TAA GOOGLE WEB TOOLKIT

Auteurs: Sylvie-Auneau/Nicolas-Evano

sommaire:

1.0-Objectif:

Développer un projet en GWT déployé sur Google App Engine. l'idée est d'afficher la position courante d'un utilisateur grâce à l'api lattitude de Google.

2.0-Les choix de conception:

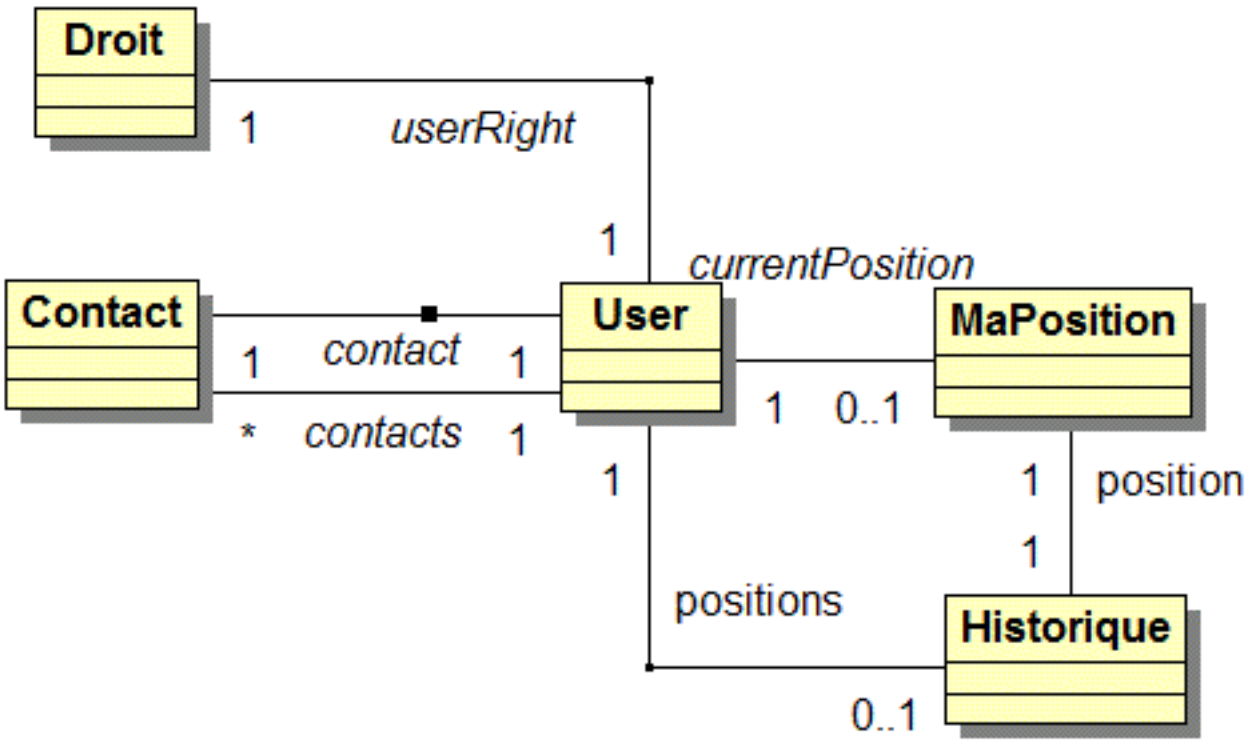
Google App Engine offre le choix entre JPA et JDO pour gérer la persistence. Nous avons choisi JDO parce qu'il offre plus de fonctionnalités et est plus stable sur la plate forme de Google.

JDO permet d'utiliser des filtres pour s'abstraire complètement des requêtes SQL.

JDO souffre pourtant d'un manque de maturité. Pour exemple il est recommandé pour les relations un à plusieurs d'utiliser des clefs primaires de type KEY d'une façon générale les long sont à proscrire. Le type String peut également être employé sous réserve d'utiliser l'annotation suivante:

@Extension(vendorName="datanucleus", key="gae.encoded-pk", value="true")

Dans le modèle métier utilisé, la table User utilise comme clef primaire un String:



modéle métier employé en TAA[1].

une fois le service RLattitudeService[2] implémenté nous avons commencé la conception du client.

3.0-L'aspect client:

GWT est en fait du code java traduit en java script par un compilateur il est donc important de bien comprendre que le résultat sera du javascript et non du "byte code" executé par le moteur javascript du navigateur du client.

Il faut également comprendre le mécanisme de callback employé par GWT pour dialoguer avec des services distants à savoir: nous sommes en mesure de prévoir quand on invoque une opération du service sur le serveur distant mais l'on ne sais pas quand le retour de l'opération nous parviendra.

L'api Maps de GWT a également été intégrée au client (api latitude en alpha pour GWT actuellement risquée).

4.0-Les fonctionnalités de l'application:

Quand un utilisateur accède à l'IMH il peut:

-Se connecter en rentrant: un login et password grâce au bouton Login.

-Il peut créer un nouvelle utilisateur grâce au bouton créer un utilisateur,bouton Create User.

1.Utilisé pour le TP1 et TP4
2.Service définit pour notre application

Quand l'utilisateur est connecté la liste de ses contacts est affichée sur le bord gauche de l'écran ainsi que son nom et prénom, il est par défaut caché des autres utilisateurs. dans cet état il peut:

-Se déconnecter, bouton Disconnect.

-Publier sa position, bouton Publish Position.

-Afficher sa position sur une carte ainsi que celle de l'ensemble de ses autres contacts si ces derniers se sont rendus visibles aux autres, bouton Display Position.

-Ajouter un contact, bouton Add Contact.

-Il peut se rendre visible des autres utilisateurs qui l'on ajoutés en contact en changeant son droit de visibilité, bouton Show Me/Hide Me.

5.0-ce que propose la base de données de l'application actuellement:

Dans la base de données actuellement il y a deux utilisateurs tmp1 (login: tmp1/pasword: tmp1) ainsi que tmp2 (login: tmp2/password: tmp2) et tmp3(login: tmp3/password: tmp3). tmp2 et tmp3 sont contacts de tmp1.

6.0-conclusion:

GWT est puissant une fois compris le mécanisme de callback utilisé avec le client pour dialoguer avec les services distants. Le choix du concept de la persistence utilisé avec GAE manque de souplesse beaucoup de limitations avec JPA. JDO reste perfectible pourtant il est plus stable et offre plus de fonctionnalités que JPA (restrint à la version 1) avec GAE. Google propose une plate-forme commode d'emploi, bien documentée.

Bibliographie

Cette bibliographie donne des références vers des livres.

Auteur	Titre
Sami-Jaber	**Programmation GWT 2**
William-H. Inmon	**Construire un Datastore opérationel**