



Projet
Application centrée utilisateur

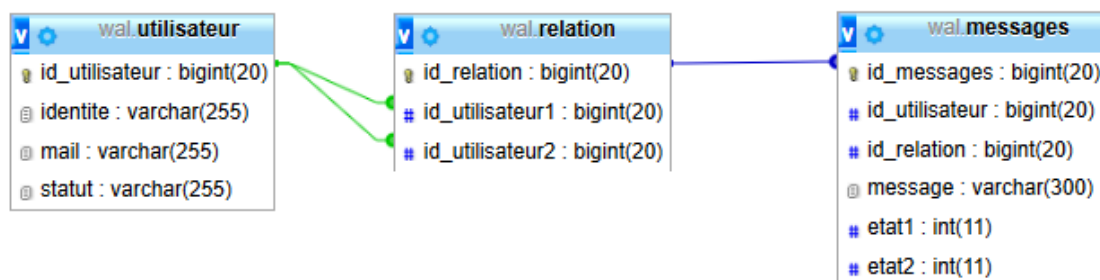
On souhaite mettre en place une application de messagerie type **WhatsApp**. Nous passons pour cela par une architecture orientée service (SOA) qui offre plus de souplesse dans le développement et en particulier dans l'adaptation de l'application aux différentes plateformes.

Notre objectif sera de proposer (i) une version logicielle de l'application pour une utilisation sur un ordinateur classique et (ii) une version WEB du même logiciel pour une utilisation sur un navigateur et en particulier sur des plateformes mobiles.

L'architecture orientée service garantit que les traitements et le stockage sont effectués coté server. Seule l'interface graphique sera recodée pour l'adapter aux différentes plateformes.

Exercice 1 : Découverte des services

Le modèle de données utilisé pour stocker les données de l'application est le suivant :



Tous les services sont accessibles depuis l'adresse :

serveur: <http://ns4004962.ip-198-27-65.net/applications/WAL>

id : valeur reçue après activation du compte

Le moteur de l'application est construit autour de 7 services qui renvoient, pour chacun, sa réponse dans un format JSON.

1. **Inscription:**

serveur /wal.php?inscription&identite=Prenom_Nom&mail=adr_mail

Retour :

```
{"etat" : { "reponse" : 1, "message" : "Un mail de validation vous a été envoyé"}}
```

Complément : Un mail est envoyé avec un code d'activation du compte.

2. **Activation :** serveur /wal.php?activation&identifiant=id

Retour :

```
{"etat" : { "reponse" : 1, "message" : "Compte activé", "identifiant" : "XXXXXXXXXX"}}
```

3. **Obtenir ses paramètres :**

serveur/wal.php?identifiant=id

Retour :

```
{"etat" : { "reponse" : 1, "message" : "identification"}, "identifiant" : XX, "identite" : "Prenom NOM" }
```

4. **Ajout d'une relation :**

serveur/wal.php?identifiant=id&lier&mail=adresse_mail

Retour :

```
{"etat" : { "reponse" : 1, "message" : "association réussit"}}
```

Complément :

L'utilisateur à mettre en relation doit préalablement s'inscrire

5. **Lister les relations :**

serveur/wal.php?identifiant=id&relations

Retour :

```
{"etat" : { "reponse" : 1, "message" : "liste des relations"}, "relations": [{"relation": XX, "identifiant": XX, "identite": "prenom NOM"}, ...]}
```

6. **Envoyer message:**

serveur/wal.php?ecrire&destinataire=id&message=mon_mess

Retour :

```
{"etat" : { "reponse" : 1, "message" : "message enregistré"}}
```

7. **Lire messages :**

serveur/wal.php?lire&relation=identifiant_relation

Retours :

```
{"etat" : { "reponse" : 1, "message" : "discussion"}, "messages": [{"identite" : "prenom NOM", "message" : "texte du message"}, ...]}
```

Complément :

messages est un tableau contenant la liste des échanges

Exercice 2 : Maquette de l'application

1. Trouvez un nom pour votre logiciel.
2. Identifier toutes les fenêtres que vous souhaitez mettre à disposition de vos utilisateurs.
3. En utilisant un outil de maquettage tel que **Balsamik** ou **Cacoo** faite des maquettes des différentes fenêtres de votre logiciel.