresa()}"> Codigo
</span></td>
          <td><span th:text="{item.getNome()}"> Nome </span></td>

```

```

        <td><span th:text="{item.getDatacadastro()}"> DataCadastro
</span></td>

        <td><span th:text="{item.getValor()}"> Valor </span></td>

        <td><span th:text="{item.getAtivo()}"> Ativo </span></td>

        <td>

                <a title="Editar" th:href="@{'TBJEmpresaForm?id=' +
${item.getIdtbjempresa()}" class="btn btn btn-default btn-sm"><i class="glyphicon glyphicon-
pencil"></i></a>

                <a title="Deletar" th:href="@{'TBJEmpresaForm?id=' +
${item.getIdtbjempresa()}" class="btn btn btn-default btn-sm"><i class="glyphicon glyphicon-
trash"></i></a>

        </td>

</tr>

</tbody>

</table>

</div>

```

```
</section>
```

```
<footer>
```

```
<p layout:fragment="custom-footer">Rodape da pagina grid</p>
```

```
</footer>
```

```
</body>
```

```
</html>
```

```
*****
```

```
<!DOCTYPE html>
```

```
<html xmlns:th="http://www.thymeleaf.org"
```

```
xmlns:layout="http://www.ultraq.net.nz/web/thymeleaf/layout"
```

```
layout:decorate="Layout">
```

```
<head>
```

```
<title>Empresa form</title>
```

```
<link href="#" th:href="@{../static/css/layout.css}" rel="stylesheet"/>
<link href="#" th:href="@{../static/css/bootstrap.min.css}" rel="stylesheet"/>
<script src="#" th:src="@{../static/js/jquery.mim.js}" ></script>
<script src="#" th:src="@{../static/js/bootstrap.min.js}" ></script>
```

```
</head>
```

```
<body>
```

```
<section layout:fragment="content">
```

```
<div class="context" >
```

```
<div class="form-horizontal">
```

```
<h4>Dados da Empresa</h4>
```

```
<hr />
```

```
<form Class="form-horizontal" action="#" th:action="@{/TBJEmpresaForm}"
th:object="${TBJEmpresa}" method="post">
```

```
<div class="form-group">
```

```
<Label class = "control-label col-md-2">Codigo</Label>
```

```
<div class="col-md-2">
```

```
<input type="text" th:field="*{idtbjempresa}" class = "form-control input-
default col-md-2" readonly />
```

```
</div>
```

```
</div>
```

```
<div class="form-group">
```

```
<Label class = "control-label col-md-2">Nome</Label>
```

```
<div class="col-md-2">
```

```
<input type="text" th:field="*{nome}" class = "form-control input-default"
required="required" />
```

```
</div>
```

```
</div>
```



```

<div class="form-group">
  <Label class = "control-label col-md-2">Data Cadastro</Label>
  <div class="col-md-2">
    <input type="date" th:field="**{datacadastro}" class = "form-control input-
default" />
  </div>
</div>
<div class="form-group">
  <Label class = "control-label col-md-2">Valor</Label>
  <div class="col-md-2">
    <input type="number" th:field="**{valor}" class = "form-control input-default"
/>
  </div>
</div>
<div class="form-group">
  <Label class = "control-label col-md-2">Ativo</Label>
  <div class="col-md-2">
    <input type="text" th:field="**{ativo}" class = "form-control input-default" />
  </div>
</div>
<div class="form-group">
  <div class="col-md-offset-2 col-md-10">
    <input type="submit" name="Salvar" value="Salvar" class="btn btn-default" />
    <input type="submit" name="Deletar" value="Deletar" class="btn btn-default"
/>
  </div>
</div>
</form>
</div>

```

```
<div>
    <a title="Voltar" href="#" class="btn btn-default news"><i class="glyphicon glyphicon-backward"></i></a>
</div>

</div>

</section>
<footer>
    <p layout:fragment="custom-footer">Rodape da pagina form</p>
</footer>
</body>
</html>
```

```
*****
*****
*****
```

Spring Security - Autenticação em memória

<https://spring.io/projects/spring-security#learn>

<https://github.com/MichelliBrito/cursospringboot>

procurar WebSecurityConfigurerAdapter

pom.xml

```
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-security</artifactId>
</dependency>
```

criar um pacote security

```
package com.api.SpringSecurity01.security;
```

criar uma classe

```
WebSecurityConfig
```

```
#####
```

```
PARA EXIBIR A TELA DE LOGIN
```

```
#####
```

```
package com.api.SpringSecurity01.security;
```

```
import org.springframework.context.annotation.Configuration;
```

```
import  
org.springframework.security.config.annotation.authentication.builders.AuthenticationManag  
erBuilder;
```

```
import org.springframework.security.config.annotation.web.builders.HttpSecurity;
```

```
import org.springframework.security.config.annotation.web.builders.WebSecurity;
```

```
import org.springframework.security.config.annotation.web.configuration.EnableWebSecurity;
```

```
import  
org.springframework.security.config.annotation.web.configuration.WebSecurityConfigurerAda  
pter;
```

```
import org.springframework.security.web.util.matcher.AntPathRequestMatcher;
```

```
@Configuration
```

```
@EnableWebSecurity
```

```
public class WebSecurityConfig extends WebSecurityConfigurerAdapter {
```

```
    @Override
```

```
    public void configure(WebSecurity web) throws Exception {
```

```
        web.ignoring().antMatchers("/resources/**");
```

```
    }
```

@Override

```
protected void configure(HttpSecurity http) throws Exception {
```

```
    http.authorizeRequests()
```

```
        .antMatchers("/", "/Index").permitAll()
```

```
        .antMatchers("/home").hasAnyRole("ADMIN")
```

```
        .antMatchers("/getEmployees").hasAnyRole("USER", "ADMIN")
```

```
        .antMatchers("/addNewEmployee").hasAnyRole("ADMIN")
```

```
        .anyRequest()
```

```
        .authenticated()
```

```
        //.and().formLogin().loginPage("/login").permitAll()
```

```
        .and().formLogin().loginPage("/login").permitAll()
```

```
.defaultSuccessUrl("/Index").usernameParameter("username").passwordParameter("password")
```

```
        //.and().logout().permitAll()
```

```
        //.loginProcessingUrl("/Index").permitAll();//para exibir a  
pagina de login do spring security
```

```
        .and().logout().logoutRequestMatcher(new  
AntPathRequestMatcher("/logout"));//para encerra a session
```

```
        //.and().exceptionHandling().accessDeniedPage("/404");
```

```
    http.csrf().disable();
```

```
    //http.httpBasic(); //nao pede a senha
```

```
}
```

@Override

```
protected void configure(AuthenticationManagerBuilder auth) throws Exception {
```

```
    auth.inMemoryAuthentication()
```

```
        .withUser("pushpak").password("{noop}12345").roles("ADMIN");
```

```
        // .and()
        // .withUser("admin").password("admin").roles("ADMIN");
    }
}
```

```
}
```

```
#####
```

```
PARA API REST
```

```
#####
```

```
.. PARA NAO EXIBIR HTML NA SOLICITAÇÃO DO REST
```

```
VAI APARECER MENSAGEM QUE ESTA ERRADO A SENHA
```

```
package com.api.SpringSecurity01.security;
```

```
import org.springframework.context.annotation.Configuration;
```

```
import org.springframework.http.HttpMethod;
```

```
import
```

```
org.springframework.security.config.annotation.authentication.builders.AuthenticationManag  
erBuilder;
```

```
import org.springframework.security.config.annotation.web.builders.HttpSecurity;
```

```
import org.springframework.security.config.annotation.web.builders.WebSecurity;
```

```
import org.springframework.security.config.annotation.web.configuration.EnableWebSecurity;
```

```
import
```

```
org.springframework.security.config.annotation.web.configuration.WebSecurityConfigurerAda  
pter;
```

```
import org.springframework.security.web.util.matcher.AntPathRequestMatcher;
```

```
@Configuration
```

```
@EnableWebSecurity
```

```
public class WebSecurityConfig extends WebSecurityConfigurerAdapter {
```

```
@Override
public void configure(WebSecurity web) throws Exception {
    web.ignoring().antMatchers("/resources/**");
}
```

```
@Override
protected void configure(HttpSecurity http) throws Exception {

    http
        //HTTP Basic authentication
        .httpBasic()
        .and()
        .authorizeRequests()
        .antMatchers(HttpMethod.GET, "/empresarest").hasRole("ADMIN")
        .antMatchers(HttpMethod.POST, "/books").hasRole("ADMIN")
        .antMatchers(HttpMethod.PUT, "/books/**").hasRole("ADMIN")
        .antMatchers(HttpMethod.PATCH, "/books/**").hasRole("ADMIN")
        .antMatchers(HttpMethod.DELETE, "/books/**").hasRole("ADMIN")
        .and()
        .csrf().disable()
        .formLogin().disable();
}
```

```
@Override
protected void configure(AuthenticationManagerBuilder auth) throws Exception {
    auth.inMemoryAuthentication()
        .withUser("pushpak").password("{noop}12345").roles("ADMIN");
    // .and()
    // .withUser("admin").password("admin").roles("ADMIN");
}
```

```
}
```

```
*****
```

```
*****
```

```
package com.api.SpringSecurity01;
```

```
import java.util.ArrayList;
```

```
import java.util.List;
```

```
public class Response<T> {
```

```
/*Com está variável podemos receber qualquer objeto*/
```

```
    private T data;
```

```
    private List<T> ListaDados;
```

```
    private List<String> errors;
```

```
    public Response(T data) {
```

```
        this.data = data;
```

```
    }
```

```
    public T getData() {
```

```
        return data;
```

```
    }
```

```
    public void setData(T data) {
```

```
        this.data = data;
```

```
    }
```

```
public List<String> getErrors() {  
    if (this.errors == null) {  
        this.errors = new ArrayList<String>();  
    }  
    return errors;  
}
```

```
public void setErrors(List<String> errors) {  
    this.errors = errors;  
}
```

```
public List<T> getListaDados() {  
    return ListaDados;  
}
```

```
public void setListaDados(List<T> listaDados) {  
    ListaDados = listaDados;  
}
```

```
}
```

```
*****
```

```
*****
```

```
package com.api.SpringSecurity01.controller;
```

```
import java.util.ArrayList;
```

```
import java.util.List;
```



```

import javax.validation.Valid;

import org.springframework.http.ResponseEntity;
import org.springframework.validation.BindingResult;
import org.springframework.web.bind.annotation.DeleteMapping;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.PutMapping;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.RestController;

import com.api.SpringSecurity01.Response;
import com.api.SpringSecurity01.modal.TBJEmpresa;

@RestController
@RequestMapping("/empresarest")
public class EmpresaRestController {

    // @RequestMapping(value = "/pessoa", method = RequestMethod.GET)
    @GetMapping
    // @Consumes(MediaType.APPLICATION_JSON)
    // @Produces(MediaType.APPLICATION_JSON)
    public ResponseEntity<List<TBJEmpresa>> Get() {
        List<TBJEmpresa> Listadados = new ArrayList<TBJEmpresa>();
        TBJEmpresa dados = new TBJEmpresa();
        dados.setIdtbjempresa(0);
        dados.setNome("Vanderlei Soares de Oliveira v");
        //dados.setDatacadastro(Comandosuteis.Data_DataAtual());
    }
}

```

```
dados.setValor(10.00);
dados.setAtivo("SIM");
Listadados.add(dados);
```

```
dados = new TBJEmpresa();
dados.setIIdtbjempresa(1);
dados.setNome("Jussara");
//dados.setDatacadastro(Comandosuteis.Data_DataAtual());
dados.setValor(10.00);
dados.setAtivo("SIM");
Listadados.add(dados);
```

```
dados = new TBJEmpresa();
dados.setIIdtbjempresa(2);
dados.setNome("Davi");
//dados.setDatacadastro(Comandosuteis.Data_DataAtual());
dados.setValor(10.00);
dados.setAtivo("SIM");
Listadados.add(dados);
```

```
dados = new TBJEmpresa();
dados.setIIdtbjempresa(3);
dados.setNome("Kayo");
//dados.setDatacadastro(Comandosuteis.Data_DataAtual());
dados.setValor(10.00);
dados.setAtivo("SIM");
Listadados.add(dados);
```

```
return ResponseEntity.ok(Listadados);
}
```

```

//@GetMapping("/{id}")
@RequestMapping(value = "/empresarest2/{id}", method = RequestMethod.GET)
public ResponseEntity<TBJEmpresa> GetById(@PathVariable String id) {

    TBJEmpresa dados = new TBJEmpresa();
    dados.setIdtbjempresa(0);
    dados.setNome("Vanderlei Soares de Oliveira v");
    //dados.setDatacadastro(Comandosuteis.Data_DataAtual());
    dados.setValor(10.00);
    dados.setAtivo("SIM");

    return ResponseEntity.ok(dados);
}

```

```

@PostMapping
public ResponseEntity<Response<TBJEmpresa>> cadastrar(@Valid @RequestBody
TBJEmpresa empresa,
    BindingResult result) {

    Response<TBJEmpresa> response = new Response<TBJEmpresa>(null);

    if (result.hasErrors()) {
        result.getAllErrors().forEach(error ->
response.getErrors().add(error.getDefaultMessage()));
        return ResponseEntity.badRequest().body(response);
    }

    empresa.setIdtbjempresa(1);
    response.setData(empresa);
    return ResponseEntity.ok(response);
}

```

```

}

//public ResponseEntity Post(TBJEmpresa dados) {

    @PutMapping
    public String Put(TBJEmpresa dados) {
        String smensagem = "";
        return smensagem;
    }

    @DeleteMapping("/{id}")
    public ResponseEntity<Response<Integer>> Delete(@PathVariable Integer id) {
        String smensagem = "";
        return ResponseEntity.ok(new Response<Integer>(1));
    }

    //@RequestMapping(value = "/pessoa", method = RequestMethod.POST)
    //@RequestMapping(value = "/pessoa/{id}", method = RequestMethod.PUT)
    //@RequestMapping(value = "/pessoa/{id}", method = RequestMethod.DELETE)

}

#####
PARA API REST - USUARIOS E SENHA BANCO DE DADOS
#####

package com.api.exemplosphtml.security;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.context.annotation.Configuration;

```

```
import org.springframework.http.HttpMethod;

import
org.springframework.security.config.annotation.authentication.builders.AuthenticationManag
erBuilder;

import org.springframework.security.config.annotation.web.builders.HttpSecurity;

import org.springframework.security.config.annotation.web.builders.WebSecurity;

import org.springframework.security.config.annotation.web.configuration.EnableWebSecurity;

import
org.springframework.security.config.annotation.web.configuration.WebSecurityConfigurerAda
pter;

import org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;
```

```
@Configuration
```

```
@EnableWebSecurity
```

```
public class WebSecurityConfig extends WebSecurityConfigurerAdapter {
```

```
    @Autowired
```

```
    private ImplementsUserService userService;
```

```
    //CONTROLE PARA EXIBIR O LOGIN MAS NAO SERVE PARA O API REST
```

```
    @Override
```

```
    protected void configure(HttpSecurity http) throws Exception {
```

```
        http
```

```
        //HTTP Basic authentication
```

```
        .httpBasic()
```

```
        .and()
```

```
        .authorizeRequests()
```

```
        .antMatchers(HttpMethod.GET, "/TBJEmpresaGrid").hasRole("ADMIN");//no banco de
dados tem de colocar ROLE_ADMIN tanto no role quanto no usuario_role
```

```
        .antMatchers(HttpMethod.GET, "/empresarest").hasRole("ADMIN")
```

```
        .antMatchers(HttpMethod.POST, "/books").hasRole("ADMIN")
```

```
.antMatchers(HttpMethod.PUT, "/books/**").hasRole("ADMIN")
.antMatchers(HttpMethod.PATCH, "/books/**").hasRole("ADMIN")
.antMatchers(HttpMethod.DELETE, "/books/**").hasRole("ADMIN")
.and()
.csrf().disable()
.formLogin().disable();
}
```

```
@Override
```

```
protected void configure(AuthenticationManagerBuilder auth) throws Exception{
    auth.userDetailsService(userDetailsService)
        .passwordEncoder(new BCryptPasswordEncoder());
}
```

```
@Override
```

```
public void configure(WebSecurity web) throws Exception{
    web.ignoring().antMatchers("/js/**", "/css/**");
}
```

```
}
```

```
*****
```

```
*****
```

```
package com.api.exemplosphtml.security;
```

```
import javax.transaction.Transactional;
```

```

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.security.core.userdetails.User;
import org.springframework.security.core.userdetails.UserDetails;
import org.springframework.security.core.userdetails.UserDetailsService;
import org.springframework.security.core.userdetails.UsernameNotFoundException;
import org.springframework.stereotype.Repository;

import com.api.exemplosphtml.modal.Usuario;
import com.api.exemplosphtml.repository.UsuarioRepository;

@Repository
@Transactional
public class ImplementsUserDetailsService implements UserDetailsService{

    @Autowired
    private UsuarioRepository ur;

    @Override
    public UserDetails loadUserByUsername(String login) throws
UsernameNotFoundException {
        Usuario usuario = ur.findByLogin(login);

        if(usuario == null){
            throw new UsernameNotFoundException("Usuario não encontrado!");
        }

        return new User(usuario.getUsername(), usuario.getPassword(), true, true,
true, true, usuario.getAuthorities());
    }
}

```

```
}
```

```
*****
```

```
*****
```

```
package com.api.exemplosphtml.repository;
```

```
import org.springframework.data.repository.CrudRepository;
```

```
import com.api.exemplosphtml.modal.Usuario;
```

```
public interface UsuarioRepository extends CrudRepository<Usuario, String>{
```

```
    Usuario findByLogin(String login);
```

```
}
```

```
*****
```

```
*****
```

```
package com.api.exemplosphtml.modal;
```

```
import java.io.Serializable;
```

```
import java.util.Collection;
```

```
import java.util.List;
```

```
import javax.persistence.Entity;
```

```
import javax.persistence.Id;
```



```

import javax.persistence.JoinColumn;
import javax.persistence.JoinTable;
import javax.persistence.ManyToMany;

import org.hibernate.validator.constraints.NotEmpty;
import org.springframework.security.core.GrantedAuthority;
import org.springframework.security.core.userdetails.UserDetails;

@SuppressWarnings("deprecation")
@Entity
public class Usuario implements UserDetails, Serializable{

private static final long serialVersionUID = 1L;

@Id
private String login;

private String nomeCompleto;

@NotEmpty
private String senha;

@ManyToMany
@JoinTable(
    name = "usuarios_roles",
    joinColumns = @JoinColumn(
        name = "usuario_id", referencedColumnName = "login"),
    inverseJoinColumns = @JoinColumn(
        name = "role_id", referencedColumnName = "nomeRole"))
private List<Role> roles;

```

```
public String getLogin() {  
    return login;  
}
```

```
public void setLogin(String login) {  
    this.login = login;  
}
```

```
public String getNomeCompleto() {  
    return nomeCompleto;  
}
```

```
public void setNomeCompleto(String nomeCompleto) {  
    this.nomeCompleto = nomeCompleto;  
}
```

```
public String getSenha() {  
    return senha;  
}
```

```
public void setSenha(String senha) {  
    this.senha = senha;  
}
```

```
public List<Role> getRoles() {  
    return roles;  
}
```

```
public void setRoles(List<Role> roles) {  
    this.roles = roles;  
}
```

```
@Override
public Collection<? extends GrantedAuthority> getAuthorities() {
    // TODO Auto-generated method stub
    return (Collection<? extends GrantedAuthority>) this.roles;
}
```

```
@Override
public String getPassword() {
    // TODO Auto-generated method stub
    return this.senha;
}
```

```
@Override
public String getUsername() {
    // TODO Auto-generated method stub
    return this.login;
}
```

```
@Override
public boolean isAccountNonExpired() {
    // TODO Auto-generated method stub
    return true;
}
```

```
@Override
public boolean isAccountNonLocked() {
    // TODO Auto-generated method stub
    return true;
}
```

```
@Override
public boolean isCredentialsNonExpired() {
    // TODO Auto-generated method stub
    return true;
}
```

```
@Override
public boolean isEnabled() {
    // TODO Auto-generated method stub
    return true;
}
```

```
}
```

```
*****
```

```
*****
```

```
package com.api.exemplosphtml.modal;
```

```
import java.util.List;
```

```
import javax.persistence.Entity;
```

```
import javax.persistence.Id;
```

```
import javax.persistence.ManyToMany;
```

```
//import javax.persistence.ManyToOne;
```

```
//import javax.persistence.OneToMany;
```

```
import org.springframework.security.core.GrantedAuthority;
```

```
@Entity
```

```
public class Role implements GrantedAuthority{
```

```
    /**
```

```
     *
```

```
    */
```

```
    private static final long serialVersionUID = -2350144649820119477L;
```

```
    @Id
```

```
    private String nomeRole;
```

```
    @ManyToMany(mappedBy = "roles")
```

```
    private List<Usuario> usuarios;
```

```
    public String getNomeRole() {
```

```
        return nomeRole;
```

```
    }
```

```
    public void setNomeRole(String nomeRole) {
```

```
        this.nomeRole = nomeRole;
```

```
    }
```

```
    public List<Usuario> getUsuarios() {
```

```
        return usuarios;
```

```
    }
```

```
    public void setUsuarios(List<Usuario> usuarios) {
```

```
        this.usuarios = usuarios;
```

```
    }
```

```
@Override
public String getAuthority() {
    // TODO Auto-generated method stub
    return this.nomeRole;
}
}
```

```
*****
```

```
*****
```

```
<!DOCTYPE html>
```

```
<html xmlns:th="http://www.thymeleaf.org"
```

```
xmlns:layout="http://www.ultraq.net.nz/web/thymeleaf/layout"
```

```
layout:decorate="Layout">
```

```
<head>
```

```
<meta charset="ISO-8859-1">
```

```
<title>Insert title here</title>
```

```
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
```

```
</head>
```

```
<body>
```

```
<div class="container">
```

```
<form method="POST" action="/login" class="form-signin">
```

```
<h2 class="form-heading">Log in</h2>
```

```
<div class="form-group ">
  <span></span>
  <input id="username" name="username" type="text" class="form-control"
placeholder="Username" autofocus="true"/>
  <input id="password" name="password" type="password" class="form-control"
placeholder="Password"/>
  <span></span>
  <label th:text="{errorMsg}" > </label><br>
  <label th:text="{msg}" > </label><br>
  <button class="btn btn-lg btn-primary btn-block" type="submit">Log In</button>
</div>

</form>

</div>

<!-- /container -->
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"></script>
<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script></body
>
</html>

*****

*****

#####

#####

*****

*****
```

na pasta templates

criar um arquivo html

o name e id dos input tem de ser o mesmo

```
<!DOCTYPE html>
```

```
<html xmlns:th="http://www.thymeleaf.org"
```

```
xmlns:layout="http://www.ultraq.net.nz/web/thymeleaf/layout"
```

```
layout:decorate="Layout">
```

```
<head>
```

```
<meta charset="ISO-8859-1">
```

```
<title>Insert title here</title>
```

```
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
```

```
</head>
```

```
<body>
```

```
<div class="container">
```

```
<form method="POST" action="/login" class="form-signin">
```

```
<h2 class="form-heading">Log in</h2>
```

```
<div class="form-group ">
```

```
<span></span>
```

```
<input id="username" name="username" type="text" class="form-control"
placeholder="Username" autofocus="true"/>
```

```
<input id="password" name="password" type="password" class="form-control"
placeholder="Password"/>
```

```
<span></span>
```

```
<label th:text="{errorMsg}" > </label><br>
```

```
<label th:text="{msg}" > </label><br>
```

```
<button class="btn btn-lg btn-primary btn-block" type="submit">Log In</button>
```



```
</div>
```

```
</form>
```

```
</div>
```

```
<!-- /container -->
```

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"></script>
```

```
<script  
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script></body  
>
```

```
</html>
```

```
*****
```

```
*****
```

```
package com.api.SpringSecurity01.controller;
```

```
import org.springframework.stereotype.Controller;
```

```
import org.springframework.ui.Model;
```

```
import org.springframework.web.bind.annotation.RequestMapping;
```

```
import org.springframework.web.bind.annotation.RequestMethod;
```

```
@Controller
```

```
public class HomeController {
```

```
    //@RequestMapping(value="/home", method= RequestMethod.GET)
```

```
        @RequestMapping(value = {"", "/", "Index", "Layout"})
```

```
        public String Index(Model model ) {
```

```
            return "Index";
```

```
        }
```

```
@RequestMapping(value="/home", method= RequestMethod.GET)
public String Home(Model model ) {

    return "home";

}

//@RequestMapping(value = "/login", method = RequestMethod.GET)
@RequestMapping(value = "/login", method = { RequestMethod.GET})
public String login(Model model, String error, String logout) {
    if (error != null)
        model.addAttribute("errorMsg", "Your username and password are invalid.");

    if (logout != null)
        model.addAttribute("msg", "You have been logged out successfully.");

    return "login";
}
```

```
}
```

```
*****
```

```
*****
```

```
no arquivo
```

```
application.properties
```

```
server.port = 8082
```

```
#=====
```

```
# = Thymeleaf configurations
```

```
#=====
```

```
#PARA USAR O JSP TEM DE MANTER ESTAS LINHAS ABAIXO COMENTADAS
```

```
spring.thymeleaf.check-template-location=true
```

```
spring.thymeleaf.prefix=classpath:/templates/
```

```
spring.thymeleaf.suffix=.html
```

```
spring.thymeleaf.servlet.content-type=text/html
```

```
spring.thymeleaf.cache=false
```

```
server.servlet.context-path=/
```

```
*****
```

```
*****
```

```
*****
```

```
*****
```

Spring Security - banco de dados - mysql

\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*

=====  
=====  
=====  
=====

JAVA DYNAMIC WEB PROJECT  
criar o projeto Dynamic Web Project

webapp  
WEB-INF  
web.xml

\*\*\*\*\*

## WEB.XML

\*\*\*\*\*

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://xmlns.jcp.org/xml/ns/javaee"
xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd" id="WebApp_ID" version="3.1">
  <display-name>Exemplo02</display-name>
```

```
<jsp-config>
```

```
  <taglib>
```

```
    <taglib-uri>http://java.sun.com/jstl/fmt</taglib-uri>
```

```
    <taglib-location>/WEB-INF/fmt.tld</taglib-location>
```

```
  </taglib>
```

```
  <taglib>
```

```
    <taglib-uri>http://java.sun.com/jstl/fmt-rt</taglib-uri>
```

```
    <taglib-location>/WEB-INF/fmt-rt.tld</taglib-location>
```

```
  </taglib>
```

```
  <taglib>
```

```
    <taglib-uri>http://java.sun.com/jstl/core</taglib-uri>
```

```
    <taglib-location>/WEB-INF/c.tld</taglib-location>
```

```
  </taglib>
```

```
  <taglib>
```

```
    <taglib-uri>http://java.sun.com/jstl/core-rt</taglib-uri>
```

```
    <taglib-location>/WEB-INF/c-rt.tld</taglib-location>
```

```
  </taglib>
```

```
<taglib>
  <taglib-uri>http://java.sun.com/jstl/sql</taglib-uri>
  <taglib-location>/WEB-INF/sql.tld</taglib-location>
</taglib>
```

```
<taglib>
  <taglib-uri>http://java.sun.com/jstl/sql-rt</taglib-uri>
  <taglib-location>/WEB-INF/sql-rt.tld</taglib-location>
</taglib>
```

```
<taglib>
  <taglib-uri>http://java.sun.com/jstl/x</taglib-uri>
  <taglib-location>/WEB-INF/x.tld</taglib-location>
</taglib>
```

```
<taglib>
  <taglib-uri>http://java.sun.com/jstl/x-rt</taglib-uri>
  <taglib-location>/WEB-INF/x-rt.tld</taglib-location>
</taglib>
```

```
</jsp-config>
```

```
<welcome-file-list>
  <welcome-file>index.html</welcome-file>
  <welcome-file>index.htm</welcome-file>
  <welcome-file>index.jsp</welcome-file>
  <welcome-file>default.html</welcome-file>
  <welcome-file>default.htm</welcome-file>
  <welcome-file>default.jsp</welcome-file>
</welcome-file-list>
```

```
</web-app>
```

\*\*\*\*\*

## GERAR O WAR

\*\*\*\*\*

clique com botão direito do mouse sobre o projeto

escolher opção Export -> Export

opção Web --> WAR file

Next

marcar:

Optimize for a sp....

pegar o apache instalado

Overwrite existing file

escolher o Destination do apache instalado

C:\Program Files\Apache Software Foundation\Tomcat 8.5\webapps\Exemplo02.war

<http://localhost:8080/Exemplo02/index>

\*\*\*\*\*

## APACHE TOMCAT

\*\*\*\*\*

abra o browser

<http://localhost:8080/>

clique no botão Manager App

automaticamente o manager vai

descompactar o war gerando a pasta

e os arquivos de acordo com a estrutura

gerando um link para teste no lado esquerdo

\*\*\*\*\*

## CONFIGURAR PACOTES

\*\*\*\*\*

Java Resources

src

com.web.Controller

com.web.Dao

com.web.Model

\*\*\*\*\*

## JSP - SERVLET

\*\*\*\*\*

fica dentro dos pacotes

clicar com botão direito sobre o pacote

New -> Servlet

Next

Url mappings pode adicionar variacoes

parameters pode passar parametros

Next

selecione

doGet

doPost

doPut

doDelete

doInit

doDestroy

Finish



\*\*\*\*\*

JSTL - PASSAR DADOS SERVLET PARA JSP

\*\*\*\*\*

001

-----

no arquivo Servlet

```
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws  
ServletException, IOException {
```

```
    // TODO Auto-generated method stub
```

```
    request.setAttribute("name", "Vanderlei Soares de Olivera");
```

```
    request.getRequestDispatcher("index.jsp").forward(request, response);
```

```
}
```

no arquivo jsp

```
<%@ page import="java.sql.ResultSet" %>
```

```
<h3> My name is ${name}</h3>
```

```
<!-- OR -->
```

```
<h3> My name is request.getAttribute("name")</h3>
```

002

-----

## Redirect

```
response.sendRedirect( "home.jsp?name=Hussein Terek" );
```

```
<h3> My name is <%= request.getParameter("name") %></h3>
```

003

-----

```
<%@ page import="com.library.controller.*"%>
```

```
<%@ page import="com.library.dao.*" %>
```

```
<%@ page import="java.util.*" %>
```

```
<%@ page import="java.lang.*" %>
```

```
<%@ page import="java.util.Date" %>
```

```
<%@page import="java.text.SimpleDateFormat"%>
```

```
<%@page import="java.util.Date"%>
```

```
<%@page import="java.util.Locale"%>
```

```
<html>
```

```
<%
```

```
String stringDate = "Fri May 13 2011 19:59:09 GMT 0530";
```

```
Date stringDate1 = new SimpleDateFormat("EEE MMM dd yyyy HH:mm:ss Z",  
Locale.ENGLISH).parse(stringDate);
```

```
String stringDate2 = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss").format(stringDate1);
```

```
out.println(stringDate2);
```

%>

<br/>

004

-----

```
Date data = new Date();

//SimpleDateFormat sm = new SimpleDateFormat("dd-mm-yyyy");
DateFormat format = new SimpleDateFormat("dd-mm-yyyy", Locale.ENGLISH);
String strDate = format.format(data);
Date dtAtual = null;
try {
    dtAtual = format.parse(strDate);
    } catch (ParseException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
aluno.setDataCadastro(dtAtual);
```

005

-----

Usando Taglibs

tem de fazer download dos arquivos

jstl-1.2.jar

standard-1.1.2.jar

e colocar na pasta

WEB-INF

lib

para funcionar

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"  
    pageEncoding="ISO-8859-1"%>
```

```
<!DOCTYPE html>
```

```
<%@ taglib prefix='c' uri='http://java.sun.com/jsp/jstl/core' %>
```

```
<%@page import="java.text.SimpleDateFormat"%>
```

```
<%@page import="java.util.Date"%>
```

```
<%@page import="java.util.Locale"%>
```

```
<%@page import="java.util.HashMap"%>
```

```
<%@page import="java.util.Locale"%>
```

```
<%@page import="java.util.Map"%>
```

```
<%@page import="java.util.Arrays"%>
```

```
<%@page import="java.util.List"%>
```

```
<%@ page import="java.util.ArrayList"%>
```

```
<%
```

```
    ArrayList<String> list = new ArrayList<String>();
```

```
    list.add("A");
```

```
    list.add("B");
```

```
    list.add("C");
```

```
    request.setAttribute("alist", list);
```

```
%>
```

```
<%
    Map<String,String> map = new HashMap<String,String>();

    map.put("key1", "Mitali");
    map.put("key2", "Vicky");
    map.put("key3", "Neil Raj");
    map.put("key4", "Aayush");
    map.put("key5", "Kajol");
    map.put("key3", "Anshu");
    map.put("key3", "Ishant");
    map.put("key3", "Ananya");

    request.setAttribute("MyMap", map);
%>

<html>
<head>
<meta charset="ISO-8859-1">
<title>Exemplo 002</title>

</head>
<body>

<h3>Exemplo de DYNAMIC WEB PROJECT</H3>
<p></p>
<ul>
<li><a href="paginas/Home.jsp">Home</a></li>
<li><a href="paginas/Aluno/AlunoIndex.jsp">AlunoIndex</a></li>
</ul>
```

```

<%@ page import="java.sql.ResultSet" %>
----- <br/>
<h3> My name is ${name}</h3>
<!-- OR -->
----- <br/>
<%
    String message = (String) request.getAttribute("message");
    out.println("Atributo exibir: "+ message);
%>
<br>
----- <br/>
<h3> dados do objeto aluno: </h3>
Id: ${alunoexib.getId()} <br/>
Nome: ${alunoexib.getNome()} <br/>
data: ${alunoexib.getDataCadastro()} <br/>
valor: ${alunoexib.getValor()} <br/>
----- <br/>
<%
String stringDate = "Fri May 13 2011 19:59:09 GMT 0530";
Date stringDate1 = new SimpleDateFormat("EEE MMM dd yyyy HH:mm:ss Z",
Locale.ENGLISH).parse(stringDate);
String stringDate2 = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss").format(stringDate1);

out.println(stringDate2);
%>
<br/>
----- <br/>
<%
    for(int j=1;j<=10;j++)
        {

```

```
%>
```

```
    <%=j%><br/>
```

```
<%
```

```
    }
```

```
%>
```

```
----- <br/>
```

```
forEach<br/>
```

```
<c:forEach var="i" begin="1" end="10" step="1">
```

```
    <c:out value="$" />
```

```
    <br />
```

```
</c:forEach>
```

```
----- <br/>
```

```
<%
```

```
int num = 4;
```

```
%>
```

```
<c:set value="34" var="num"/>
```

```
<br/>
```

```
<c:out value="{num % 2 eq 0 ? 'even': 'odd'}"/>
```

```
----- <br/>
```

```
<%
```

```
    String sNome = "Vanderlei";
```

```
%>
```

```
<c:if test="{sNome == 'Vanderlei'}"> <c:out value="é igual pode exibir" /> </c:if>
```

```
----- <br/>
```

```
<c:out value="exibir vanderlei" />
```

```
<br/>
```

```
----- <br/>
```

```
<c:if test = "${sNome != 'TR1'}">
```

```
  //process
```

```
  <c:out value="quando for diferente exibir vanderlei" /> <br/>
```

```
</c:if>
```

```
----- <br/>
```

```
<c:choose>
```

```
  <c:when test="${sNome == 'Vanderlei'}">
```

```
    pizza.
```

```
  <br />
```

```
</c:when>
```

```
<c:otherwise>
```

```
  pizzas2.
```

```
  <br />
```

```
</c:otherwise>
```

```
</c:choose>
```

```
----- <br/>
```

```
<input type="text" name="email" value="<c:out value="${sNome}"/>" />
```

```
<br />
```

```
----- <br/>
```



```
<select name="day">
  <c:forEach begin="1" end="31" var="day">
    <option><c:out value="\${day}"/></option>
  </c:forEach>
</select>
<br />
<select name="year">
  <c:forEach begin="1930" end="2003" var="year">
    <option><c:out value="\${year}"/></option>
  </c:forEach>
</select>
<br />
```

----- <br/>

<br />

<br />

```
<select id="food" name="fooditems">
  <c:forEach items="\${foods}" var="food">
    <option value="\${food.key}">
      \${food.value}
    </option>
  </c:forEach>
</select>
<br />
<br />
```

----- <br/>

<%

```
String[] columnHeaders = {"Banana", "Apple", "Carrot", "Orange", "Lychee", "Permisson"};
request.setAttribute("columnHeaders", columnHeaders);
```

```

%>
<br />
<table>
    <tr>
        <td></td>
    </tr>
<c:forEach var="columnHeader" items="{columnHeaders}">
    <td>
        <c:out value="{columnHeader}" /><br />
    </td>
</c:forEach>
</table>
<br />

```

----- <br/>

```

<%
    List<String> Listafrutas = Arrays.asList("Banana", "Apple", "Carrot", "Orange",
"Lychee", "Permisson");
%>
<br />
<c:forEach var="columnHeader" items="{Listafrutas}">
    <td>
        <c:out value="{columnHeader}" />
    </td>
</c:forEach>
<br />
----- <br/>

```

```
----- <br/>
```

```
<br />
```

```
<br />
```

```
<style>
```

```
table, th, td {
```

```
border: 1px solid black;
```

```
}
```

```
</style>
```

```
<table>
```

```
<tr>
```

```
<td>id</td>
```

```
<td>nome</td>
```

```
</tr>
```

```
<c:forEach var="p" items="${ListaAluno}">
```

```
<tr>
```

```
<td><c:out value="{p.id}" /></td>
```

```
<td><c:out value="{p.nome}" /></td>
```

```
</tr>
```

```
</c:forEach>
```

```
</table>
```

```
<br />
```

```
----- <br/>
```

```
<c:forEach var="counter" begin="1" end="10">
```

```
<c:out value="{counter}" />
```

```
</c:forEach>
```

```
----- <br/>
```

```
<br />
```

```
<%
```

```
String[][] employeeData = {
```

```
 {"101", "Mukesh", "123", "Chennai"},
```

```
 {"102", "Ravi", "258", "Gurgaon"},
```

```
 {"103", "Amit", "695", "Gurgaon"},
```

```
 {"104", "Vicky", "897", "Noida"},
```

```
 {"105", "Golu", "745", "Patna"}
```

```
};
```

```
request.setAttribute("empdata", employeeData);
```

```
%>
```

```
Employee Details
```

```
<br />
```

```
<table border="1">
```

```
<tr>
```

```
<th>EMP ID</th>
```

```
<th>Name</th>
```

```
<th>Phone</th>
```

```
<th>City</th>
```

```
</tr>
```

```
<c:forEach var="employee" items="{empdata}">
```

```
<tr>
```

```
<td>${employee[0]}</td>
```

```
<td>${employee[1]}</td>
```

```
<td align="center">${employee[2]}</td>
```

```
<td align="center">${employee[3]}</td>
```

```
</tr>
```

```
</c:forEach>
```

```
</table>
```

```
<br />
```

```
----- <br/>
```

```
<br />
```

```
<c:forEach items="{alist}" var="listItem">
```

```
    ${listItem} <br />
```

```
</c:forEach>
```

```
<br />
```

```
----- <br/>
```

```
<br />
```

```
<c:forEach items="{MyMap}" var="mapItem">
```

```
    ${mapItem.key} ${mapItem.value} <br/>
```

```
</c:forEach>
```

```
<br />
```

```
----- <br/>
```

```
<br />
```

```
<%
```

```
    ArrayList<HashMap<String,String>> listOfMap = new ArrayList<HashMap<String,String>>();
```

```
HashMap<String,String> map2;
```

```
for(int i=0;i<3;i++){
```

```
    map2 = new HashMap<String, String>();
```

```
    map2.put("key1", "value1"+i);
```

```
    map2.put("key2", "value2"+i);
```

```

listOfMap.add(map2);
}

    request.setAttribute("listOfMap", listOfMap);
%>
<c:forEach items="{listOfMap}" var="maps">
    <c:forEach items="{maps}" var="mapItem">
        ${mapItem.key} ${mapItem.value} <br />
    </c:forEach>
</c:forEach>
<br />
----- <br />
<br />
<%
    HashMap<String,ArrayList<String>> mapOfList = new HashMap<String,ArrayList<String>>();

ArrayList<String> list2;

for(int i=0;i<2;i++){
    list2 = new ArrayList<String>();

    list2.add("Mitali");
    list2.add("Vicky");
    list2.add("Neil");
    list2.add("Aayush");
    list2.add("Kajol");
    list2.add("Anshu");
    list2.add("Ishant");
list2.add("Ananya");

mapOfList.put("C"+i,list2);

```

```
}
```

```
request.setAttribute("mapOfList", mapOfList);
```

```
%>
```

```
<c:forEach items="${mapOfList}" var="list">
```

```
    <c:forEach items="${list.value}" var="listItem">
```

```
        ${listItem} <br/>
```

```
    </c:forEach>
```

```
</c:forEach>
```

```
<br />
```

```
</body>
```

```
</html>
```

```
=====
```

```
=====
```

```
=====
```

```
=====
```

para gerar o projeto SPRING

<https://start.spring.io/>

INSTALANDO A IDE PARA TRABALHAR NO PROJETO

<https://spring.io/tools>

Server=179.188.16.6;Database=sindsmart2;Uid=sindsmart2;Pwd=odavi240175

Os componentes-base do Spring Boot

O Spring Boot possui 3 principais componentes que permitem toda essa “magia” acontecer:

- Spring Boot Starter;
- //Spring Boot AutoConfigurator;
- //Spring Boot Actuator.

O Spring Boot Starter é o coração de uma aplicação Spring. Sua função principal é combinar as várias dependências advindas de um projeto Spring Boot em uma única dependência, retirando-se a necessidade de configuração de múltiplas dependências no Maven ou no Gradle.

Por exemplo: se quisermos criar uma aplicação Spring web “convencional”, nós precisaríamos de, pelo menos, as seguintes dependências:

- Spring Core;
- Spring Web;
- Spring Web MVC;
- Servlet API.

\*\*\*\*\*



NOVO PROJETO

\*\*\*\*\*

File -> Spring Starter project

Type = Maven

Packaging = war

java Version = 8

Group = com.api

...

Next

Spring Boot Version: 2.2.0(snapshot)

MySQL Driver

Spring Data JPA

Spring Web Starter

//testar

spring boot devtools

spring data jpa

mysql driver

spring security

spring web starter

Finish

\*\*\*\*\*

ABRIR ARQUIVO

SPRINGSMARTBDAPPLICATION.JAVA

\*\*\*\*\*

package com.api.springsmartbd2;

```
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.EnableAutoConfiguration;
import org.springframework.boot.builder.SpringApplicationBuilder;
import org.springframework.boot.web.support.SpringBootServletInitializer;
import org.springframework.context.annotation.ComponentScan;
import org.springframework.context.annotation.Configuration;

//@SpringBootApplication
@Configuration
@ComponentScan(basePackages = {"com.api.springsmartbd2"})

//@EntityScan(basePackages = { "com.devmedia.mvc.entity" })
//@EnableJpaRepositories(basePackages = { "com.devmedia.mvc.repository" })
//@ComponentScan(basePackages = {"com.devmedia.mvc.controller"})

@EnableAutoConfiguration
public class Springsmartbd2Application extends SpringBootServletInitializer {

    @Override
    protected SpringApplicationBuilder configure(SpringApplicationBuilder application) {
        return application.sources(Springsmartbd2Application.class);
    }

    public static void main(String[] args) {
        SpringApplication.run(Springsmartbd2Application.class, args);
    }
}
```

\*\*\*\*\*

APPLICATION.PROPERTIES

\*\*\*\*\*

#spring.resources.static-locations=classpath:/custom/

#PORTA ONDE VAI SER EXECUTADO O NOSSO TOMCAT

server.port = 8090

#INFORMAÇÕES PARA CONEXÃO COM O BANCO DE DADOS MYSQL

spring.datasource.url=jdbc:mysql://179.188.16.6:3306/sindsmart2

spring.datasource.username=sindsmart2

spring.datasource.password=odavi240175

spring.datasource.driver-class-name=com.mysql.jdbc.Driver

#PARA CRIAR A TABELA NO BANCO BASTA COLOCAR update cria as tabelas

spring.jpa.hibernate.ddl-auto=update

# LazyInitializationException = spring.jpa.open-in-view=false

spring.jpa.open-in-view=false

spring.jpa.generate-ddl=true

spring.jpa.show-sql=true

spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL5Dialect

spring.jpa.properties.hibernate.format\_sql=true

#CAMINHO DOS ARQUIVOS DE EXIBIÇÕES

#spring.mvc.view.prefix=/WEB-INF/jsp/

#spring.mvc.view.suffix=.jsp

#=====

# = Thymeleaf configurations

#=====

spring.thymeleaf.check-template-location=true

spring.thymeleaf.prefix=classpath:/templates/

```
spring.thymeleaf.suffix=.html
spring.thymeleaf.content-type=text/html
spring.thymeleaf.cache=false
server.contextPath=/
#=====
```

```
#For detailed logging during development
logging.level.org.springframework=TRACE
logging.level.com=TRACE
```

```
*****
```

```
POM.XML
```

```
*****
```

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
  http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>1.5.21.RELEASE</version>
    <relativePath/> <!-- lookup parent from repository -->
  </parent>
  <groupId>com.api</groupId>
  <artifactId>springmartbd2</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <packaging>war</packaging>
  <name>springmartbd2</name>
  <description>Demo mysql</description>
```

```
<properties>
    <java.version>1.8</java.version>
```

```
</properties>
```

```
<dependencies>
```

```
    <dependency>
```

```
        <groupId>org.hibernate</groupId>
```

```
        <artifactId>hibernate-core</artifactId>
```

```
    </dependency>
```

```
<dependency>
```

```
    <groupId>org.hibernate</groupId>
```

```
    <artifactId>hibernate-entitymanager</artifactId>
```

```
    </dependency>
```

```
    <dependency>
```

```
        <groupId>org.springframework.boot</groupId>
```

```
        <artifactId>spring-boot-starter-data-jpa</artifactId>
```

```
    </dependency>
```

```
    <dependency>
```

```
        <groupId>org.springframework.boot</groupId>
```

```
        <artifactId>spring-boot-starter-web</artifactId>
```

```
    </dependency>
```

```
    <dependency>
```

```
        <groupId>org.springframework.boot</groupId>
```

```
        <artifactId>spring-boot-devtools</artifactId>
```

```
        <scope>runtime</scope>
```

```
    </dependency>
```

```
<dependency>
  <groupId>com.h2database</groupId>
  <artifactId>h2</artifactId>
  <scope>runtime</scope>
</dependency>
```

```
<dependency>
  <groupId>mysql</groupId>
  <artifactId>mysql-connector-java</artifactId>
  <scope>runtime</scope>
</dependency>
```

```
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-tomcat</artifactId>
  <scope>provided</scope>
```

```
</dependency>
```

```
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-thymeleaf</artifactId>
</dependency>
```

```
<!-- JSTL for JSP -->
```

```
<dependency>
  <groupId>javax.servlet</groupId>
  <artifactId>jstl</artifactId>
</dependency>
```

```
<!-- Need this to compile JSP -->
```

```
<dependency>
  <groupId>org.apache.tomcat.embed</groupId>
```

```
        <artifactId>tomcat-embed-jasper</artifactId>
        <scope>provided</scope>
    </dependency>

    <!-- Need this to compile JSP, -->

    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-test</artifactId>
        <scope>test</scope>
    </dependency>

</dependencies>

<build>
    <plugins>
        <plugin>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-maven-plugin</artifactId>
        </plugin>
    </plugins>
</build>

</project>
```

\*\*\*\*\*

CRIANDO OS PACOTES

\*\*\*\*\*

na pasta = src/main/java

exemplo

com.api.springsmartbd

com.api.springsmartbd.controller

com.api.springsmartbd.model

com.api.springsmartbd.repository

com.api.springsmartbd.service

com.api.springsmartbd.dao

\*\*\*\*\*

homeController.java

\*\*\*\*\*

package com.api.springsmartbd2;

import org.springframework.stereotype.Controller;

import org.springframework.ui.Model;

import org.springframework.web.bind.annotation.RequestMapping;

@Controller

public class homeController {

    @RequestMapping("/")

    public String index(){

        return "index";

    }

    @RequestMapping("/Index")

    public String Index(Model model) {

        return "Index";

    }



```
@RequestMapping("/home")
public String Home(Model model) {

    // System.out.println("home exibir");

    //model.addAttribute("TextoExibir", "Vanderlei exibir os dados");
    return "home";

}

}
```

\*\*\*\*\*

Na pasta templates

\*\*\*\*\*

A PASTA CSS JS IMAGEM DEVEM FICAR NA PASTA

RESOURCES / STATIC

dentro colocar os arquivos e pasta

+++ Index.html +++

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
<title>Bootstrap Example</title>
```

```
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/css/bootstrap.min.css" />
```

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
```

```
<script  
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/js/bootstrap.min.js"></script>
```

```
<script type="text/javascript">
```

```
</script>
```

```
</head>
```

```
<body>
```

```
<nav class="navbar navbar-inverse">
```

```
<div class="container-fluid">
```

```
<div class="navbar-header">
```

```
<a class="navbar-brand" href="#">WebSiteName</a>
```

```
</div>
```

```
<ul class="nav navbar-nav">
```

```
<li class="active"><a href="#">Home</a></li>
```

```
<li class="dropdown"><a class="dropdown-toggle" data-toggle="dropdown"  
href="#">Controle Paginas <span class="caret"></span></a>
```

```
<ul class="dropdown-menu">
```

```
<li><a href="#">Page 1-1</a></li>
```

```
<li><a href="#">Page 1-2</a></li>
```

```
<li><a href="#">Page 1-3</a></li>
```

```
</ul>
```

```
</li>
```

```
<li><a target="iframeIndex" href="/TESTEBD2">Mensagem</a></li>
```

```
<li><a target="iframeIndex" href="/TESTEBD">Listar Dados</a></li>
```

```
</ul>
```

```
<ul class="nav navbar-nav navbar-right">
```

```
<li><a href="#"><span class="glyphicon glyphicon-user"></span> Sign Up</a></li>
```

```
</li><a href="#"><span class="glyphicon glyphicon-log-in"></span> Login</a></li>
</ul>
</div>
</nav>

<div class="container">
  <div class="starter-template">
    <iframe id="iframeIndex" name="iframeIndex" class="embed-responsive-item"
    src="/home" onload="this.style.height=this.contentDocument.body.scrollHeight+50 +'px';"
    style="border:0;overflow:hidden;width:100%;" allowfullscreen scrolling='NO' framborder="0"
    ></iframe>
  </div>
</div>
<footer>
<hr>
Direitos
</footer>
</body>
</html>
```

+++ home.html +++

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Bootstrap Example</title>

</head>
<body>
```

```
<div class="container">
  <div class="starter-template">
    Home pagina exibindo
    <div style="height:900px;width: 400px;" >

  </div>
</div>
</div>
</body>
</html>
```

```
*****
```

```
CONTROLLER
```

```
*****
```

```
package com.api.springmartbd2;

import java.util.Date;
import java.util.List;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.beans.factory.annotation.Value;
import org.springframework.http.MediaType;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestMapping;
```

```
import org.springframework.web.bind.annotation.RequestMethod;
import com.api.springmartbd2.model.TESTEBD;
import com.api.springmartbd2.service.*;
```

```
@Controller
```

```
public class testebdController {
```

```
    @Autowired
```

```
    private TESTEBDService testebdservico = new TESTEBDService();
```

```
    @Value("${dados.home:default-value}")
```

```
    private String dados = "dados em branco";
```

```
    //TESTE DA MENSAGEM ENVIANDO DADOS PARA A TELA
```

```
    @RequestMapping("/TESTEBD2")
```

```
    public String listar(Model model) {
```

```
        List<TESTEBD> Listadados = testebdservico.consultar();
```

```
        System.out.println(Listadados);
```

```
        System.out.println("teste exibir");
```

```
        /*
```

```
        * COLOCAR ESTAS DUAS LINHAS ABAIXO DENTRO DA CLASSE FORA DOS
```

METODOS

```
        * DESTA FORMA CONSEGUE PASSAR DADOS PARA O HTML
```

```
        * @Value("${message.home:default-value}")
```

```
        private String message = "mensagem para exibir";
```

```
        *
```

```
        */
```

```
        //model.addAttribute("Listadados", Listadados);
```

```

        //model.addAttribute("message", "Vanderlei Testando !!");
        //model.addAttribute("mensagemdois", "Vanderlei Testando 2222 !!");

        return "exibirtestebd";

    }

    @RequestMapping(value="/TESTEBD", method = RequestMethod.GET,
        produces=MediaType.APPLICATION_JSON_UTF8_VALUE)
    public String TESTEBDgrid(Model model) {

        List<TESTEBD> Listadados = testebdservico.consultar();

        model.addAttribute("Listadados", Listadados);

        return "TESTEBDgrid";

    }

    @GetMapping("/TESTEBDform")
    public String TESTEBDform(Model model) {

        System.out.println("TESTEBDform exhibir");
        TESTEBD dados = new TESTEBD();
        dados.setID(0);
        dados.setNOME("");
        dados.setCATEGORIA("CASADO");
        dados.setATIVO("SIM");
        dados.setVALOR(0.0);

        //DateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd
HH:mm:ss");

```

```

        Date date = new Date();
        dados.setDATACADASTRO(date);
        //System.out.println(dateFormat.format(date)); //2016/11/16 12:08:43
        //testebd.setDATACADASTRO(dateFormat.format(date));

        model.addAttribute("dados", dados);

        return "TESTEBDform";
    }

    @GetMapping("/TESTEBDform/{id}")
    public String TESTEBDform(Model model, @PathVariable("id") int Idpes) {

        System.out.println("TESTEBDform exhibir");
        TESTEBD dados = testebdservico.buscar(Idpes);

        model.addAttribute("dados", dados);

        return "TESTEBDform";
    }

    @PostMapping("/TESTEBDform")
    public String TESTEBDform(@ModelAttribute("dados") TESTEBD dados, Model model) {

        testebdservico.salvar(dados);

        return "redirect:/TESTEBD";
    }

```

```
}
```

```
*****
```

```
PACONTE MODEL
```

```
NO PACOTE MODAL CRIANDO AS CLASSES QUANDO EXECUTA O SISTEM
```

```
O HIBERNATE VAI GERAR AS TABELAS NO BANCO
```

```
O BANCO DE DADOS TEM DE ESTA CRIADO
```

```
*****
```

```
+++ ResponseModel.java +++
```

```
package com.api.springsmartbd2.model;
```

```
public class ResponseModel {
```

```
    private int codigo;
```

```
    private String mensagem;
```

```
    public ResponseModel() {}
```

```
    public ResponseModel(int codigo, String mensagem) {
```

```
        this.codigo = codigo;
```

```
        this.mensagem = mensagem;
```

```
    }
```



```
        public int getCodigo() {
            return codigo;
        }

        public void setCodigo(int codigo) {
            this.codigo = codigo;
        }

        public String getMensagem() {
            return mensagem;
        }

        public void setMensagem(String mensagem) {
            this.mensagem = mensagem;
        }
    }
}
```

++ Tabela Classe +++

++ TESTEBD.java ++

```
package com.api.springmartbd2.model;
```

```
import java.io.Serializable;
```

```
import java.util.Date;
```

```
import javax.persistence.Column;
```

```
import javax.persistence.Entity;
```

```
import javax.persistence.GeneratedValue;
```

```
import javax.persistence.GenerationType;
```

```
import javax.persistence.Id;
```

```
import javax.persistence.NamedNativeQueries;
import javax.persistence.NamedNativeQuery;
import javax.persistence.Table;
import javax.persistence.Temporal;
import javax.persistence.TemporalType;
import javax.validation.constraints.NotNull;

import org.hibernate.validator.constraints.NotBlank;
import org.springframework.format.annotation.DateTimeFormat;
import org.springframework.format.annotation.NumberFormat;
```

```
@Entity
```

```
@NamedNativeQueries({
```

```
    @NamedNativeQuery(
```

```
        name = "TESTEBD.findByNome",
```

```
        query = " SELECT * FROM testebd t WHERE t.nome = 'VANDERLEI SOARES' ",
```

```
        resultClass=TESTEBD.class
```

```
    ),
```

```
    @NamedNativeQuery(
```

```
        name = "TESTEBD.findByCategoria",
```

```
        query = " SELECT * FROM testebd t WHERE t.categoria = ?1 ",
```

```
        resultClass=TESTEBD.class
```

```
    )
```

```
}}
```

```
@Table(name = "TESTEBD")
```

```
public class TESTEBD implements Serializable {
```

```

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

//@Id
//@SequenceGenerator(name = "post_seq", sequenceName = "post_seq")
//@GeneratedValue(strategy = GenerationType.SEQUENCE, generator = "post_seq")
//private int ID;

@Id
@GeneratedValue(strategy= GenerationType.AUTO)
@Column(name="ID")
private int ID;

@Column(nullable = false, length = 200)
@NotBlank(message = "Autor é uma informação obrigatória.")
private String NOME;

@Column(nullable = true, length = 45)
@NotBlank(message = "CATEGORIA pode ser nula.")
private String CATEGORIA;

@Column(nullable = false)
@NumberFormat(style=NumberFormat.Style.CURRENCY )
@NotNull(message = "Valor é uma informação obrigatória.")
private Double VALOR;

@Column(nullable = false)

```

```

@Temporal(TemporalType.DATE)

@DateTimeFormat(pattern = "yyyy-MM-dd")

@NotNull(message = "Data é uma informação obrigatória.")

private Date DATACADASTRO;

@Column(nullable = false, length = 3)

@NotBlank(message = "Ativo é uma informação obrigatória.")

private String ATIVO;

@Override

public String toString() {

    StringBuilder builder = new StringBuilder();

    builder.append("City{ ")

        .append(", id=").append(ID)

        .append(", name=").append(NOME)

        .append(", categoria=").append(CATEGORIA)

        .append(", ativo=").append(ATIVO)

        .append(", datacadastro=").append(DATACADASTRO)

        .append(", valor=").append(VALOR)

        .append("}");

    return builder.toString();
}

//GET E SET

public int getID() {

    return ID;

}

public void setID(int iD) {

```

```
        ID = iD;
    }

    public String getNOME() {
        return NOME;
    }

    public void setNOME(String nOME) {
        NOME = nOME;
    }

    public String getCATEGORIA() {
        return CATEGORIA;
    }

    public void setCATEGORIA(String cATEGORIA) {
        CATEGORIA = cATEGORIA;
    }

    public Double getVALOR() {
        return VALOR;
    }

    public void setVALOR(Double vALOR) {
        VALOR = vALOR;
    }

    public Date getDATACADASTRO() {
        return DATACADASTRO;
    }
}
```

```

        public void setDataCADASTRO(Date dATACADASTRO) {
            DATA CADASTRO = dATA CADASTRO;
        }

        public String getATIVO() {
            return ATIVO;
        }

        public void setATIVO(String aTIVO) {
            ATIVO = aTIVO;
        }
    }

    *****

    PACOTE REPOSITORY -- USANDO HIBERNATE
    *****

    package com.api.springsmartbd2.repository;

    import java.util.List;

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

    import com.api.springsmartbd2.model.TESTEBD;

    //public interface Produtos extends JpaRepository<TESTEBD, Long> {
    public interface TESTEBDRepository extends JpaRepository<TESTEBD, Integer> {

```

```
//void save(TESTEBD testebd);

void delete(TESTEBD testebd);

List<TESTEBD> findAll();

TESTEBD findOne(Integer id);

public TESTEBD findByName();

    @Query(nativeQuery = true)
public List<TESTEBD> findByCategoria(String ppesc);

}

*****

PACOTE SERVICE

*****

package com.api.springsmartbd2.service;

import java.util.*;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.MediaType;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RequestMapping;
```

```
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.ResponseBody;
import org.springframework.web.bind.annotation.RestController;

import com.api.springmartbd2.model.ResponseModel;
import com.api.springmartbd2.model.TESTEBD;
import com.api.springmartbd2.repository.TESTEBDRepository;

@RestController
@RequestMapping("/service")
public class TESTEBDService {

    @Autowired
    private TESTEBDRepository repository;

    /**
     * SALVAR UM NOVO REGISTRO
     * @param pessoa
     * @return
     */
    @RequestMapping(value="/TESTEBD", method = RequestMethod.POST,
consumes=MediaType.APPLICATION_JSON_UTF8_VALUE,produces=MediaType.APPLICATION_
JSON_UTF8_VALUE)
    public @ResponseBody ResponseModel salvar(@RequestBody TESTEBD testebd){

        try {

            this.repository.save(testebd);

            return new ResponseModel(1,"Registro salvo com sucesso!");
        }
    }
}
```



```

        }catch(Exception e) {

                return new ResponseModel(0,e.getMessage());

        }
}

/**
 * ATUALIZAR O REGISTRO DE UMA TESTEBD
 * @param pessoa
 * @return
 */
@RequestMapping(value="/TESTEBD", method = RequestMethod.PUT,
consumes=MediaType.APPLICATION_JSON_UTF8_VALUE)
public @ResponseBody ResponseModel atualizar(@RequestBody TESTEBD testebd){

        try {

                this.repository.save(testebd);

                return new ResponseModel(1,"Registro atualizado com sucesso!");

        }catch(Exception e) {

                return new ResponseModel(0,e.getMessage());

        }
}

/**
 * CONSULTAR TODAS AS PESSOAS
 * @return
 */

```

```

public @ResponseBody List<TESTEBD> consultar(){

    List<TESTEBD> ListaDados = new ArrayList<TESTEBD>();

    System.out.println("passou aqui consultar");

    ListaDados = this.repository.findAll();

    return ListaDados;

}

/**
 * BUSCAR UMA TESTEBD PELO ID
 * @param id
 * @return
 */
@RequestMapping(value="/TESTEBD/{id}", method = RequestMethod.GET,
produces=MediaType.APPLICATION_JSON_UTF8_VALUE)
public @ResponseBody TESTEBD buscar(@PathVariable("id") Integer codigo){

    return this.repository.findOne(codigo);

}

/**
 *QUERY CONSULTAR POR NOME
 *@return
 */

@RequestMapping(value="/QUERYTESTE", method = RequestMethod.GET,
produces=MediaType.APPLICATION_JSON_UTF8_VALUE)
public @ResponseBody TESTEBD consultaquery(){

    TESTEBD Dados = new TESTEBD();

    System.out.println("consultaquery");

```

```

        Dados = this.repository.findByNome();

        return Dados;

    }

    /**
     * QUERY CONSULTAR TODAS POR CATEGORIA
     * @return
     */
    @RequestMapping(value="/QUERYTESTENOVO", method = RequestMethod.GET,
        produces=MediaType.APPLICATION_JSON_UTF8_VALUE)
    public @ResponseBody List<TESTEBD> consultaqueryNative(){

        List<TESTEBD> ListaDados = new ArrayList<TESTEBD>();
        System.out.println("passou aqui consultaqueryNative");
        ListaDados = this.repository.findByCategoria("CASADO");
        return ListaDados;

    }

    /**
     * EXCLUIR UM REGISTRO PELO ID
     * @param id
     * @return
     */
    @RequestMapping(value="/TESTEBD/{id}", method = RequestMethod.DELETE,
        produces=MediaType.APPLICATION_JSON_UTF8_VALUE)
    public @ResponseBody ResponseModel excluir(@PathVariable("id") Integer codigo){

        TESTEBD testebd = repository.findOne(codigo);

```

```
try {

    repository.delete(testebd);

    return new ResponseModel(1, "Registro excluido com sucesso!");

} catch (Exception e) {

    return new ResponseModel(0, e.getMessage());

}

}
```

\*\*\*\*\*

REST

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

FIM SPRING BOOT - REST E HTML

\*\*\*\*\*

=====

=====

=====

=====

-- SPRING TOO SUIT --

para gerar o projeto SPRING

<https://start.spring.io/>

INSTALANDO A IDE PARA TRABALHAR NO PROJETO

<https://spring.io/tools>

PARA TESTAR REST

faça download

wiztools restclient

Server=179.188.16.6;Database=sindsmart2;Uid=sindsmart2;Pwd=odavi240175

Os componentes-base do Spring Boot

O Spring Boot possui 3 principais componentes que

permitem toda essa “magia” acontecer:

Spring Boot Starter;

Spring Boot AutoConfigurator;

Spring Boot Actuator.

O Spring Boot Starter é o coração de uma aplicação Spring. Sua função principal é combinar as várias dependências advindas de um projeto Spring Boot em uma única dependência, retirando-se a necessidade de configuração de múltiplas dependências no Maven ou no Gradle.

Por exemplo: se quisermos criar uma aplicação Spring web “convencional”, nós precisaríamos de, pelo menos, as seguintes dependências:

Spring Core;

Spring Web;

Spring Web MVC;

Servlet API.

\*\*\*\*\*

APPLICATION.PROPERTIES

\*\*\*\*\*

abrir o arquivo

application.properties = Outros código fonte - pacote default

BANCO DE DADOS

ARQUIVO: APPLICATION.PROPERTIES

MYSQL -----

# BANCO DE DADOS MYSQL

#spring.resources.static-locations=classpath:/custom/

#PORTA ONDE VAI SER EXECUTADO O NOSSO TOMCAT

server.port = 8090

#INFORMAÇÕES PARA CONEXÃO COM O BANCO DE DADOS MYSQL

spring.datasource.url=jdbc:mysql://179.188.16.6:3306/sindsmart2

spring.datasource.username=sindsmart2

spring.datasource.password=odavi240175

spring.datasource.driver-class-name=com.mysql.jdbc.Driver

#PARA CRIAR A TABELA NO BANCO BASTA COLOCAR update cria as tabelas

spring.jpa.hibernate.ddl-auto=update

# LazyInitializationException = spring.jpa.open-in-view=false

spring.jpa.open-in-view=false

spring.jpa.generate-ddl=true

spring.jpa.show-sql=true

spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL5Dialect

spring.jpa.properties.hibernate.format\_sql=true

#CAMINHO DOS ARQUIVOS DE EXIBIÇÕES

#spring.mvc.view.prefix=/WEB-INF/jsp/

#spring.mvc.view.suffix=.jsp

#=====

# = Thymeleaf configurations

```
#=====
spring.thymeleaf.check-template-location=true
spring.thymeleaf.prefix=classpath:/templates/
spring.thymeleaf.suffix=.html
spring.thymeleaf.content-type=text/html
spring.thymeleaf.cache=false
server.contextPath=/
```

```
#spring.thymeleaf.check-template-location=true
#spring.thymeleaf.prefix=classpath:/templates/
#spring.thymeleaf.suffix=.html
#spring.thymeleaf.mode=LEGACYHTML5
#spring.thymeleaf.encoding=UTF-8
#spring.thymeleaf.content-type=text/html
#spring.thymeleaf.cache=false
```

```
#=====

#For detailed logging during development
logging.level.org.springframework=TRACE
logging.level.com=TRACE
```

```
-----
spring.jpa.hibernate.ddl-auto=update
```

As opções são essas:

validate: validar o schema, não faz mudanças no banco de dados.

update: faz update o schema.

create: cria o schema, destruindo dados anteriores.

create-drop: drop o schema quando ao terminar a sessão.

```
-----
```



POSTGRESQL -----

spring.jpa.properties.hibernate.jdbc.lob.non\_contextual\_creation=true

spring.jpa.hibernate.ddl-auto=update

spring.datasource.url=jdbc:postgresql://localhost:3306/your\_database

spring.datasource.username=root

spring.datasource.password=root

FIM POSTGRESQL -----

++Criar pacotes

com.api.springsmartbd

com.api.springsmartbd.controller

com.api.springsmartbd.model

com.api.springsmartbd.repository

com.api.springsmartbd.service

++

\*\*\*\*\*

DEPEDENCIA

\*\*\*\*\*

ARQUIVO

ABRIR O ARQUIVO POM.XML

```
<?xml version="1.0" encoding="UTF-8"?>

<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">

  <modelVersion>4.0.0</modelVersion>

  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>1.5.21.RELEASE</version>
    <relativePath/> <!-- lookup parent from repository -->
  </parent>

  <groupId>com.api</groupId>
  <artifactId>springmartbd2</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <packaging>war</packaging>
  <name>springmartbd2</name>
  <description>Demo mysql</description>

  <properties>
    <java.version>1.8</java.version>
  </properties>

  <dependencies>
    <dependency>
      <groupId>org.hibernate</groupId>
      <artifactId>hibernate-core</artifactId>
    </dependency>

    <dependency>
      <groupId>org.hibernate</groupId>
      <artifactId>hibernate-entitymanager</artifactId>
```

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-data-jpa</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-web</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-devtools</artifactId>

<scope>runtime</scope>

</dependency>

<dependency>

<groupId>com.h2database</groupId>

<artifactId>h2</artifactId>

<scope>runtime</scope>

</dependency>

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

<scope>runtime</scope>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-tomcat</artifactId>

<scope>provided</scope>

```
</dependency>  
<dependency>  
  <groupId>org.springframework.boot</groupId>  
  <artifactId>spring-boot-starter-thymeleaf</artifactId>  
</dependency>
```

```
<!-- JSTL for JSP -->  
<dependency>  
  <groupId>javax.servlet</groupId>  
  <artifactId>jstl</artifactId>  
</dependency>
```

```
<!-- Need this to compile JSP -->  
<dependency>  
  <groupId>org.apache.tomcat.embed</groupId>  
  <artifactId>tomcat-embed-jasper</artifactId>  
  <scope>provided</scope>  
</dependency>
```

```
<!-- Need this to compile JSP, -->  
  
<dependency>  
  <groupId>org.springframework.boot</groupId>  
  <artifactId>spring-boot-starter-test</artifactId>  
  <scope>test</scope>  
</dependency>
```

```
</dependencies>
```

```
<build>
  <plugins>
    <plugin>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-maven-plugin</artifactId>
    </plugin>
  </plugins>
</build>

</project>
```

\*\*\*\*\*

EXEMPLO 001 HELLO WORLD ABRIR HTML

\*\*\*\*\*

src/main/java

No pacote principal

abrir o arquivo ...Application.java

```
package com.api.springmartnew;
```

```
import org.springframework.boot.SpringApplication;
```

```
import org.springframework.boot.autoconfigure.EnableAutoConfiguration;
```

```
import org.springframework.context.annotation.ComponentScan;
```

```
import org.springframework.context.annotation.Configuration;
```

```
//@SpringBootApplication
```

```
@Configuration
```

```
@ComponentScan(basePackages = {"com.api.springmartnew"})
```

```
@EnableAutoConfiguration
```

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

-----

-----

com.api.controller

com.api.modal

com.api.dao

com.api.Service -- facade

com.api.Repository

-----

-----

Criar um novo arquivo exemplo

HELLO WORLD

```
package com.api.springmartnew;
```

```
import org.springframework.stereotype.Controller;
```

```
import org.springframework.web.bind.annotation.RequestMapping;
```

```
@Controller
```

```
//@RequestMapping("/")
```

```
public class HelloWorldController {
```

```
        //@RequestMapping(value="/helloWorld", method=RequestMethod.GET)
        @RequestMapping("/helloWorld")
public String helloWorld()
{
    System.out.println("passou aqui");
    return "helloWorldPage";
}

}
```

-----

-----

no caminho = src/main/resources

dentro da pasta templates

criar um html

helloWorldPage.html

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<meta charset="ISO-8859-1">
```

```
<title>Insert title here</title>
```

```
</head>
```

```
<body>
```

```
    Hello world...
```

```
</body>
```

```
</html>
```

\*\*\*\*\*

JSON

\*\*\*\*\*

```
<dependency>  
  <groupId>com.fasterxml.jackson.core</groupId>  
  <artifactId>jackson-databind</artifactId>  
  <version>2.4.1</version>  
</dependency>
```

Criar a Class

```
package com.api.springmartnew;
```

```
public class Curso {
```

```
    private Integer ID;
```

```
    private String NOME;
```

```
    private String DURACAO;
```

```
    public Curso(Integer id, String nome, String duracao) {
```

```
        this.ID = id;
```

```
        this.NOME = nome;
```

```
        this.DURACAO = duracao;
```

```
    }
```

```
    public Integer getID() {
```

```
        return ID;
```

```
    }
```



```
public void setID(Integer iD) {  
    ID = iD;  
}
```

```
public String getNOME() {  
    return NOME;  
}
```

```
public void setNOME(String nOME) {  
    NOME = nOME;  
}
```

```
public String getDURACAO() {  
    return DURACAO;  
}
```

```
public void setDURACAO(String dURACAO) {  
    DURACAO = dURACAO;  
}
```

```
}
```

```
*****
```

```
*****
```

Criar a Class

```
package com.api.springsmartnew;
```

```
import java.util.*;
```

```
import org.slf4j.Logger;
```

```
import org.slf4j.LoggerFactory;
```

```
import org.springframework.http.HttpStatus;
```

```
import org.springframework.http.ResponseEntity;
```

```
import org.springframework.web.bind.annotation.PathVariable;
```

```
import org.springframework.web.bind.annotation.RequestBody;
```

```
import org.springframework.web.bind.annotation.RequestMapping;
```

```
import org.springframework.web.bind.annotation.RequestMethod;
```

```
import org.springframework.web.bind.annotation.RestController;
```

```
@RestController
```

```
public class CursoController {
```

```
    public static final Logger logger = LoggerFactory.getLogger(CursoController.class);
```

```
    private Map<Integer, Curso> cursos;
```

```
    public CursoController() {
```

```
        cursos = new HashMap<Integer, Curso>();
```

```
        Curso c1 = new Curso(1, "Workshop Rest", "24hs");
```

```
        Curso c2 = new Curso(2, "Workshop Spring MVC", "24hs");
```

```
        Curso c3 = new Curso(3, "Desenvolvimento Web com JSF 2", "60hs");
```

```
        cursos.put(1, c1);
```

```
        cursos.put(2, c2);
```

```

        cursos.put(3, c3);
    }

// -----Retrieve All Cursos-----

    @RequestMapping(value = "/cursos", method = RequestMethod.GET)
    public ResponseEntity<List<Curso>> listar() {
        return new ResponseEntity<List<Curso>>(new ArrayList<Curso>(cursos.values()),
HttpStatus.OK);
    }

// -----Retrieve Single Curso-----

    @RequestMapping(value = "/cursos/{id}", method = RequestMethod.GET)
    public ResponseEntity<Curso> buscar(@PathVariable("id") Integer id) {
        Curso curso = cursos.get(id);

        if (curso == null) {
            return new ResponseEntity<>(HttpStatus.NOT_FOUND);
        }

        return new ResponseEntity<Curso>(curso, HttpStatus.OK);
    }

// ----- Delete a Curso-----

    @RequestMapping(value = "/cursos/{id}", method = RequestMethod.DELETE)

```

```

public ResponseEntity<?> deletar(@PathVariable("id") int id) {
    Curso curso = cursos.remove(id);

    if (curso == null) {
        return new ResponseEntity<>(HttpStatus.NOT_FOUND);
    }

    return new ResponseEntity<>(HttpStatus.NO_CONTENT);
}

// -----Create a Curso-----
@RequestMapping(value = "/update", method = RequestMethod.POST)
public ResponseEntity<Curso> update(@RequestBody Curso curso) {

    //if (car != null) {
    //    car.setMiles(car.getMiles() + 100);
    //}

    // TODO: call persistence layer to update
    return new ResponseEntity<Curso>(curso, HttpStatus.OK);
}

// ----- Update a Curso -----

@RequestMapping(value = "/updateCurso/{id}", method = RequestMethod.PUT)
public ResponseEntity<?> updateCurso(@PathVariable("id") long id, @RequestBody
Curso curso) {
    logger.info("Updating User with id {}", id);
    Curso currentCurso = new Curso(null, null, null);
}

```

```
//User currentUser = userService.findById(id);

/*
if (currentUser == null) {
    logger.error("Unable to update. User with id {} not found.", id);
    return new ResponseEntity(new CustomErrorType("Unable to upate. User with
id " + id + " not found."),
        HttpStatus.NOT_FOUND);
}

currentUser.setName(user.getName());
currentUser.setAge(user.getAge());
currentUser.setSalary(user.getSalary());

userService.updateUser(currentUser);
*/
return new ResponseEntity<Curso>(currentCurso, HttpStatus.OK);
}

}
```

\*\*\*\*\*

POM.XML

\*\*\*\*\*

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
  http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>2.1.6.RELEASE</version>
    <relativePath/> <!-- lookup parent from repository -->
  </parent>
  <groupId>com.api</groupId>
  <artifactId>springmartnew</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <name>springmartnew</name>
  <description>Demo project for Spring Boot</description>

  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>
    <java.version>1.8</java.version>
  </properties>

  <dependencies>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-thymeleaf</artifactId>
    </dependency>

    <dependency>
```

```
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-web</artifactId>
</dependency>
```

```
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-devtools</artifactId>
    <scope>runtime</scope>
    <optional>true</optional>
</dependency>
```

```
<!-- DEPENDÊNCIAS DO SPRING -->
```

```
<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-core</artifactId>
</dependency>
```

```
<!-- JSON Processor -->
```

```
<dependency>
    <groupId>com.fasterxml.jackson.core</groupId>
    <artifactId>jackson-databind</artifactId>
</dependency>
```

```
<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-web</artifactId>
</dependency>
```

```
<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-webmvc</artifactId>
```

```
</dependency>
```

```
<!-- DEPENDÊNCIAS DO SPRING -->
```

```
<dependency>
```

```
  <groupId>javax.servlet.jsp.jstl</groupId>
```

```
  <artifactId>jstl-api</artifactId>
```

```
  <version>1.2</version>
```

```
  <scope>compile</scope>
```

```
  <exclusions>
```

```
    <exclusion>
```

```
      <artifactId>servlet-api</artifactId>
```

```
      <groupId>javax.servlet</groupId>
```

```
    </exclusion>
```

```
    <exclusion>
```

```
      <artifactId>jsp-api</artifactId>
```

```
      <groupId>javax.servlet.jsp</groupId>
```

```
    </exclusion>
```

```
  </exclusions>
```

```
</dependency>
```

```
<dependency>
```

```
  <groupId>org.springframework.boot</groupId>
```

```
  <artifactId>spring-boot-starter-test</artifactId>
```

```
  <scope>test</scope>
```

```
</dependency>
```

```
</dependencies>
```

```
<build>
```

```
  <plugins>
```

```
    <plugin>
```

```
      <groupId>org.springframework.boot</groupId>
```



```
                <artifactId>spring-boot-maven-plugin</artifactId>
            </plugin>
        </plugins>
    </build>

</project>
```

```
=====
```

```
HTML - MENSAGEM PARA O HTML
```

```
=====
```

```
package com.api.springsmartbd2;
```

```
import java.util.List;
```

```
import org.springframework.beans.factory.annotation.Autowired;
```

```
import org.springframework.beans.factory.annotation.Value;
```

```
import org.springframework.stereotype.Controller;
```

```
import org.springframework.ui.Model;
```

```
import org.springframework.web.bind.annotation.RequestMapping;
```

```
import com.api.springsmartbd2.model.TESTEBD;
```

```
import com.api.springsmartbd2.service.*;
```

```
@Controller
```

```
public class testebdController {
```

```
    @Autowired
```

```
    private TESTEBDService testebdservico = new TESTEBDService();
```

```
    @Value("${messages.home:default-value}")
```

```
private String message = "Hello World";
```

```
//TESTE DA MENSAGEM ENVIANDO DADOS PARA A TELA
```

```
@RequestMapping("/TESTEBD")
```

```
public String listar(Model model) {
```

```
    List<TESTEBD> Listados = testebdservico.consultar();
```

```
    System.out.println(Listados);
```

```
    System.out.println("teste exibir");
```

```
    //model.addAttribute("Listados", Listados);
```

```
    model.addAttribute("message", "Vanderlei Testando !!");
```

```
    model.addAttribute("mensagemdois", "Vanderlei Testando 2222 !!");
```

```
    return "exibirtestebd";
```

```
}
```

```
}
```

```
+++++
```

```
+++++
```

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
<title>Bootstrap Example</title>
```

```
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/css/bootstrap.min.css" />
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/js/bootstrap.min.js"></script>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<meta http-equiv="X-UA-Compatible" content="IE=edge"/>
<meta name="viewport" content="width=device-width, initial-scale=1"/>
<!-- END META SECTION -->
```

```
<!-- BEGIN STYLE -->
```

```
<style>
```

```
table, th, td {
    border: 1px solid black;
    padding: 1px;
}
```

```
</style>
```

```
<!-- END STYLE -->
```

```
</head>
```

```
<body>
```

```
<nav class="navbar navbar-inverse">
```

```
<div class="container-fluid">
```

```
<div class="navbar-header">
```

```
<a class="navbar-brand" href="#">WebSiteName</a>
```

```
</div>
```

```
<ul class="nav navbar-nav">
```

```
<li class="active"><a href="#">Home</a></li>
```

```
<li class="dropdown"><a class="dropdown-toggle" data-toggle="dropdown" href="#">Page
1 <span class="caret"></span></a>
  <ul class="dropdown-menu">
    <li><a href="#">Page 1-1</a></li>
    <li><a href="#">Page 1-2</a></li>
    <li><a href="#">Page 1-3</a></li>
  </ul>
</li>
<li><a href="#">Page 2</a></li>
</ul>
<ul class="nav navbar-nav navbar-right">
  <li><a href="#"><span class="glyphicon glyphicon-user"></span> Sign Up</a></li>
  <li><a href="#"><span class="glyphicon glyphicon-log-in"></span> Login</a></li>
</ul>
</div>
</nav>

<div class="container">
  <div class="starter-template">
    <h1>Spring Boot Configuration Example</h1>
    <h2 th:text="{message}"></h2>
    <h2 th:text="{messagemdois}"></h2>

  </div>
</div>
</body>
</html>
```

=====

=====

\*\*\*\*\*

JAVA NETBEANS - Maven

\*\*\*\*\*

<http://localhost:29588/crmprimaface/>

\*\*\*\*\*

\*\*\*\*\*

Criar novo

Novo projeto

maven

Aplicação Web

(se nao existir clicar no menu Ferramentas -> Plug-ins -

procurar todos maven e instalar todos reiniciar o netbeans)

clicar om o botão do direito do mouse

sobre o projeto clicar na opção

propriedade

agora selecionar Frameworks

botão Adicionar

escolher um de cada vez

Spring Web MVC

JavaServer Faces

Struts

Hibernate 4

qual a diferença entre facelets e jpa

\*\*\*\*\*

\*\*\*\*\*

Criar os pacotes

na pasta Pacotes de codigos-fonte

com.web.crmprimaface

com.web.dao

com.web.modal

\*\*\*\*\*

\*\*\*\*\*

PrimeFaces

<https://www.primefaces.org/>

pagar o no site na area do

download

```
<dependency>
```

```
  <groupId>org.primefaces</groupId>
```

```
  <artifactId>primefaces</artifactId>
```

```
  <version>7.0</version>
```

```
</dependency>
```

tem de recompilar e fechar o sistema abrir novamente

para que vc possa usar o primefaces

ai vc pode usar os codigo pegar no site o exemplo

\*\*\*\*\*

\*\*\*\*\*

model.xhtml

```
<?xml version='1.0' encoding='UTF-8' ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://xmlns.jcp.org/jsf/html"
      xmlns:f="http://xmlns.jcp.org/jsf/core"
      xmlns:ui="http://java.sun.com/jsf/facelets"
      xmlns:p="http://primefaces.org/ui">
  <h:head>
    <title>Facelet Title</title>
  </h:head>
  <h:body>
    <f:view>

    </f:view>
  </h:body>
</html>
```

\*\*\*\*\*

Servlet

\*\*\*\*\*

```
package com.web.Controller;

import java.io.IOException;
import java.text.DateFormat;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
```

```
import java.util.Arrays;

import java.util.Date;

import java.util.HashMap;

import java.util.List;

import java.util.Locale;

import java.util.Map;

import javax.servlet.ServletConfig;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import com.web.Model.Aluno;

/**
 * Servlet implementation class homeController
 */

@WebServlet({ "/index", "/Index", "/" })

public class homeController extends HttpServlet {

    private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */

    public homeController() {

        super();

        // TODO Auto-generated constructor stub

    }

}
```



```

/**
 * @see Servlet#init(ServletConfig)
 */
public void init(ServletConfig config) throws ServletException {
    // TODO Auto-generated method stub
}

/**
 * @see Servlet#destroy()
 */
public void destroy() {
    // TODO Auto-generated method stub
}

/**
 * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse
response)
 */
protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
    // TODO Auto-generated method stub
    //response.getWriter().append("Served at:
").append(request.getContextPath());
    request.setAttribute("name", "Vanderlei Soares de Olivera");
    String message = "Novo exemplo de atributo.";
    request.setAttribute("message", message);

Aluno aluno = new Aluno();
aluno.setId(1);

```

```
aluno.setNome("Vanderlei");

Date data = new Date();
//SimpleDateFormat sm = new SimpleDateFormat("dd-mm-yyyy");
DateFormat format = new SimpleDateFormat("dd-mm-yyyy", Locale.ENGLISH);
String strDate = format.format(data);
Date dtAtual = null;
try {
    dtAtual = format.parse(strDate);
    } catch (ParseException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
aluno.setDataCadastro(dtAtual);

//System.out.println(dtAtual);

aluno.setValor(10.0);
request.setAttribute("alunoexib", aluno);

//List<String> columnHeaders = Arrays.asList("Banana", "Apple", "Carrot", "Orange",
"Lychee", "Permisson");

Map<String, String> foods = new HashMap<String, String>();
    foods.put("man", "mango");
    foods.put("app", "apple");
    foods.put("gra", "grapes");
    request.setAttribute("foods", foods);

List<Aluno> ListaAluno = new ArrayList<Aluno>();
```

```

        aluno = new Aluno();
aluno.setId(1);
aluno.setNome("Vanderlei");
aluno.setValor(10.0);
ListaAluno.add(aluno);

aluno = new Aluno();
aluno.setId(2);
aluno.setNome("Davi");
aluno.setValor(10.0);
ListaAluno.add(aluno);

aluno = new Aluno();
aluno.setId(3);
aluno.setNome("Kayo");
aluno.setValor(10.0);
ListaAluno.add(aluno);

request.setAttribute("ListaAluno", ListaAluno);

        request.getRequestDispatcher("index.jsp").forward(request, response);

    }

    /**
     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse
response)
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {

        // TODO Auto-generated method stub

```

```

        doGet(request, response);
    }

    /**
     * @see HttpServlet#doPut(HttpServletRequest, HttpServletResponse)
     */
    protected void doPut(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
        // TODO Auto-generated method stub
    }

    /**
     * @see HttpServlet#doDelete(HttpServletRequest, HttpServletResponse)
     */
    protected void doDelete(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
        // TODO Auto-generated method stub
    }
}

```

\*\*\*\*\*

JSTL 1.1 : simple insert, update, delete example

\*\*\*\*\*

```

CREATE DATABASE IF NOT EXISTS loginjdbc;
CREATE TABLE `product` (
  `id` int(10) unsigned NOT NULL auto_increment,
  `pname` varchar(45) NOT NULL,

```

```
`quantity` int(10) unsigned NOT NULL,  
PRIMARY KEY (`id`)  
);
```

```
INSERT INTO `product` (`id`,`pname`,`quantity`) VALUES  
(1,'Mouse',50),  
(2,'Keyboard',5),  
(3,'Monitor',34);
```

```
+++++
```

```
index.jsp
```

```
+++++
```

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
```

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
```

```
    <title>JSP Page</title>
```

```
  </head>
```

```
  <body>
```

```
    <h1>Choose Option</h1>
```

```
    <a href="insert.jsp">Insert Record</a><p></p>
```

```
    <a href="display.jsp">Display Record</a>
```

```
  </body>
```

```
</html>
```

```
+++++
```

```
insert.jsp
```

```
+++++
```

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>JSP Page</title>
  </head>
  <body>
    <form action="insertdb.jsp" method="post">
      <table border="0" cellspacing="2" cellpadding="5">
        <thead>
          <tr>
            <th colspan="2">Purchase Product</th>
          </tr>
        </thead>
        <tbody>
          <tr>
            <td><label>Product Name</label></td>
            <td><input type="text" name="pname"/></td>
          </tr>
          <tr>
            <td><label>Quantity</label></td>
            <td><input type="text" name="qty"/></td>
          </tr>
          <tr>
            <td><input type="submit" value="Save" /></td>
            <td><input type="reset" value="reset"/></td>
          </tr>
        </tbody>
      </table>
```

```

</form>

<font color="red"><c:if test="\${not empty param.errMsg}">
  <c:out value="\${param.errMsg}" />
  <a href="index.jsp">Go Back</a>
</c:if></font>

<font color="green"><c:if test="\${not empty param.susMsg}">
  <c:out value="\${param.susMsg}" />
  <a href="index.jsp">Go Back</a>
</c:if></font>

</body>
</html>

+++++
insertdb.jsp
+++++

<%@ page import="java.io.*,java.util.*,java.sql.*"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/sql" prefix="sql"%>

<html>
  <head>
    <title>JINSERT Operation</title>
  </head>
  <body>
    <c:if test="\${ empty param.pname or empty param.qty}">
      <c:redirect url="insert.jsp" >
        <c:param name="errMsg" value="Please Enter Product and Quantity" />
      </c:redirect>
    </c:if>
  </body>
</html>

```

```
</c:if>

<sql:setDataSource var="dbsource" driver="com.mysql.jdbc.Driver"
    url="jdbc:mysql://localhost/loginjdbc"
    user="root" password="pass"/>

<sql:update dataSource="{dbsource}" var="result">
    INSERT INTO product(pname, quantity) VALUES (?,?);
    <sql:param value="{param.pname}" />
    <sql:param value="{param.qty}" />
</sql:update>

<c:if test="{result}>=1">
    <font size="5" color='green'> Congratulations ! Data inserted
    successfully.</font>

    <c:redirect url="insert.jsp" >
        <c:param name="susMsg" value="Congratulations ! Data inserted
        successfully." />
    </c:redirect>
</c:if>
```

```
</body>
</html>
```

```
+++++++
display.jsp
+++++++
```

```
<%@ page import="java.io.*,java.util.*,java.sql.*"%>
```



```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
```

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/sql" prefix="sql"%>
```

```
<html>
```

```
<head>
```

```
<title>SELECT Operation</title>
```

```
<script>
```

```
function confirmGo(m,u) {
```

```
    if ( confirm(m) ) {
```

```
        window.location = u;
```

```
    }
```

```
}
```

```
</script>
```

```
</head>
```

```
<body>
```

```
<sql:setDataSource var="dbsource" driver="com.mysql.jdbc.Driver"
```

```
    url="jdbc:mysql://localhost/loginjdbc"
```

```
    user="root" password="pass"/>
```

```
<sql:query dataSource="${dbsource}" var="result">
```

```
    SELECT * from product;
```

```
</sql:query>
```

```
<center>
```

```
<form>
```

```
<table border="1" width="40%">
```

```
<caption>Product List</caption>
```

```
<tr>
```

```
<th>Product ID</th>
```

```
<th>Product Name</th>
```

```
<th>Quantity</th>
```

```

        <th colspan="2">Action</th>
    </tr>
    <c:forEach var="row" items="${result.rows}">
        <tr>
            <td><c:out value="${row.id}"/></td>
            <td><c:out value="${row.pname}"/></td>
            <td><c:out value="${row.quantity}"/></td>
            <td><a href="update.jsp?id=<c:out value="${row.id}"/>">Update</a></td>
            <td><a href="javascript:confirmGo('Sure to delete this
record?', 'deletedb.jsp?id=<c:out value="${row.id}"/>')">Delete</a></td>

        </tr>
    </c:forEach>
</table>
</form>
<a href="index.jsp">Go Home</a>
</center>
</body>
</html>

```

```

+++++++

```

```

update.jsp

```

```

+++++++

```

```

<%@page contentType="text/html" pageEncoding="UTF-8"%>

```

```

<%@ page import="java.io.*,java.util.*,java.sql.*"%>

```

```

<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>

```

```

<%@ taglib uri="http://java.sun.com/jsp/jstl/sql" prefix="sql"%>

```

```

<!DOCTYPE html>

```

```

<html>

```

```
<head>

  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

  <title>JSP Page</title>

</head>

<body>

  <sql:setDataSource var="dbsource" driver="com.mysql.jdbc.Driver"
    url="jdbc:mysql://localhost/loginjdbc"
    user="root" password="pass"/>

  <sql:query dataSource="${dbsource}" var="result">
    SELECT * from product where id=?;
    <sql:param value="${param.id}" />
  </sql:query>

  <form action="updatedb.jsp" method="post">

    <table border="0" width="40%">
      <caption>Update Product</caption>
      <tr>
        <th>Product Name</th>
        <th>Quantity</th>
      </tr>
      <c:forEach var="row" items="${result.rows}">
        <tr>
          <td><input type="hidden" value="${param.id}" name="id"/>
          <input type="text" value="${row.pname}" name="pname"/></td>
          <td><input type="text" value="${row.quantity}" name="qty"/></td>
          <td><input type="submit" value="Update"/></td>
        </tr>
      </c:forEach>
    </table>

    <a href="index.jsp">Go Home</a>
```

```
</form>
</body>
</html>
```

```
+++++
updatedb.jsp
+++++
```

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@ page import="java.io.*,java.util.*,java.sql.*"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/sql" prefix="sql"%>
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>JSP Page</title>
  </head>
  <body>
    <sql:setDataSource var="dbsource" driver="com.mysql.jdbc.Driver"
      url="jdbc:mysql://localhost/loginjdbc"
      user="root" password="pass"/>
    <sql:update dataSource="${dbsource}" var="count">
      UPDATE product SET pname = ?, quantity=?
      WHERE id='${param.id}'
      <sql:param value="${param.pname}" />
      <sql:param value="${param.qty}" />
    </sql:update>
    <c:if test="${count}>=1">
      <font size="5" color='green'> Congratulations ! Data updated
```

```
        successfully.</font>
        <a href="index.jsp">Go Home</a>
    </c:if>
</body>
</html>
```

```
+++++
```

```
deletedb.jsp
```

```
+++++
```

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@ page import="java.io.*,java.util.*,java.sql.*"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/sql" prefix="sql"%>
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>JSP Page</title>
  </head>
  <body>
    <sql:setDataSource var="dbsource" driver="com.mysql.jdbc.Driver"
      url="jdbc:mysql://localhost/loginjdbc"
      user="root" password="pass"/>
    <sql:update dataSource="${dbsource}" var="count">
      DELETE FROM product
      WHERE id='${param.id}'
    </sql:update>
    <c:if test="${count}>=1">
      <font size="5" color='green'> Congratulations ! Data deleted
```

```
        successfully.</font>
        <a href="index.jsp">Go Home</a>
    </c:if>
</body>
</html>
```

+++++

form post e get

+++++

```
<html>
<head><title>Página de Login</title></head>
<body>
<form action=" ${pageContext.request.contextPath}/login.do" method="get">
    Nome.: <input type="text" name="nome" />
<br />
    Sobrenome.: <input type="text" name="sobrenome" />
<br />
<br />
<input type="submit" value="Enviar"/>
</form>
</body>
</html>
```

```
<html>
<head><title>Página de Login</title></head>
<body>
<form action=" ${pageContext.request.contextPath}/login.do" method="post">
```

```
Nome.: <input type="text" name="nome" />
<br />
Sobrenome.: <input type="text" name="sobrenome" />
<br />
<br />
<input type="submit" value="Enviar"/>
</form>
</body>
</html>
```

```
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

@WebServlet("/Ola")
public class Ola extends HttpServlet {
    private static final long serialVersionUID = 1L;
    public Ola() {
        super();
    }

    protected void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
        doPost(request, response);
    }

    protected void doPost(HttpServletRequest request, HttpServletResponse response)
```

```

throws ServletException, IOException {
    String nome = request.getParameter("nome");
    String sobrenome = request.getParameter("sobrenome");
    response.setContentType("text/html");
    PrintWriter out = response.getWriter();
    out.println("Bem Vindo<h3>"+nome+" " +sobrenome+"</h3>");
    out.close();
}
}

```

```
<%@ page import = "java.io.*,java.util.*" %>
```

```

<html>
<head>
<title>HTTP Header Request Example</title>
</head>

<body>
<center>
<h2>HTTP Header Request Example</h2>
<table width = "100%" border = "1" align = "center">
<tr bgcolor = "#949494">
<th>Param Name</th>
<th>Param Value(s)</th>
</tr>
<%
Enumeration paramNames = request.getParameterNames();
while(paramNames.hasMoreElements()) {
String paramName = (String)paramNames.nextElement();
out.print("<tr><td>" + paramName + "</td>\n");

```



```
        String paramValue = request.getHeader(paramName);
        out.println("<td> " + paramValue + "</td></tr>\n");
    }
%>
</table>
</center>

</body>
</html>
```

```
=====
=====
```

PROJETO EXIBIR CONSOLE

New java project

digitar o nome do projeto

Criar uma classe no src ele vai criar um pacote default

nesta classe vc tem de criar um metodo

```
    public static void main(String[] args) {
    }
```

para o sistema poder chamar ai vc pode criar outras classes

e metodos e chamar neste ponto ...

=====  
=====

## PROJETO DESCKTOP

New project

other

java

Java Project

Projeto gerado

clicar na pasta src com o botão direito new other

escolher WindowBuilder

Swing Designer

Application Window

clicar em Next

deixar o Package vazio pra ele gerar o default

digitar o nome do projeto

Criar os pacotes

clicar botao direito na pasta src

com.desc.View

com.desc.Model

com.desc.Controller

no pacote default

na classe principal alterar

=====

CLASSE DEFAULT

=====

```
import java.awt.EventQueue;
```

```
import javax.swing.JFrame;
```

```
import com.desc.Principal.Principal;
```

```
public class Descktop01 {
```

```
    private JFrame frame;
```

```
    /**
```

```
     * Launch the application.
```

```
     */
```

```
    public static void main(String[] args) {
```

```
        EventQueue.invokeLater(new Runnable() {
```

```
            public void run() {
```

```
                try {
```

```
                    Principal window = new Principal();
```

```
                    window.setVisible(true);
```

```
                } catch (Exception e) {
```

```
                    e.printStackTrace();
```

```
                }
```

```
            }
```

```
        });
```

```
    }
```

```

/**
 * Create the application.
 */
public Desktop01() {
    initialize();
}

/**
 * Initialize the contents of the frame.
 */
private void initialize() {
    frame = new JFrame();
    frame.setBounds(100, 100, 450, 300);
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
}
}

```

```

=====
TELA PAI
=====

```

```

package com.desc.Principal;

import javax.swing.JDesktopPane;
import javax.swing.JMenu;
import javax.swing.JMenuItem;

```

```
import com.desc.View.EmpresaGrid;

import javax.swing.JMenuBar;
import javax.swing.JFrame;

import java.awt.event.*;
import java.awt.*;

public class Principal extends JFrame implements ActionListener
{
    /**
     *
     */
    private static final long serialVersionUID = 1L;
    // Declara componetes
    JDesktopPane painel;

    public Principal()
    {
        // herda classe
        super("Java - Formulários - MDI - JFrame / JDesktopPane / JInternalFrame");

        // Define tamanho da janela principal
        int bordas = 50;
        Dimension tela = Toolkit.getDefaultToolkit().getScreenSize();
        setBounds(bordas, bordas, tela.width - bordas * 2 , tela.height - bordas * 2);

        // Cria painel principal
        painel = new JDesktopPane();
        painel.setBackground(Color.LIGHT_GRAY);
    }
}
```

```
setContentPane(painel);
setJMenuBar(createMenuBar());
}
```

```
protected JMenuBar createMenuBar()
```

```
{
    // Cria barra de menu
    JMenuBar barra = new JMenuBar();

    //=====
    // MENU CADASTRO
    //=====
    // Cria menu
    JMenu menu = new JMenu("Cadastro");
    barra.add(menu);

    // Cria itens dos menus
    JMenuItem item = new JMenuItem("Cliente");
    item.setActionCommand("btCliente");
    item.addActionListener(this);
    menu.add(item);

    //Cria segundo item
    item = new JMenuItem("Fornecedor");
    item.setActionCommand("btFornecedor");
    item.addActionListener(this);
    menu.add(item);

    //=====
    // MENU CADASTRO
```

```
//=====
// Cria menu
menu = new JMenu("Administrador");
barra.add(menu);

// Cria itens dos menus
item = new JMenuItem("Empresa");
item.setActionCommand("btEmpresa");
item.addActionListener(this);
menu.add(item);

//Cria segundo item
item = new JMenuItem("Filial");
item.setActionCommand("btFilial");
item.addActionListener(this);
menu.add(item);

//=====
// MENU Menu 001
//=====
// Cria menu
menu = new JMenu("Menu001");
barra.add(menu);

// Cria itens dos menus
item = new JMenuItem("Item001");
item.setActionCommand("btItem001");
item.addActionListener(this);
menu.add(item);
menu.addSeparator();
```

```
//Cria segundo item
item = new JMenuItem("Item002");
item.setActionCommand("btItem002");
item.addActionListener(this);
menu.add(item);

JMenu submenu = new JMenu("SubMenu");
menu.add(submenu);

JMenuItem Subitem = new JMenuItem("SubMenu001");
Subitem.setActionCommand("btSubMenu001");
Subitem.addActionListener(this);
submenu.add(Subitem);

//=====
// MENU ADMINISTRADOR
//=====

// Cria menu
menu = new JMenu("Ajuda");
barra.add(menu);

// Cria itens dos menus
item = new JMenuItem("Sair");
item.setActionCommand("btSair");
item.addActionListener(this);
menu.add(item);

return barra;
```



```
}
```

```
public void actionPerformed(ActionEvent e)
```

```
{
```

```
    switch(e.getActionCommand()){
```

```
        case "btEmpresa":
```

```
            frmEmpresaGrid();
```

```
            break;
```

```
        case "btFilial":
```

```
            //faz outra coisa
```

```
            break;
```

```
        default:
```

```
            sair();
```

```
    }
```

```
}
```

```
protected void sair()
```

```
{
```

```
    // Fecha sistema
```

```
    System.exit(0);
```

```
}
```

```
protected void frmEmpresaGrid()
```

```
{
```

```
// Cria formulario filho
    EmpresaGrid frame = new EmpresaGrid();
frame.setVisible(true);

// Adiciona janela filho na janela pai
painel.add(frame);
try
{
    // Seta foco
    frame.setSelected(true);
}
catch (java.beans.PropertyVetoException e)
{

}
}

private static void criaGUI()
{
    // Decora barra da janela
    JFrame.setDefaultLookAndFeelDecorated(true);

    // Cria formulario filho
    Principal formulario = new Principal();
    formulario.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    // Exibe formulario
    formulario.setVisible(true);
}
```

```
public static void main(String[] args)
{
    javax.swing.SwingUtilities.invokeLater(new Runnable()
    {
        public void run()
        {
            // Chama GUI
            criaGUI();
        }
    });
}

}
```

=====

=====

TELA FILHA - GRID

clicar no botao direito do mouse sobre o pacote

view

new -> other -> WindowsBuilder -> Swing Designer -> JInternalFrame

obs.. para poder organizar os componentes na tela coloque

um JPanel e nos componentes escolher

Layouts -> Absolute layout

desta forma pode movimentar os componentes de acordo com a  
necessidade

```
package com.desc.View;
```

```
import javax.swing.JInternalFrame;
```

```
import javax.swing.JLabel;
```

```
import javax.swing.JOptionPane;
```

```
import java.awt.Container;
```

```
import java.awt.Dimension;
```

```
import java.awt.GridLayout;
```

```
import javax.swing.BorderFactory;
```

```
import javax.swing.Box;
```

```
import javax.swing.BoxLayout;
```

```
import javax.swing.JButton;
```

```
import javax.swing.JFrame;
```

```
import javax.swing.JPanel;
```

```
import javax.swing.JScrollPane;
```

```
import com.desc.Controller.TBJEmpresaController;
```

```
import com.desc.Model.TBJEmpresa;
```

```
import com.desc.Principal.Principal;
```

```
import javax.swing.JTable;
```

```
import javax.swing.JTextField;
```

```
import javax.swing.SwingConstants;
```

```
import javax.swing.event.ListSelectionListener;
```

```
import javax.swing.table.DefaultTableModel;
```

```
import javax.swing.table.TableModel;

import java.awt.FlowLayout;
import java.awt.event.ActionListener;
import java.awt.event.MouseListener;
import java.util.ArrayList;
import java.util.List;
import java.util.Vector;
import java.awt.event.ActionEvent;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;

public class EmpresaGrid extends JFrame
{
    JLabel rotulo;
    JTextField Filtrocampo;
    JButton botao;
    JButton btNovo;

    JPanel painelFundo;
    JTable tabela;
    JScrollPane barraRolagem;

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

    // Declara variaveis
    static int formulario = 0;
    static int posicao = 30;
```

```

private JPanel panel;

private JPanel panel_1;

public EmpresaGrid()
{
    // Herda da classe filho
    super("Formulario Filho - Grid :" + (++formulario), true, true, true, true);
    pack();

    setContentPane(criaPainel());
    pack();

    // Define posição do formulario
    setLocation(posicao * formulario, posicao * formulario);
}

public Container criaPainel()
{
    // Cria painel
    JPanel painel = new JPanel();

    // Define layout
    painel.setLayout(new BorderLayout(painel, BorderLayout.PAGE_AXIS));
    painel.setBorder(BorderFactory.createEmptyBorder(20,20,20,20));
    Filtrocampo = new JTextField();

    // define tamanho do campo
    Filtrocampo.setPreferredSize(new Dimension(300,20));
}

```

```

panel_1 = new JPanel();
FlowLayout flowLayout_1 = (FlowLayout) panel_1.getLayout();
flowLayout_1.setAlignment(FlowLayout.LEFT);
painei.add(panel_1);

// Cria componentes
rotulo = new JLabel("Filtros:");
panel_1.add(rotulo);
rotulo.setHorizontalAlignment(SwingConstants.LEFT);
painei.add(Filtrocampo);

panel = new JPanel();
FlowLayout flowLayout = (FlowLayout) panel.getLayout();
flowLayout.setAlignment(FlowLayout.LEFT);
painei.add(panel);
botao = new JButton("Filtrar");
botao.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {

    }
});
botao.setHorizontalAlignment(SwingConstants.LEFT);
panel.add(botao);
btNovo = new JButton("Novo");
btNovo.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {

        //JOptionPane.showMessageDialog(null,"MENSAGEM AQUI");
        //JOptionPane.showMessageDialog(null,"MENSAGEM AQUI","TITULO",
JOptionPane.INFO_MESSAGE);

```

```

        /*
            int reply = JOptionPane.showConfirmDialog(null, "mensagem exibir ", title,
JOptionPane.YES_NO_OPTION);

            if (reply == JOptionPane.YES_OPTION) {
                JOptionPane.showMessageDialog(null, "HELLO");
            }
            else {
                JOptionPane.showMessageDialog(null, "GOODBYE");
                //System.exit(0);
            }
        */

        /*

            int n = JOptionPane.showConfirmDialog(null, "Yes No Cancel", "YesNoCancel",
JOptionPane.YES_NO_CANCEL_OPTION);

                                if(n == 0)
                                {
                                    JOptionPane.showConfirmDialog(null, "You
pressed YES\n"+"Pressed value is = "+n);
                                }
                                else if(n == 1)
                                {
                                    JOptionPane.showConfirmDialog(null, "You
pressed NO\n"+"Pressed value is = "+n);
                                }
                                else if (n == 2)
                                {
                                    JOptionPane.showConfirmDialog(null, "You
pressed CANCEL\n"+"Pressed value is = "+n);
                                }
                                else if (n == -1)
                                {

```



```
pressed X\n"+"Pressed value is = "+n);
        JOptionPane.showConfirmDialog(null, "You
    }
}
```

```
*/
```

```
//JFrame.setDefaultLookAndFeelDecorated(true);
    EmpresaForm frm = new EmpresaForm();
    //frm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frm.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
    frm.setVisible(true);
```

```
    }
```

```
});
```

```
panel.add(btNovo);
```

```
panel.add(Box.createVerticalStrut(10));//criar espaço
```

```
panel.add(GetDadosGrid());
```

```
return painel;
```

```
}
```

```
private static void criaGUI()
```

```
{
```

```
    // Cria GUI (Janela Filho)
```

```
    JFrame.setDefaultLookAndFeelDecorated(true);
```

```
    Principal formulario = new Principal();
```

```
    formulario.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

```
    formulario.setVisible(true);
```

```
}
```

```
public static void main(String[] args)
```

```
{
```

```
    javax.swing.SwingUtilities.invokeLater(new Runnable()
```

```
    {
```

```
        public void run()
```

```
        {
```

```
            // Chama GUI
```

```
            criaGUI();
```

```
        }
```

```
    });
```

```
}
```

```
private JPanel GetDadosGrid() {
```

```
    TBJEmpresaController contr = new TBJEmpresaController();
```

```
    List<TBJEmpresa> Lista = new ArrayList<TBJEmpresa>();
```

```
    Lista = contr.ListA();
```

```
    DefaultTableModel model = new DefaultTableModel();
```

```
    tabela = new JTable(model);
```

```
        model.addColumn("Codigo");
```

```
        model.addColumn("Nome");
```

```
        model.addColumn("DataCadastro");
```

```
        model.addColumn("Valor");
```

```
        model.addColumn("Ativo");
```

```

        for(TBJEmpresa item : Lista) {
            model.addRow(new Object[]
{item.getIdtbjempresa(),item.getNome(),item.getDatacadastro(),
item.getValor(),item.getAtivo()});
        }

//model.addRow(new Object[] { "v1", "v2" });
//model.addRow(new Object[] { "v1" });
//model.addRow(new Object[] { "v1", "v2", "v3" });

//tabela = new JTable(dados,colunas);
painelFundo = new JPanel();
painelFundo.setLayout(new GridLayout(1, 1));
//para criar este comando clique com o botao direito do mouse
//sobre a tabela e depois selecione add event handle -> mouse -> mouseClicked
tabela.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent e) {
        int coditem = Integer.parseInt(tabela.getValueAt(tabela.getSelectedRow(),
0).toString());

        JOptionPane.showMessageDialog(null, coditem);

    }
});

barraRolagem = new JScrollPane(tabela);
painelFundo.add(barraRolagem);

/*
Object [][] dados = {
    {"Ana Monteiro", "48 9923-7898", "ana.monteiro@gmail.com"},

```

```

        {"João da Silva", "48 8890-3345", "joaosilva@hotmail.com"},
        {"Pedro Cascaes", "48 9870-5634", "pedrinho@gmail.com"}
    };

    String [] colunas = {"Nome", "Telefone", "Email"};

    painelFundo = new JPanel();
    painelFundo.setLayout(new GridLayout(1, 1));
    tabela = new JTable(dados, colunas);
    tabela.addMouseListener(new MouseAdapter() {
        @Override
        public void mouseClicked(MouseEvent e) {
            int coditem = Integer.parseInt(tabela.getValueAt(tabela.getSelectedRow(),
0).toString());

            JOptionPane.showMessageDialog(null, coditem);

        }
    });
    barraRolagem = new JScrollPane(tabela);
    painelFundo.add(barraRolagem);
    */

    return painelFundo;

}

```

```
}
```

```
/*
```

```
String hql = "from Product p inner join p.category";
```

```
Query query = session.createQuery(hql);
```

```
List<Object[]> listResult = query.list();
```

```
for (Object[] aRow : listResult) {
```

```
    Product product = (Product) aRow[0];
```

```
    Category category = (Category) aRow[1];
```

```
    System.out.println(product.getName() + " - " + category.getName());
```

```
}
```

```
=====
```

```
String hql = "select sum(p.price), p.category.name from Product p group by category";
```

```
Query query = session.createQuery(hql);
```

```
List<Object[]> listResult = query.list();
```

```
for (Object[] aRow : listResult) {
```

```
    Double sum = (Double) aRow[0];
```

```
    String category = (String) aRow[1];
```

```
    System.out.println(category + " - " + sum);
```

```
}
```

```
=====
```

```
String hql = "from Product where description like :keyword";
```

```
String keyword = "New";  
Query query = session.createQuery(hql);  
query.setParameter("keyword", "%" + keyword + "%");  
  
List<Product> listProducts = query.list();  
  
for (Product aProduct : listProducts) {  
    System.out.println(aProduct.getName());  
}
```

=====

\*/

=====

=====

TELA FILHA - FORM

```
package com.desc.View;
```

```
import java.awt.Container;
```

```
import java.awt.EventQueue;
```

```
import javax.swing.BorderFactory;
```

```
import javax.swing.JPanel;
```

```
import javax.swing.JLabel;

import javax.swing.JOptionPane;

import javax.swing.JTextField;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JSpinner;

import com.desc.Controller.TBJEmpresaController;

import com.desc.Model.TBJEmpresa;

import com.toedter.calendar.JCalendar;

import com.toedter.calendar.JDateChooser;

import java.awt.event.ActionListener;

import java.util.Date;

import java.awt.event.ActionEvent;

import javax.swing.JComboBox;

public class EmpresaForm extends JFrame {

    static int formulario = 0;

    static int posicao = 30;

    private JPanel panel;

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

    private JTextField textCodigo;

    private JTextField textNome;

    /**
     * Launch the application.
     */
}
```

```

*/
public static void main(String[] args) {
    EventQueue.invokeLater(new Runnable() {
        public void run() {
            try {
                EmpresaForm frame = new EmpresaForm();
                frame.setSize(500, 500);
                frame.setLocation(50, 50);
                frame.setVisible(true);
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    });
}

```

```
/**
```

```
* Create the frame.
```

```
*/
```

```

public EmpresaForm() {
    //setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setType(Type.POPUP);
    setBounds(100, 100, 600, 600);
    setContentPane(criaPainelFundo());

    JComboBox<String> comboBox = new JComboBox<String>();
    comboBox.setBounds(111, 113, 175, 20);
    comboBox.removeAllItems();
    comboBox.addItem("--Escolha um Item--");
    comboBox.addItem("Item 1");
    comboBox.addItem("Sair");
}

```



```

panel.add(comboBox);
        setSize(600, 600);
setLocation(posicao * formulario, posicao * formulario);
//setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
    }

    public Container criaPainelFundo()
{

        panel = new JPanel();
        panel.setBorder(BorderFactory.createEmptyBorder(05,05,05,05));
        panel.setSize(400, 400);
        //panel.setLocation(50, 50);

panel.setLayout(null);

JLabel lblCodigo = new JLabel("Codigo");
lblCodigo.setBounds(10, 11, 46, 14);
panel.add(lblCodigo);

textCodigo = new JTextField();
textCodigo.setBounds(10, 27, 86, 20);
panel.add(textCodigo);
textCodigo.setColumns(10);

JLabel lblNome = new JLabel("Nome");
lblNome.setBounds(10, 53, 46, 14);
panel.add(lblNome);

textNome = new JTextField();

```

```
textNome.setBounds(10, 68, 523, 20);  
panel.add(textNome);  
textNome.setColumns(10);  
  
JLabel lblDataCadastro = new JLabel("Data Cadastro");  
lblDataCadastro.setBounds(10, 99, 96, 14);  
panel.add(lblDataCadastro);  
  
JLabel lblValor = new JLabel("Valor");  
lblValor.setBounds(10, 156, 46, 14);  
panel.add(lblValor);  
  
JButton btnSalvar = new JButton("Salvar");  
btnSalvar.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent arg0) {  
  
        SalvarDados(arg0);  
  
    }  
});  
btnSalvar.setBounds(10, 303, 89, 23);  
panel.add(btnSalvar);  
  
JSpinner spinner = new JSpinner();  
spinner.setBounds(10, 172, 123, 20);  
panel.add(spinner);  
  
JCalendar calendar = new JCalendar();  
calendar.setBounds(315, 156, 191, 153);  
panel.add(calendar);
```

```
JDateChooser dateChooser = new JDateChooser();  
dateChooser.setBounds(9, 113, 87, 20);  
panel.add(dateChooser);  
  
return panel;  
}
```

```
public void SalvarDados(ActionEvent arg0) {
```

```
    //String sTexto = "";  
    //sTexto = " codigo:"+textCodigo.getText();  
    //sTexto += " \n Nome:"+textNome.getText();  
    //JOptionPane.showMessageDialog(null,sTexto);
```

```
    TBJEmpresaController contr = new TBJEmpresaController();
```

```
    TBJEmpresa dados = new TBJEmpresa();
```

```
    //dados.setIdtbjempresa(0);
```

```
    dados.setNome("Vanderlei");
```

```
    dados.setDatacadastro(new Date());
```

```
    dados.setValor(10.00);
```

```
    dados.setAtivo("SIM");
```

```
    contr.Salvar(dados);
```

```
    JOptionPane.showMessageDialog(null,"Dados salvo corretamente...");
```

```
    }
```

```
}
```

=====

=====

## HIBERNATE JPA CRUD

download dos arquivos necessarios

encontra no meu OneDrive

JPA +Mysql + Hibernate.zip

baixar e colocar em uma pasta base

depois abrir o projeto e clicar com o botão direito do mouse

sobre o projeto

escolher Build Path -> Configure Build Path

aba libraries

Add Library -> User Library -> User Libraries -> New

selecione a libraries

Add External JARs..

pronto o hibernate esta liberado

agora vamos fazer o mesmo para mysql

agora criar o xml de pesistencia

clicar com botao direito do mouse sobre src

criar uma pasta META-INF

copiar a o arquivo persistence.xml para esta pasta

neste arquivo tem as configurações do banco de dados

```
<persistence-unit name="mysqlpesistencia" transaction-type="RESOURCE_LOCAL">
```

os dados de conexao...

=====

=====

persistence.xml

```
<persistence xmlns="http://java.sun.com/xml/ns/persistence"
```

```
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://java.sun.com/xml/ns/persistence
http://java.sun.com/xml/ns/persistence/persistence_2_0.xsd"
version="2.0">

<persistence-unit name="mysqlEntity" transaction-type="RESOURCE_LOCAL">

    <!-- provedor/implementacao do JPA -->
    <provider>org.hibernate.ejb.HibernatePersistence</provider>

    <!-- entidade mapeada -->
    <class>com.desc.Model.TBJEmpresa</class>

    <properties>
    <!-- dados da conexao -->
    <property name="hibernate.connection.driver_class" value="com.mysql.jdbc.Driver" />
    <property name="hibernate.connection.url"
value="jdbc:mysql://179.188.16.6:3306/sindsmart2" />
    <property name="hibernate.connection.username" value="sindsmart2" />
    <property name="hibernate.connection.password" value="odavi240175" />

    <!-- propriedades do hibernate -->
    <property name="hibernate.dialect" value="org.hibernate.dialect.MySQL5InnoDBDialect" />
    <property name="hibernate.show_sql" value="false" />
    <property name="hibernate.format_sql" value="false" />

    <!-- atualiza o banco, gera as tabelas se for preciso -->
    <property name="hibernate.hbm2ddl.auto" value="update" />

    </properties>
</persistence-unit>
```

</persistence>

=====

=====

CRIAR O MODAL

```
package com.desc.Model;
```

```
import java.util.Date;
```

```
import java.util.List;
```

```
import java.util.Set;
```

```
import javax.persistence.CascadeType;
```

```
import javax.persistence.Column;
```

```
import javax.persistence.GeneratedValue;
```

```
import javax.persistence.Id;
```

```
import javax.persistence.JoinColumn;
```

```
import javax.persistence.JoinTable;
```

```
import javax.persistence.OneToOne;
```

```
import javax.persistence.Table;
```

```
import javax.persistence.Temporal;
```

```
import javax.persistence.TemporalType;
```

```
import javax.persistence.Entity;
```

```
import javax.persistence.FetchType;
```

```
@Entity
```

```
@Table(name="TBJEmpresa")
```

```
public class TBJEmpresa {
```

```

@Id @GeneratedValue
@Column(name="idtbjempresa")
private int idtbjempresa;

@Column(name="nome")
@NotEmpty(message="O nome nao pode esta vazio")
private String nome;

@Column(name="datacadastro", nullable=false)
@Temporal(TemporalType.DATE)
//@Temporal(TemporalType.TIMESTAMP)
private Date datacadastro;

@Column(name="valor")
private Double valor;

@Column(name="ativo")
private String ativo;

//TEM DE CRIAR O CAMPO DE LIGAÇÃO NA OUTRA TABELA IGUAL A FILIAL
@OneToMany(mappedBy = "tbjEmpresa", targetEntity = TBJFilial.class, fetch =
FetchType.LAZY, cascade = CascadeType.ALL)
@JoinColumn(name = "idTBJFilial")
private List<TBJFilial> tbjFilial;

//vai gerar uma tabela pra fazer o relacionamento
//na tabela onde vai ser relacionado tem de criar o campo tambem
//@ManyToMany
//@OneToMany
//@JoinTable(name = "TBJEmpresa_TBJFilial",

```

```
//    joinColumns = @JoinColumn(name = "idtbjempresa", referencedColumnName =  
"idtbjempresa"),  
//    inverseJoinColumns = @JoinColumn(name = "idTBJFilial", referencedColumnName =  
"idTBJFilial"))  
//private Set<TBJFilial> ListaTBJFilial;
```

```
public int getIdtbjempresa() {  
    return idtbjempresa;  
}
```

```
public void setIdtbjempresa(int idtbjempresa) {  
    this.idtbjempresa = idtbjempresa;  
}
```

```
public String getNome() {  
    return nome;  
}
```

```
public void setNome(String nome) {  
    this.nome = nome;  
}
```

```
public Date getDatacadastro() {  
    return datacadastro;  
}
```

```
public void setDatacadastro(Date datacadastro) {  
    this.datacadastro = datacadastro;  
}
```



```
    public Double getValor() {
        return valor;
    }

    public void setValor(Double valor) {
        this.valor = valor;
    }

    public String getAtivo() {
        return ativo;
    }

    public void setAtivo(String ativo) {
        this.ativo = ativo;
    }

}

/**
 * para fazer um forkey usar esta anotações
 */
@Column(unique=true, nullable=false, insertable=true, updatable=true)
@Column(name="nome", precision=2)
@Column(name = "MY_FIELD_1", length=13)
@Length(min = 13, max = 13)
@Column(columnDefinition="TEXT")
@OneToMany(mappedBy="universidade", fetch = FetchType.LAZY)
@Cascade(CascadeType.ALL)
```

```
private Collection<Centro> centros;

*/

/*

Universidade univ = new Universidade();
univ.setNome("Universidade Federal do Rio Grande do Norte");
Centro centro1 = new Centro();
centro1.setNome("Centro de Tecnologia");
centro1.setUniversidade(univ);
Centro centro2 = new Centro();
centro2.setNome("Centro de Humanas");
centro2.setUniversidade(univ);
univ.setCentros(new HashSet<Centro>());
univ.getCentros().add(centro1);
univ.getCentros().add(centro2);
session.save(univ);

*/

+++++
+++++
+++++

package com.desc.Model;

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.FetchType;
import javax.persistence.GeneratedValue;
import javax.persistence.Id;
```

```

import javax.persistence.JoinColumn;
import javax.persistence.ManyToOne;
import javax.persistence.Table;

@Entity
@Table(name="TBJFilial")
public class TBJFilial {

    @Id @GeneratedValue
    @Column(name="idTBJFilial")
    private int idTBJFilial;
    @Column(name="Nome")
    private String Nome;
    @Column(name="Documento")
    private String Documento;

    //fazendo a referencia para montar forikey
    @ManyToOne(fetch = FetchType.LAZY)
    @JoinColumn
(name="TBJEmpresa_idtbjempresa",referencedColumnName="idtbjempresa",nullable=false,unique=true)
    private TBJEmpresa tbjEmpresa;

    public int getIdTBJFilial() {
        return idTBJFilial;
    }
    public void setIdTBJFilial(int idTBJFilial) {
        this.idTBJFilial = idTBJFilial;
    }
    public String getNome() {
        return Nome;
    }
}

```

```
        public void setNome(String nome) {
            Nome = nome;
        }
        public String getDocumento() {
            return Documento;
        }
        public void setDocumento(String documento) {
            Documento = documento;
        }
    }
}
```

=====

=====

CRIAR O CONTROLLER

```
package com.desc.Controller;
```

```
import java.util.List;
```

```
import javax.persistence.EntityManager;
```

```
import javax.persistence.EntityManagerFactory;
```

```
import javax.persistence.Persistence;
```

```
import javax.persistence.Query;
```

```
import com.desc.Model.TBJEmpresa;
```

```
import com.desc.Model.TBJFilial;
```

```

public class TBJEmpresaController {

    EntityManagerFactory emf;

    EntityManager em;

    public TBJEmpresaController() {

        System.out.println("inicio");

        emf = Persistence.createEntityManagerFactory("mysqlEntity");

        em = emf.createEntityManager();

        System.out.println("final");

    }

    public void Salvar(TBJEmpresa dados) {

        em.getTransaction().begin();

        em.merge(dados);

        em.getTransaction().commit();

        em.close();

    }

    public void Deletar(TBJEmpresa dados) {

        em.getTransaction().begin();

        Query q = (Query) em.createNamedQuery("delete from TBJEmpresa where
idtbjempresa = "+dados.getIdtbjempresa());

        q.executeUpdate();

        em.getTransaction().commit();

        em.close();

    }
}

```

```

@SuppressWarnings("unchecked")
public List<TBJEmpresa> ListA(){
    List<TBJEmpresa> lista = null;

    String hql = "from TBJEmpresa";
    Query query = em.createQuery(hql);
    lista = query.getResultList();

    for (TBJEmpresa item : lista) {
        System.out.println(item.getNome());
        List<TBJFilial> Listfil = item.tbjFilial;
        for(TBJFilial itemfil : Listfil) {
            System.out.println(itemfil.getNome());
        }
    }

    return lista;
}

```

```

/*

```

```

    String str1 = "Select * from PRODUTOS where CLASSIFICACOES = " +
ResultadoPesquisaClassificacoesID.getText();
    ResultSet rs1 = cnn.ExecutarResult(str1);
    System.out.println(str1);
    List<String> Arraylista = new ArrayList<String>();
    while(rs1.next()){
        Arraylista.add(rs1.getString("DESC_PROD"));
    }

```

```
*/
```

```
}
```

```
=====
```

```
=====
```

#### JAVA ANOTATION

```
import java.util.Collection;
import javax.persistence.*;
import org.hibernate.annotations.Cascade;
import org.hibernate.annotations.CascadeType;
@Entity@Table(name="departamento", schema="anotacoes")
publicclass Departamento {
    @Id@GeneratedValue(strategy = GenerationType.SEQUENCE)
    @Column(name="id_departamento")
    privateintid;
    @Column(nullable=false, insertable=true, updatable=true)
    private String nome;
    @Column(nullable=false, insertable=true, updatable=true, length=6)
    private String sigla;
    @ManyToOne(fetch=FetchType.EAGER)
    @JoinColumn(name="id_centro", insertable=true, updatable=true)
    @Cascade(CascadeType.SAVE_UPDATE)
    private Centro centro;
    @ManyToMany(fetch=FetchType.LAZY)
    @JoinTable(name="departamento_curso", schema="anotacoes",
    joinColumns=@JoinColumn(name="id_departamento"),
```

```
inverseJoinColumn=@JoinColumn(name="id_curso"))
private Collection<Curso> cursos;
//Métodos setters e getters
}
```

#### EXEMPLO DE USO

```
Departamento depart = new Departamento();
depart.setNome("DEPARTAMENTO 1");
depart.setSigla("DEP.01");
//Busca objeto centro na base de dados com
//identificador igual a 110
depart.setCentro((Centro)session.get(Centro.class, 110));
//Busca objeto curso na base de dados com
//identificador igual a 220
Curso curso = (Curso)session.get(Curso.class, 220);
depart.setCursos(new ArrayList<Curso>());
//Adiciona curso na coleção do departamento
depart.getCursos().add(curso);
//Persiste departamento
session.save(depart);
tx.commit();
```

#### EXEMPLO DE USO 2

```
Tarefa tarefa = new Tarefa();
tarefa.setDescricao("Estudar JPA");
tarefa.setFinalizado(true);
tarefa.setDataFinalizacao(Calendar.getInstance());
```

```
EntityManagerFactory factory = Persistence.createEntityManagerFactory("tarefas");
```



```
EntityManager manager = factory.createEntityManager();
```

```
manager.getTransaction().begin();
```

```
manager.persist(tarefa);
```

```
manager.getTransaction().commit();
```

```
System.out.println("ID da tarefa: " + tarefa.getId());
```

```
manager.close();
```

### EXEMPLO 3

```
EntityManagerFactory factory = Persistence.createEntityManagerFactory("tarefas");
```

```
EntityManager manager = factory.createEntityManager();
```

```
Tarefa encontrada = manager.find(Tarefa.class, 1L);
```

```
System.out.println(encontrada.getDescricao());
```

### EXEMPLO 4

```
Tarefa tarefa = new Tarefa();
```

```
    tarefa.setDescricao("Estudar JPA e Hibernate");
```

```
    tarefa.setFinalizado(true);
```

```
    tarefa.setDataFinalizacao(Calendar.getInstance());
```

```
EntityManagerFactory factory = Persistence.
```

```
    createEntityManagerFactory("tarefas");
```

```
EntityManager manager = factory.createEntityManager();
```

```
manager.getTransaction().begin();  
manager.persist(tarefa);  
manager.getTransaction().commit();  
  
System.out.println("ID da tarefa: " + tarefa.getId());  
  
manager.close();
```

=====

=====

XAMPP PARA INSTALAR O APACHE

=====

=====

INSTALACAO DO PROJETO NO CLIENTE

DISTRIBUIR PROJETO

projeto pronto

clique com botao direito do mouse sobre o projeto

escolha exportar -> java -> Runnable JAR file

Launch configuration escolher o projeto inicial man

Export destination colocar o caminho e nome onde vai gera o jar

finish

criar um bat na pasta onde vai ficar o jar

coloque este comando para executar o jar

@echo off

start javaw -jar C:\Temp\desctop01.jar

exit

criar um icone para o bat e colocar

atalho do bat para o usuario executar o sistema

=====  
=====

## SQL CONTROLE

```
select SYSDATE from tabela nome
```

```
select to_date(year || month || day, 'YYYYMMDD') from dual
```

```
select to_date(to_char(effective_date, 'YYYY') || '-' || to_char(point_date, 'MM-DD'), 'YYYY-MM-DD')
```

```
SELECT EXTRACT(YEAR FROM DATE '2017-12-01') FROM DUAL;
```

2017

```
SELECT EXTRACT(MONTH FROM DATE '2017-12-01') FROM DUAL;
```

12

```
SELECT *
```

```
FROM myschema.mytablename
```

```
WHERE TO_CHAR(myDATE, 'YYYY') = '2017';
```

```
SELECT *
```

```
FROM myschema.mytablename
```

```
WHERE TO_CHAR(myDATE, 'YYYY') = '2017' AND TO_CHAR(myDate, 'MM') = '07';
```

```
SELECT *
FROM myschema.mytablename
WHERE TO_CHAR(myDATE, 'YYYY') = '2017' AND TO_CHAR(myDate, 'MM') = '07';
```

```
SELECT SYSDATE,
LAST_DAY(SYSDATE) "Last",
LAST_DAY(SYSDATE) - SYSDATE "Days Left"
FROM DUAL;
```

MONTHS\_BETWEEN( date1, date2 )

```
SELECT MONTHS_BETWEEN
(TO_DATE('02-02-2015','MM-DD-YYYY'),
TO_DATE('12-01-2014','MM-DD-YYYY') ) "Months"
FROM DUAL;.
```

```
SELECT TO_CHAR(ADD_MONTHS(credito,1),'DD-MON-YYYY') "Proximo mes"
FROM clientes
WHERE nome = 'FRANK';
```

=====

=====

COMPONENTE - INSTALAR E USUAR

JCALENDAR

Download - <https://toedter.com/jcalendar/>

copiar o jar para pasta F:\Trabalho\Java\Componentes\JCalendar (exemplo)

criar um projeto desktop depois clicar botao direito no projeto

escolher propriedade no lado esquerdo escolher

Java Build Path -> Libraries -> Add External JARs...

buscar o jar

Exibir o componente na paleta de componentes

abra a tela no modo Design

na aba Palette

feche todas as pastas de componentes

Clique com botão direito do mouse

na paleta e escolha

Paletter Manager

clicar no botão Add Category...

digitar um nome exemplo JCALENDAR

ok

clicar botão Importar Jar...

no jar Archive escolher FileSystem

buscar o caminho do jar

selecionar todos e ok

ok

Pronto instalado o novo componente

=====  
=====

=====  
=====

Spring MVC 4 e NetBeans

+++++

GERANDO O WAR

projeto pronto clicar no menu

Executar

Limpar e Construir Projeto

vai na pasta do projeto la foi criado uma pasta

dist

pronto war criado

+++++

Menu Arquivo -> Novo -> Java Web -> Aplicação Web

dar um nome pro projeto

proximo escolher o servidor

a versao

Caminho do Contexto deixar o que esta exibindo

para ele fazer as rotas

colocar o servidor tom cat ou glasfit

proximo

escolher o Spring Web MVC

na aba biblioteca escolher spring framework 4.0.1

na aba Configuração

Mapeamento de emissores colocar /

(com a barra fica por rota)

executar primeira vez

se o tomcat estiver ativo ele vai criar o projeto no maneger

da uma olhada

http://localhost:8081/manager/html

http://localhost:8081/NetBeansExemplo01/index

abrir o arquivo dispatcher-servlet.xml

manter

```
<bean id="viewResolver"  
    class="org.springframework.web.servlet.view.InternalResourceViewResolver"  
    p:prefix="/WEB-INF/jsp/"  
    p:suffix=".jsp" />
```

o restante apagar

apagar o arquivo redirect.jsp

apagar o arquivo index.jsp na pasta web-inf -> jsp

Criar os pacotes na pasta

Pacotes de código-fonte

exemplo

com.api.controller

com.api.model

com.api.repository

com.api.service

com.api.dao

=====

abrir o arquivo dispatcher-servlet.xml

incluir

```
<mvc:annotation-driven />  
<context:component-scan base-package="com.api.*" />
```

alterar

```

<beans xmlns="http://www.springframework.org/schema/beans"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:p="http://www.springframework.org/schema/p"
  xmlns:aop="http://www.springframework.org/schema/aop"
  xmlns:tx="http://www.springframework.org/schema/tx"
  xmlns:mvc="http://www.springframework.org/schema/mvc"
  xmlns:context="http://www.springframework.org/schema/context"
  xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans-4.0.xsd
  http://www.springframework.org/schema/aop
http://www.springframework.org/schema/aop/spring-aop-4.0.xsd
  http://www.springframework.org/schema/tx
http://www.springframework.org/schema/tx/spring-tx-4.0.xsd
  http://www.springframework.org/schema/mvc
http://www.springframework.org/schema/mvc/spring-mvc-4.0.xsd
  http://www.springframework.org/schema/context
http://www.springframework.org/schema/context/spring-context-4.0.xsd ">

```

=====

no pacote controller criar uma class

homeController

@Controller

```
public class HomeController {
```

```
  @RequestMapping("/index")
```

```
    @RequestMapping(value={"/", "/", "/index"})
```

```
    public String index(){
```

```
        return "index";
```

```
    }
```

```
}
```



na pasta jsp

criar o arquivo index.jsp

Copilar e testar

<http://localhost:8081/manager/html>

=====

BOOTSTRAP - PASTAS DE CONTROLE

Indetação automática

É ALT+SHIFT+F,

ou Código-fonte>>Formatar

criar o local para colocar css - js - imagens pastas para o html

clique no botão direito na página web nova pasta

resources

- css

- js

- download

- imagem

- upload

- tema

download bootstrap copiar os arquivos css e js

abrir o arquivo dispatcher-servlet.xml

incluir

```
<mvc:resources mapping="/resources/**" location="/resources/" />
```

no head dos html jsp

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
  <head>
    <link href="<c:url value="/resources/css/bootstrap.css" />" rel="stylesheet">
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>JSP Page</title>
  </head>
  <body>
    <h1>Hello World!</h1>
  </body>
</html>
```

=====

CONEXAO JPA

Criar a classe

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package com.api.dao;
```

```
import java.sql.Connection;
```

```
import java.sql.DriverManager;
```

```

/**
 *
 * @author sovan
 */
public class DBConnection {

    public static Connection createConnection() {

        //para funcionar a conexao tem de adicionar a biblioteca do jdbc mysql
        //clicar com botão direito do mouse sobre a pasta Bibliotecas -> Adicionar Biblioteca
        //escolher Driver JDBC do MySQL - pronto vai funcionar agora...

        Connection con = null;

        String url =
"jdbc:mysql://179.188.16.6:3306/sindsmart2?zeroDateTimeBehavior=convertToNull";
//MySQL URL and followed by the database name

        String username = "sindsmart2"; //MySQL username
        String password = "odavi240175"; //MySQL password

        try {
            try {
                //DriverManager.registerDriver(new com.mysql.jdbc.Driver());
                //Class.forName("com.mysql.jdbc.Driver");
                Class.forName("com.mysql.jdbc.Driver"); //loading mysql driver
            } catch (ClassNotFoundException e) {
                e.printStackTrace();
            }

            con = DriverManager.getConnection(url, username, password); //attempting to connect
to MySQL database

            System.out.println("Printing connection object " + con);
        } catch (Exception e) {

```

```
        e.printStackTrace();
    }
    return con;
}
```

```
}
```

```
=====
```

```
tbjempresaDal
```

```
package com.api.dao;
```

```
import com.api.model.TBJEmpresa;
```

```
import com.google.gson.Gson;
```

```
import java.sql.PreparedStatement;
```

```
import java.sql.ResultSet;
```

```
import java.sql.SQLException;
```

```
import java.sql.Statement;
```

```
import java.util.ArrayList;
```

```
import java.util.Calendar;
```

```
import java.util.Date;
```

```
import java.util.List;
```

```
public class tbjempresaDal {
```

```

public String adicionar(TBJEmpresa dados) throws Exception{
    String sRetorno = "";

    StringBuilder sql = new StringBuilder();
    //sql.append("insert into TBJEmpresa (idtbjempresa, nome, datacadastro,valor,ativo) ");
    //sql.append("values (?, ?, ?, ?, ?)");//now()

    try {
        //sql puro
        sql.append("insert into TBJEmpresa (idtbjempresa, nome, datacadastro,valor,ativo) ");
        sql.append("values (").append(dados.getIdtbjempresa()).append(",");
        sql.append(dados.getNome()).append(",");
        sql.append(dados.getDatacadastro()).append(",");
        sql.append(dados.getValor()).append(",");
        sql.append(dados.getAtivo()).append(")");

        Statement stmt = DBConnection.createConnection().createStatement();
        stmt.executeUpdate(sql.toString());
        sRetorno = sql.toString();

        //ResultSet rs = stmt.executeQuery(sql.toString());
        //statement.executeUpdate("INSERT INTO Customers " + "VALUES (1001, 'Simpson',
'Mr.', 'Springfield', 2001)");

        /*

        PreparedStatement pstmt = conexao.getConexao().prepareStatement(sql.toString());
        pstmt.setInt(1, 1);
        pstmt.setString(2, "vanderlei soares");

```

```

Date date = new Date();

pstmt.setDate(3, (java.sql.Date) date);

pstmt.setDouble(4, 10.00);

pstmt.setString(5, "SIM");

*/

//sRetorno = pstmt.toString();

//pstmt.execute();

//pstmt.close();
} catch (SQLException e) {
//System.out.println(e.getMessage());

throw new Exception("Preparando para Inserir.", e);

//throw new DemoiselleException("Não foi possível gravar o log de operação");
}

return sRetorno;
}

```

```

public void alterando(TBJEmpresa dados) throws Exception{

```

```

    StringBuilder sql = new StringBuilder();

    sql.append(" update TBJEmpresa set idtbjempresa=?, nome=?,
datacadastro=?,valor=?,ativo=? where idtbjempresa = ? ");//now()

```

```

    try {

```

```

        PreparedStatement pstmt =
        DBConnection.createConnection().prepareStatement(sql.toString());

        pstmt.setInt(1, dados.getIdtbjempresa());

        pstmt.setString(2, dados.getNome());

```

```

        pstmt.setDate(3, dados.getDatacadastro());
        pstmt.setDouble(4, dados.getValor());
        pstmt.setString(5, dados.getAtivo());
        pstmt.setInt(6, dados.getIdtbjempresa());
        pstmt.executeUpdate();
        pstmt.close();
    } catch (SQLException e) {
        //throw new DemoiselleException("Não foi possível gravar o log de operação");
    }
}

```

```

public TBJEmpresa findById(int pld) throws Exception {
    StringBuilder sql = new StringBuilder();
    sql.append(" select * from TBJEmpresa where idtbjempresa = ")
.append(Integer.toString(pld));

    PreparedStatement pstmt =
    DBConnection.createConnection().prepareStatement(sql.toString());

    ResultSet rs = pstmt.executeQuery();
    TBJEmpresa result = new TBJEmpresa();

    while (rs.next()) {
        TBJEmpresa dados = new TBJEmpresa();
        result.setIdtbjempresa(rs.getInt("idtbjempresa"));
        result.setNome(rs.getString("nome"));
        result.setDatacadastro(rs.getDate("datacadastro"));
        result.setValor(rs.getDouble("valor"));
        result.setAtivo(rs.getString("ativo"));
    }
}

```

```
rs.close();  
pstmt.close();  
return result;  
}
```

```
public List<TBJEmpresa> findAll() throws Exception {
```

```
    StringBuilder sql = new StringBuilder();  
    sql.append(" select * from TBJEmpresa ");
```

```
    PreparedStatement pstmt =  
    DBConnection.createConnection().prepareStatement(sql.toString());
```

```
    ResultSet rs = pstmt.executeQuery();
```

```
    List<TBJEmpresa> result = new ArrayList<TBJEmpresa>();
```

```
    while (rs.next()) {
```

```
        TBJEmpresa dados = new TBJEmpresa();  
        dados.setIdtbjempresa(rs.getInt("idtbjempresa"));  
        dados.setNome(rs.getString("nome"));  
        dados.setDatacadastro(rs.getDate("datacadastro"));  
        dados.setValor(rs.getDouble("valor"));  
        dados.setAtivo(rs.getString("ativo"));  
        result.add(dados);
```

```
    }
```

```
rs.close();
```

```
pstmt.close();
```

```
return result;
```

```
}
```



```

public String GetJson(){

    String jsonretorno = "";

    TBJEmpresa dadosretorno = new TBJEmpresa();
    dadosretorno.setIldtbjempresa(1);
    dadosretorno.setNome("Vanderlei Soares de Oliveira");
    java.util.Date date = Calendar.getInstance().getTime();
    java.sql.Date sqlDate = new java.sql.Date(date.getTime());
    dadosretorno.setDatacadastro(sqlDate);
    dadosretorno.setValor(10.00);
    dadosretorno.setAtivo("SIM");

    Gson gson = new Gson();
    jsonretorno = gson.toJson(dadosretorno);

    return jsonretorno;
}
}

```

=====

controller

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

```
*/
```

```
package com.api.controller;
```

```
import com.api.dao.DBConnection;
```

```
import com.api.dao.tbjempresaDal;
```

```
import com.api.model.TBJEmpresa;
```

```
import java.sql.ResultSet;
```

```
import java.sql.Statement;
```

```
import java.util.ArrayList;
```

```
import java.util.Calendar;
```

```
import java.util.List;
```

```
import javax.servlet.http.HttpServletRequest;
```

```
import org.springframework.stereotype.Controller;
```

```
import org.springframework.ui.Model;
```

```
import org.springframework.web.bind.annotation.PathVariable;
```

```
import org.springframework.web.bind.annotation.RequestMapping;
```

```
import org.springframework.web.bind.annotation.RequestParam;
```

```
/**
```

```
*
```

```
* @author sovan
```

```
*/
```

```
@Controller
```

```
public class tbjEmpresaController {
```

```
    @RequestMapping("/tbjEmpresaIndex")
```

```
    public String tbjEmpresaIndex(Model model, HttpServletRequest request) throws Exception  
{
```

```
        List<TBJEmpresa> ListaDados = new ArrayList<TBJEmpresa>();
```

```
DBConnection conexao = new DBConnection();
Statement stmt = conexao.createConnection().createStatement();
ResultSet rs = stmt.executeQuery(" SELECT * FROM TBJEmpresa ");
while (rs.next()) {
    //String lastName = rs.getString("Lname");
    //System.out.println(lastName + "\n");
    TBJEmpresa dadosadd = new TBJEmpresa();
    dadosadd.setIdtbjempresa(rs.getInt("Idtbjempresa"));
    dadosadd.setNome(rs.getString("nome"));
    //dadosadd.setDatacadastro(datacadastro);
    dadosadd.setValor(rs.getDouble("Valor"));
    dadosadd.setAtivo(rs.getString("Ativo"));
    ListaDados.add(dadosadd);
}
}
```

```
String suname = "";
// HttpSession session=request.getSession(false);
//if(session!=null){
// suname = (String)session.getAttribute("uname");
//}
TBJEmpresa dadosadd = new TBJEmpresa();
dadosadd.setIdtbjempresa(1);
dadosadd.setNome("Vanderlei Soares de Oliveira");
//dadosadd.setDatacadastro(datacadastro);
dadosadd.setValor(10.00);
dadosadd.setAtivo("SIM");
//ListaDados.add(dadosadd);
```

```

        dadosadd = new TBJEmpresa();
        dadosadd.setIldtbjempresa(2);
        dadosadd.setNome("Jussara Storch");
        //dadosadd.setDatacadastro(datacadastro);
        dadosadd.setValor(10.00);
        dadosadd.setAtivo("SIM");
        //ListaDados.add(dadosadd);

        dadosadd = new TBJEmpresa();
        dadosadd.setIldtbjempresa(3);
        dadosadd.setNome("Davi");
        //dadosadd.setDatacadastro(datacadastro);
        dadosadd.setValor(10.00);
        dadosadd.setAtivo("SIM");
        //ListaDados.add(dadosadd);

        model.addAttribute("ListaEmpresa",ListaDados);

        return "/tbjEmpresa/tbjEmpresaIndex";
    }

    @RequestMapping("/tbjEmpresaCreate")
    public String tbjEmpresaCreate() {

        return "/tbjEmpresa/tbjEmpresaCreate";
    }

    //pegando dados do parametros
    //@RequestMapping("/tbjEmpresaEdit")
    @RequestMapping(value="/tbjEmpresaEdit")
    public String tbjEmpresaEdit(@RequestParam int id, Model model) throws Exception {

```

```
//model.addAttribute("idExibir",id);  
tbjempresaDal dal = new tbjempresaDal();  
TBJEmpresa dados = new TBJEmpresa();  
dados = dal.findById(id);  
model.addAttribute("dadosEmpr",dados);  
  
return "/tbjEmpresa/tbjEmpresaEdit";  
}
```

```
@RequestMapping("/saveTBJEmpresa")  
public String saveTBJEmpresa(Model model, HttpServletRequest request ) {  
    //HttpServletResponse response  
    //String nomeCliente = request.getParameter("txtnome");  
  
    tbjempresaDal dal = new tbjempresaDal();  
    TBJEmpresa dadosadd = new TBJEmpresa();  
    String stexto = "";  
  
    try{  
  
        dadosadd.setIdtbjempresa(Integer.parseInt(request.getParameter("txtidtbjempresa")));  
        dadosadd.setNome(request.getParameter("txtnome"));  
        java.util.Date date = Calendar.getInstance().getTime();  
        java.sql.Date sqlDate = new java.sql.Date(date.getTime());  
        dadosadd.setDatacadastro(sqlDate);  
        dadosadd.setValor(10.00);  
        dadosadd.setAtivo("SIM");  
  
        stexto = dal.adicionar(dadosadd);  
    }  
}
```

```

}catch(Exception ex){
    // throw ex;
}

//HttpSession session = request.getSession();
//session.setAttribute("uname","vanderlei veio do create");

//stexto = "Id="+Integer.toString(dadosadd.getIdtbjempresa());
//stexto += "Nome="+dadosadd.getNome();

model.addAttribute("textoexibir",stexto);

//redirect url
return "redirect:/tbjEmpresalIndex";

}

@RequestMapping("/alterarTBJEmpresa")
public String alterarTBJEmpresa(Model model, HttpServletRequest request ) {

    tbjempresaDal dal = new tbjempresaDal();
    TBJEmpresa dadosadd = new TBJEmpresa();
    String stexto = "";

    try{

        dadosadd.setIdtbjempresa(Integer.parseInt(request.getParameter("txtidtbjempresa")));

```

```

dadosadd.setNome(request.getParameter("txtnome"));
java.util.Date date = Calendar.getInstance().getTime();
java.sql.Date sqlDate = new java.sql.Date(date.getTime());
dadosadd.setDatacadastro(sqlDate);
dadosadd.setValor(10.00);
dadosadd.setAtivo("SIM");

dal.alterando(dadosadd);

}catch(Exception ex){
    // throw ex;
}

//return "/tbjEmpresa/tbjEmpresalIndex";
return "redirect:/tbjEmpresalIndex";

}

}

```

=====

dispatcher-servlet.xml

```

<?xml version='1.0' encoding='UTF-8' ?>
<!-- was: <?xml version="1.0" encoding="UTF-8"?> -->
<beans xmlns="http://www.springframework.org/schema/beans"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

```

```
xmlns:p="http://www.springframework.org/schema/p"
xmlns:aop="http://www.springframework.org/schema/aop"
xmlns:tx="http://www.springframework.org/schema/tx"
xmlns:mvc="http://www.springframework.org/schema/mvc"
xmlns:context="http://www.springframework.org/schema/context"

xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans-4.0.xsd

http://www.springframework.org/schema/aop
http://www.springframework.org/schema/aop/spring-aop-4.0.xsd

http://www.springframework.org/schema/tx
http://www.springframework.org/schema/tx/spring-tx-4.0.xsd

http://www.springframework.org/schema/mvc
http://www.springframework.org/schema/mvc/spring-mvc-4.0.xsd

http://www.springframework.org/schema/context
http://www.springframework.org/schema/context/spring-context-4.0.xsd ">
```

```
<mvc:annotation-driven />
```

```
<context:component-scan base-package="com.api.*" />
```

```
<bean id="viewResolver"
```

```
class="org.springframework.web.servlet.view.InternalResourceViewResolver"
```

```
p:prefix="/WEB-INF/jsp/"
```

```
p:suffix=".jsp" />
```

```
<mvc:resources mapping="/resources/**" location="/resources/" />
```

```
<context:component-scan base-package="org.springframework.rest" />
```

```
</beans>
```

```
=====
```



web.xml

```
<?xml version="1.0" encoding="UTF-8"?>

<web-app version="3.1" xmlns="http://xmlns.jcp.org/xml/ns/javaee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd">

  <context-param>

    <param-name>contextConfigLocation</param-name>

    <param-value>/WEB-INF/applicationContext.xml</param-value>

  </context-param>

  <listener>

    <listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>

  </listener>

  <servlet>

    <servlet-name>dispatcher</servlet-name>

    <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>

    <load-on-startup>2</load-on-startup>

  </servlet>

  <servlet>

    <servlet-name>NewServlet</servlet-name>

    <servlet-class>com.api.controller.NewServlet</servlet-class>

  </servlet>

  <servlet-mapping>

    <servlet-name>dispatcher</servlet-name>

    <url-pattern>/</url-pattern>

  </servlet-mapping>

  <servlet-mapping>

    <servlet-name>NewServlet</servlet-name>

    <url-pattern>/NewServlet</url-pattern>

  </servlet-mapping>

  <session-config>
```

```
<session-timeout>
    30
</session-timeout>
</session-config>
<welcome-file-list>
    <welcome-file>redirect.jsp</welcome-file>
</welcome-file-list>
</web-app>
```

=====

Index.jsp

```
<%--
```

```
Document : index
```

```
Created on : 07/09/2019, 17:58:41
```

```
Author : sovan
```

```
--%>
```

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
```

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
```

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
```

```
<title>JSP Page</title>
```

```
<link rel="stylesheet" href="<c:url value="/resources/css/bootstrap.css" />">
```

```
<link rel="stylesheet" href="<c:url value="/resources/css/bootstrap.min.css" />">
```

```
<script src="<c:url value="/resources/js/jquery.min.js" />"></script>
```

```
<script src="<c:url value="/resources/js/bootstrap.min.js" />"></script>
<style>
/* Remove the navbar's default margin-bottom and rounded borders */
.navbar {
  margin-bottom: 0;
  border-radius: 0;
}

/* Set height of the grid so .sidenav can be 100% (adjust as needed) */
.row.content {height: 450px}

/* Set gray background color and 100% height */
.sidenav {
  padding-top: 20px;
  background-color: #f1f1f1;
  height: 100%;
}

/* Set black background color, white text and some padding */
footer {
  background-color: #555;
  color: white;
  padding: 15px;
}

/* On small screens, set height to 'auto' for sidenav and grid */
@media screen and (max-width: 767px) {
  .sidenav {
    height: auto;
    padding: 15px;
  }
}
```

```
.row.content {height:auto;}
}
</style>
</head>
<body>
  <nav class="navbar navbar-inverse">
<div class="container-fluid">
  <div class="navbar-header">
    <button type="button" class="navbar-toggle" data-toggle="collapse" data-
target="#myNavbar">
      <span class="icon-bar"></span>
      <span class="icon-bar"></span>
      <span class="icon-bar"></span>
    </button>
    <a class="navbar-brand" href="#">Logo</a>
  </div>
<div class="collapse navbar-collapse" id="myNavbar">
  <ul class="nav navbar-nav">
    <li class="active"><a href="#home">Principal</a></li>
    <li><a href="tbjEmpresaIndex">Empresa</a></li>
    <li><a href="#">Projects</a></li>
    <li><a href="#">Contact</a></li>
  </ul>
  <ul class="nav navbar-nav navbar-right">
    <li><a href="#"><span class="glyphicon glyphicon-log-in"></span> Login</a></li>
  </ul>
</div>
</div>
</nav>

<div class="container-fluid text-center">
```

```
<div class="row content">
```

```
  <div class="col-sm-2 sidenav">
```

```
    <p><a href="#">Link</a></p>
```

```
    <p><a href="#">Link</a></p>
```

```
    <p><a href="#">Link</a></p>
```

```
  </div>
```

```
  <div class="col-sm-8 text-left">
```

```
    <h1>Welcome</h1>
```

```
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.</p>
```

```
    <hr>
```

```
    <h3>Test</h3>
```

```
    <p>Lorem ipsum...</p>
```

```
  </div>
```

```
<div class="col-sm-2 sidenav">
```

```
  <div class="well">
```

```
    <p>ADS</p>
```

```
  </div>
```

```
  <div class="well">
```

```
    <p>ADS</p>
```

```
  </div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
<footer class="container-fluid text-center">
```

```
  <p>Footer Text</p>
```

```
</footer>
```

```
</body>
</html>
```

```
=====
```

Criar uma pasta empresa

```
<%--
```

```
Document : tbjEmpresalIndex
```

```
Created on : 09/09/2019, 09:29:21
```

```
Author : sovan
```

```
--%>
```

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
```

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
```

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
```

```
<title>JSP Page</title>
```

```
<link rel="stylesheet" href="<c:url value="/resources/css/bootstrap.css" />">
```

```
<link rel="stylesheet" href="<c:url value="/resources/css/bootstrap.min.css" />">
```

```
<script src="<c:url value="/resources/js/jquery.min.js" />"></script>
```

```
<script src="<c:url value="/resources/js/bootstrap.min.js" />"></script>
```

```
</head>
```

```
<body>
```

```
<h2>Lista Usuários</h2>
```

```
<div class="container"
```

```
<div>
```

```

<p>
    <a title="Novo" href="tbjEmpresaCreate" class="btn btn btn-default command-
news"><i class="glyphicon glyphicon-new-window"></i></a>

    <a title="Atualizar" href="tbjEmpresalIndex" class="btn btn btn-default command-
news"><i class="glyphicon glyphicon-refresh"></i></a>

    ${textoexibir}
</p>
<br>

<br />

<table id="grid-command-buttons" class="table table-condensed table-hover table-
striped bootgrid-table table-bordered" aria-busy="false">
    <thead>
        <tr>
            <th data-column-id="id" data-identifier="false" data-type="numeric">
                Codigo
            </th>
            <th data-column-id="NOME">
                Nome Empresa
            </th>
            <th data-column-id="ATIVO">
                Ativo
            </th>
            <th data-column-id="commands" data-formatter="commands" data-
sortable="false">Controles</th>
        </tr>
    </thead>
    <tbody>

        <c:forEach var="itemdados" items="${ListaEmpresa}">
            <tr>

```

```
<td>
<c:out value="\${itemdados.getIdtbjempresa()}" />
</td>
<td>
<c:out value="\${itemdados.getNome()}" />
</td>
<td>
<c:out value="\${itemdados.getAtivo()}" />
</td>
<td>
<a title="Atualizar"
href="tbjEmpresaEdit?id=\${itemdados.getIdtbjempresa()}" >clique aqui</a>
</td>
</tr>
```

```
</c:forEach>
```

```
</tbody>
```

```
</table>
```

```
</div>
```

```
</div>
```

```
</body>
```

```
</html>
```

```
=====
```

```
<%--
```



Document : tbjEmpresaEdit

Created on : 09/09/2019, 09:43:13

Author : sovan

--%>

<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>JSP Page</title>

<link rel="stylesheet" href="<c:url value="/resources/css/bootstrap.css" />">

<link rel="stylesheet" href="<c:url value="/resources/css/bootstrap.min.css" />">

<script src="<c:url value="/resources/js/jquery.min.js" />"></script>

<script src="<c:url value="/resources/js/bootstrap.min.js" />"></script>

</head>

<body>

<h2>Edite</h2>

<div class="form-horizontal">

<h4>Dados</h4>

<hr />

<form action="alterarTBJEmpresa" cssClass="form-horizontal" method="POST" >

<div class="col-lg-9">

<div class="form-group">

<Label class = "control-label col-md-2">Codigo</Label>

<div class="col-md-2">

<input type="text" id="txtidtbjempresa" name="txtidtbjempresa" class =  
"form-control input-default" readonly value="{dadosEmpr.getIdtbjempresa()}" />

</div>

```
</div>
```

```
<div class="form-group">
```

```
  <Label class = "control-label col-md-2">Nome</Label>
```

```
  <div class="col-md-2">
```

```
    <input type="text" id="txtnome" name="txtnome" class = "form-control input-  
default" required="" value="{dadosEmpr.getNome()}" />
```

```
  </div>
```

```
</div>
```

```
<div class="form-group">
```

```
  <Label class = "control-label col-md-2">datacadastro</Label>
```

```
  <div class="col-md-2">
```

```
    <input type="text" id="txtdatacadastro" name="txtdatacadastro" class = "form-  
control input-default" value="{dadosEmpr.getDatacadastro()}" />
```

```
  </div>
```

```
</div>
```

```
<div class="form-group">
```

```
  <Label class = "control-label col-md-2">valor</Label>
```

```
  <div class="col-md-2">
```

```
    <input type="text" id="txtvalor" name="txtvalor" class = "form-control input-  
default" value="{dadosEmpr.getValor()}" />
```

```
  </div>
```

```
</div>
```

```
<div class="form-group">
```

```
  <Label class = "control-label col-md-2">ativo</Label>
```

```
  <div class="col-md-2">
```

```
    <input type="text" id="txtativo" name="txtativo" ativo = "form-control input-  
default" value="{dadosEmpr.getAtivo()}" />
```

```
  </div>
```

```
</div>
```

```
</div>

<div class="form-group">
  <div class="col-md-offset-2 col-md-10">
    <input type="submit" value="Salvar" class="btn btn-default" />
  </div>
</div>
</form>

</div>

<div>
  <a title="Voltar" href="#" class="btn btn-default news"><i class="glyphicon glyphicon-backward"></i></a>
</div>

</body>
</html>
```

=====

<%--

Document : tbjEmpresaCreate

Created on : 09/09/2019, 09:31:10

Author : sovan

--%>

<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>

<%@page contentType="text/html" pageEncoding="UTF-8"%>

```
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>JSP Page</title>
    <link rel="stylesheet" href="<c:url value="/resources/css/bootstrap.css" />">
    <link rel="stylesheet" href="<c:url value="/resources/css/bootstrap.min.css" />">
    <script src="<c:url value="/resources/js/jquery.min.js" />"></script>
    <script src="<c:url value="/resources/js/bootstrap.min.js" />"></script>
  </head>
  <body>
    <h2>Novo</h2>

    <div class="form-horizontal">
      <h4>Dados</h4>
      <hr />
      <form action="saveTBJEmpresa" cssClass="form-horizontal" method="POST" >
        <div class="col-lg-9">

          <div class="form-group">
            <Label class = "control-label col-md-2">Codigo</Label>
            <div class="col-md-2">
              <input type="text" id="txtidtbjempresa" name="txtidtbjempresa" class = "form-
control input-default" required="" value="" />
            </div>
          </div>

          <div class="form-group">
            <Label class = "control-label col-md-2">Nome</Label>
            <div class="col-md-2">
              <input type="text" id="txtnome" name="txtnome" class = "form-control input-
default" required="" value="" />
            </div>
          </div>
        </div>
      </form>
    </div>
  </body>
</html>
```

```
</div>
```

```
</div>
```

```
<div class="form-group">
```

```
<Label class = "control-label col-md-2">datacadastro</Label>
```

```
<div class="col-md-2">
```

```
<input type="text" id="txtdatacadastro" name="txtdatacadastro" class = "form-control input-default" value="" />
```

```
</div>
```

```
</div>
```

```
<div class="form-group">
```

```
<Label class = "control-label col-md-2">valor</Label>
```

```
<div class="col-md-2">
```

```
<input type="text" id="txtvalor" name="txtvalor" class = "form-control input-default" value="" />
```

```
</div>
```

```
</div>
```

```
<div class="form-group">
```

```
<Label class = "control-label col-md-2">ativo</Label>
```

```
<div class="col-md-2">
```

```
<input type="text" id="txtativo" name="txtativo" ativo = "form-control input-default" value="" />
```

```
</div>
```

```
</div>
```

```
</div>
```

```
<div class="form-group">
```

```
<div class="col-md-offset-2 col-md-10">
```

```
        <input type="submit" value="Salvar" class="btn btn-default" />
    </div>
</div>
</form>

</div>

<div>
    <a title="Voltar" href="#" class="btn btn-default news"><i class="glyphicon glyphicon-backward"></i></a>
</div>

</body>
</html>
```

=====

=====

NetBeans - SOAP Spring MVC

Tem de por neste arquivos

Colocar dispatcher-servlet.xml

```
<context:component-scan base-package="org.springframework.rest" />
```

criar classe normal

que vai ser a controller

```
/*
```

```
* To change this license header, choose License Headers in Project Properties.
```

```
* To change this template file, choose Tools | Templates
```

```
* and open the template in the editor.
```

```
*/
```

```
package com.api.controller;
```

```
import com.api.dao.tbjempresaDal;
```

```
import com.api.modelCurso;
```

```
import com.api.model.TBJEmpresa;
```

```
import java.util.Calendar;
```

```
import org.springframework.beans.factory.annotation.Autowired;
```

```
import org.springframework.stereotype.Controller;
```

```
import org.springframework.web.bind.annotation.RequestMapping;
```

```
import org.springframework.web.bind.annotation.RequestMethod;
```

```
import org.springframework.web.bind.annotation.ResponseBody;
```

```
/**
```

```
*
```

```
* @author sovan
```

```
*/
```

```
@Controller
```

```
public class restEmpresaController {
```

```
    @Autowired
```

```
    private tbjempresaDal dal;
```

```

//private JSONObject jsonObject;

//tem de por neste arquivos

//colocar dispatcher-servlet.xml

//<context:component-scan base-package="org.springframework.rest" />

//@Autowired

//private tbjempresaDal dal = new tbjempresaDal();

//private Map<Integer, Curso> cursos;

/**
 * Simply selects the home view to render by returning its name.
 */

@RequestMapping(value = "/empresarest", method = RequestMethod.GET)
@ResponseBody
public TBJEmpresa exibirDadosEmpresa() {
    TBJEmpresa dadosretorno = new TBJEmpresa();
    dadosretorno.setIdtbjempresa(1);
    dadosretorno.setNome("Vanderlei Soares de Oliveira");
    java.util.Date date = Calendar.getInstance().getTime();
    java.sql.Date sqlDate = new java.sql.Date(date.getTime());
    dadosretorno.setDatacadastro(sqlDate);
    dadosretorno.setValor(10.00);
    dadosretorno.setAtivo("SIM");
    return dadosretorno;
}

@RequestMapping(value = "/empresarestjson", method = RequestMethod.GET)
@ResponseBody
public String exibirDadosEmpresajson() {
    TBJEmpresa dadosretorno = new TBJEmpresa();

```



```

dadosretorno.setIIdtobjempresa(1);

dadosretorno.setNome("Vanderlei Soares de Oliveira");

java.util.Date date = Calendar.getInstance().getTime();

java.sql.Date sqlDate = new java.sql.Date(date.getTime());

dadosretorno.setDatacadastro(sqlDate);

dadosretorno.setValor(10.00);

dadosretorno.setAtivo("SIM");

String myJson = "{\n" +
    "  user: {\n" +
    "    id: 123456,\n" +
    "    name: \"Neto Marin\",\n" +
    "    username: \"netomarin\",\n" +
    "    email: \"netomarin@globalcode.com.br\"\n" +
    "  }\n" +
    "}";

//Gson gson = new Gson();

//myJson = gson.toJson(dadosretorno);

//dal = new tbjempresaDal();

//myJson =dal.GetJson();

return myJson;

}

//@RequestMapping(value = "/cursos", method = RequestMethod.GET)
public Curso listar() {

    //ResponseEntity<List<Curso>>

    //cursos = new HashMap<Integer, Curso>();

    Curso c1 = new Curso(1, "Workshop Rest", "24hs");

```

```

//Curso c2 = new Curso(2, "Workshop Spring MVC", "24hs");
//Curso c3 = new Curso(3, "Desenvolvimento Web com JSF 2", "60hs");

//cursos.put(1, c1);
//cursos.put(2, c2);
//cursos.put(3, c3);

//return new ResponseEntity<List<Curso>>(new ArrayList<Curso>(cursos.values()),
HttpStatus.OK);
return c1;

}

}

```

\*\*\*\*\*

class modal para usar no soap

```

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

```

```

package com.api.model;

```

```

import java.sql.Date;
import javax.xml.bind.annotation.XmlRootElement;

```

```

/**
 *
 * @author sovan
 */

```

@XmlElement

```
public class TBJEmpresa {
```

```
    private int idtbjempresa;
```

```
    private String nome;
```

```
    private Date datacadastro;
```

```
    private Double valor;
```

```
    private String ativo;
```

```
    public int getIdtbjempresa() {
```

```
        return idtbjempresa;
```

```
    }
```

```
    public void setIdtbjempresa(int idtbjempresa) {
```

```
        this.idtbjempresa = idtbjempresa;
```

```
    }
```

```
    public String getNome() {
```

```
        return nome;
```

```
    }
```

```
    public void setNome(String nome) {
```

```
        this.nome = nome;
```

```
    }
```

```
    public Date getDatacadastro() {
```

```
        return datacadastro;
```

```
    }
```

```
public void setDatacadastro(Date datacadastro) {  
    this.datacadastro = datacadastro;  
}
```

```
public Double getValor() {  
    return valor;  
}
```

```
public void setValor(Double valor) {  
    this.valor = valor;  
}
```

```
public String getAtivo() {  
    return ativo;  
}
```

```
public void setAtivo(String ativo) {  
    this.ativo = ativo;  
}
```

```
}
```

```
=====
```

```
=====
```

```
<%@ taglib prefix="spring" uri="http://www.springframework.org/tags"%>
<%@ taglib prefix="form" uri="http://www.springframework.org/tags/form"%>
```

```
    <form:form method="post" modelAttribute="userForm" action="${userActionUrl}">
        <form:input path="name" type="text" /> <!-- bind to user.name-->
        <form:errors path="name" />
    </form:form>
```

```
=====
```

```
<%@ page session="false"%>
<%@ taglib prefix="spring" uri="http://www.springframework.org/tags"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt"%>
```

```
<jsp:include page="../fragments/header.jsp" />
```

```
=====
```

```
<c:if test="${not empty msg}">
    <div class="alert alert-${css} alert-dismissible" role="alert">
        <button type="button" class="close" data-dismiss="alert"
            aria-label="Close">
            <span aria-hidden="true">x</span>
        </button>
        <strong>${msg}</strong>
    </div>
</c:if>
```

=====

```
<button class="btn btn-info" onclick="location.href='${userUrl}'">Query</button>
```

=====

```
<spring:bind path="address">
  <div class="form-group ${status.error ? 'has-error' : ''}">
    <label class="col-sm-2 control-label">Address</label>
    <div class="col-sm-10">
      <form:textarea path="address" rows="5" class="form-control"
        id="address" placeholder="address" />
      <form:errors path="address" class="control-label" />
    </div>
  </div>
</spring:bind>
```

```
<spring:bind path="framework">
  <div class="form-group ${status.error ? 'has-error' : ''}">
    <label class="col-sm-2 control-label">Web Frameworks</label>
    <div class="col-sm-10">
      <form:checkboxes path="framework" items="${frameworkList}"
        element="label class='checkbox-inline'" />
      <br />
      <form:errors path="framework" class="control-label" />
    </div>
  </div>
</spring:bind>
```

```
<div class="form-group">
```

```
<div class="col-sm-offset-2 col-sm-10">
  <c:choose>
    <c:when test="{userForm['new']}">
      <button type="submit" class="btn-lg btn-primary pull-right">Add
    </button>
    </c:when>
    <c:otherwise>
      <button type="submit" class="btn-lg btn-primary pull-right">Update
    </button>
    </c:otherwise>
  </c:choose>
</div>
</div>
```

=====

```
<c:choose>
  <c:when test="{userForm['new']}">
    <h1>Add User</h1>
  </c:when>
  <c:otherwise>
    <h1>Update User</h1>
  </c:otherwise>
</c:choose>
<br />
```

=====

```
<%@ page session="false"%>
<%@ taglib prefix="spring" uri="http://www.springframework.org/tags"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<%@ taglib prefix="form" uri="http://www.springframework.org/tags/form"%>
<%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt"%>
```

```
<!DOCTYPE html>
<html lang="en">
```

```
<jsp:include page="../../fragments/header.jsp" />
```

```
<div class="container">
```

```
    <c:choose>
        <c:when test="${userForm['new']}">
            <h1>Add User</h1>
        </c:when>
        <c:otherwise>
            <h1>Update User</h1>
        </c:otherwise>
    </c:choose>
```

```
</div>
```

```
<br />
```

```
<spring:url value="/users" var="userActionUrl" />
```

```
<form:form class="form-horizontal" method="post"
    modelAttribute="userForm" action="${userActionUrl}">
```

```
    <form:hidden path="id" />
```

```
    <spring:bind path="name">
```



```
<div class="form-group ${status.error ? 'has-error' : ''}">
  <label class="col-sm-2 control-label">Name</label>
  <div class="col-sm-10">
    <form:input path="name" type="text" class="form-control"
  id="name" placeholder="Name" />
    <form:errors path="name" class="control-label" />
  </div>
</div>
</spring:bind>
```

```
<spring:bind path="email">
  <div class="form-group ${status.error ? 'has-error' : ''}">
    <label class="col-sm-2 control-label">Email</label>
    <div class="col-sm-10">
      <form:input path="email" class="form-control"
  id="email" placeholder="Email" />
      <form:errors path="email" class="control-label" />
    </div>
  </div>
</spring:bind>
```

```
<spring:bind path="password">
  <div class="form-group ${status.error ? 'has-error' : ''}">
    <label class="col-sm-2 control-label">Password</label>
    <div class="col-sm-10">
      <form:password path="password" class="form-control"
  id="password" placeholder="password" />
      <form:errors path="password" class="control-label" />
    </div>
  </div>
</spring:bind>
```

```
<spring:bind path="confirmPassword">
  <div class="form-group ${status.error ? 'has-error' : ''}">
    <label class="col-sm-2 control-label">confirm Password</label>
    <div class="col-sm-10">
      <form:password path="confirmPassword" class="form-control"
id="password" placeholder="password" />
      <form:errors path="confirmPassword" class="control-label" />
    </div>
  </div>
</spring:bind>
```

```
<spring:bind path="address">
  <div class="form-group ${status.error ? 'has-error' : ''}">
    <label class="col-sm-2 control-label">Address</label>
    <div class="col-sm-10">
      <form:textarea path="address" rows="5" class="form-control"
id="address" placeholder="address" />
      <form:errors path="address" class="control-label" />
    </div>
  </div>
</spring:bind>
```

```
<spring:bind path="newsletter">
  <div class="form-group ${status.error ? 'has-error' : ''}">
    <label class="col-sm-2 control-label">Newsletter</label>
    <div class="col-sm-10">
      <div class="checkbox">
        <label>
          <form:checkbox path="newsletter" id="newsletter" />
        </label>
      </div>
    </div>
  </div>
</spring:bind>
```

```

        <form:errors path="newsletter" class="control-label" />
    </div>
</div>
</div>
</spring:bind>

<spring:bind path="framework">
    <div class="form-group ${status.error ? 'has-error' : ''}">
        <label class="col-sm-2 control-label">Web Frameworks</label>
        <div class="col-sm-10">
            <form:checkboxes path="framework"
items="${frameworkList}"
            element="label class='checkbox-inline'" />
            <br />
            <form:errors path="framework" class="control-label" />
        </div>
    </div>
</spring:bind>

<spring:bind path="sex">
    <div class="form-group ${status.error ? 'has-error' : ''}">
        <label class="col-sm-2 control-label">Sex</label>
        <div class="col-sm-10">
            <label class="radio-inline">
                <form:radio button path="sex" value="M" /> Male
            </label>
            <label class="radio-inline">
                <form:radio button path="sex" value="F" /> Female
            </label> <br />
            <form:errors path="sex" class="control-label" />
        </div>
    </div>
</spring:bind>

```

```

</div>
</spring:bind>

<spring:bind path="number">
  <div class="form-group ${status.error ? 'has-error' : ''}">
    <label class="col-sm-2 control-label">Number</label>
    <div class="col-sm-10">
      <form:radiobuttons path="number" items="${numberList}"
element="label class='radio-inline'" />
      <br />
      <form:errors path="number" class="control-label" />
    </div>
  </div>
</spring:bind>

<spring:bind path="country">
  <div class="form-group ${status.error ? 'has-error' : ''}">
    <label class="col-sm-2 control-label">Country</label>
    <div class="col-sm-5">
      <form:select path="country" class="form-control">
        <form:option value="NONE" label="--- Select ---" />
        <form:options items="${countryList}" />
      </form:select>
      <form:errors path="country" class="control-label" />
    </div>
  </div>
</spring:bind>

<spring:bind path="skill">
  <div class="form-group ${status.error ? 'has-error' : ''}">

```

```
<label class="col-sm-2 control-label">Java Skills</label>

<div class="col-sm-5">
    <form:select path="skill" items="{javaSkillList}"
multiple="true" size="5" class="form-control" />
    <form:errors path="skill" class="control-label" />
</div>
<div class="col-sm-5"></div>
</div>
</spring:bind>

<div class="form-group">
<div class="col-sm-offset-2 col-sm-10">
    <c:choose>
        <c:when test="{userForm['new']}">
            <button type="submit" class="btn-lg btn-primary pull-right">Add
</button>
        </c:when>
        <c:otherwise>
            <button type="submit" class="btn-lg btn-primary pull-right">Update
</button>
        </c:otherwise>
    </c:choose>
</div>
</div>
</form:form>

</div>

<jsp:include page="../../fragments/footer.jsp" />

</body>
```

```
</html>
```

```
=====
```

```
<form:form method="POST" action="${pageContext.request.contextPath}/showPage">
```

```
  <input type="submit" value="Click here!" name="button" />
```

```
</form:form>
```

```
=====
```

```
<form:select path="house">
```

```
  <form:option value="Gryffindor" />
```

```
  <form:option value="Hufflepuff" />
```

```
  <form:option value="Ravenclaw" />
```

```
  <form:option value="Slytherin" />
```

```
</form:select>
```

```
=====
```

```
<form:select path="country">
```

```
  <form:option value="-" label="--Please Select" />
```

```
  <form:options items="${countryList}" itemValue="code" itemLabel="name" />
```

```
</form:select>
```

=====

```
<tr>
  <td>Sex:</td>
  <td>
    Male:
    <form:radio button path="sex" value="M" /> <br/> Female:
    <form:radio button path="sex" value="F" />
  </td>
</tr>
```

=====

```
<tr>
  <td>Sex:</td>
  <td>
    <form:radio buttons path="sex" items="{sexOptions}" />
  </td>
</tr>
```

=====

```
<tr>
  <td>Password:</td>
  <td>
    <form:password path="password" />
  </td>
</tr>
```

=====

```
<tr>
  <td>House:</td>
  <td>
```

```

<form:select path="house">
    <form:option value="Gryffindor" />
    <form:option value="Hufflepuff" />
    <form:option value="Ravenclaw" />
    <form:option value="Slytherin" />
</form:select>
</td>
</tr>

```

=====

```

<tr>
    <td>Notes:</td>
    <td>
        <form:textarea path="notes" rows="3" cols="20" />
    </td>
    <td>
        <form:errors path="notes" />
    </td>
</tr>

```

=====

```

public class UserValidator implements Validator {

    public boolean supports(Class candidate) {
        return User.class.isAssignableFrom(candidate);
    }

    public void validate(Object obj, Errors errors) {
        ValidationUtils.rejectIfEmptyOrWhitespace(errors, "firstName", "required", "Field is
required.");
        ValidationUtils.rejectIfEmptyOrWhitespace(errors, "lastName", "required", "Field is
required.");
    }
}

```



```
}  
}
```

```
<form:form>
```

```
<table>
```

```
<tr>
```

```
<td>First Name:</td>
```

```
<td>
```

```
<form:input path="firstName" />
```

```
</td>
```

```
<%-- Show errors for firstName field --%>
```

```
<td>
```

```
<form:errors path="firstName" />
```

```
</td>
```

```
</tr>
```

```
<tr>
```

```
<td>Last Name:</td>
```

```
<td>
```

```
<form:input path="lastName" />
```

```
</td>
```

```
<%-- Show errors for lastName field --%>
```

```
<td>
```

```
<form:errors path="lastName" />
```

```
</td>
```

```
</tr>
```

```
<tr>
```

```
<td colspan="3">
```

```
<input type="submit" value="Save Changes" />
```

```
</td>
```

```
</tr>
```

```
</table>
</form:form>
```

```
<form method="POST">
  <table>
    <tr>
      <td>First Name:</td>
      <td><input name="firstName" type="text" value="" /></td>
      <!-- Associated errors to firstName field displayed --%>
      <td><span name="firstName.errors">Field is required.</span></td>
    </tr>

    <tr>
      <td>Last Name:</td>
      <td><input name="lastName" type="text" value="" /></td>
      <!-- Associated errors to lastName field displayed --%>
      <td><span name="lastName.errors">Field is required.</span></td>
    </tr>

    <tr>
      <td colspan="3">
        <input type="submit" value="Save Changes" />
      </td>
    </tr>
  </table>
</form>
```

=====

=====

=====

=====

=====

```
package br.com.treinaweb.springbootapi.controller;

import java.util.List;
import java.util.Optional;

import javax.validation.Valid;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.RestController;

import br.com.treinaweb.springbootapi.entity.Pessoa;
import br.com.treinaweb.springbootapi.repository.PessoaRepository;

@RestController
public class PessoaController {

    @Autowired
    private PessoaRepository _pessoaRepository;
```

```
@RequestMapping(value = "/pessoa", method = RequestMethod.GET)
```

```
public List<Pessoa> Get() {  
    return _pessoaRepository.findAll();  
}
```

```
@RequestMapping(value = "/pessoa/{id}", method = RequestMethod.GET)
```

```
public ResponseEntity<Pessoa> GetById(@PathVariable(value = "id") long id)  
{  
    Optional<Pessoa> pessoa = _pessoaRepository.findById(id);  
    if(pessoa.isPresent())  
        return new ResponseEntity<Pessoa>(pessoa.get(), HttpStatus.OK);  
    else  
        return new ResponseEntity<>(HttpStatus.NOT_FOUND);  
}
```

```
@RequestMapping(value = "/pessoa", method = RequestMethod.POST)
```

```
public Pessoa Post(@Valid @RequestBody Pessoa pessoa)  
{  
    return _pessoaRepository.save(pessoa);  
}
```

```
@RequestMapping(value = "/pessoa/{id}", method = RequestMethod.PUT)
```

```
public ResponseEntity<Pessoa> Put(@PathVariable(value = "id") long id, @Valid  
@RequestBody Pessoa newPessoa)  
{  
    Optional<Pessoa> oldPessoa = _pessoaRepository.findById(id);  
    if(oldPessoa.isPresent()){  
        Pessoa pessoa = oldPessoa.get();  
        pessoa.setNome(newPessoa.getNome());  
        _pessoaRepository.save(pessoa);  
    }  
}
```

```
        return new ResponseEntity<Pessoa>(pessoa, HttpStatus.OK);
    }
    else
        return new ResponseEntity<>(HttpStatus.NOT_FOUND);
    }

    @RequestMapping(value = "/pessoa/{id}", method = RequestMethod.DELETE)
    public ResponseEntity<Object> Delete(@PathVariable(value = "id") long id)
    {
        Optional<Pessoa> pessoa = _pessoaRepository.findById(id);
        if(pessoa.isPresent()){
            _pessoaRepository.delete(pessoa.get());
            return new ResponseEntity<>(HttpStatus.OK);
        }
        else
            return new ResponseEntity<>(HttpStatus.NOT_FOUND);
    }
}
```

=====

=====

AUTOMAÇÃO DE TESTE

JUNIT

criar o projeto

<https://start.spring.io/>

++++++

+++++

## CONTROLLER

criar o controller normalmente

depois para fazer o teste vai na pasta

src/test/java

criar o pacote controller

criar a class com o mesmo no do controller mas colocar

o Teste na frente do nome

Ex: TesteHomeController.java

```
package com.api.springbootJUnit02.controller;
```

```
import org.springframework.stereotype.Controller;
```

```
import org.springframework.ui.Model;
```

```
import org.springframework.web.bind.annotation.RequestMapping;
```

```
@Controller
```

```
public class HomeController {
```

```
    @RequestMapping(value= {"", "/", "/index"})
```

```
    public String home(Model model) {
```

```
        System.out.println("controller lendo");
```

```
        return "index";
```

```
    }
```

```
}
```

+++++

```
package com.api.springbootJunit02.controller;

import static org.springframework.test.web.servlet.request.MockMvcRequestBuilders.get;
import static org.springframework.test.web.servlet.result.MockMvcResultMatchers.content;
import static org.springframework.test.web.servlet.result.MockMvcResultMatchers.status;

import org.junit.Before;
import org.junit.Test;
import org.junit.runner.RunWith;
import org.mockito.InjectMocks;
import org.mockito.MockitoAnnotations;
import org.springframework.boot.SpringBootConfiguration;
import org.springframework.http.MediaType;
import org.springframework.test.context.junit4.SpringRunner;
import org.springframework.test.web.servlet.MockMvc;
import org.springframework.test.web.servlet.setup.MockMvcBuilders;

@RunWith(SpringRunner.class)
@SpringBootConfiguration
public class TesteHomeController {

    private MockMvc mockMvc;

    @InjectMocks
    private HomeController homeController;

    @Before
```

```
public void setUp() throws Exception {  
    MockitoAnnotations.initMocks(this);  
    mockMvc = MockMvcBuilders.standaloneSetup(homeController).build();  
}
```

```
@Test
```

```
public void Teste_home() throws Exception {  
    System.out.println("Teste controller 001");  
    mockMvc.perform(get("/index")  
        .accept(MediaType.APPLICATION_JSON))  
        .andExpect(status().isOk())  
        .andExpect(content().string("index"));  
  
    System.out.println("Teste controller 002");  
    mockMvc.perform(get("/index")  
        .accept(MediaType.APPLICATION_JSON))  
        .andExpect(status().isOk())  
        .andExpect(content().string("index"));  
}
```



```
}
```

```
COMANDO
```

```
@Ignore
```

```
@Test
```

```
public void testName() {
```

```
}
```

```
@BeforeClass
```

```
public void setup() {
```

```
    dataBase = new DataBase();
```

```
    dataBaseConnection.open();
```

```
    dataBase.populate();
```

```
    list_name = new ArrayList<>();
```

```
    index = 0;
```

```
}
```

```
@AfterClass
```

```
public void close() {
```

```
    dataBase.drop();
```

```
    dataBaseConnection.close();
```

```
    dataBase = null;
```

```
    list_name = null;
```

```
    index = null;
```

```
}
```

```
@After
```

```
public void finalize() {
```

```
    list_name = calendar.getAllNames();
```

```
    index = current_index;
}
```

```
@Test(timeout = 500)
public void fastTestCase() {
    Assert.assertEquals(1500, calendar.getSize());
}
```

```
//TESTAR EXEÇÃO
```

```
@Test(expected=NullPointerException.class)
public void testException(int input) {
    int month = 4;
    Assert.assertNotNull(calendar.getAllDays(month));
}
```

```
@BeforeClass
```

```
public static void setup() {
}
```

```
@AfterClass
```

```
public static void close() {
}
```

```
import static org.junit.jupiter.api.Assertions.fail;
```

```
import static org.junit.jupiter.api.Assumptions.assumeTrue;
```

```
import org.junit.jupiter.api.AfterAll;
```

```
import org.junit.jupiter.api.AfterEach;
```

```
import org.junit.jupiter.api.BeforeAll;
import org.junit.jupiter.api.BeforeEach;
import org.junit.jupiter.api.Disabled;
import org.junit.jupiter.api.Test;
```

```
class StandardTests {
```

```
    @BeforeAll
```

```
    static void initAll() {
```

```
    }
```

```
    @BeforeEach
```

```
    void init() {
```

```
    }
```

```
    @Test
```

```
    void succeedingTest() {
```

```
    }
```

```
    @Test
```

```
    void failingTest() {
```

```
        fail("a failing test");
```

```
    }
```

```
    @Test
```

```
    @Disabled("for demonstration purposes")
```

```
    void skippedTest() {
```

```
        // not executed
```

```
    }
```

```
    @Test
```

```
void abortedTest() {
    assertTrue("abc".contains("Z"));
    fail("test should have been aborted");
}

@AfterEach
void tearDown() {
}

@AfterAll
static void tearDownAll() {
}

}
```

++++++

+++++

SERVICE - REPOSITORY

=====

=====

WEBDRIVE JAVA

=====

=====

=====

=====

=====

=====

=====

=====

=====

=====

=====

=====

=====

=====

=====

=====

=====

=====



=====

=====

=====

=====

=====

=====

=====

=====

=====

=====

=====

=====

=====

=====

=====

=====

=====

=====

=====

=====

=====

=====

=====

=====

CRIAR UM BAT PARA EXECUTAR JAR

@echo off

```
start javaw -jar C:\Temp\desctop01.jar
```

```
exit
```

```
=====
=====
```

```
COMANDO JAVA
```

```
+++++
```

```
com.api.springboot06
```

```
com.api.springboot06.controller
```

```
com.api.springboot06.email
```

```
com.api.springboot06.modal
```

```
com.api.springboot06.repository
```

```
com.api.springboot06.service
```

```
+++++
```

```
List<Integer> itens = Arrays.asList( 11, 10, 16, 5, 85 );
```

```
for(Integer item : itens){
```

```
    System.out.print(item);
```

```
}
```

```
+++++
```

```
List<Integer> itens = Arrays.asList( 11, 10, 16, 5, 85 );
```

```
items.forEach(item->System.out.println(item));
```

```
+++++
```

```
List<Integer> items = Arrays.asList( 11, 10, 16, 5, 85 );
```

```
items.forEach(System.out::println);
```

```
+++++
```

```
List<Integer> items = Arrays.asList( 11, 10, 16, 5, 85 );
```

```
for(Integer item : items){
```

```
    if ( item == 16 )
```

```
        System.out.println(item);
```

```
}
```

```
+++++
```

```
List<Integer> items = Arrays.asList( 11, 10, 16, 5, 85 );
```

```
items.stream().filter(i -> i == 16).forEach(System.out::println);
```

```
+++++
```

```
List<Integer> items = Arrays.asList( 11, 10, 16, 5, 85 );
```

```
items.stream().filter(i -> i > 16).forEach(System.out::println);
```

```
+++++
```

```
List<Object> aList;
```

```
for (int i = 0; i < aList.size(); i++)
```

```
{
    object = aList.get(i);
    //do something with i
}
```

+++++

```
final int size = aList.size();
for (int i = 0; i < size; i++)
{
    object = aList.get(i);
    //do something with i
}
```

+++++

```
ArrayList<String> namesList = new ArrayList<String>(Arrays.asList( "alex", "brian", "charles" ) );
```

```
ListIterator<String> listItr = namesList.listIterator();
```

```
while(listItr.hasNext())
{
    System.out.println(listItr.next());
}
```

+++++

```
List<String> items = new ArrayList<>();
```

```
    items.add("A");
```

```
    items.add("B");
```

```
    items.add("C");
```

```
    items.add("D");
```

```
    items.add("E");
```

```
//lambda
//Output : A,B,C,D,E
items.forEach(item->System.out.println(item));
```

```
//Output : C
items.forEach(item->{
    if("C".equals(item)){
        System.out.println(item);
    }
});
```

```
//method reference
//Output : A,B,C,D,E
items.forEach(System.out::println);
```

```
//Stream and filter
//Output : B
items.stream()
    .filter(s->s.contains("B"))
    .forEach(System.out::println);
```

```
+++++
```

```
switch(expression) {
```

```
case x:
```

```
    // code block
```

```
    break;
```

```
case y:
```

```
    // code block
```

```
    break;
```

```
default:  
    // code block  
}
```

```
+++++++  
int day = 4;  
switch (day) {  
    case 6:  
        System.out.println("Today is Saturday");  
        break;  
    case 7:  
        System.out.println("Today is Sunday");  
        break;  
    default:  
        System.out.println("Looking forward to the Weekend");  
}  
// Outputs "Looking forward to the Weekend"
```

```
+++++++  
  
try {  
    comando(s);  
} catch (Exception e) {  
    comando(s);  
}
```

```
+++++++  
void saveNetDevices() {  
    try {  
        FileOutputStream fos = new
```



```

FileOutputStream("NetDevices.txt");

ObjectOutputStream oos = new ObjectOutputStream(fos);

oos.writeObject(mNetDevices);

oos.close();

fos.close();

} catch (NullPointerException nexc) {

    JOptionPane.showMessageDialog(null, "NetDevices list is null - No devices saved", "Saving
NetDevices", JOptionPane.INFORMATION_MESSAGE);

} catch (InvalidClassException nexc) {

    JOptionPane.showMessageDialog(null, "NetDevices list is invalid - No devices saved", "Saving
NetDevices", JOptionPane.INFORMATION_MESSAGE);

} catch (NotSerializableException nexc) {

    JOptionPane.showMessageDialog(null, "NetDevices list is not serializable - No devices
saved", "Saving NetDevices", JOptionPane.INFORMATION_MESSAGE);

} catch (IOException nexc) {

    JOptionPane.showMessageDialog(null, "IO Exception saving NetDevices list - No devices
saved", "Saving NetDevices", JOptionPane.INFORMATION_MESSAGE);

}

}

```

+++++

```
try
```

```
{
```

```
    //...
```

```
}
```

```
catch(tipoExcessão_1 e)
```

```
{
```

```
    throw new novoTipoExcecao(e);
```

```
}
```

+++++

```
b.open;
```

```
try {
    A a = b.read(); // b lança exception
} catch (Exception e) {
    // imprimindo o erro;
    e.printStackTrace(); // ou coisa do gênero... não lembro
    // decidi que devo finalizar o programa por causa do erro
    System.exit(-1);
} finally {
    b.close();
}
```

```
+++++
```

```
import java.io.IOException;
import java.nio.file.Files;
import java.nio.file.Path;
import java.nio.file.Paths;
import java.util.List;
```

```
public class Main {
```

```
    public static void main(String[] args) throws IOException {
```

```
        Path path = Paths.get("c:/temp/arquivo.txt");
```

```
        List<String> linhas = Files.readAllLines(path);
```

```
        String novoConteudo =
```

```
            linhas.get(0).substring(0, 7) + "conteudo" + linhas.get(0).substring(15);
```

```
        linhas.remove(0);
```

```
        linhas.add(0, novoConteudo);
```

```
        Files.write(path, linhas);
    }
}
```

+++++

CSV

<https://dicasdejava.com.br/como-escrever-arquivos-csv-em-java/>

```
<dependency>
  <groupId>com.opencsv</groupId>
  <artifactId>opencsv</artifactId>
  <version>4.2</version>
</dependency>
```

```
import com.opencsv.CSVWriter;
```

```
import java.io.IOException;
```

```
import java.io.Writer;
```

```
import java.nio.file.Files;
```

```
import java.nio.file.Paths;
```

```
import java.util.ArrayList;
```

```
import java.util.List;
```

```
public class CriarCsvComStringsOpenCsv {
```

```
    public static void main(String[] args) throws IOException {
```

```
        String[] cabecalho = {"nome", "idade", "telefone"};
```

```

List<String[]> linhas = new ArrayList<>();
linhas.add(new String[]{"Joao", "35", "joao@dicasdejava.com.br"});
linhas.add(new String[]{"Maria", "23", "maria@dicasdeprogramacao.com.br"});
linhas.add(new String[]{"Ana", "25", "ana@dicasdejava.com.br"});

Writer writer = Files.newBufferedWriter(Paths.get("pessoas.csv"));
CSVWriter csvWriter = new CSVWriter(writer);

csvWriter.writeNext(cabecalho);
csvWriter.writeAll(linhas);

csvWriter.flush();
writer.close();

}

}

+++++++

public class CsvPessoa {
    private String nome;
    private Integer idade;
    private String email;

    public CsvPessoa(String nome, Integer idade, String email) {
        this.nome = nome;
        this.idade = idade;
        this.email = email;
    }
}

```

```

// Gets e Sets omitidos para ficar menor

}

import com.opencsv.bean.StatefulBeanToCsv;
import com.opencsv.bean.StatefulBeanToCsvBuilder;
import com.opencsv.exceptions.CsvDataTypeMismatchException;
import com.opencsv.exceptions.CsvRequiredFieldEmptyException;

import java.io.IOException;
import java.io.Writer;
import java.nio.file.Files;
import java.nio.file.Paths;
import java.util.ArrayList;
import java.util.List;

public class CriarCsvComObjetosOpenCsv {

    public static void main(String[] args) throws IOException, CsvDataTypeMismatchException,
CsvRequiredFieldEmptyException {

        List<CsvPessoa> pessoas = new ArrayList<>();
        pessoas.add(new CsvPessoa("Joao",35,"joao@dicasdejava.com.br"));
        pessoas.add(new CsvPessoa("Maria",23,"maria@dicasdeprogramacao.com.br"));
        pessoas.add(new CsvPessoa("Ana",25,"ana@dicasdejava.com.br"));

        Writer writer = Files.newBufferedWriter(Paths.get("pessoas.csv"));

        StatefulBeanToCsv<CsvPessoa> beanToCsv = new
StatefulBeanToCsvBuilder(writer).build();

        beanToCsv.write(pessoas);

```

```
writer.flush();
```

```
writer.close();
```

```
}
```

```
}
```

```
+++++
```

```
+++++
```

```
+++++
```

```
+++++
```

```
+++++
```

```
+++++
```

```
+++++
```

```
+++++
```

```
+++++
```

```
+++++
```

=====  
=====  
CERTIFICAÇÃO JAVA

Java SE 8 Programmer I | 1Z0-808

=====  
  
public class TryMe {  
  
 public static void printB(String str) {  
 System.out.print(Boolean.valueOf(str) ? "true" : "false");  
 }  
  
 public static void main(String args[]) {  
 printB("tRuE");  
 printB("false");  
 }  
}

=====  
  
What can directly access and change the value of the variable roomNr?

package com.mycompany;

public class Hotel {  
 public int roomNr = 100;

```
}
```

```
=====
```

Which piece of code will allow this class to be correctly serialized and deserialized? Drag and drop the correct fragment.

```
import java.io.*;
```

```
public class Car implements Serializable {
```

```
    public int speeds;
```

```
    public int wheels;
```

```
    private void writeObject(ObjectOutputStream oos)
```

```
        throws IOException {
```

```
            oos.writeInt(speeds);
```

```
            oos.writeInt(wheels);
```

```
        }
```

```
    private void readObject(ObjectInputStream ois)
```

```
        throws IOException {
```

```
            . . . . . Place here . . . . .
```

```
        }
```

```
}
```

```
ois.defaultReadObject();
```

```
this = ois.defaultReadObject();
```

```
speeds = ois.readInt(); wheels = ois.readInt();
```

```
wheels = ois.readInt(); speeds = ois.readInt();
```



=====

Given the code. What is true about it? (Choose all that apply)

```
enum State{ACTIVE, INACTIVE, DELETED}
```

- A) State.ACTIVE == State.ACTIVE is true
- B) State.ACTIVE == State.INACTIVE is false
- C) State.ACTIVE.equals(State.ACTIVE) is true
- D) State.ACTIVE < State.INACTIVE is true

=====

Give a piece of code. What is true?

```
public void waitForSomething() {  
    SomeClass o = new SomeClass();  
    synchronized (o) {  
        o.wait();  
        o.notify();  
    }  
}
```

- A) This code may throw an InterruptedException
- B) This code may throw an IllegalStateException
- C) This code may throw a TimeoutException
- D) Reversing the order of o.wait() and o.notify() will cause this method to complete normally.

=====

Given two classes defined in two different files. What is required at line marked "//some code goes here" to process the method doStuff() of a class A?

```
// The first file
package pack1;

public class ClassA {
    public static void doStuff() {
        System.out.println("doStuff");
    }
}

// The second file
package pack2;

public class ClassB {
    public static void main(String args[]) {
        //some code goes here
    }
}
```

- A) ClassA.doStuff();
- B) pack1.ClassA.doStuff();
- C) doStuff();
- D) It is impossible to use the method doStuff() in the class B.
- E) import pack1.A.\*; doStuff();

=====

Someone is writing a class MyKey, that will be used as a key in the java.util.HashMap. Which methods should be overridden to guarantee that MyKey works as HashMap key? (Select all that apply)

- A) public int has()
- B) public int hashCode()
- C) public boolean equals(MyKey)
- D) public boolean equals(Object)
- E) public int compareTo(Object)

F) public int compareTo(MyKey)

=====

Given the code. What is the result?

```
public class Hotel {  
  
    private static void book() {  
        System.out.print("book");  
    }  
  
    public static void main(String[] args) {  
        Thread.sleep(1);  
        book();  
    }  
}
```

- A) Compilation fails.
- B) An exception is thrown at runtime.
- C) "book" is printed.
- D) The code executes normally, but nothing is printed.

=====

Give a piece of code. Which two are possible results? (Select two)

```
public class Cruiser {  
    private int a = 0;  
  
    public void foo() {  
        Runnable r = new LittleCruiser();  
        new Thread(r).start();  
        new Thread(r).start();  
    }  
}
```

```
}
```

```
public static void main(String arg[]) {  
    Cruiser c = new Cruiser();  
    c.foo();  
}
```

```
public class LittleCruiser implements Runnable {  
    public void run() {  
        int current = 0;  
        for (int i = 0; i < 4; i++) {  
            current = a;  
            System.out.print(current + ", ");  
            a = current + 2;  
        }  
    }  
}
```

- A) 0, 2, 4, 0, 2, 4, 6, 6,
- B) 0, 2, 4, 6, 8, 10, 12, 14,
- C) 0, 2, 4, 6, 8, 10, 2, 4,
- D) 0, 0, 2, 2, 4, 4, 6, 6, 8, 8, 10, 10, 12, 12, 14, 14,
- E) 0, 2, 4, 6, 8, 10, 12, 14, 0, 2, 4, 6, 8, 10, 12, 14,

=====

Given the code. What is the output?

```
public class Test {  
    int a = 10;  
  
    public void doStuff(int a) {
```

```
    a += 1;
    System.out.println(++a);
}
public static void main(String args[]) {
    Test t = new Test();
    t.doStuff(3);
}
}
```

- A) 4
- B) 5
- C) 12
- D) 11

=====

Given the code. What is the result?

```
public class Hotel {

    private static void book() {
        System.out.print("book");
    }

    public static void main(String[] args) throws InterruptedException {
        Thread.sleep(1);
        book();
    }
}
```

- A) Compilation fails.
- B) An exception is thrown at runtime.
- C) "book" is printed
- D) The code executes normally, but nothing is printed.

=====

Give the code. What is the result?

```
class Hotel {
    public int bookings;
    public void book() {
        bookings++;
    }
}

public class SuperHotel extends Hotel {
    public void book() {
        bookings--;
    }

    public void book(int size) {
        book();
        super.book();
        bookings += size;
    }

    public static void main(String args[]) {
        SuperHotel hotel = new SuperHotel();
        hotel.book(2);
        System.out.print(hotel.bookings);
    }
}
```

- A) Compilation fails.
- B) An exception is thrown at runtime.
- C) 0
- D) 1
- E) 2

F) -1

=====

Given the code. What is the result?

```
public class TrickyNum<X extends Object> {  
  
    private X x;  
  
    public TrickyNum(X x) {  
        this.x = x;  
    }  
  
    private double getDouble() {  
        return x.doubleValue();  
    }  
  
    public static void main(String args[]) {  
        TrickyNum<Integer> a = new TrickyNum<Integer>(new Integer(1));  
        System.out.print(a.getDouble());  
    }  
}
```

- A) Compilation fails.
- B) An exception is thrown at runtime.
- C) "1.0" is printed.
- D) "1" is printed.

=====

Given the code. What is the result?

```
import java.io.*;  
  
public class Hotel implements Serializable {  
    private Room room = new Room();
```

```

public static void main(String[] args) {
    Hotel h = new Hotel();
    try {
        FileOutputStream fos = new FileOutputStream("Hotel.dat");
        ObjectOutputStream oos = new ObjectOutputStream(fos);
        oos.writeObject(h);
        oos.close();
    } catch(Exception ex) {
        ex.printStackTrace();
    }
}
}

```

```

class Room {
}

```

- A) Compilation fails.
- B) An instance of Hotel is serialized.
- C) An instance of Hotel and an instance of Room are both serialized.
- D) An exception is thrown at runtime.

=====

Given the code. What is the result?

```

class Vehicle {
    public void printSound() {
        System.out.print("vehicle");
    }
}

```

```

class Car extends Vehicle {
    public void printSound() {
        System.out.print("car");
    }
}

```



```
}
```

```
class Bike extends Vehicle {  
    public void printSound() {  
        System.out.print("bike");  
    }  
}
```

```
public class Test {  
    public static void main(String[] args) {  
        Vehicle v = new Car();  
        Car c = (Car) v;  
  
        v.printSound();  
        c.printSound();  
    }  
}
```

- A) Compilation fails.
- B) An exception is thrown at runtime.
- C) "vehiclecar" is printed.
- D) "vehiclebike" is printed.
- E) "carcar" is printed.
- F) "bikebike" is printed

=====

Given the exhibit. What is the result?

```
public class Hotel {  
  
    public static void book(short a) {  
        System.out.print("short ");  
    }  
}
```

```
public static void book(Short a) {  
    System.out.print("SHORT ");  
}
```

```
public static void book(Long a) {  
    System.out.print("LONG ");  
}
```

```
public static void main(String[] args) {  
    short shortRoom = 1;  
    int intRoom = 2;  
  
    book(shortRoom);  
    book(intRoom);  
}
```

```
}
```

A) SHORT LONG

B) short LONG

C) Compilation fails

D) An exception is thrown at runtime

=====

Given the code. Which line of code marks the earliest point that an object referenced by myInt becomes a candidate for garbage collection?

```
1 public void doStuff() {  
2     Integer arr[] = new Integer[5];  
3     for (int i = 0; i < arr.length; i++) {  
4         Integer myInt = new Integer(i);  
5         arr[i] = myInt;  
6     }  
7     System.out.println("end");  
8 }
```

- A) Line 4
- B) Line 5
- C) Line 6
- D) Line 7

=====

Given the code. What is the result?

1. public static void main(String args[]) {
2.     Object myObj = new String[]{"one", "two", "three"} {
3.         for (String s : (String[])myObj) System.out.print(s + ".");
4.     }
5. }

- A) one.two.three.
- B) Compilation fails because of an error at line 2
- C) Compilation fails because of an error at line 3
- D) An exception is thrown at runtime.

=====

Given the code. What is the result?

```
public static void main(String args[]) {  
    String str = null;  
    if (str == null) {  
        System.out.print("1");  
    } else (str.length() == 0) {  
        System.out.print("2");  
    } else {  
        System.out.print("3");  
    }  
}
```

- A) Compilation fails.
- B) "1" is printed.
- C) "2" is printed.

D) "3" is printed.

E) An exception is thrown at runtime.

=====

Complete the class that maps Strings to Integers.

```
import java.util.*;
```

```
public class StringToInteger {
```

```
    HashMap . . . . . Place here . . . . .
```

```
    hm = new HashMap<String, Integer>();
```

```
        public void add(. . . . . Place here . . . . .
```

```
    a, . . . . . Place here . . . . .
```

```
    b) {
```

```
        hm.put(a, b);
```

```
    }
```

```
        public . . . . . Place here . . . . .
```

```
    getKeys() {
```

```
        return hm.keySet();
```

```
    }
```

```
        public . . . . . Place here . . . . .
```

```
    getValues() {
```

```
        return hm.values();
```

```
    }
```

```
    public static void main(String args[]) {
```

```
        StringToInteger si = new StringToInteger();
```

```
        si.add("one", 1);
```

```
        si.add("two", 2);
```

```

        System.out.println(si.getKeys());
        System.out.println(si.getValues());
    }
}

```

<String, Integer>

<Integer, String>

Integer

String

Set<String>

Set<Integer>

Collection<String>

Collection<Integer>

=====

A developer is creating a class A, that needs to access a class B. The B class is deployed in a jar named superLib.jar. Which three, will allow the developer to use the B class while compiling the A class. (choose three)

A) The jar file is located at /myfolder/superLib.jar and the A class is compiled using "javac -classpath /myfolder/superLib.jar A.java"

B) The jar file is located at /myfolder/superLib.jar and the A class is compiled using "javac -d /myfolder/superLib.jar A.java"

C) The jar file is located at /myfolder/superLib.jar and the A class is compiled using "javac -cp /myfolder/superLib.jar/B A.java"

D) The jar file is located at /myfolder/superLib.jar and a classpath environment variable is set that includes /myfolder/superLib.jar

E) The jar file is located at /myfolder/superLib.jar and a classpath environment variable is set that includes /myfolder/superLib.jar/B.class

F) The jar file is located at \$JAVA\_HOME/jre/lib/ext/superLib.jar

G) The jar file is located at \$JAVA\_HOME/jre/classes/superLib.jar

=====

Given the code. What is the result?

```

class Small {
    public Small() {
        System.out.print("a ");
    }
}

```

```
}
```

```
class Small2 extends Small {  
    public Small2() {  
        System.out.print("b ");  
    }  
}
```

```
class Small3 extends Small2 {  
    public Small3() {  
        System.out.print("c ");  
    }  
}
```

```
public class Test {  
    public static void main(String args[]) {  
        new Small3();  
    }  
}
```

A) a

B) c

C) a b c

D) c b a

E) The code runs without output.

=====

Given the code. What is the result?

```
public class TrickyNum<X extends Number> {
```

```
    private X x;
```

```
    public TrickyNum(X x) {
```

```

    this.x = x;
}

private double getDouble() {
    return x.doubleValue();
}

public static void main(String args[]) {
    TrickyNum<Integer> a = new TrickyNum<Integer>(new Integer(1));
    System.out.print(a.getDouble());
}
}

```

- A) Compilation fails.
- B) An exception is thrown at runtime.
- C) "1.0" is printed.
- D) "1" is printed.

=====

Which three will compile without exception? (Choose three)

- A) private synchronized SomeClass a;
- B) void book() { synchronized () {} }
- C) public synchronized void book() {}
- D) public synchronized(this) void book() {}
- E) public void book() { synchronized(Cruiser.class) {} }
- F) public void book() {synchronized(a){}}

=====

Given the code. What is the result?

```
import java.io.*;
```

```

public class Hotel implements Serializable {
    private transient Room room = new Room();
}

```

```

public static void main(String[] args) {
    Hotel h = new Hotel();
    try {
        FileOutputStream fos = new FileOutputStream("Hotel.dat");
        ObjectOutputStream oos = new ObjectOutputStream(fos);
        oos.writeObject(h);
        oos.close();
    } catch(Exception ex) {
        ex.printStackTrace();
    }
}
}

```

```

class Room {
}

```

- A) Compilation fails.
- B) An exception is thrown at runtime.
- C) An instance of Hotel is serialized.
- D) An instance of Hotel and an instance of Room are both serialized.

=====

Given the code. What is the result?

```

class Vehicle {
    public void printSound() {
        System.out.print("vehicle");
    }
}

```

```

class Car extends Vehicle {
    public void printSound() {
        System.out.print("car");
    }
}

```



```
}
```

```
class Bike extends Vehicle {  
    public void printSound() {  
        System.out.print("bike");  
    }  
}
```

```
public class Test {  
    public static void main(String[] args) {  
        Vehicle v = new Car();  
        Bike b = (Bike) v;  
  
        v.printSound();  
        b.printSound();  
    }  
}
```

- A) Compilation fails.
- B) An exception is thrown at runtime.
- C) "vehiclecar" is printed.
- D) "vehiclebike" is printed.
- E) "carcar" is printed.
- F) "bikebike" is printed

=====

Given the code. What is the result?

```
import java.util.Arrays;
```

```
public class HashTest {  
  
    private String str;
```

```

public HashTest(String str) {
    this.str = str;
}

@Override
public String toString() {
    return this.str;
}

public static void main(String args[]) {
    HashTest h1 = new HashTest("2");
    String s1 = new String("1");

    Object arr[] = new Object[2];
    arr[0] = h1;
    arr[1] = s1;
    Arrays.sort(arr);
    for (Object o : arr) {
        System.out.print(o + " ");
    }
}
}

```

- A) "2 1" is printed.
- B) "1 2" is printed.
- C) Compilation fails.
- D) An exception is thrown at runtime.

=====

Given the code. What is the result?

```

String test = "This is a test string";
String[] tokens = test.split("\\s");
System.out.println(tokens.length);

```

- A) 0
- B) 5
- C) 21
- D) Compilation fails
- E) An exception is thrown at runtime

=====

Given the code. What is the result?

```
public class TrickyNum<X extends Object> {  
  
    private X x;  
  
    public TrickyNum(X x) {  
        this.x = x;  
    }  
  
    private double getDouble() {  
        return ((Double) x).doubleValue();  
    }  
  
    public static void main(String args[]) {  
        TrickyNum<Integer> a = new TrickyNum<Integer>(new Integer(1));  
        System.out.print(a.getDouble());  
    }  
}
```

- A) Compilation fails.
- B) An exception is thrown at runtime.
- C) "1.0" is printed.
- D) "1" is printed.

=====

Given the code. What is the result?

```

import java.util.Collections;

import java.util.Iterator;

import java.util.LinkedList;

import java.util.List;

public class TryMe {

    public static void main(String args[]) {

        List list = new LinkedList<String>();

        list.add("one");

        list.add("two");

        list.add("three");

        Collections.reverse(list);

        Iterator iter = list.iterator();

        for (Object o : iter) {

            System.out.print(o + " ");

        }

    }

}

```

- A) "three two one " is printed
- B) "one two three " is printed
- C) Nothing is printed
- D) Compilation fails
- E) An exception is thrown at runtime

=====

Given the code. What is the result?

```

public class Cruiser implements Runnable {

    public static void main(String[] args) throws InterruptedException {

        Thread a = new Thread(new Cruiser());

        a.start();

    }

}

```

```
System.out.print("Begin");  
a.join();  
System.out.print("End");  
}
```

```
public void run() {  
    System.out.print("Run");  
}  
}
```

- A) Compilation fails.
- B) An exception is thrown at runtime.
- C) "BeginRunEnd" is printed.
- D) "BeginEndRun" is printed.
- E) "BeginEnd" is printed.

=====

Given the code. What is the result?

```
class Hotel {  
    public void book() throws Exception {  
        throw new Exception();  
    }  
}
```

```
public class SuperHotel extends Hotel {  
    public void book() {  
        System.out.print("booked");  
    }  
}
```

```
public static void main(String args[]) {  
    Hotel h = new SuperHotel();  
    h.book();  
}
```

```
}  
}
```

- A) An exception is thrown at runtime.
- B) Compilation fails.
- C) The code runs without any output.
- D) "boked" is printed.

=====

Given the code. What is the result?

```
public class EmptyStringsTest {  
    public static boolean isEmpty(String s) {  
        return (s == null | s.length() == 0);  
    }  
  
    public static void main(String args[]) {  
        if (isEmpty(null)) {  
            System.out.print("empty ");  
        } else {  
            System.out.print("not_empty ");  
        }  
    }  
}
```

- A) "empty" is printed
- B) "not\_empty" is printed
- C) Compilation fails
- D) An exception is thrown at runtime

=====

Given the code. What is the result?

```
import java.util.Collections;  
import java.util.LinkedList;  
import java.util.List;
```

```

public class HashTest {

    private String str;

    public HashTest(String str) {
        this.str = str;
    }

    @Override
    public String toString() {
        return this.str;
    }

    public static void main(String args[]) {
        HashTest h1 = new HashTest("2");
        String s1 = new String("1");

        List<Object> list = new LinkedList<Object>();
        list.add(h1);
        list.add(s1);

        Collections.sort(list);
        for (Object o : list) {
            System.out.print(o + " ");
        }
    }
}

```

- A) "2 1" is printed.
- B) "1 2" is printed.
- C) Compilation fails.
- D) An exception is thrown at runtime.

=====

When comparing java.io.BufferedWriter to java.io.FileWriter, which capability exist as a method in only one of the two?

- A) Closing the stream
- B) flushing the stream
- C) writing to the stream
- D) marking a location in the stream
- E) writing a line separator to the stream

=====

Given:

```
System.out.printf("Pi = %f and E = %b", Math.PI, Math.E).
```

Place values where they would appear in the output

Pi = . . . . . Place here . . . . .

and E = . . . . . Place here . . . . .

3

2

3.141593

2.718282

true

false

=====

Given the code. What is the result?

```
1 public class TryMe {
2     Integer A;
3     int a;
4     public TryMe(int a) {
5         this.a = A + a;
6         System.out.print(this.a);
7     }
8     public static void main(String args[]) {
```



```
9     Integer A = new Integer(1);
10    TryMe t = new TryMe(A);
11    }
12 }
```

- A) The value "1" is printed
- B) Compilation fails because of an error in line 5
- C) A NullPointerException occurs at runtime
- D) A NumberFormatException occurs at runtime
- E) An IllegalStateException occurs at runtime

=====

Given two classes. What should be inserted at the beginning of the class B to compile and run it correctly?

```
// class A
```

```
package mycompany;
```

```
public class A {
    public static void doStuff() {
        System.out.println("doStuff");
    }
}
```

```
// class B
```

```
..... Place here .....
```

```
public class B {
    public static void main(String args[]) {
        doStuff();
    }
}
```

```
import mycompany;
```

```
import mycompany.*;
static import mycompany.*;
import mycompany.A;
static import mycompany.A;
import static mycompany.A.doStuff;
import mycompany.A.doStuff;
```

=====

Which three statements concerning the use of the `java.io.Serializable` interface are true?  
(Choose three)

- A) object from classes that use aggregation cannot be serialized.
- B) An object serialized on one JVM can be deserialized on a different JVM.
- C) The values in fields with the `volatile` modifier will not survive serialization and deserialization.
- D) The values in field with the `transient` modifier will not survive serialization and deserialization.
- E) It is legal to serialize an object of a type that has a supertype that does not implement `java.io.Serializable`

=====

Given the code. What is the result?

```
public static void main(String args[]) {
    String str = null;
    if (str.length() == 0) {
        System.out.print("1");
    } else if (str == null) {
        System.out.print("2");
    } else {
        System.out.print("3");
    }
}
```

- A) Compilation fails.
- B) "1" is printed.
- C) "2" is printed.

D) "3" is printed.

E) An exception is thrown at runtime.

=====

Given the code. What is the output?

```
public class Test {  
    int a = 10;  
  
    public void doStuff(int a) {  
        a += 1;  
        System.out.println(a++);  
    }  
    public static void main(String args[]) {  
        Test t = new Test();  
        t.doStuff(3);  
    }  
}
```

A) 11

B) 12

C) 4

D) 5

=====

Given the code. What is the result?

```
import java.util.HashSet;
```

```
public class HashTest {  
  
    private String str;  
  
    public HashTest(String str) {  
        this.str = str;  
    }  
}
```

```
@Override
```

```
public int hashCode() {  
    return this.str.hashCode();  
}
```

```
@Override
```

```
public boolean equals(Object obj) {  
    return this.str.equals(obj);  
}
```

```
public static void main(String args[]) {
```

```
    HashTest h1 = new HashTest("1");
```

```
    HashTest h2 = new HashTest("1");
```

```
    String s1 = new String("2");
```

```
    String s2 = new String("2");
```

```
    HashSet<Object> hs = new HashSet<Object>();
```

```
    hs.add(h1);
```

```
    hs.add(h2);
```

```
    hs.add(s1);
```

```
    hs.add(s2);
```

```
    System.out.print(hs.size());
```

```
}
```

```
}
```

A) "4" is printed.

B) "3" is printed.

C) "2" is printed.

D) Compilation fails.

E) An exception is thrown at runtime.

=====

What do you need to do to correct compilation errors? (Select two)

```
public class Creature {
    private int legCount;
    private int wingCount;

    public Creature(int legCount) {
        this.legCount = this.legCount;
        this.wingCount = 0;
    }

    public String toString() {
        return "legs=" + this.legCount + " wings=" + wingCount;
    }
}

public class Animal extends Creature {
    public Animal(int legCount) {
        this.wingCount = 0;
    }
}
```

- A) insert a call to super() into Creature constructor.
- B) insert a call to super() into Animal constructor.
- C) insert a call to this() into Animal constructor.
- D) insert a call to super(legCount) into Animal constructor.
- E) change the wingCount variable in the class Creature to protected.
- F) change the string "this.wingCount = 0" in the class Animal to "super.wingCount = 0"

=====

Given the code. Which statements are true? (Select two)

```
public class Hotel {
```

```
public static void book() {  
    //some code goes here  
}
```

```
public void cancelBooking() {  
    //some code goes here  
}
```

```
}
```

- A) Method book() can directly call method cancelBooking()
- B) Method cancelBooking() can directly call method book()
- C) Hotel.book() is a valid invocation of book()
- D) Hotel.cancelBooking() is a valid invocation of cancelBooking()

=====

Given the code. What is the result?

```
public class Cruiser implements Runnable {  
    public static void main(String[] args) {  
        Thread a = new Thread(new Cruiser());  
        a.start();  
  
        System.out.print("Begin");  
        a.join();  
        System.out.print("End");  
    }  
}
```

```
public void run() {  
    System.out.print("Run");  
}  
}
```

- A) Compilation fails.
- B) An exception is thrown at runtime.

- C) "BeginRunEnd" is printed.
- D) "BeginEndRun" is printed.
- E) "BeginEnd" is printed.

=====

Given the code. What is the result?

```
DateFormat df = DateFormat.getDateInstance(DateFormat.LONG, Locale.US);
Calendar c = Calendar.getInstance();
c.set(Calendar.YEAR, 2009);
c.set(Calendar.MONTH, 6);
c.set(Calendar.DAY_OF_MONTH, 1);
String formattedDate = df.format(c.getTime());
System.out.println(formattedDate);
```

- A) July 1, 2009
- B) June 1, 2009
- C) Compilation fails
- D) An exception is thrown at runtime

=====

Given the code. What is true?

```
public class Room {
    public int roomNr;
    private Date beginDtm;
    private Date endDttm;

    public void book(int roomNr, Date beginDttm, Date endDttm) {
        this.roomNr = roomNr;
        this.beginDtm = beginDttm;
        this.endDttm = endDttm;
    }
}
```

- A) The code demonstrates polymorphism.
- B) The class is fully encapsulated.

- C) The variable roomNr breaks encapsulation.
- D) Variables beginDttm and endDttm break polymorphism.
- E) The method book breaks encapsulation.

=====

Given the code. What is the result after the class TryMe execution?

```
class A {  
    public void doA() {  
        B b = new B();  
        b.dobB();  
        System.out.print("doA");  
    }  
}
```

```
class B {  
    public void dobB() {  
        C c = new C();  
        c.doC();  
        System.out.print("doB");  
    }  
}
```

```
class C {  
    public void doC() {  
        if (true)  
            throw new NullPointerException();  
        System.out.print("doC");  
    }  
}
```

```
public class TryMe {
```



```

public static void main(String args[]) {
    try {
        A a = new A();
        a.doA();
    } catch (Exception ex) {
        System.out.print("error");
    }
}

```

- A) "doCdoBdoA" is printed
- B) "doAdoBdoC" is printed
- C) "doBdoAerror" is printed
- D) "error" is printed
- E) nothing is printed

=====

Given the code. Select correct calls of methods. (Select all that apply)

```
import java.util.*;
```

```
class Empty {
}

```

```
class Extended extends Empty {
}

```

```
public class TryMe {
    public static void doStuff1(List<Empty> list) {
        // some code
    }
    public static void doStuff2(List list) {
        // some code
    }
}

```

```
public static void doStuff3(List<? extends Empty> list) {  
    // some code  
}
```

```
public static void main(String args[]) {  
    List<Empty> list1 = new LinkedList<Empty>();  
    List<Extended> list2 = new LinkedList<Extended>();  
  
    // more code here  
}
```

- A) doStuff1(list1);
- B) doStuff2(list2);
- C) doStuff2(list1);
- D) doStuff3(list1);
- E) doStuff3(list2);
- F) doStuff1(list2);

=====

What can directly access and change the value of the variable roomNr?

```
package com.mycompany;
```

```
public class Hotel {  
    private int roomNr = 100;  
}
```

- A) Only the Hotel class.
- B) Any class.
- C) Any class in com.mycompany package.
- D) Any class that extends Hotel.

=====

Given:

d is a valid, non-null java.util.Date object

df is a valid, non-null java.text.DateFormat object set to the current locale.

What outputs the current locale's country name and the appropriate version of date?

A)

```
Locale l = Locale.getDefault();  
System.out.println(l.getDisplayCountry() + " " + df.format(d));
```

B)

```
Locale l = Locale.getLocale();  
System.out.println(l.getDisplayCountry());
```

C)

```
Locale l = Locale.getLocale();  
System.out.println(l.getDisplayCountry() + " " + df.setDateFormat(d));
```

D)

```
Locale l = Locale.getDefault();  
System.out.println(l.getDisplayCountry() + " " + df.setDateFormat(d));
```

=====

Given the code. What is the result?

```
public class SomeClass {  
    private int value = 1;  
  
    public int getValue() {  
        return value;  
    }  
  
    public void changeVal(int value) {  
        value = value;  
    }  
  
    public static void main(String args[]) {  
        int a = 2;  
        SomeClass c = new SomeClass();  
        c.changeVal(a);  
    }  
}
```

```
        System.out.print(c.getValue());
    }
}
```

- A) "1" is printed
- B) "2" is printed
- C) Compilation fails
- D) An exception is thrown at runtime

=====

Given the code. What is the result?

```
import java.util.Collections;
```

```
import java.util.LinkedList;
```

```
public class TryMe {
    public static void main(String args[]) {
        LinkedList<String> list = new LinkedList<String>();
        list.add("BbB1");
        list.add("bBb2");
        list.add("bbB3");
        list.add("BBb4");
        Collections.sort(list);
        for (String str : list) {
            System.out.print(str + ":");
        }
    }
}
```

- A) "BbB1:bBb2:bbB3:BBb4:" is printed.
- B) "BBb4:bbB3:bBb2:BbB1:" is printed.
- C) "BBb4:BbB1:bBb2:bbB3:" is printed.
- D) "bbB3:bBb2:BbB1:BBb4:" is printed
- E) Compilation fails.

F) An exception is thrown at runtime.

=====

Given the code. What is the result?

```
class Hotel {  
    public int bookings;  
    public void book() {  
        bookings++;  
    }  
}  
  
public class SuperHotel extends Hotel {  
    public void book() {  
        bookings--;  
    }  
  
    public void book(int size) {  
        book();  
        super.book();  
        bookings += size;  
    }  
  
    public static void main(String args[]) {  
        Hotel hotel = new SuperHotel();  
        hotel.book(2);  
        System.out.print(hotel.bookings);  
    }  
}
```

A) Compilation fails.

B) An exception is thrown at runtime.

C) 0

D) 1

E) 2

F) -1

=====

Given the code. What is the result?

```
public static void main(String args[]) {  
    try {  
        String arr[] = new String[10];  
        arr = null;  
        arr[0] = "one";  
        System.out.print(arr[0]);  
    } catch(NullPointerException nex) {  
        System.out.print("null pointer exception");  
    } catch(Exception ex) {  
        System.out.print("exception");  
    }  
}
```

- A) "one" is printed.
- B) "exception" is printed.
- C) "null pointer exception" is printed.
- D) Compilation fails.

=====

A Java bean component has the following field:

```
private boolean done;
```

Which method declarations follow the JavaBean standards for getting/settings this field?  
(Choose 3)

- A) public void setDone(boolean done)
- B) public boolean setDone(boolean done)
- C) private boolean setDone(boolean done)
- D) public void setDone()
- E) public boolean getDone()

- F) public boolean isDone()
- G) public boolean getDone(boolean done)
- H) public void isDone()

=====

Given the code. What is the result?

```
import java.util.*;

public class TryMe {
    public static void main(String args[]) {
        Queue<String> q = new PriorityQueue<String>();
        q.add("3");
        q.add("1");
        q.add("2");
        System.out.print(q.poll() + " ");
        System.out.print(q.peek() + " ");
        System.out.print(q.peek());
    }
}
```

- A) 1 2 3
- B) 3 2 1
- C) 1 2 2
- D) 3 1 1
- E) 2 3 3
- F) 1 1 2

=====

Given the code. What is the result?

```
public static void main(String args[]) {
    String str = "null";
    if (str == null) {
```

```

        System.out.print("1");
    } else if (str.length() == 0) {
        System.out.print("2");
    } else {
        System.out.print("3");
    }
}

```

- A) Compilation fails.
- B) "1" is printed.
- C) "2" is printed.
- D) "3" is printed.
- E) An exception is thrown at runtime.

=====

Given the code. What is the result?

```

public class Test {
    private static void doStuff(String str) {
        int var = 4;
        if (var == str.length()) {
            System.out.print(str.charAt(--var) + " ");
        }
        else {
            System.out.print(str.charAt(0) + " ");
        }
    }
}

public static void main(String args[]) {
    doStuff("abcd");
    doStuff("efg");
    doStuff("hi");
}
}

```

- A) Compilation fails.



- B) An exception is thrown at runtime.
- C) d e h
- D) d f i
- E) c f i
- F) c e h

=====

Given the code. What is the output?

1. public static void main(String args[]) {
2.     Object myObj = new String[]{"one", "two", "three"};
3.         for (String s : (String[])myObj) System.out.print(s + ".");
4.     }
5. }

- A) one.two.three.
- B) Compilation fails because of an error at line 2
- C) Compilation fails because of an error at line 3
- D) An exception is thrown at runtime.

=====

What is true? (Choose three)

- A) A method with the same signature as a private final method in class Z can be implemented in a subclass of Z.
- B) A final method in class Z can be abstract if and only if Z is abstract.
- C) A protected method in class Z can be overridden by any subclass of Z.
- D) A private static method can be called only within other static methods in class Z.
- E) A non-static public final method in class Z can be overridden in any subclass of Z.
- F) A public static method in class Z can be called by a subclass of Z without explicitly referencing the class Z.

=====

endpoint

=====

EL implicit object gives access to the parameters stored in servletContext

=====

EE 7 JMS message-driven bean with bean-managed transactions

=====

=====  
=====

=====  
=====

=====  
=====

```
=====
=====

FTP - CONTROLE
```

```
*****
```

```
*****
```

```
import java.io.IOException;
```

```
import java.net.SocketException;
```

```
import java.util.List;
```

```
public class FtpPrincipal {
```

```
    public static void main(String[] args) {
```

```
        // TODO Auto-generated method stub
```

```
        System.out.println("Inicio do sistema...");
```

```
        try {
```

```
            //FTPControle.ListarArquivosFromFTP();
```

```
            List<String> Lista =
```

```
FTPControle.ListarArquivosDiretoriosFromFTP().getListaArquivos();
```

```
            for (String f : Lista) {
```

```
                System.out.println(f);
```

```
            }
```

```
System.out.println("Total Arquivos: "+Lista.size());
```

```
    } catch (SocketException e) {  
        // TODO Auto-generated catch block  
        e.printStackTrace();  
    } catch (IOException e) {  
        // TODO Auto-generated catch block  
        e.printStackTrace();  
    }  
}
```

```
}
```

```
}
```

```
*****
```

```
*****
```

```
import java.util.List;
```

```
public class FTPDados {

    private String ftpConexao;

    private String ftpLogin;

    private String ftpSenha;

    private int ftpPorta;

    private String ftpCaminhoServidor;

    private String caminhoOrigem;

    private String caminhoDestino;

    private List<String> ListaArquivos;

    private List<String> ListaDiretorios;

    public String getFtpConexao() {

        return ftpConexao;

    }

    public void setFtpConexao(String ftpConexao) {

        this.ftpConexao = ftpConexao;

    }

    public String getFtpLogin() {

        return ftpLogin;

    }

    public void setFtpLogin(String ftpLogin) {

        this.ftpLogin = ftpLogin;

    }

    public String getFtpSenha() {

        return ftpSenha;

    }

    public void setFtpSenha(String ftpSenha) {

        this.ftpSenha = ftpSenha;

    }

    public int getFtpPorta() {
```

```
        return ftpPorta;
    }
    public void setFtpPorta(int ftpPorta) {
        this.ftpPorta = ftpPorta;
    }
    public String getFtpCaminhoServidor() {
        return ftpCaminhoServidor;
    }
    public void setFtpCaminhoServidor(String ftpCaminhoServidor) {
        this.ftpCaminhoServidor = ftpCaminhoServidor;
    }
    public String getCaminhoOrigem() {
        return caminhoOrigem;
    }
    public void setCaminhoOrigem(String caminhoOrigem) {
        this.caminhoOrigem = caminhoOrigem;
    }
    public String getCaminhoDestino() {
        return caminhoDestino;
    }
    public void setCaminhoDestino(String caminhoDestino) {
        this.caminhoDestino = caminhoDestino;
    }
    public List<String> getListaArquivos() {
        return ListaArquivos;
    }
    public void setListaArquivos(List<String> listaArquivos) {
        ListaArquivos = listaArquivos;
    }
    public List<String> getListaDiretorios() {
        return ListaDiretorios;
    }
}
```

```
}  
public void setListaDiretorios(List<String> listaDiretorios) {  
    ListaDiretorios = listaDiretorios;  
}
```

```
}
```

```
*****
```

```
*****
```

```
#Crunchify Properties
```

```
ftpConexao=ftp.websalt.com.br
```

```
ftpLogin=sindsmart
```

```
ftpSenha=odavi240175
```

```
ftpPorta=0
```

```
ftpCaminhoServidor=/Web/PHP/
```

```
caminhoOrigem=
```

```
caminhoDestino=
```

```
*****
```

\*\*\*\*\*

```
import java.io.FileNotFoundException;
```

```
import java.io.IOException;
```

```
import java.io.InputStream;
```

```
import java.util.Properties;
```

```
public class GetPropertyValues {
```

```
    InputStream inputStream;
```

```
    public FTPDados getPropValues() throws IOException {
```

```
        FTPDados retorno = new FTPDados();
```

```
        try {
```

```
            Properties prop = new Properties();
```

```
            String propFileName = "config.properties"; //CRIAR UMA PASTA  
resources NA RAIS DO PROJETO ABAIXO src
```

```
            inputStream =  
getClass().getClassLoader().getResourceAsStream(propFileName);
```

```
            if (inputStream != null) {
```

```
                prop.load(inputStream);
```

```
            } else {
```

```
                throw new FileNotFoundException(" " + propFileName + "  
não foi encontrado no projeto...");
```

```
            }
```

```
            //Date time = new Date(System.currentTimeMillis());
```

```
            String ftpConexao = prop.getProperty("ftpConexao");
```



```
String ftpLogin = prop.getProperty("ftpLogin");
String ftpSenha = prop.getProperty("ftpSenha");
String ftpPorta = prop.getProperty("ftpPorta");
String ftpCaminhoServidor = prop.getProperty("ftpCaminhoServidor");
String caminhoOrigem = prop.getProperty("caminhoOrigem");
String caminhoDestino = prop.getProperty("caminhoDestino");
```

```
retorno.setFtpConexao(ftpConexao);
retorno.setFtpLogin(ftpLogin);
retorno.setFtpSenha(ftpSenha);
retorno.setFtpPorta(Integer.parseInt(ftpPorta));
retorno.setFtpCaminhoServidor(ftpCaminhoServidor);
retorno.setCaminhoOrigem(caminhoOrigem);
retorno.setCaminhoDestino(caminhoDestino);
```

```
} catch (Exception e) {
    System.out.println("ERRO: " + e.getMessage());
} finally {
    inputStream.close();
}

return retorno;
```

```
}
```

```
}
```

```
*****
```

```
*****
```

```
import java.io.FileInputStream;
```

```
import java.io.FileOutputStream;
```

```
import java.io.IOException;
```

```
import java.net.SocketException;
```

```
import java.util.ArrayList;
```

```
import java.util.List;
```

```
import org.apache.commons.net.ftp.FTPClient;
```

```
//baixar o jar commons-net-1.4.1.jar
```

```
// import org.apache.commons.net.ftp.FTPClient;
```

```
// https://commons.apache.org/proper/commons-net/download_net.cgi
```

```
// http://www.java2s.com/Code/Jar/a/Downloadapachecommonsnetjar.htm
```

```
public class FTPControle {
```

```
    /**
```

```
     * Listaos arquivos existente em um caminho no servidor do FTP
```

```
     *
```

```

* @return um objetos FTPDados alimentado com um lista de arquivos
* @throws SocketException
* @throws IOException
*/
public static FTPDados ListarArquivosDiretoriosFromFTP() throws SocketException,
IOException {

    FTPDados dadosftp = new FTPDados();

    List<String> ListaArquivo = new ArrayList<String>();
    List<String> ListaDiretorio = new ArrayList<String>();
    GetPropertyValues prop = new GetPropertyValues();
    FTPDados ftpdados = prop.getPropValues();

    FTPClient ftp = new FTPClient();

    ftp.connect(ftpdados.getFtpConexao());
    ftp.login(ftpdados.getFtpLogin(), ftpdados.getFtpSenha());
    // porta 21

    ftp.changeWorkingDirectory(ftpdados.getFtpCaminhoServidor());

    // System.out.print(ftp.getReplyString());

    String[] arq = ftp.listNames();

    System.out.println("Listando arquivos: \n");

    for (String item : arq) {

        // System.out.println(f);

```

```

        if (item.indexOf(".") > 0) {
            ListaArquivo.add(item);
        } else {
            ListaDiretorio.add(item);
        }
    }

    if (ListaArquivo.size() > 0) {
        dadosftp.setListaArquivos(ListaArquivo);
    }

    if (ListaDiretorio.size() > 0) {
        dadosftp.setListaDiretorios(ListaDiretorio);
    }

    return dadosftp;
}

public static void EnviarArquivosParaFTP() throws SocketException, IOException {

    FTPClient ftp = new FTPClient();

    GetPropertyValues prop = new GetPropertyValues();
    FTPDados ftpdados = prop.getPropValues();

    ftp.connect(ftpdados.getFtpConexao());
    ftp.login(ftpdados.getFtpLogin(), ftpdados.getFtpSenha());
}

```

```
        FileInputStream arqEnviar = new
FileInputStream("/Web/PHP/arquivoenviar.doc");

        if (ftp.storeFile("meuarquivo.doc", arqEnviar))
            System.out.println("Arquivo armazenado com sucesso!");
        else
            System.out.println("Erro ao armazenar o arquivo.");

    }
}
```

```
public static void BaixarArquivosDoFTP(String[] args) throws SocketException,
IOException {
```

```
    FTPClient ftp = new FTPClient();
    GetPropertyValues prop = new GetPropertyValues();
    FTPDados ftpdados = prop.getPropValues();

    ftp.connect(ftpdados.getFtpConexao());
    ftp.login(ftpdados.getFtpLogin(), ftpdados.getFtpSenha());

    ftp.changeWorkingDirectory("grdt_websigem");

    String[] arq = ftp.listNames();

    FileOutputStream fos = new
FileOutputStream("/Web/PHP/arquivoftpdownload.doc");

    if (ftp.retrieveFile("RD-4410-00412[192017].log", fos))
        System.out.println("Download efetuado com sucesso!");
}
```

```
else
```

```
System.out.println("Erro ao efetuar download do arquivo.");
```

```
}
```

```
}
```

```
*****
```

```
*****
```

```
=====
```

```
=====
```

=====

=====

=====

