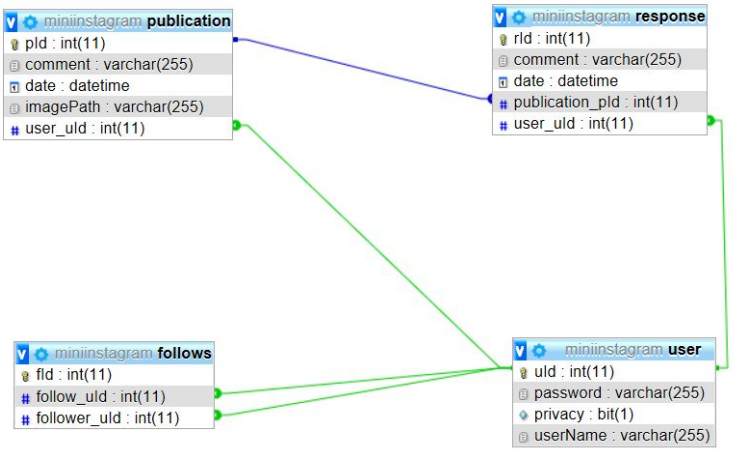
Architectures des systèmes d'information

TP noté Sujet 3 -

Web Service d'Instagram REST et SOAP

ABULIMITI Alafate (DI4-SI2)
LI Yuanyuan (DI4-SI2)
20/05/2018

1. Schéma de la base de données



2. L'architecture de l'application

1. Modèle

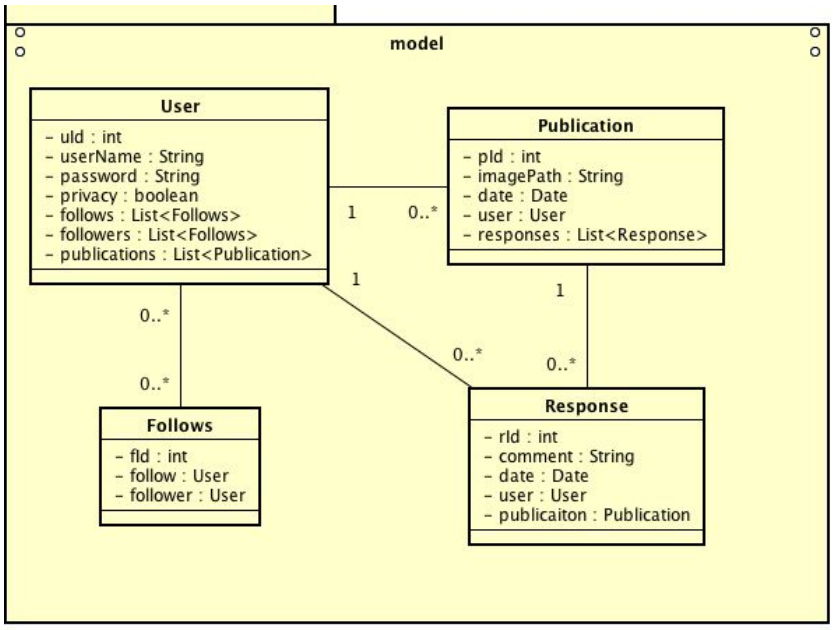
Nous utilisons le framework hibernate pour construire la structure de base de données, il y a quatre classes : Publication , User, Follows, Response

Publication: les informations d'une publicaition

User : les inforamations d'un utilisateur

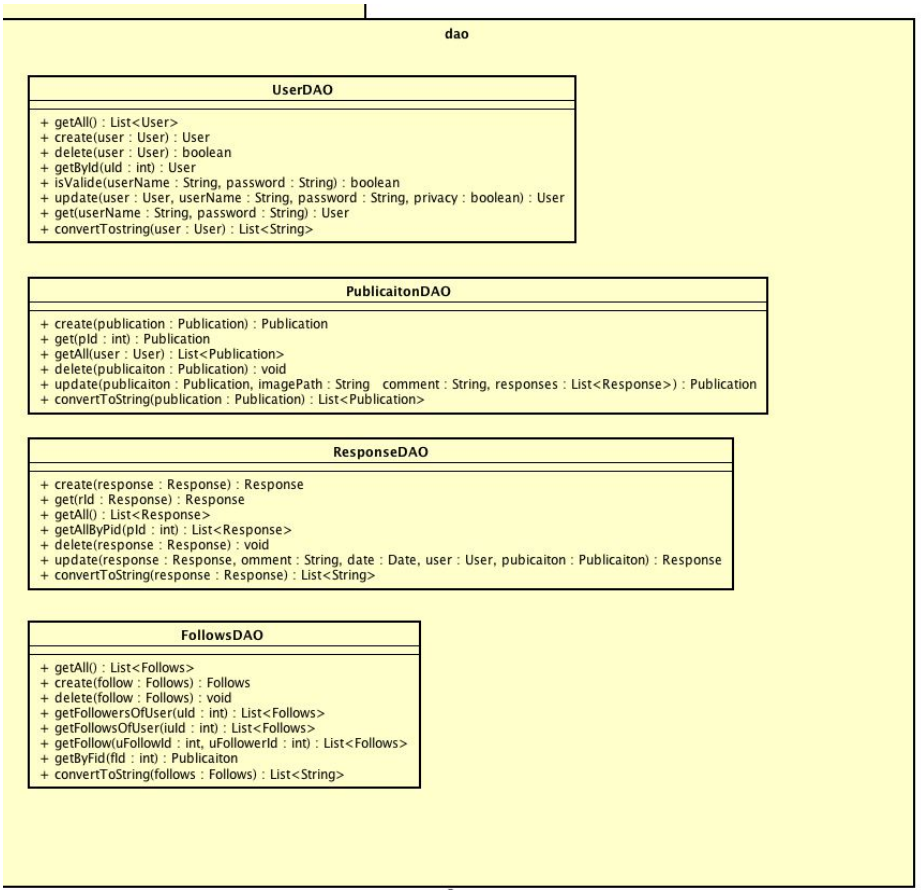
Follows: les interactions entre les utilisateur

Response : les actions des utilisateur.



2. DAO

Dans la partie de contrôleur, nous avons le dao (DATA ACCESS OBJECT) : pour gérer les données.

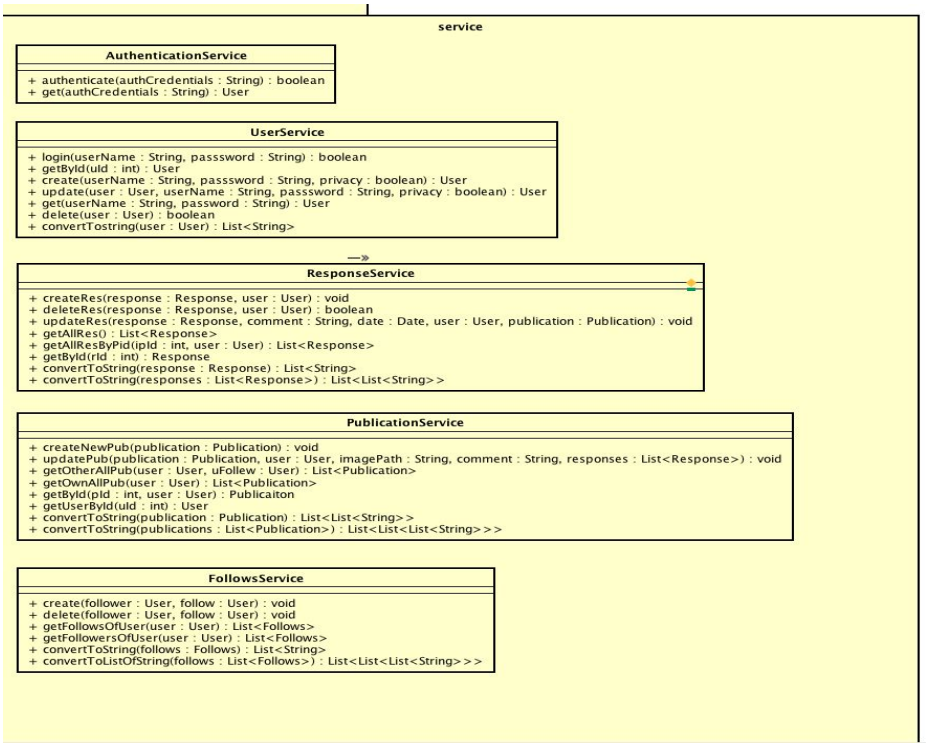


Dans la partie de web service, nous avons découpé à deux partie :

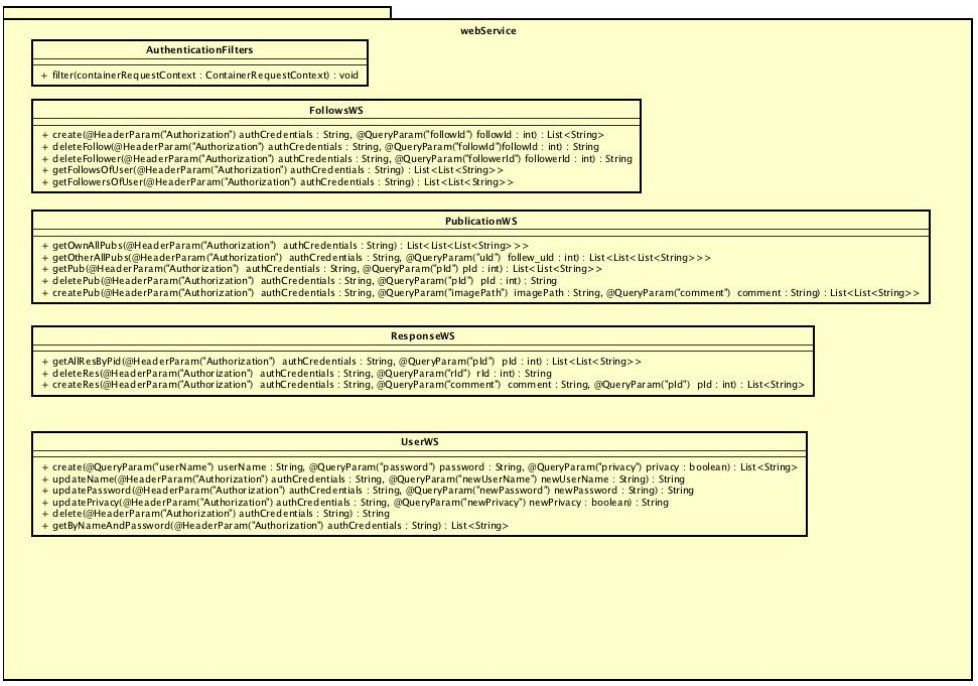
Serivce : pour faciltiter à modifier les fonctions et la maintenir.

WebService : pour gérer les webservices.

3. Serivce



4. Web Service



3. Les spécifications de REST

Tous les notre web service retournent du type ‘APPLICATION_JSON’ ou ‘TEXT_PLAIN’. Et on teste notre web service vers console ‘System.out.println’. Il y a plein de services dans notre serveur webservice REST. On juste teste quelques fonctions pour présenter notre service.

Il y a deux type d’utilisateur: privé et publique. On ajoute un attribut de “privacy”, si privacy = true, les publications sont visibles seulement pour les followers , puisque les ‘followers’ peuvent les répondre. Si privacy = false, tous le monde peuvent les voir et les répondre. Donc la possibilité est comme ci-dessous:

Tableau 3-1 La possibilité de Visualisation de publication et de répondre

	Suivre (follow)	**Ne suivre pas (do not follow)**
Publique	Oui	Oui
Privé	Oui	Non

3.1. Créer un utilisateur

Du côté client, on d’abord créer un utilisateur. On put regarder que le client ‘Li’ est créé.

```
44
45      /** Test: webService create a user: @POST */
46      WebTarget userWebTargetcreate = userWebTarget.queryParam("userName", "Li");
47      WebTarget userWebTargetcreateName = userWebTargetcreate.queryParam("password", "Yuanyuan");
48      WebTarget userWebTargetcreatePrivacy = userWebTargetcreateName.queryParam("privacy", true);
49      Invocation.Builder invocationBuilderCreateUser = userWebTargetcreatePrivacy.request(MediaType.APPLICATION_JSON);
50      invocationBuilderCreateUser.header("some-header", "true");
51      Response responseCreate = invocationBuilderCreateUser.post(Entity.entity("create", MediaType.APPLICATION_JSON));
52      System.out.println(responseCreate.getStatus());
53      System.out.println(responseCreate.readEntity(String.class));
```

```
Console  Markers  Servers  Data Source Explorer  Snippets  Problems  Search  JUnit
<terminated> RestClient [Java Application] F:\informatique\JDK\jdk-64\bin\javaw.exe (2018年5月20日 下午8:27:58)
200
["userName is: Li","password is: Yuanyuan","privacy or not: true"]
```

Aussi, on peut vérifier dans notre base de données, l’utilisateur ‘Li’ est créé.

3.2. Changer le nom d’utilisateur

Teste 2: update - On peut changer le nom. Pour cela, il faut attacher HEADER de ‘Basic Authentication’ pour ‘logIn Instagram’.

Par exemple, si on n'attache pas de HEADER ou le mot de passe n’est pas correcte, on vais recevoir le message ci-dessous:

```
25        HttpAuthenticationFeature feature = HttpAuthenticationFeature.basic("Li", "Yuanyuan");
26        client.register(feature);
```

Console Markers Servers Data Source Explorer Snippets Problems Search JUnit

```
<terminated> RestClient [Java Application] F:\informatique\JDK\jdk-64\bin\javaw.exe (2018年5月20日 下午8:40:58)
401
User cannot access the resource, please logIn by using HEADERS: Basic Authentication.
```

Si le mot de passe est correcte, on peut recevoir le message comme:

```
55        /**
56         * Test: webService change the name of user: @PUT. user can only change his own
57         * account.
58         */
59        WebTarget userWebTarChange = userWebTarget.path("userName");
60        WebTarget userWebTargetChangeName = userWebTarChange.queryParam("newUserName", "LIYuanyuan");
61        Invocation.Builder invocationBuilderUpdateUser = userWebTargetChangeName.request(MediaType.TEXT_PLAIN);
62        invocationBuilderUpdateUser.header("some-header", "true");
63        Response responseUpdate = invocationBuilderUpdateUser.put(Entity.entity("create", MediaType.APPLICATION_JSON));
64        System.out.println(responseUpdate.getStatus());
65        System.out.println(responseUpdate.readEntity(String.class));
```

Console Markers Servers Data Source Explorer Snippets Problems Search JUnit

```
<terminated> RestClient [Java Application] F:\informatique\JDK\jdk-64\bin\javaw.exe (2018年5月20日 下午8:44:58)
200
New name: LIYuanyuan
```

3.3. Rechercher tous les publications d'utilisateur

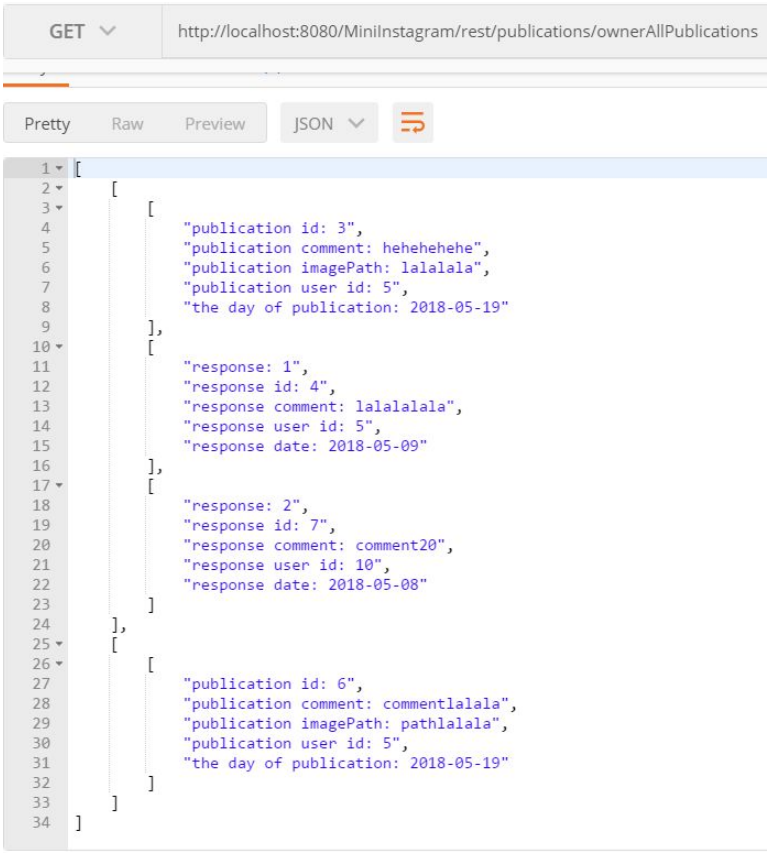
On peut aussi obtenir tous les publications de soi-même.

```
73        /** Test: webService get all the publication of user's own: @GET */
74        WebTarget publicationWebTarget = webTarget.path("publications");
75        WebTarget publicationWebTargetAll = publicationWebTarget.path("ownerAllPublications");
76        Invocation.Builder invocationBuilderPublication = publicationWebTargetAll.request(MediaType.APPLICATION_JSON);
77        invocationBuilderPublication.header("some-header", "true");
78        Response responsePublication = invocationBuilderPublication.get();
79        System.out.println(responsePublication.getStatus());
80        System.out.println(responsePublication.readEntity(String.class));
```

Console Markers Servers Data Source Explorer Snippets Problems Search JUnit

```
<terminated> RestClient [Java Application] F:\informatique\JDK\jdk-64\bin\javaw.exe (2018年5月20日 下午8:57:50)
200
[[["publication id: 3","publication comment: hehehehehe","publication imagePath: lalalala","publication user id: 5","the day of
```

Aussi on peut vérifier vers utiliser '**Postman**'. par exemple comme ci-dessous:



Si on veut regarder les publications d'autre utilisateur, si on ne le 'follow' pas, on ne peut pas regarder, par exemple: on veut regarder l'utilisateur 2:

on reçoit rien, mais il y a une publication d'utilisateur 2:

Donc, pour regarder, on 'follow' utilisateur 2 , et après, on peut regarder.

3.3. Publier une publication

On aussi tester de publier une publication:

```
97        /** Test: webService create a user: @POST */
98        WebTarget publicationWebTarget = webTarget.path("publications");
99        WebTarget userWebTargetImage = publicationWebTarget.queryParam("imagePath", "pathLi");
100       WebTarget userWebTargetComment = userWebTargetImage.queryParam("comment", "comment Yuanyuan");
101       Invocation.Builder invocationBuilderPublier= userWebTargetComment.request(MediaType.APPLICATION_JSON);
102       invocationBuilderPublier.header("some-header", "true");
103       Response responsePost = invocationBuilderPublier.post(Entity.entity("pust publication", MediaType.APPLICATION_JSON
104       System.out.println(responsePost.getStatus());
105       System.out.println(responsePost.readEntity(String.class));
106
```

Console ⊠ Markers Servers Data Source Explorer Snippets Problems Search JUnit

```
<terminated> RestClient [Java Application] F:\informatique\JDK\jdk-64\bin\javaw.exe (2018年5月20日 下午9:05:17)
200
[["publication id: 12","publication comment: comment Yuanyuan","publication imagePath: pathLi","publication user id: 5","the day c
```

Vérifier dans la base de données:

☐	Edit Copy Delete	11	super hard	2018-05-20 07:10:10	Highlight	15
☐	Edit Copy Delete	12	comment Yuanyuan	2018-05-20 21:05:21	pathLi	5

3.4. Répondre une publication

Si on veut répondre à publication ‘9’, cette publication est crée par utilisateur ‘14’, on ne ‘follow’ pas cet utilisateur, donc, si on veut répondre, on vais recevoir le message comme ci-dessous:

```
107       /**
108        * Test: webService response a publication: @POST. Description: if you do not
109        * follow other user,and other user is private, you can not response the
110        * publications of other user.
111        */
112       WebTarget responseWebTarget = webTarget.path("responses");
113       WebTarget responseWebTargetComment = responseWebTarget.queryParam("comment", "comment response");
114       WebTarget responseWebTargetPId = responseWebTargetComment.queryParam("pId", 9);
115       Invocation.Builder invocationBuilderResponse = responseWebTargetPId.request(MediaType.APPLICATION_JSON);
116       invocationBuilderResponse.header("some-header", "true");
117       Response responseResponse = invocationBuilderResponse
118               .post(Entity.entity("response a publication", MediaType.APPLICATION_JSON));
119       System.out.println(responseResponse.getStatus());
120       System.out.println(responseResponse.readEntity(String.class));
121   }
```

Console ⊠ Markers Servers Data Source Explorer Snippets Problems Search JUnit

```
<terminated> RestClient [Java Application] F:\informatique\JDK\jdk-64\bin\javaw.exe (2018年5月20日 下午9:29:54)
200
["You do not have the permission to response this publication."]
```

Donc, on ‘follow’ utilisateur 9 pour répondre, et après, on peut répondre publication 9:

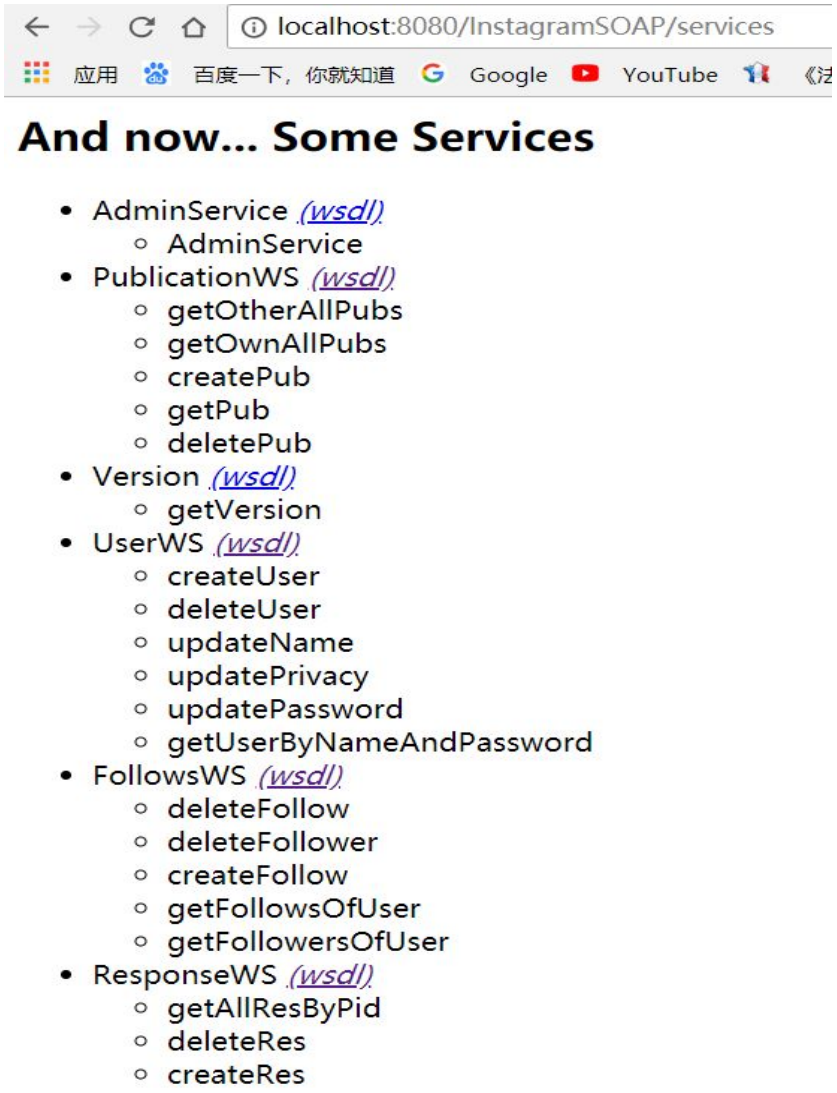
```
107       /**
108        * Test: webService response a publication: @POST. Description: if you do not
109        * follow other user,and other user is private, you can not response the
110        * publications of other user.
111        */
112       WebTarget responseWebTarget = webTarget.path("responses");
113       WebTarget responseWebTargetComment = responseWebTarget.queryParam("comment", "comment response");
114       WebTarget responseWebTargetPId = responseWebTargetComment.queryParam("pId", 9);
115       Invocation.Builder invocationBuilderResponse = responseWebTargetPId.request(MediaType.APPLICATION_JSON);
116       invocationBuilderResponse.header("some-header", "true");
117       Response responseResponse = invocationBuilderResponse
118               .post(Entity.entity("response a publication", MediaType.APPLICATION_JSON));
119       System.out.println(responseResponse.getStatus());
120       System.out.println(responseResponse.readEntity(String.class));
121   }
```

Console ⊠ Markers Servers Data Source Explorer Snippets Problems Search JUnit

```
<terminated> RestClient [Java Application] F:\informatique\JDK\jdk-64\bin\javaw.exe (2018年5月20日 下午9:34:27)
200
["response Id: 12","response comment: comment response","response user Id: 5","the date of response: 2018-05-20","publication Id: 9",
```

4. Les spécifications de SOAP

Pour web service SOAP, les services comme ci-dessous.



Aussi, du côté client, on teste juste quelques services pour présenter.

4.1. Publier une publication

publier:

```
11        /** test create a publication */
12        try {
13            PublicationWSProxy publicationWSProxy = new PublicationWSProxy();
14            String publication = publicationWSProxy.createPub("Li", "Yuanyuan", "image path", "instagram");
15            System.out.println(publication);
16        } catch (RemoteException e) {
17            // TODO Auto-generated catch block
18            e.printStackTrace();
19        }
```

```
Console  Markers  Servers  Data Source Explorer  Snippets  Problems  Search  JUnit
<terminated> Main (23) [Java Application] F:\informatique\JDK\jdk-64\bin\javaw.exe (2018年5月20日 下午9:53:13)
五月 20, 2018 9:53:13 下午 org.apache.axis.utils.JavaUtils isAttachmentSupported
警告: Unable to find required classes (javax.activation.DataHandler and javax.mail.internet.MimeMultipart). Attach
publication id: 14, publication comment: instagram, publication imagePath: image path, publication user id: 15, t
```

Si le mot de passe n'est pas correcte, on reçoit null, c'est à dire, cette publication n'est pas créé.:

```
11        /** test create a publication */
12        try {
13            PublicationWSProxy publicationWSProxy = new PublicationWSProxy();
14            String publication = publicationWSProxy.createPub("Li", "Yuanyuan", "image path", "instagram");
15            System.out.println(publication);
16        } catch (RemoteException e) {
17            // TODO Auto-generated catch block
18            e.printStackTrace();
19        }
```

```
Console  Markers  Servers  Data Source Explorer  Snippets  Problems  Search  JUnit
<terminated> Main (23) [Java Application] F:\informatique\JDK\jdk-64\bin\javaw.exe (2018年5月20日 下午9:54:43)
五月 20, 2018 9:54:43 下午 org.apache.axis.utils.JavaUtils isAttachmentSupported
警告: Unable to find required classes (javax.activation.DataHandler and javax.mail.internet.MimeMultipart). Attac
null
```

4.2. Visualiser les publication d'autre personne

Si d'autre utilisateur n'est pas privé, on peut visualiser tous les publication:

```
33        /**
34         * test look the publication of other user. if you do not follow other user, and
35         * other user is private, you can not see the publications of other user.
36         */
37        try {
38            PublicationWSProxy publicationWSProxy = new PublicationWSProxy();
39            String[] publications = publicationWSProxy.getOtherAllPubs("Li", "Yuanyuan", 13);
40            if(publications!=null)
41            {
42                for (String publication : publications) {
43                    System.out.println(publication);
44                }
45            }
46            else {
47                System.out.println("no publications.");
48            }
49        } catch (RemoteException e) {
50            // TODO Auto-generated catch block
```

```
Console  Markers  Servers  Data Source Explorer  Snippets  Problems  Search  JUnit
<terminated> Main (23) [Java Application] F:\informatique\JDK\jdk-64\bin\javaw.exe (2018年5月20日 下午10:17:56)
五月 20, 2018 10:18:00 下午 org.apache.axis.utils.JavaUtils isAttachmentSupported
警告: Unable to find required classes (javax.activation.DataHandler and javax.mail.internet.MimeMultipart). Attachment support is disabled.
publication id: 4, publication comment: hahaha, publication imagePath: hehehehe, publication user id: 13, the day of publication: 2018-05-19
publication id: 5, publication comment: hahaha, publication imagePath: hehehehe, publication user id: 13, the day of publication: 2018-05-19
```

Si on chage l'utilisateur 13 est prive, et on ne 'follow' pas d'utilisateur 13, apres:

```
33        /**
34         * test look the publication of other user. if you do not follow other user, and
35         * other user is private, you can not see the publications of other user.
36         */
37        try {
38            PublicationWSProxy publicationWSProxy = new PublicationWSProxy();
39            String[] publications = publicationWSProxy.getOtherAllPubs("Li", "Yuanyuan", 13);
40            if(publications!=null)
41            {
42                for (String publication : publications) {
43                    System.out.println(publication);
44                }
45            }
46            else {
47                System.out.println("no publications.");
48            }
49        } catch (RemoteException e) {
50            // TODO Auto-generated catch block
51            e.printStackTrace();
52        }
```

```
Console  Markers  Servers  Data Source Explorer  Snippets  Problems  Search  JUnit
<terminated> Main (23) [Java Application] F:\informatique\JDK\jdk-64\bin\javaw.exe (2018年5月20日 下午10:20:05)
五月 20, 2018 10:20:06 下午 org.apache.axis.utils.JavaUtils isAttachmentSupported
警告: Unable to find required classes (javax.activation.DataHandler and javax.mail.internet.MimeMultipart
no publications.
```

Après, on 'follow' utilisateur 13, on peut visualiser tous les publications d'utilisateur 13.

4.3. Répondre a une publication

Si on veut répondre à publication 5, cette publication est créé par utilisateur 13, utilisateur 13 est privé, on ne 'follow' pas d'utilisateur 13, donc on ne peut pas répondre:

```
54        /**
55         * test response a publication. if you do not follow other user, and other user
56         * is private, you can not response the publications of other user.
57         */
58        try {
59            ResponseWSProxy responseWSProxy = new ResponseWSProxy();
60            String[] responses = responseWSProxy.createRes("Li", "Yuanyuan", "comment 20180520", 5);
61            for (String response : responses) {
62                System.out.println(response);
63            }
64        } catch (RemoteException e) {
65            // TODO Auto-generated catch block
66            e.printStackTrace();
67            }
```

```
Console  Markers  Servers  Data Source Explorer  Snippets  Problems  Search  JUnit
<terminated> Main (23) [Java Application] F:\informatique\JDK\jdk-64\bin\javaw.exe (2018年5月20日 下午10:22:26)
五月 20, 2018 10:22:28 下午 org.apache.axis.utils.JavaUtils isAttachmentSupported
警告: Unable to find required classes (javax.activation.DataHandler and javax.mail.internet.MimeMultipart)
You do not have the permission to response this publication.
```

Donc, on 'follow' utilisateur 13, après, on peut répondre:

```
54        /**
55         * test response a publication. if you do not follow other user, and other user
56         * is private, you can not response the publications of other user.
57         */
58        try {
59            ResponseWSProxy responseWSProxy = new ResponseWSProxy();
60            String[] responses = responseWSProxy.createRes("daqing", "daqing", "comment 20180520", 5);
61            for (String response : responses) {
62                System.out.println(response);
63            }
64        } catch (RemoteException e) {
65            // TODO Auto-generated catch block
66            e.printStackTrace();
67            }
68    }
```

```
Console  Markers  Servers  Data Source Explorer  Snippets  Problems  Search  JUnit
<terminated> Main (23) [Java Application] F:\informatique\JDK\jdk-64\bin\javaw.exe (2018年5月20日 下午10:41:06)
五月 20, 2018 10:41:08 下午 org.apache.axis.utils.JavaUtils isAttachmentSupported
警告: Unable to find required classes (javax.activation.DataHandler and javax.mail.internet.MimeMultipart). A
response Id: 13
response comment: comment 20180520
response user Id: 14
the date of response: 2018-05-20
publication Id: 5
publication comment: hahaha
publication user Id: 13
```

5. Sécurité

5.1. REST

Utilisateur: On utilise l'interface '**ContainerRequestFilter**' pour vérifier si l'utilisateur est logIn. Donc, chaque fois le client doit attacher HEADER 'Basic Authentication' avec le nom et les mots de passe pour accéder les données. Si le client n'attache pas de HEADER ou attache avec les informations est ne sont pas correctes, notre service intercepte automatiquement.

5.2. SOAP

Utilisateur: sur chaque service de notre projet SOAP, le client doit utiliser le nom et les mots de passe pour accéder les données (dans chaque fonction, il y a deux variables : ‘userName’ et ‘password’).

6. Conclusion

Notre web service offre les services comme ci-dessous:

1. Vérification de login.
2. Gestion d’utilisateur:
   (1). Création.
   (2). Mise à jour des information: nom, mot de passe, privé ou non.
   (3). Supprission.
3. Gestion de relation entre les utilisateurs (suivre / follow):
   (1). Suivre d’autre utilisateur.
   (2). Ne pas suivre d’autre utilisateur.
   (3). Supprission le lien avec d’autre utilisateur.
4. Gestion de publication:
   (1). Création les publications. (publier)
   (2). Modification: le chemin d’image.
   (3). Supprission.
   (4). Les visualisation des publications (la possibilité de la visualisation est comme tableau 3-1).
5. Gestion de réponse:
   (1). Réponse de les publications. (répondre: la possibilité de répondre à une publication est comme tableau 3-1).
   (4). Supprission.
   (4). Les visualisation des responses (la possibilité de la visualisation est comme tableau 3-1).