Explications liées au programme Test2

Dans cet exemple, on passe un objet (bean) de requête en requête

@Data
@Component
public class Personne {
private String firstname, lastname;
private String type, sexe;
private String langue;
}
	@Data : annotation Lombok qui évite de devoir coder explicitement les getters et setters et autres (https://fxrobin.developpez.com/tutoriels/java/lombok-retour-experience/)
	
	@Component indique qu'il s'agit d'un bean qui sera géré par Spring (IoC : Inversion of Control)

@Controller
public class HomeController {
 	
@Autowired
Personne personne;
 	
@GetMapping("/")
public String home(**Model model**)
model.addAttribute("user", personne);
return "home";}
 	
@PostMapping("/urlCtrl")
public String page(**Personne personne**,
Model model)
{ **model.addAttribute("user", personne);**
return "suite";
}
}
	@Autowired : annotation principale qui permet l'injection de dépendances. Permet à Spring de résoudre et d'injecter des beans collaboratifs dans un bean. Il faut que **@SpringBootApplication** soit appliquée sur la classe principale de l'application. On peut utiliser **@Autowired** sur des attributs, des setters et des constructeurs.
	
	La requête Get permet d'appeler la vue home.html. Elle contient le formulaire d'encodage d'un objet de la classe personne qui est instancié automatiquement par Spring et mis dans le modèle de la vue home
	
	La requête Post est initialisée par la validation du formulaire. L'objet Personne est donc posté et reçu comme paramètre de la méthode page et mis dans le modèle de la page suite

<body>
  <**form th:action="@{/urlCtrl}"**
  th:object="${user}" method="post">
 	
  Nom : <input type="text" th:field="*{firstname}"/>
 	
  Prenom: <input type="text" th:field="*{lastname}"/>
 	
  <input type="checkbox" th:field="*{type}"
  th:value="Bachelier"/> Bachelier
  <input type="checkbox" th:field="*{type}"
  th:value="Master"/> Master
 	
  <input type="checkbox" th:field="*{sexe}"
  th:value="M"/> Masculin
  <input type="checkbox" th:field="*{sexe}"
  th:value="F"/> Féminin
 	
  <select th:field="*{langue}" >
  <option th:value="Fr"> Français </option>
  <option th:value="Ang"> Anglais </option>
  <option th:value="Néer"> Néerlandais </option>
  </select>
 	
  <button type="submit" > Envoyer</button>
 	
  </form>
</body>
	Page home.html qui utilise le moteur de templates Thymeleaf
	
	th:action="@{/urlCtrl}" :
	path associé à la requête
	
	method="post" : requête Post
	
	➔ @PostMapping("/urlCtrl")
	
	th:object="${user}" :le formulaire utilise l'attribut du modèle user qui est un bean Personne
	
	th:field="*{firstname}" : le champ input est associé à l'attribut user.firstname. On peut d'ailleurs utiliser la notation **th:field="${user.firstname}"**

<body <span th:text="${user.firstname}"/> <span th:text="${user.lastname}"/> <span th:text="${user.type}"/> <span th:text="${user.sexe}"/> </body>	Page suite.html