**Raddio button:**

* **Parameter**:

<form th:action="register" method="post">  
 FirstName <input type="text" th:name="firstName">  
 <br>  
 LastName <input type="text" th:name="lastName">  
 <br>  
 <input type="radio" th:name="country" th:value="Moldova" th:text="Moldova"/>  
 <br>  
 <input type="radio" th:name="country" th:value="Romania" th:text="Romania"/>  
 <br>  
 <input type="radio" th:name="country" th:value="USA" th:text="USA"/>  
 <br>  
 <input type="submit" value="Submit">  
</form>

th:name e ca si name

th:value e value

* **(Attribute)From a HashMap:**
* public class Student {  
   private String firstName="null";  
   private String lastName="null";  
   private String country;  
    
   @Value("#{factoryBean}")  
   private HashMap<String,String> countriesList;

<form th:action="register" method="post" th:object="${student}">  
 FirstName <input type="text" th:field="\*{firstName}">  
 <br>  
 LastName <input type="text" th:field="\*{lastName}">  
  
 <input type="radio" th:field="\*{country}" th:each="pairs : \*{countriesList}" th:name="country" th:value="${pairs.key}" th:utext="'<br>'+${pairs.value}">  
  
 <input type="submit" value="Submit">  
</form>

* Putem folosi si aici un HashMap.
* intr-un foreach, hashmap returneaza un obiect ce contine si key si value, deci ne putem referi la key prin .key si value .value
* th:utext permite punerea si la taguri html. La th:text ‘<br>’ asa si ar aparea, nu ar fi new line, ci simplue string <br>

**Select**

* **Param**
* <form action="/register" method="post" >  
  <select th:name="country">  
   <option th:value="SUA" th:text="SUA"/>  
   <option th:value="MD" th:text="MD"/>  
   <option th:value="RO" th:text="RO"/>  
  </select>  
   <input type="submit">  
  </form>
* **Attribute(HashMap)**
* <select th:field="\*{country}">  
   <option th:each="pairs : \*{countriesList}" th:label="${pairs.value}" th:value="${pairs.key}"></option>  
  </select>
* th:field – asta e obiect de unde se ia si se pune valoarea, De luat nu are ce lua de la el aici, deci doar va seta
* th:label e exact ca label

**CheckBox**

* **Parametru**:

@PostMapping("/register")  
public String register(@RequestParam("country")String[] country){  
 System.*out*.println(Arrays.*toString*(country));  
 return "info";  
}

<form action="/register" method="post" t>  
<input type="checkbox" th:name="country" th:value="MD" th:text="Moldova"/>  
 <input type="checkbox" th:name="country" th:value="SUA" th:text="America"/>  
 <input type="checkbox" th:name="country" th:value="FR" th:text="France"/>  
 <input type="submit">  
</form>

* **Attribute(HashMap)**
* private String firstName="null";  
  private String lastName="null";  
  private String[] country;

<input type="checkbox" th:name="country" th:each="pairs : \*{countriesList}" th:value="${pairs.key}" th:text="${pairs.value}"/>

**Validation Message**

<**p** th:if="${#fields.hasErrors('age')}">Invalid Age</**p**>

#fields – este un obiect venit de la MVC de tip RequestBinding

Daca vrem sa afisam si mesajul errorii folosim th:errors=”\*{field}”

<p th:if="${#fields.hasErrors('email')}" th:errors="\*{email}"></p>