

1 Contexte

Nous nous plaçons dans le cadre du développement d'une application permettant de gérer la liste des participants à des réunions/colloques. Ces événements sont en général organisés par des groupes de personnes, autour d'un thème donné. L'application doit permettre de gérer les flux d'inscriptions des participants. Nous ne nous intéresserons ici qu'à une petite partie de l'application.

2 Objectifs de ce TP

- Découvrir la gestion de la **persistance des objets** Java à l'aide du framework **Hibernate** et des annotations JPA (Java Persistence API).
 - *Documentation officielle Hibernate* : <http://hibernate.org/orm/documentation/5.4/>
 - *Hibernate Annotations* : http://docs.jboss.org/hibernate/orm/5.4/quickstart/html_single/
- Construire une classe persistante « Participant » avec ce framework et l'utiliser dans une application Java ;
- Faire le lien avec une autre classe persistante « Evenement »

3 Configuration de base du projet

- Sur votre serveur de base de données PostgreSQL, créer une base sans aucune table ;
- Dans votre IDE, créer un projet Maven, nommé « colloque »
- Dans l'arborescence du projet, créer 3 packages :
 - `colloque.metier` (qui contiendra les classes « métier »)
 - `colloque.services` (qui contiendra les classes permettant d'assurer la persistance)
 - `colloque.session` (qui contiendra le main du projet, et l'initialisation de la sessionFactory Hibernate)
- Ajouter les dépendances Maven pour PostgreSQL et pour Hibernate 5.4 (core ORM)
A chercher ici : <https://mvnrepository.com/>
- Créer le fichier de configuration `hibernate.cfg.xml` comme dans le TP précédent

4 Persistance d'une classe

- Dans le package `colloque.metier`, créer une classe **Participant** comme indiquée dans le schéma de la section suivante.

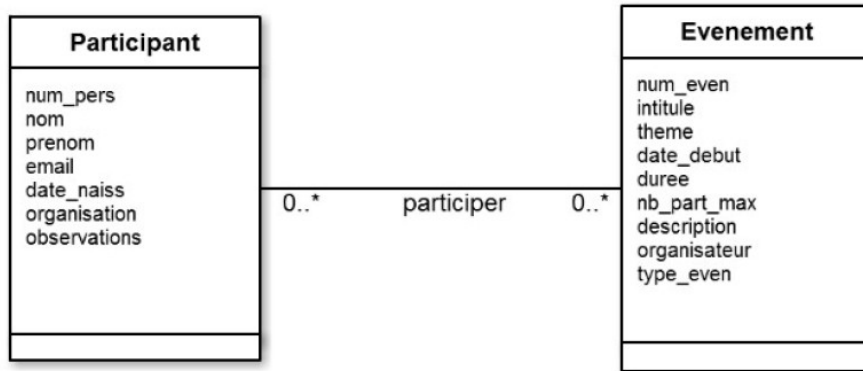
Vous utiliserez les annotations Hibernate pour paramétrer la persistance :

http://docs.jboss.org/hibernate/orm/5.4/quickstart/html_single/#tutorial_annotations

- Dans le package `colloque.services`, créer une classe **ParticipantService** permettant d'utiliser les services de persistance proposés par Hibernate (se reporter au document du cours).
- Dans le package `colloque.session`,
 - créer une méthode permettant d'initialiser la sessionFactory d'Hibernate
 - dans le main, créer quelques participants et insérer-les dans la base de données en utilisant la classe « ParticipantService »
- Exécuter votre projet
- Se connecter à Postgresql pour observer ce qui s'est passé dans la base.

5 Liaison entre deux tables

Voici un diagramme de classe plus précis représentant la situation décrite précédemment.



- Modifier votre application afin de représenter cette situation en Java. On supposera un mapping uni-directionnel et une responsabilité sur la classe Evenement (les participants devront être supprimés si un événement est supprimé).
- Les fonctionnalités demandées pour la classe Participant dans le TP précédent devront être implémentées
- Modifier votre projet afin d'intégrer la classe Evenement dans la persistance des données ainsi que l'association entre les deux classes Participant et Evenement. Il faudra penser également modifier le fichier XML de configuration Hibernate pour y faire apparaître le mapping de la classe Evenement.
- Ajouter quelques événements, et quelques participants à ces événements.

6 Héritage

On suppose désormais que certains participants sont des "Intervenants" qui prendront la parole pendant les réunions.

- Créer une classe Intervenant qui hérite de la classe Participant en y ajoutant deux champs : un numéro de téléphone, et un titre (qui précise sa fonction).
- Ajouter les annotations de persistance, notamment le choix de la stratégie de traduction de l'héritage.
- Ajouter quelques Intervenants dans le programme et vérifier qu'ils apparaissent dans la base de données