



# **Le rapport I.K.Y.A.**

Projet Réseaux Middlewares

Yishuo LYU

Zunzun WANG

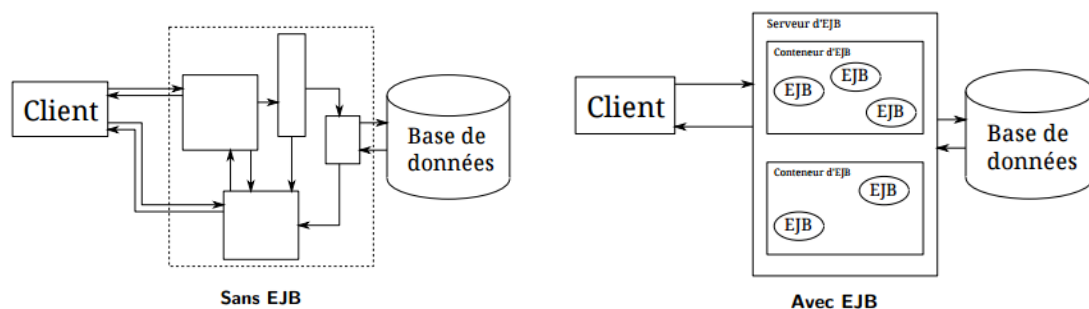
Sabrine TIZAOUI

Laodice MELLITI

## L'architecture de I.K.Y.A. :

Notre projet est composé par 5 parties, Client, ejb-projetClient, ejb-projet, ejb-projetEAR, jpa-projet. Dans ces 5 parties, Client réalise principalement l'affichage des fenêtres et l'appelle des interfaces, ejb-projetClient définit principalement les interfaces liées avec Client, ejb-projet réalise principalement toutes les fonctions, ejb-projetEAR (Entreprise Application Projet) a pour but d'intégrer l'environnement des 3 projets au-dessus et jpa-projet réalise principalement la persistance et à connecter la base de données.

La description et la fonction du projet EJB (Entreprise Java Beans) :



- ✓ Ils sont éléments de Java EE, et permettent de développer des applications distribuées.
- ✓ Ce sont des composants réutilisables, persistants et transactionnels qui proposent des services.
- ✓ Ces "briques" permettent de construire une application distribuée complexe.
- ✓ Ils ne peuvent s'exécuter que dans un environnement serveur dédié : un serveur EJB.

La description et la fonction du projet JPA (Persistence Java) :

Il existe plusieurs méthodes de persistance : Le driver de la base de données (jdbc), les interfaces ORM (Hibernate, Toplink) et les API (JPA, Morphia).

1. La persistance permet de :
  - i. Sauvegarder les données d'un programme
  - ii. Sauvegarder l'état d'un programme
  - iii. Restaurer les données et l'état du programme
  - iv. Stopper le programme sans perdre ses données ou son état
2. La sauvegarde peut être :
  - i. Locale ou distante
  - ii. Dans un fichier (xml, json) ou dans une base de données
3. Le mapping des données est primordial

## La description de chaque EJBs (sessions et entités) :

Dans notre projet, il existe 4 entités : User, List\_contact, list\_defisapply et list\_friendapply.

noté: Marin est un test entité

Table	Action	Lignes	Type	Interclassement	Taille	Perte
List_contact	Afficher Structure Rechercher Insérer Vider Supprimer	4	InnoDB	latin1_swedish_ci	16 Kio	-
list_defisapply	Afficher Structure Rechercher Insérer Vider Supprimer	2	InnoDB	latin1_swedish_ci	16 Kio	-
list_friendapply	Afficher Structure Rechercher Insérer Vider Supprimer	2	InnoDB	latin1_swedish_ci	16 Kio	-
Marin	Afficher Structure Rechercher Insérer Vider Supprimer	0	InnoDB	latin1_swedish_ci	16 Kio	-
USER	Afficher Structure Rechercher Insérer Vider Supprimer	9	InnoDB	latin1_swedish_ci	16 Kio	-
5 table(s)	Somme	17	InnoDB	latin1_swedish_ci	80 Kio	0 o

La structure de l'entité User : (USERID est la clé primaire.)

#	Nom	Type	Interclassement	Attributs	Null	Défaut	Extra
1	USERID	int(11)			Non	Aucune	AUTO_INCREMENT
2	ISONLINE	tinyint(1)			Oui	0	
3	USERPASSWORD	varchar(255)			Oui	NULL	
4	USERSEX	varchar(255)			Oui	NULL	
5	USERNAME	varchar(255)			Oui	NULL	
6	USEREMAIL	varchar(255)			Oui	NULL	
7	USERACCOUNT	varchar(255)			Oui	NULL	
8	UserRecord	int(255)			Non	0	

La structure de l'entité List\_contact : (contact\_id est la clé primaire.)

#	Nom	Type	Interclassement	Attributs	Null	Défaut	Extra
1	contact_id	int(11)			Non	Aucune	AUTO_INCREMENT
2	USERID1	int(11)			Oui	NULL	
3	USERID2	int(11)			Oui	NULL	

La structure de l'entité list\_defisapply : (id est la clé primaire.)

← Serveur: 127.0.0.1 » Base de données: BDD\_Middleware » Table: list\_defisapply

Afficher Structure SQL Rechercher Insérer Exporter

Structure de table Vue relationnelle

#	Nom	Type	Interclassement	Attributs	Null	Défaut	Extra
<input type="checkbox"/> 1	<b>id</b>	int(255)			Non	Aucune	AUTO_INCREMENT
<input type="checkbox"/> 2	<b>from_userId</b>	int(255)			Non	Aucune	
<input type="checkbox"/> 3	<b>to_userId</b>	int(255)			Non	Aucune	
<input type="checkbox"/> 4	<b>date</b>	datetime			Non	Aucune	
<input type="checkbox"/> 5	<b>isRead</b>	tinyint(1)			Non	0	

La structure de l'entité list\_friendapply : (ID est la clé primaire.)

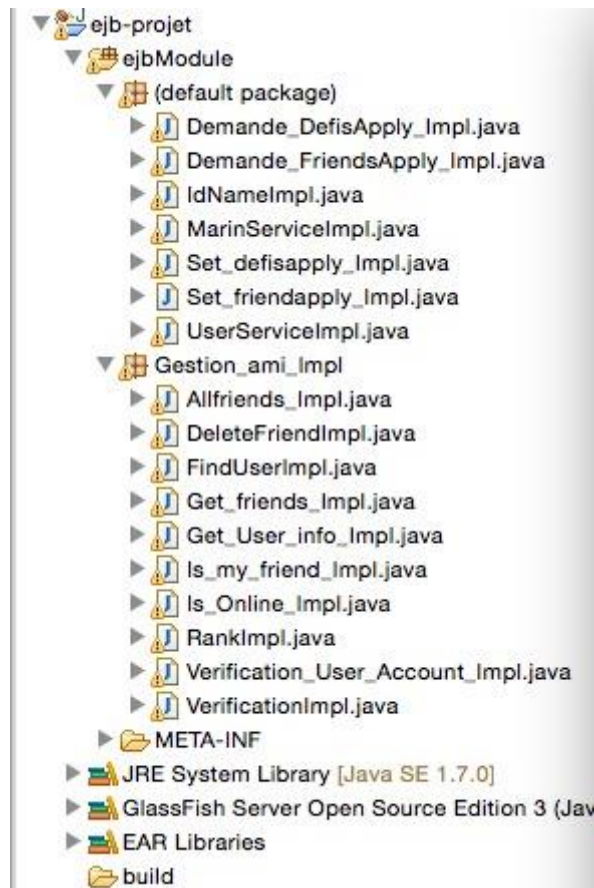
← Serveur: 127.0.0.1 » Base de données: BDD\_Middleware » Table: list\_friendapply

Afficher Structure SQL Rechercher Insérer Exporter

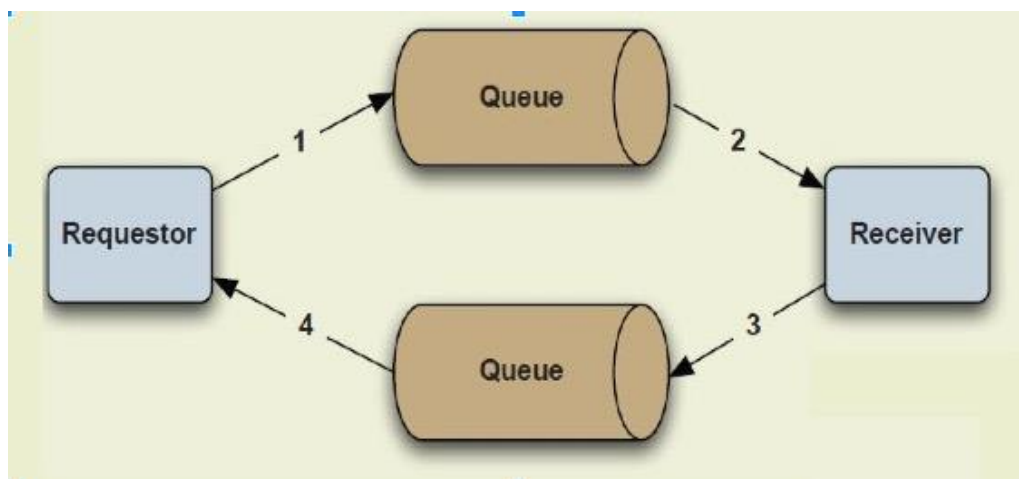
Structure de table Vue relationnelle

#	Nom	Type	Interclassement	Attributs	Null	Défaut	Extra
<input type="checkbox"/> 1	<b>ID</b>	int(11)			Non	Aucune	AUTO_INCREMENT
<input type="checkbox"/> 2	<b>TO_USERID</b>	int(11)			Oui	NULL	
<input type="checkbox"/> 3	<b>FROM_USERID</b>	int(11)			Oui	NULL	
<input type="checkbox"/> 4	<b>DATE</b>	date			Oui	NULL	
<input type="checkbox"/> 5	<b>ISREAD</b>	tinyint(4)			Oui	NULL	

Et il existe plusieurs sessions pour réaliser les différentes fonctions :



La session de Messagerie instantanée : (réalisé par Glassfish4)



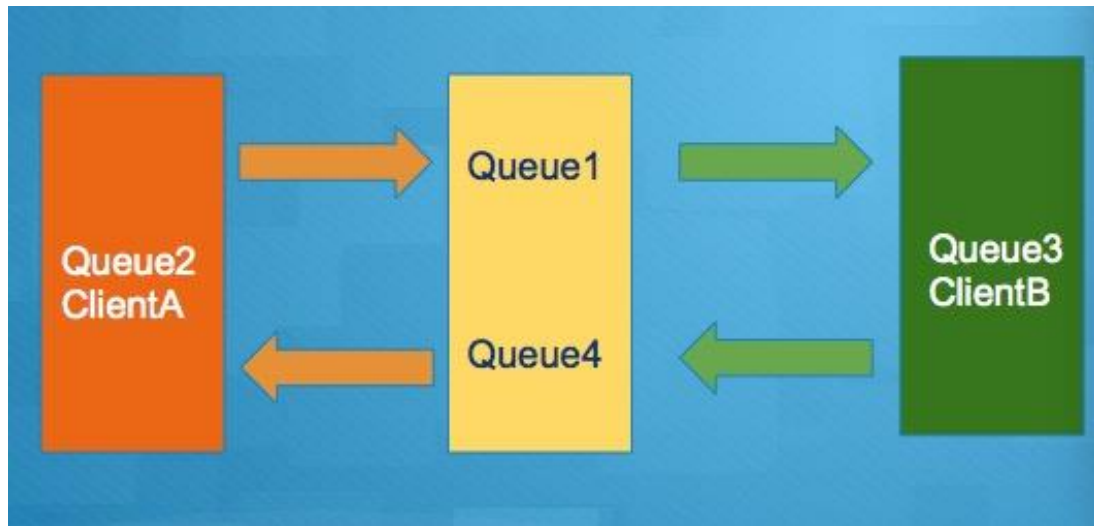
Queue1 et queue4 sont les files d'écoute du projet EJB, queue2 et queue3 sont les files d'écoute du projet Client.

Dans notre projet, Client A écoute queue2 et Client B écoute queue3.

C'est un modèle P2P, Client A envoie un message à queue1, EJB va fonctionner quand il écoute la nouvelle information de la queue1 (*function onMessage*). Nous mettons la fonction onMessage pour transmettre la même information à queue3, parce que l'écoute de Client B est queue3 par le serveur.

De même, Client B envoie un message à queue4, EJB fonctionne la fonction onMessage pour l'envoyer à queue2 quand il écoute ce message venant de queue4. Donc Client A peut recevoir le message que client B a envoyé par l'écoute de la queue2.

Nous pouvons réaliser la communication entre Client A et Client B par la processus au-dessus.

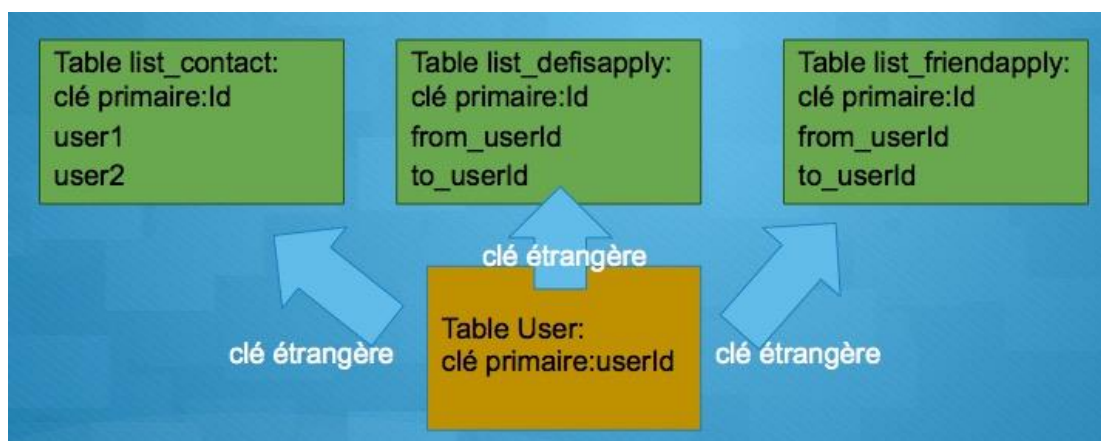


## La justification de votre choix pour la base de données :

Dans notre projet, nous choisissons Mysql comme la base de données. Mais en fait la persistance a une grande influence sur le serveur Mysql et il y a plusieurs trappes sur l'intégration avec les autres parties. Nous perdons beaucoup de temps pour configurer l'environnement d'exécution sur JEE.

Par exemple, pour ces 5 parties dans notre projet, il faut utiliser jdk 1.6 pour configurer le projet JPA et jdk 1.7 pour configurer les autres parties.

## Un schéma de votre base de données





## Les problèmes que vous avez rencontrés et les améliorations que vous pourriez prévoir :

Dans notre projet, il y a deux problèmes que nous rencontrons dans la processus de faire le projet :

1. Nous réalisons Messagerie instantanée par Glassfish 4.0, mais les autres fonctions sont réalisés par Glassfish Server Open Source Edition 3, c'est à cause de différences entre Glassfish 3 et 4 (par exemple, les interfaces de jmscontext, Connectionfactory etc.). Parce que les deux version de Glassfish ne sont pas compatible, les deux parties (la partie Messagerie instantanée et le reste) sont distincts.

2. Le Glassfish est instable et souvent se bloque. Toutes les données dans la base de données sont supprimés quand le Glassfish se bloque.

Solution : Les commandes Mysql dans Java ont un peu de différences avec les commandes Mysql, il faut essayer d'éviter des erreurs des commandes Mysql dans Java lorsque l'exécution de la programme.

Par exemple : (BASIC JPA DEVELOPMENT – QUERYING – JPQL)

### Select Queries

Select queries can be used to read objects from the database. Select queries can return a single object or data element, a list of objects or data elements, or an object array of multiple objects and data.

Select query examples

```
// Query for a List of objects.
Query query = em.createQuery("Select e FROM Employee e WHERE e.salary > 100000");
List<Employee> result = query.getResultList();

// Query for a single object.
Query query = em.createQuery("Select e FROM Employee e WHERE e.id = :id");
query.setParameter("id", id);
Employee result2 = (Employee) query.getSingleResult();

// Query for a single data element.
Query query = em.createQuery("Select MAX(e.salary) FROM Employee e");
BigDecimal result3 = (BigDecimal) query.getSingleResult();

// Query for a List of data elements.
Query query = em.createQuery("Select e.firstName FROM Employee e");
List<String> result4 = query.getResultList();

// Query for a List of element arrays.
Query query = em.createQuery("Select e.firstName, e.lastName FROM Employee e");
List<Object[]> result5 = query.getResultList();
```

Cf.:

[http://wiki.eclipse.org/EclipseLink/UserGuide/JPA/Basic\\_JPA\\_Development/Querying/JPQL](http://wiki.eclipse.org/EclipseLink/UserGuide/JPA/Basic_JPA_Development/Querying/JPQL)