

4ème année G.INFO

2019/2020

**TP1 Génie Logiciel**

**Environnement de travail :**

Installer Maven

Télécharger eclipse 2019-12

Installer les plugins spring tools pour eclipse :

[Spring Tools 3 Add-On for Spring Tools 4](https://marketplace.eclipse.org/content/spring-tools-3-add-spring-tools-4)

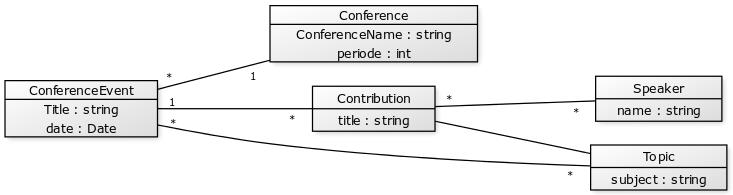
[Spring Tools 4 (aka Spring Tool Suite 4)](https://marketplace.eclipse.org/content/spring-tools-4-aka-spring-tool-suite-4)

Un serveur de base de données Mysql

**But de Tp**

Le But de notre application est la gestion d’une conférence scientifique. La conférence est un événement qui se déroule périodiquement (chaque année, toutes les deux années…). Chaque évenement discute plusieurs topic ou sujet. Et dans chaque sujet plusieurs contributions vont être présentées.

Soit le diagramme de classe suivant :



Le but de ce Tp est de se familiariser avec l’environnent d’inversion de contrôle spring en faisant une première configuration xml puis une autre configuration java annotations.

1. Créer un *Maven Project*, séléctionner *Create a simple project (skip archetype selection)*. Définir *Group Id* et *Artifact Id*
2. Configurer le pom.xml Project Object Model de telle sorte à importer les dépendence suivant :

hibernate-core version 5.4.10.Final

spring-context version 5.2.3.RELEASE

spring-tx version 5.2.3.RELEASE

spring-data-commons version 2.2.1.RELEASE

spring-data-jpa version 2.2.1.RELEASE

hibernate-entitymanager 5.4.10.Final

mysql-connector-java version 5.1.34

1. Créer les entités de persistence
2. Créer Spring Bean Configuration File, choisir les names spaces : beans, context, tx et jpa
3. Dans ce fichier de configuration définir les beans :

Datasoure :

<bean id=*"dataSource"*

class=*"org.springframework.jdbc.datasource.DriverManagerDataSource"*>

<property name=*"driverClassName"* value=*"com.mysql.jdbc.Driver"* />

<property name=*"url"* value=*"jdbc:mysql://localhost:3306/spring\_jpa"* />

<property name=*"username"* value=*"root"* />

<property name=*"password"* value=*""* />

</bean>

entityManagerFactory :

<bean id=*"myEmf"*

class=*"org.springframework.orm.jpa.LocalContainerEntityManagerFactoryBean"*>

<property name=*"dataSource"* ref=*"dataSource"* />

<property name=*"packagesToScan"* value=*"ma.ensa.entities,ma.ensa.dao"* />

<property name=*"jpaVendorAdapter"*>

<bean class=*"org.springframework.orm.jpa.vendor.HibernateJpaVendorAdapter"* />

</property>

<property name=*"jpaProperties"*>

<props>

<prop key=*"hibernate.hbm2ddl.auto"*>create</prop>

<prop key=*"hibernate.dialect"*>org.hibernate.dialect.MySQL5Dialect</prop>

</props>

</property>

</bean>

Un gestionnaire de transaction :

<bean id=*"transactionManager"* class=*"org.springframework.orm.jpa.JpaTransactionManager"*>

<property name=*"entityManagerFactory"* ref=*"myEmf"* />

</bean>

<tx:annotation-driven/>

 activer la génération automatique des implémentations

<jpa:repositories base-package=*"ma.ensa.dao"*

entity-manager-factory-ref=*"myEmf"* >

</jpa:repositories>

1. Utiliser spring Data pour créer l’interface Repository de persistence.
2. Créer un classe de teste : créer une instance du conteneur, recupèrer le bean ConferenceRepository. Tester les méthodes de ce repository