A voir dans ce tutoriel (Pouiller)

|  |  |  |  |
| --- | --- | --- | --- |
| Identification du Document | | | |
| Client : | CEREMA/DTecITM/CITS | Projet : | tutoriel-web-spring-pouiller |
| Rédacteur : | Daniel Lévy | Directeur de Projets :  Expert : | Daniel Lévy |
| Approbateur : |  | Directeur : |  |

|  |  |  |
| --- | --- | --- |
| Historique des Modifications | | |
| 10/10/2017 | V00.00.00 | Initialisation |
|  |  |  |
|  |  |  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| Diffusion | | |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| Documents de Référence | | |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Table des matières

[1 Renseignements généraux à propos du tutoriel 4](#_Toc496818125)

[2 Objectifs du tutoriel 4](#_Toc496818126)

[2.1 Utiliser l'internationalisation avec SPRING 5](#_Toc496818127)

[2.2 Utiliser les expression Language (EL) et PlaceHolders 5](#_Toc496818128)

[2.3 Créer une base de données 6](#_Toc496818129)

[3 Objectifs poursuivis par le Cerema 6](#_Toc496818130)

[3.1 Adapter le tutoriel en fonction des composants modernes retenus 6](#_Toc496818131)

[3.2 Tester le tutoriel avec les composants modernes 7](#_Toc496818132)

[3.3 Tester le tutoriel avec les serveurs de BD Postgresql, MySql et HsqlDB 7](#_Toc496818133)

[3.4 Réaliser un POM.XML modèle réutilisable 7](#_Toc496818134)

[3.5 Définir les stratégies d'implémentation à retenir pour TraficWeb 7](#_Toc496818135)

[4 Travail réalisé 7](#_Toc496818136)

[5 Capitalisation 7](#_Toc496818137)

[6 Tâches à réaliser par Clemessy 8](#_Toc496818138)

# Renseignements généraux à propos du tutoriel

**Site web** : <http://rpouiller.developpez.com/tutoriels/spring/application-web-spring-hibernate/>

|  |  |
| --- | --- |
| **Type de renseignement** | **Renseignement** |
| Nom du projet Eclipse | **tutoriel-web-spring-pouiller** |
| Site Web du tutorial : | <http://rpouiller.developpez.com/tutoriels/spring/application-web-spring-hibernate/> |
| Localisation du tutorial au format pdf sur le dépôt GITHUB | tutoriel-web-spring-pouiller/conception\_appli/tuto\_pouiller\_spring-application-web.pdf |
| Type de projet | Java 1.8.144, WEB J2EE, MAVEN 3.5.0, SPRING 5.0.0, JPA 2.1, Hibernate 5.2.11 |
| Serveurs de Base de Données | Postgresql 9.6.4, mySql 5.7.19.0, HsqlDB 2.4.0 |
| Nom de la base de données | *base-tutoriel-web-spring-pouiller* |
| Diagramme des classes |  |

# Objectifs du tutoriel

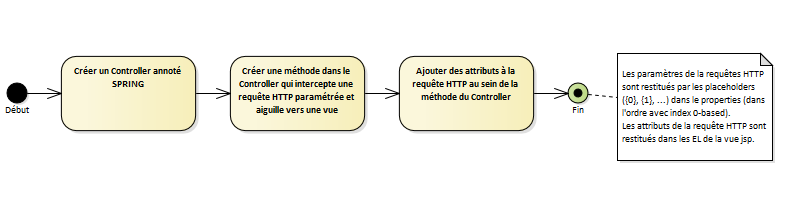
1. **Déclarer le Controller SPRING** dans le descripteur de déploiement d'une application **web.xml**.
2. **Utiliser la configuration par annotations de SPRING** (context : component-scan dans **applicationContext.xml**).
3. **Utiliser l'internationalisation de SPRING** (bean ReloadableResourceBundleMessageSource dans le **applicationContext.xml**).
4. **Utiliser les taglibs de SPRING** dans les jsp.
5. **Créer un Controller annoté par SPRING** interceptant une action et aiguillant vers une ressource (jsp).
6. **Alimenter les attributs d'une requête** dans un Controller annoté SPRING.
7. **Utiliser des Expression Language (EL)** **dans une jsp** pour restituer les attributs d'une requête.
8. **Utiliser des Expression Language (EL) appelées PLACEHOLDERS dans les messages contenus dans les properties** SPRING et les alimenter avec les paramètres d'une requête.
9. **Créer une base de données** (BD)
10. Afficher des données provenant d'une base

## Utiliser l'internationalisation avec SPRING



1. **Déclarer le bean** ReloadableResourceBundleMessageSource dans applicationContext.xml
2. **Préciser le nom de base** des properties contenant les messages SPRING dans applicationContext.xml
3. **Créer les properties** contenant les messages SPRING *sous le classpath*
4. **Incorporer la directive d'inclusion des taglibs SPRING** dans les vues jsp affichant les messages SPRING
5. **Incorporer les balises <spring:message code="key"/>** dans les vues jsp affichant les messages SPRING

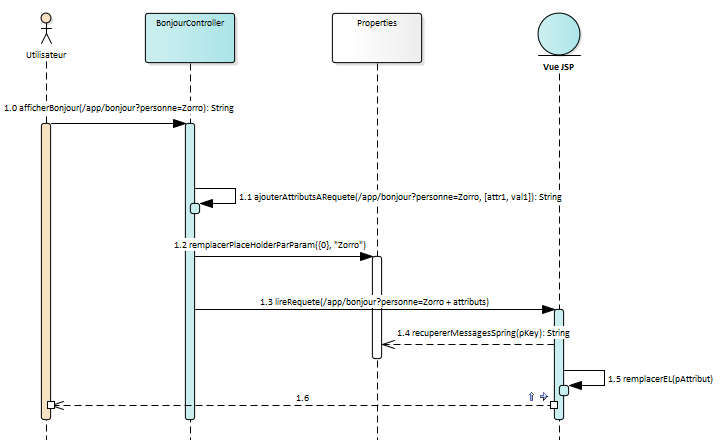
## Utiliser les expression Language (EL) et PlaceHolders



1. **Créer un Controller** annoté SPRING
2. **Créer une méthode dans le Controller** qui intercepte une **requête HTTP paramétrée** et aiguille vers une vue
3. **Ajouter des attributs à la requête HTTP** au sein de la méthode du Controller

Les paramètres de la requêtes HTTP sont restitués par les placeholders ({0}, {1}, ...) dans le properties (dans l'ordre avec index 0-based).

Les attributs de la requête HTTP sont restitués dans les EL de la vue jsp.



## Créer une base de données

Copier les jar des connecteurs des BD (Postgresql, MySql, HsqlDB) utilisées dans le répertoire lib du serveur web TOMCAT (si ce n'est déjà fait).

Créer la BD dans le Gestionnaire (pgAdmin4 pour Postgresql, ...)

# Objectifs poursuivis par le Cerema

Utiliser SPRING 5.0.0

Appliquer Maven 5.0.0

Appliquer HTML5

## Adapter le tutoriel en fonction des composants modernes retenus

## Tester le tutoriel avec les composants modernes

## Tester le tutoriel avec les serveurs de BD Postgresql, MySql et HsqlDB

## Réaliser un POM.XML modèle réutilisable

## Définir les stratégies d'implémentation à retenir pour TraficWeb

# Travail réalisé

# Capitalisation

L'utilisation de l'archétype MAVEN "**maven-archetype-webapp**" est à **proscrire**. Le POM résultant est totalement inutilisable car les composants utilisés sont tous dépréciés (Java SE1.5 !!!). Il faut utiliser directement le POM établi par le Cerema.

# Tâches à réaliser par Clemessy

Cf le classeur Excel "**Suivi\_Collaboration\_Cerema\_Clemessy.xlsx**", onglet "**tâches à réaliser**".

* Parcourir le présent document.
* Relire et Valider les modifications apportées par le Cerema au readme.txt.
* Déterminer si les placeholders dans les properties sont du J2EE pur ou du Spring.
* Voir rapidement si MySql 5.7.19.0 peut être utilisée. La notion de DataBase semble avoir disparu dans MySQL (pour la notion de schéma), la notion de schéma semble être l'équivalent de la Database dans Postgresql, la connexion à établir dans le context.xml pour MySQL est compliquée (pas de SSL, problème de TimeZone (url=*"jdbc:mysql://localhost:3306/base-tutoriel-web-spring-pouiller****?serverTimezone=UTC&amp;useSSL=FALSE"***), …).
* Essayer de comprendre le pourquoi du message de démarrage de TOMCAT "INFOS: At least one JAR was scanned for TLDs yet contained no TLDs. Enable debug logging for this logger for a complete list of JARs that were scanned but no TLDs were found in them. Skipping unneeded JARs during scanning can improve startup time and JSP compilation time."
* Essayer de voir si il serait possible de lancer "partiellement" une application sans instancier le bean de Spring *LocalContainerEntityManagerFactoryBean*. En clair, de lancer une application sans la connexion à la base de données.