<!DOCTYPE HTML>  
<%@ **page** contentType="text/html;charset=UTF-8" language="java" %>  
<%@ **taglib** prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>  
<%@ **page** isELIgnored="false" %>  
<html>  
<head>  
 <title>List Customers</title>  
</head>  
<body>  
<div id="wrapper">  
 <div id="header">  
 <h2>CRM - Customer RelationShip Manager</h2>  
 </div>  
</div>  
  
<div id="container">  
 <div id="content">  
 <table>  
 <tr>  
 <th>First Name</th>  
 <th>Last Name</th>  
 <th>Email</th>  
 </tr>  
 <**c:forEach** var="customer" items="${customers}">  
 <tr>  
 <td>${customer.firstName}</td>  
 <td>${customer.lastName}</td>  
 <td>${customer.email}</td>  
 </tr>  
 </**c:forEach**>  
 </table>  
 </div>  
</div>  
  
</body>  
</html>

* pentru a putea adauga

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

trebuie sa punem in pom.xml:

<dependency>  
 <groupId>javax.servlet</groupId>  
 <artifactId>jstl</artifactId>  
 <version>1.2</version>  
</dependency>

* forEach nu are in tag modelAttribute ca form, insa putem accesa atributul a lui Model object venit de la Controller cu ${}, doar ca nu se va trimite atributul la Controller.

**Adaugare la CSS/JS etc. in MVC project**

* **Punem mere /nume/ , adica / la inceput si / la final**

**mvc:resources**

1. In webapp cream un folder cu orice nume,de ex resources, si punem in el foldere la css,images, js etc.
2. In applicationContext.xml adugam

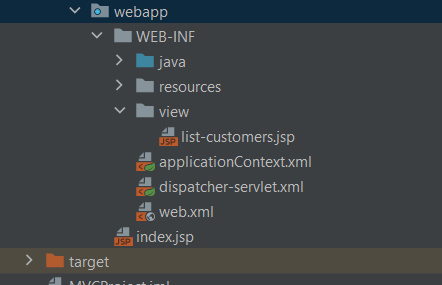
<mvc:resources mapping="/p/\*\*" location="/WEB-INF/resources/"/>

**mapping** – aici indicam care va fi request pentru care se va cauta ceea ce e in location

**location** – aici indicam propriu zis care este locatia la care se refera mapping. Atentie! Indicam **fullpath**!

Deci, daca de ex vom avea **href=”/p/styles.css”,** MVC **nu va inlocui pur si simplu /p/ cu /WEB-INF/resources/**

dar va cauta anume ceea ce vom scrie dupa /p/ in /WEB-INF/resources/ , caci are nevoie de fullpath si va oferi acea resursa



**folderul de la care se incepe e webapp**

**Evita folosirea lui href singur! Foloseste-l cu <c:url>**

**SAU**

Pentru a accesa acum calea unde se afla folderul, putem folosi:

**${pageContext.request.contextPath}/WEB-INF/resources/CSS/styles.css**

**<link rel="stylesheet" type="text/css"**

**href="${pageContext.request.contextPath}/WEB-INF/CSS/CSS.css**

**">**

**<img src="${pageContext.request.contextPath}/resources/images/spring-logo.png" />**

* **${pageContext.request.contextPath} – se refera la folderul radacina al proiectului, adica webapp, si va avea adresa /src/main/webapp**

**Sau**

Utilizam pur si simplu

"../../resources/css/styles.css"

**SAU!!!**

Spring Security sau MVC poate trata toate aceste / ca normal request, si deci sa nu poata citi fisierele noatre. In asa caz,nu mai putem lucra cu WEB-INF folder, asa ca trebuie sa il abandonam.Deci lasam in context app neaparat mvc:resources!

<mvc:resources mapping="/resources/\*\*" location="/resources/"/>

Si vom adauga css neaparat asa:

<link rel="stylesheet" type="text/css" href="<**c:url** value="/resources/CSS/styles.css"/>"/>

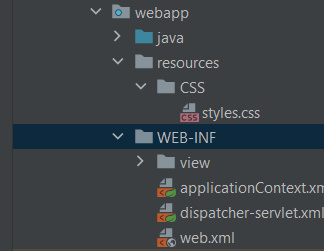
Pentru asta, in JSP mai adaugam

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

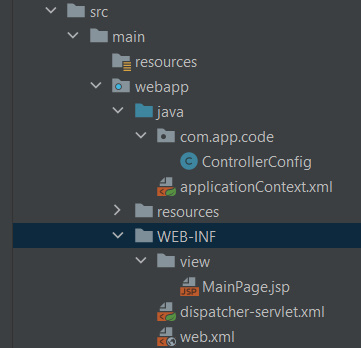
Si in pom.xml

<dependency>  
 <groupId>javax.servlet</groupId>  
 <artifactId>jstl</artifactId>  
 <version>1.2</version>  
</dependency>

Dar vom crea un folder resources in webapp folder, si acolo vom adauga CSS si tot ce avem nevoie.



applicationContext poate fi si in java folder, nu se va schimba nimic,



**Welcome File**

Welcome file sau page in cazul nostru, pot fi setate din web.xml

<welcome-file-list>  
 <welcome-file>index.jsp</welcome-file>  
 <welcome-file>index.html</welcome-file>  
</welcome-file-list>

Primul gasit e folosit.

Putem face ca pagina noastra sa ne trimita direct la cea dorita. Pune in index.jsp doar asta:

<% response.sendRedirect("customer/list"); %>

**Button**

Putem adauga un button ce trimite un request asa:

<input type="button" value="Add Customer" onclick="***window***.location.href='showFormForAdd'; return false;" class="add-button"/>