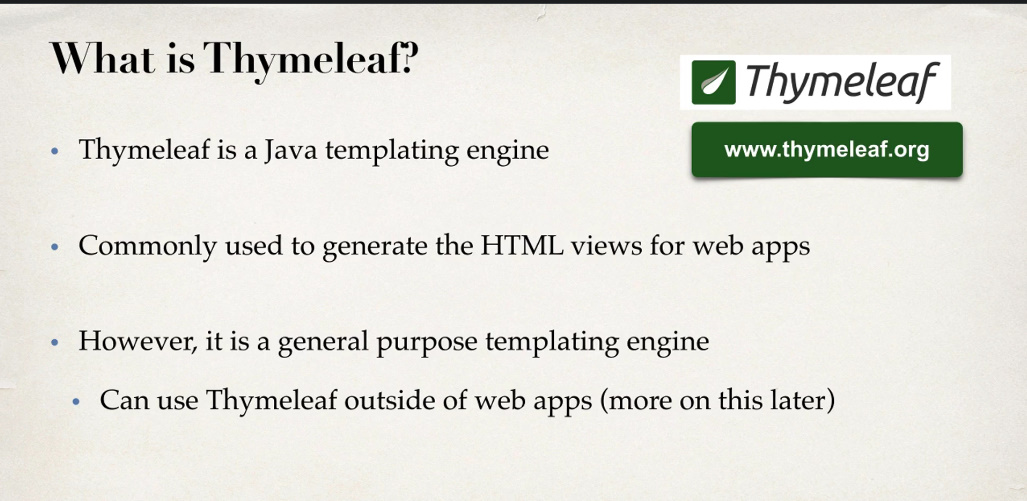
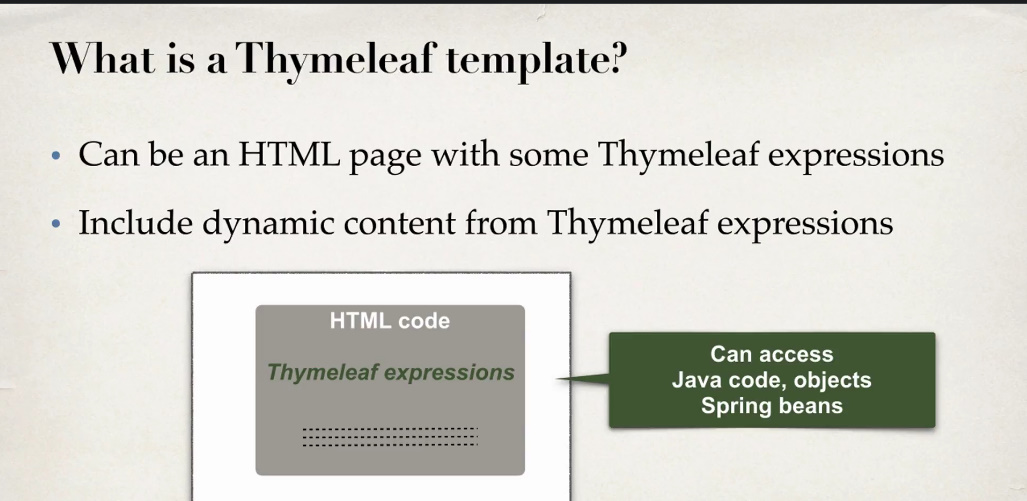
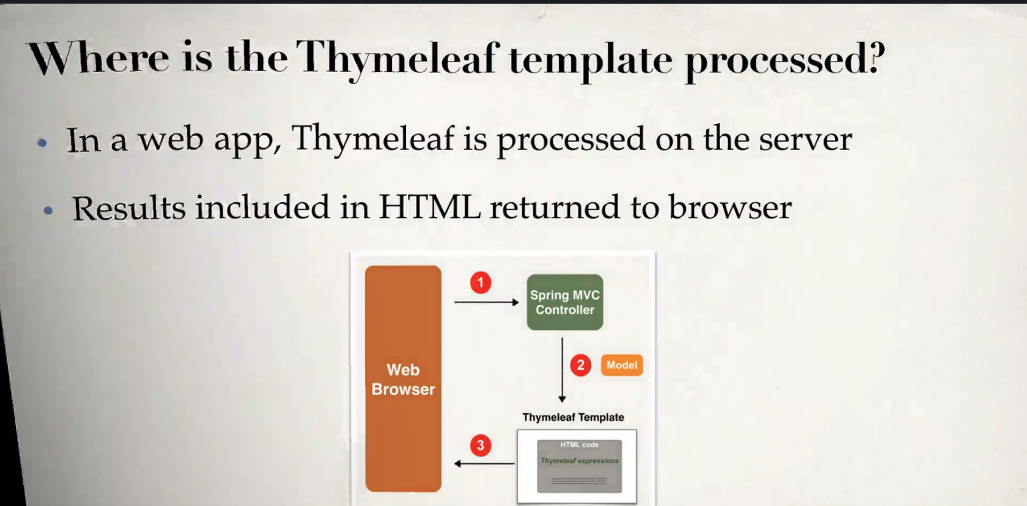
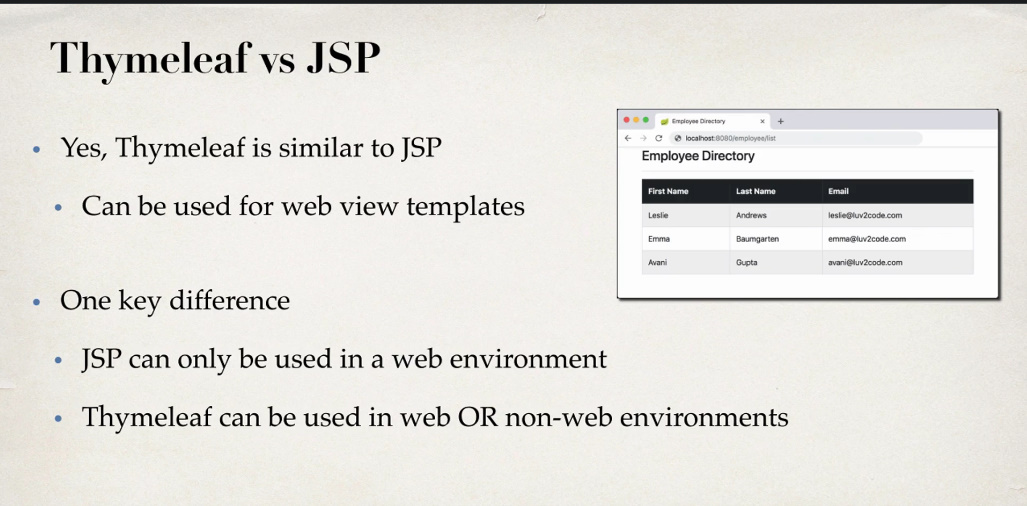
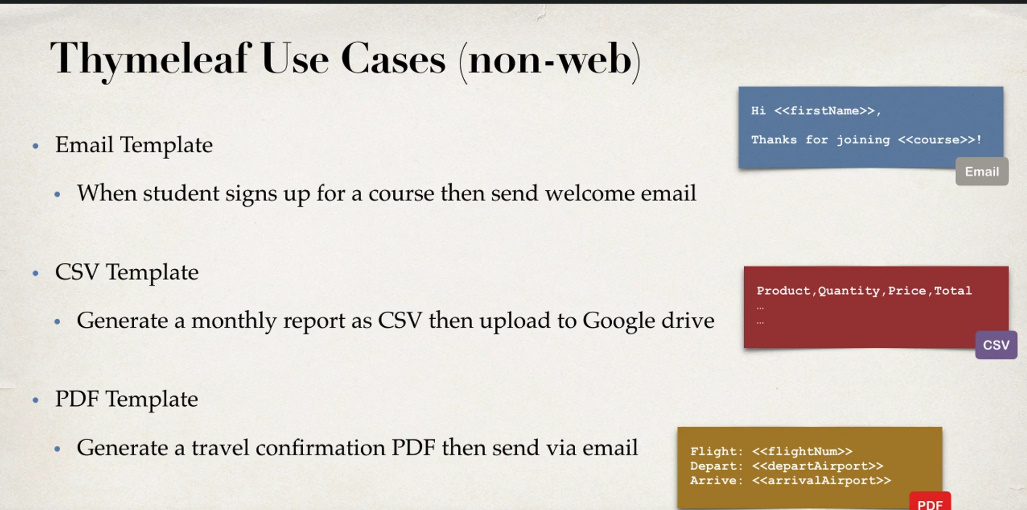
**Ce este**

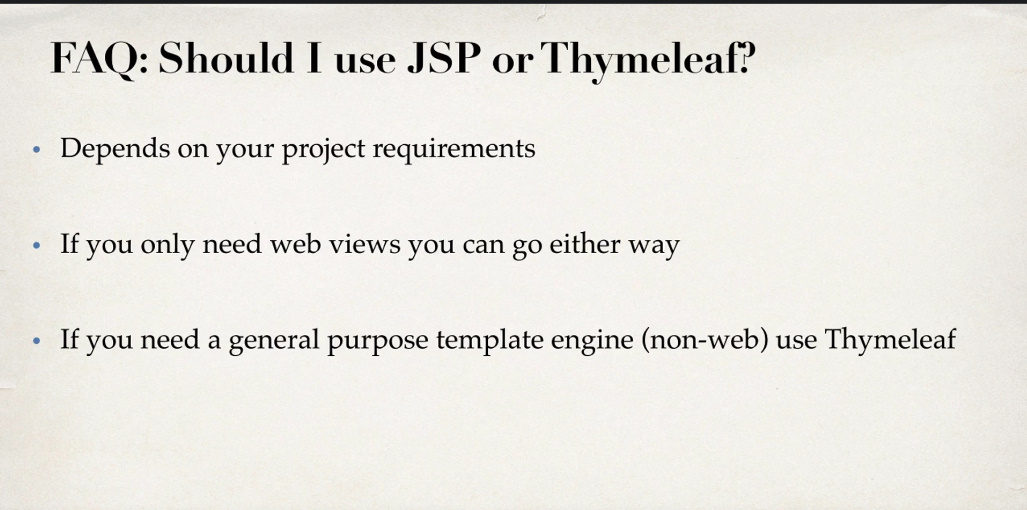


* Este un proiect separat ce nu are legatura cu spring
* Ca si JSP, e facut pentru a face o pagina HTML ce interactioneaza cu codul de backend
* Putem crea webapp cu thymeleaf, fara spring
* Totusi, spring si thymeleaf pot fi combinte
* 

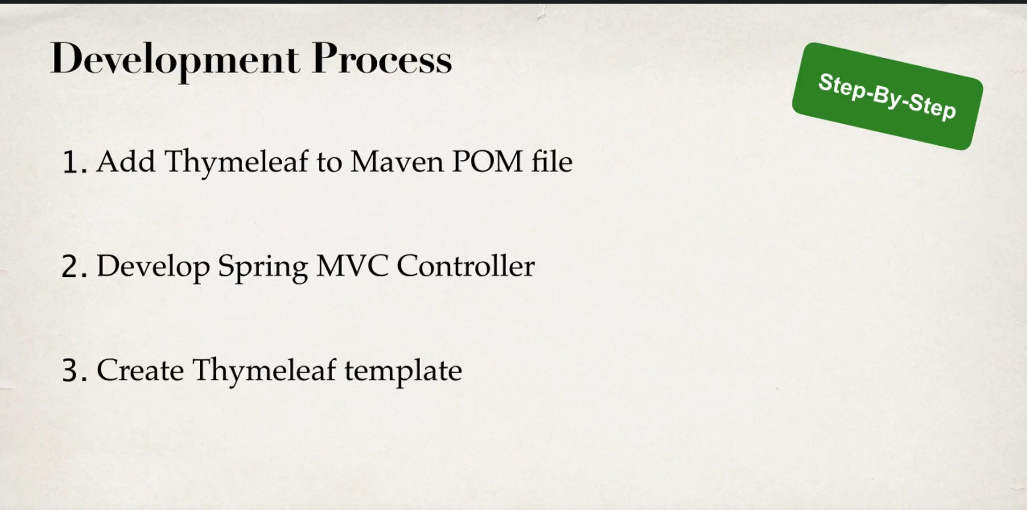


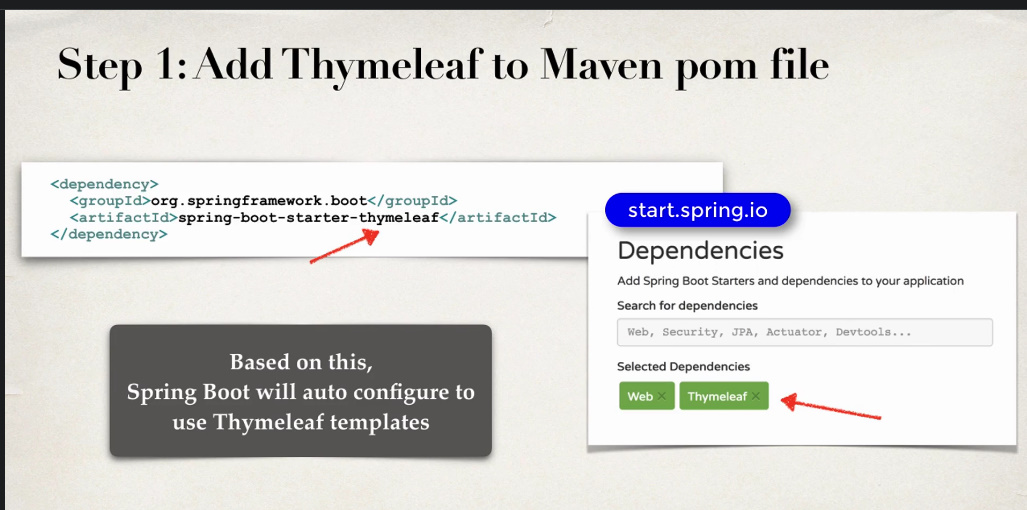


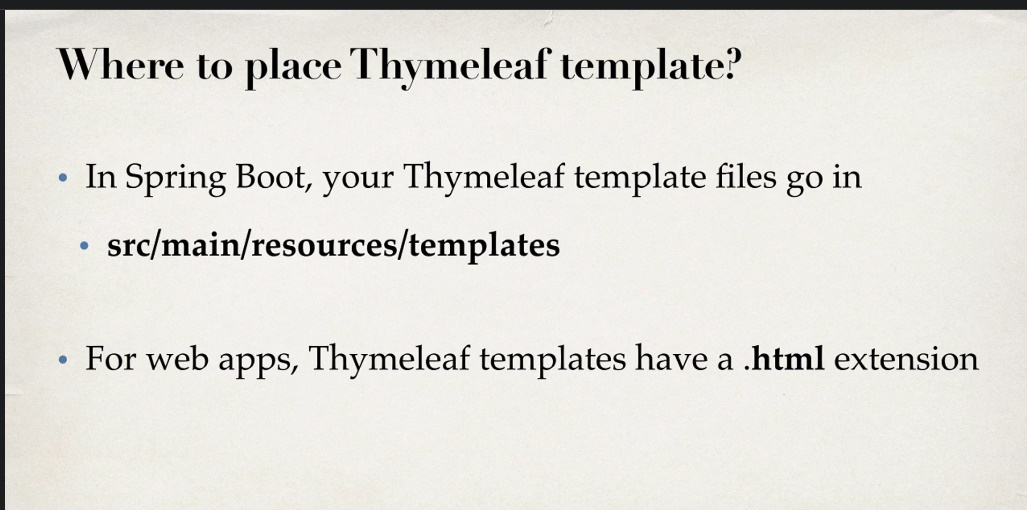




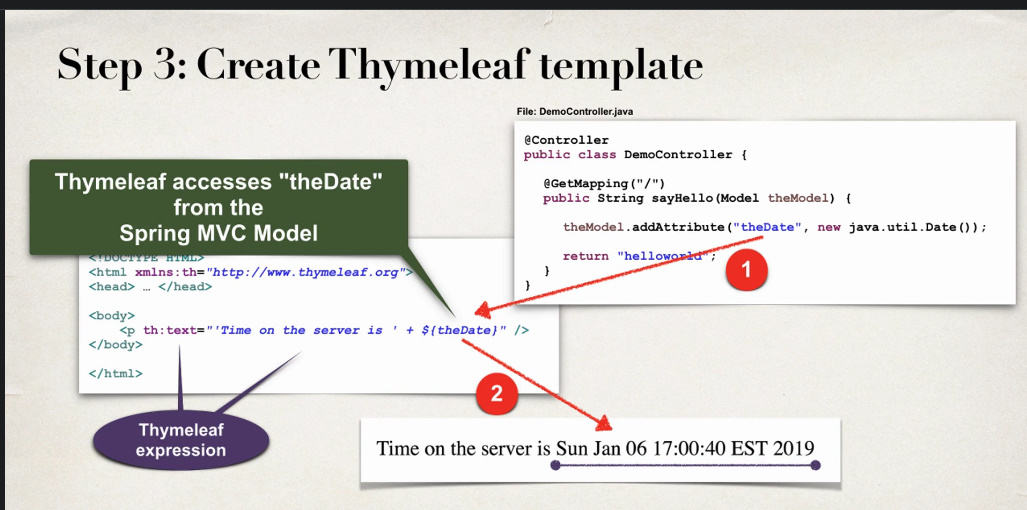
**Proiect**

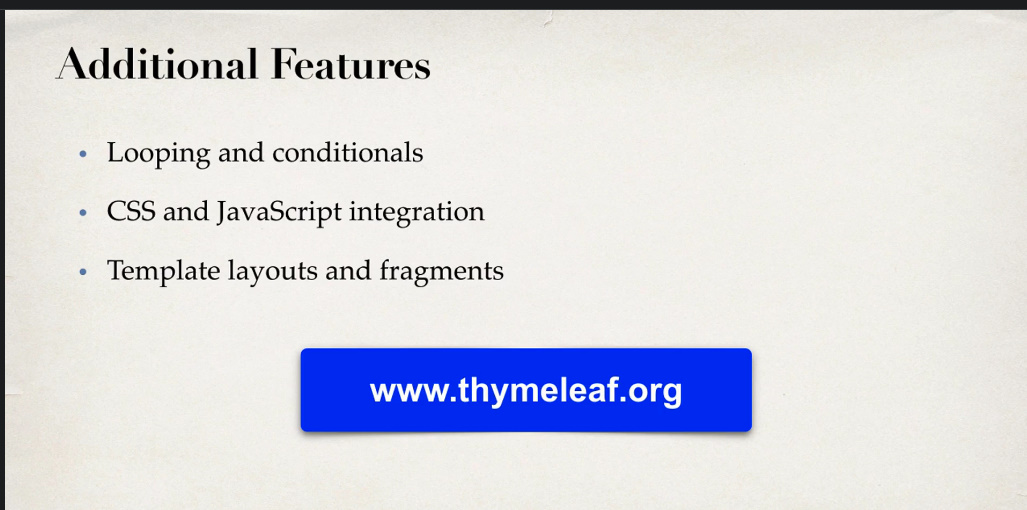


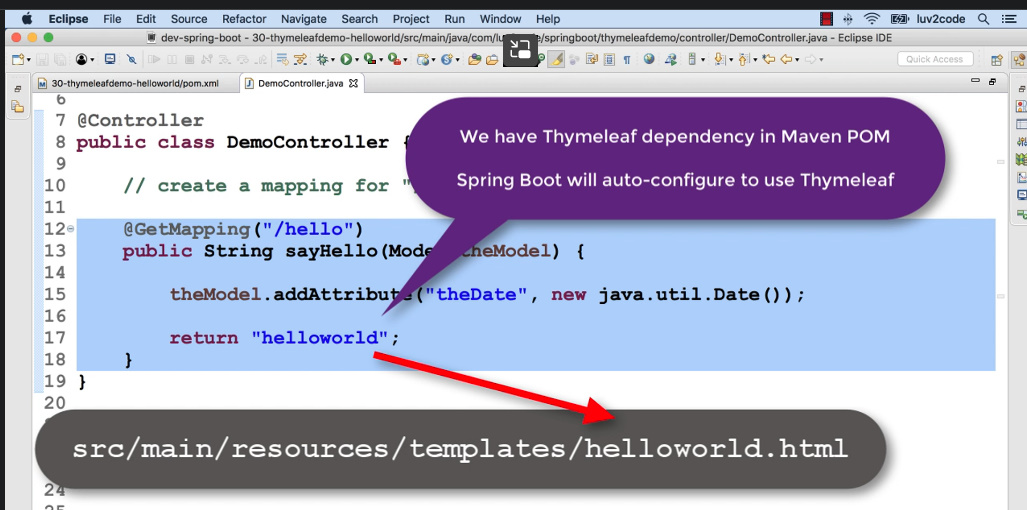




**Deci, odata ce e pusa dependenta pentru thymeleaf, toate paginile vor fi cautate automat in resources/templates!!!Nu in resources/static!!!!Nu trebuie sa mai punem .html la return, e optional.**







Deci, spring va cauta automat paginile html in templates si le va pune .html la urma

**Exemplu**

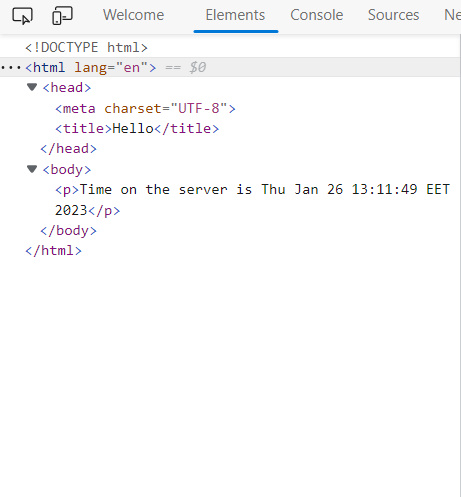
Adaugam in template hello.html:

<!DOCTYPE html>  
<html lang="en" xmlns:th="http://www.thymeleaf.org">  
<head>  
 <meta charset="UTF-8">  
 <title>Hello</title>  
</head>  
<body>  
<p th:text="'Time on the server is '+${theDate}"/>  
</body>  
</html>



**Observam ca**

* xmlns:th= e necesar pentru a folosi thymeleaf expressions(**Nu e obligatoriu, asa cum va fi autogenerat)**
* nu scriem intre <p> </p> ci direct in <p/> tag. Asa functioneaza thymeleaf. El automat va pune acel text, cand se va genera pagina, intre <p> </p>
* Daca inspectam sursa, vedem <p> </p>





* atributele la fel se acceseaza prin ${}

**Usee CSS**

Putem importa o foaie CSS in 2 moduri:

* folosim una locala
* folosim una remote
* Varianta locala:

Spring va cauta CSS tot in resources/static



th:href=”@{/css/demo.css}”

@ - se refera la contextPath, asa incat nu mai trebuie sa scrim acel lung link



Deci in resources putem adauga si:

/META-INF/resources

/resources

/static

/public

Spring le va lua si pe ele in considerare,si va cauta de sus in jos, **dar asta e valabil doar pentru th:href!**

* Remote CSS:

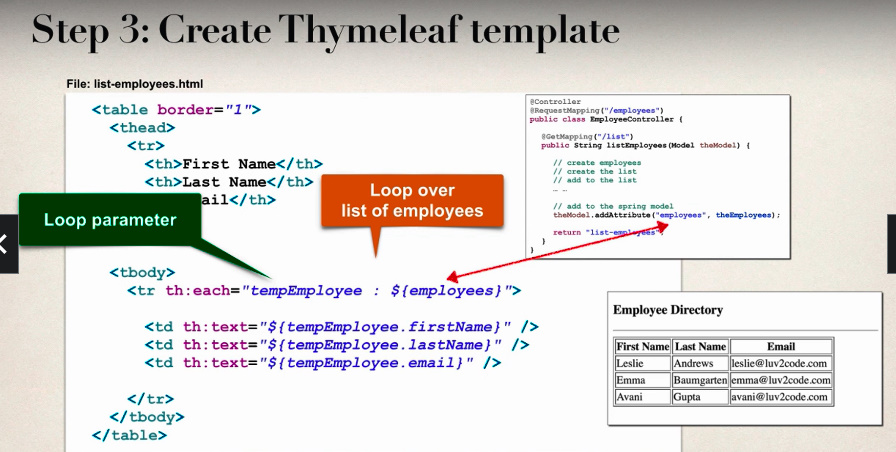


Sau daca vrem sa luam de pe net folosim pur si simplu href:



**HTML Tables**

Loop:

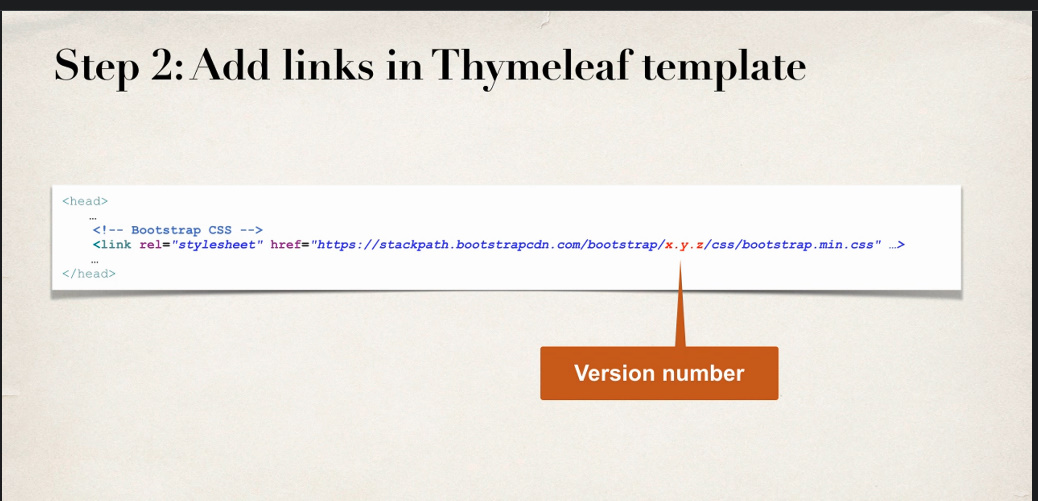


Deci, <tr> va fi repetat cu tot ce se afla intre el de atatea ori cat va fi necesar. Sintaxa e exact ca a forEach in Java

Loop: **th:each**

**get tot se pune automat, de aia punem tempEmployee.firstName**

**Use Bootstrap**



Nu ne trebuie ceva de la thymeleaf pentru a importa ceva din bootstrap. doar href e suficient, asa cum facem in orice HTML project.



**ViewResolver**

* Atentie! ViewResolver este doar pentru JSP. Daca folosim thymeleaf si vrem sa adaugam prefix si sufix, in .properties adaugam asa:
* spring.thymeleaf.prefix=classpath:/templates/views/  
  spring.thymeleaf.suffix=.html