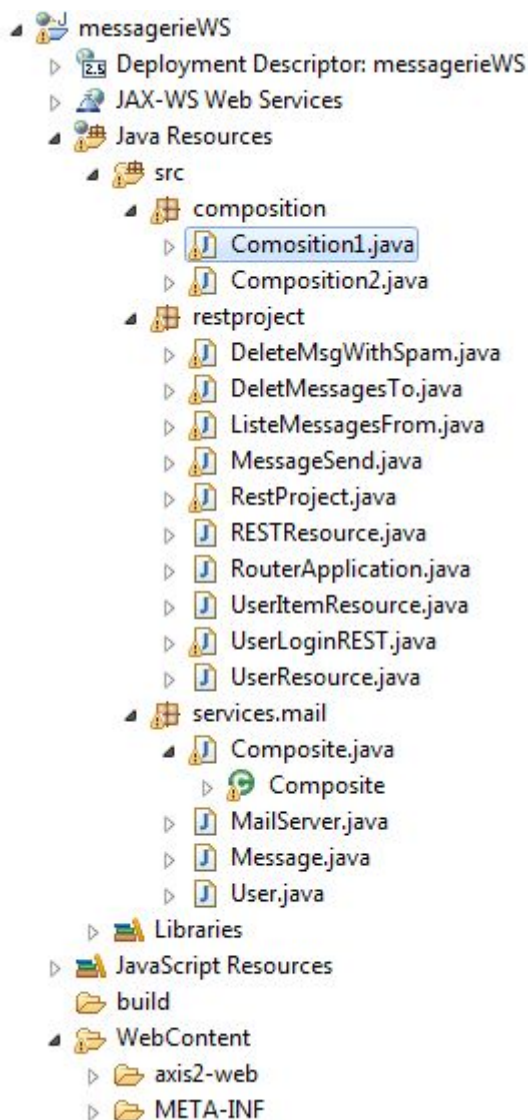
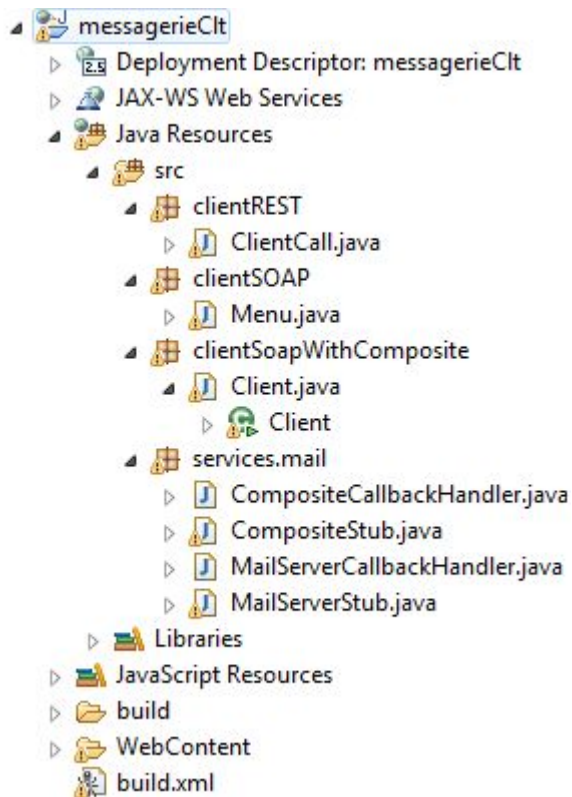


# Projet SOA - Messagerie

## Structure générale de notre projet:





## Service commun

### (MailServer):

Ce service (**messagerieWS/src/service/mail/MailServer.java**) se présente sous une classe java contenant les opérations nécessaires pour les menus SOAP et REST.

Cette classe contient les méthodes:

- `public String sendMessage (String fromp, String top, String messagep) //envoie un message avec l'heure courante, pour qu'on puisse utiliser les compositions.`
- `public List<Message> getMessages (String to)`
- `public String removeMessages (String to)`
- `public String authentifier (String nom, String email, String psw) // pour l'authentification qui sera utilisée dans les Menus pour accéder au boîtes aux lettres.`

et les attributs:

- `public static List <Message> all = new ArrayList<Message>(); //liste contenant tous les messages.`

- public static List<User> users = new ArrayList<User>(); //liste des utilisateurs.

Le fichier WSDL correspondant à cette service(MailServer):

```
http://localhost:8080/messagerieWS/services/Composite?wsdl
<?xml version="1.0" encoding="UTF-8"?>
- <wsdl:definitions targetNamespace="http://mail.services" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:ns1="http://org.apache.axis2/xsd" xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
  xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
  xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/" xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"
  xmlns:ns="http://mail.services" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:documentation> Please Type your service description here </wsdl:documentation>
  - <wsdl:types>
    - <xs:schema targetNamespace="http://mail.services" elementFormDefault="qualified"
      attributeFormDefault="qualified">
      - <xs:element name="ChangePerior">
        - <xs:complexType>
          - <xs:sequence>
            <xs:element name="nom" type="xs:string" nillable="true" minOccurs="0"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      - <xs:element name="ChangePeriorResponse">
        - <xs:complexType>
          - <xs:sequence>
            <xs:element name="return" type="xs:string" nillable="true" minOccurs="0"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      - <xs:element name="accuseBienRecu">
        - <xs:complexType>
          - <xs:sequence>
            <xs:element name="exp" type="xs:string" nillable="true" minOccurs="0"/>
            <xs:element name="nom" type="xs:string" nillable="true" minOccurs="0"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      - <xs:element name="accuseBienRecuResponse">
        - <xs:complexType>
```

## Clients:

Les clients (menus) utilisent le service (MailServer) pour effectuer les 6 tâches décrits dans l'énoncé. Nous avons développé un client SOAP et un client REST.

### Client SOAP (Menu):

Ce client ( **messagerieClt/src/clientSOAP/Menu.java**) utilise MailServerStub.java qui est générée automatiquement avec Axis2 - 1.5.1 pour accéder aux méthodes du services expliqués précédemment. Le menu est implémenté par un bloc switch qui pour chaque choix de l'utilisateur, effectue le travail correspondant.

SOAP: instancie la classe stub, appelle la méthode voulue, instancie la classe de la méthode voulue, affecte les variables via les setters et récupère le résultat par la méthode get\_result().

### Client REST (Menu):(messagerieClt/src/clientREST/ClientCall.java)

La même travail.

## REST:

**Serveur:** Créer une classe RouterApplication.java et attacher une url aux classes qui font les fonctions métier, implémenter ces classes en se référant aux méthodes créées dans le MailServer.

Types des méthodes utilisés:

- Méthodes @get: les paramètres sont passés dans l'url.
- Méthodes @post et @delete: les paramètres sont passés dans un Form qu'on crée dans le client.

**Client:** Nous avons implémenté la classe ClientCall.java qui contient l'url de la méthode (classe) implémentée dans le serveur, la Forms qui contient la valeur avec le key, qui va récupérer par la classe attaché par son url dans le cas des méthodes @post et le Ressource qui va invoquer ces méthodes qu'on récupère leurs résultats avec write.

## Compositions:

Les compositions sont des classes qui effectuent un travail plus complexe qui utilise des méthodes du service principal.

### Première composition: (MessagerieWS/src/composition/composition1.java)

Supprime les SPAM en utilisant getMessage() et removeMessages() du service MailServer.java

```
package composition;

import java.rmi.RemoteException;

public class Comosition1 {

    public void spam (String top) throws RemoteException {
        MailServer ms = new MailServer();
        String msg;
        /*int i=0;
        while(i<ms.all.size()){*/
        for (int i=0;i<ms.all.size();i++){
            if (ms.all.get(i).getTo().equals(top)){
                String fromp=ms.all.get(i).getFrom();
                for (int j=0;j<ms.users.size();j++){
                    if ((ms.users.get(j).getNom().equals(fromp))&&(!"Admin".equals(fromp))){
                        //u.setPassword(mail.users.get(j).getPassword());
                        String spam=ms.users.get(j).getPassword();
                        msg=ms.all.get(i).getMessage();
                        if (msg.contains(spam)){
                            ms.all.get(i).setIsNew(false);
                            //ms.all.remove(ms.all.get(i));
                            String res=ms.removeMessages(top);
                            i--;
                        }
                    }
                }
            }
        }
    }
}
```

## Deuxième composition:(MessagerieWS/src/composition/composition2.java)

Envoie un accusé de réception en utilisant sendMessage() et removeMessages() du service MailServer.java

```
package composition;

import java.rmi.RemoteException;

public class Composition2 {

    public void accuse (String exp, String nom) throws RemoteException {

        MailServer ms = new MailServer();

        for (int i=0;i<ms.all.size();i++) {
            if(ms.all.get(i).getFrom().equals(exp)) {
                ms.sendMessage(nom,exp,"Bien reçu !");
            }
            else {
                ms.all.get(i).setIsNew(false);
                String to=ms.all.get(i).getTo();
                String res=ms.removeMessages(to);//on p pas suppr le msg avant etre lire
                i--;
            }
        }

    }

}
```

## Composite:(MessagerieWS/src/service/mai/composite.java)

Le composite est un service (classe JAVA) qui utilise les 2 compositions implémentés précédemment pour aboutir à des fonctionnalités plus complexes.

Cette classe contient les méthodes:

- changerPriorité
- removeSPAM
- accuséReception



```
public class Composite {  
    Composition1 c1 = new Composition1();  
    Composition2 c2 = new Composition2();  
    public String ChangePerior(String nom){  
        MailServer ms=new MailServer();  
        System.out.println("**** mettre priorité haute entre 20h et 24h");  
        for (int i=0;i<ms.all.size();i++) {  
            if (ms.all.get(i).getTo().equals(nom)){  
                if(ms.all.get(i).getSendTime().startsWith("20") || ms.all.get(i).getSendTime().startsWith("21") || ms.all.get(i).getSe  
                    ms.all.get(i).setPriorite("high");  
            }  
        }  
        return "la modification de la periorite est succes";  
    }  
    public String removeSpam(String top){  
        try {  
            c1.spam(top);  
        } catch (RemoteException e) {  
            // TODO Auto-generated catch block  
            e.printStackTrace();  
        }  
        return "Spam delete succesfully";  
    }  
    public String accuseBienRecu(String exp,String nom){  
        try {  
            c2.accuse(exp, nom);  
        } catch (RemoteException e) {  
            // TODO Auto-generated catch block  
            e.printStackTrace();  
        }  
        return "msg Bien Envoyer";  
    }  
}
```

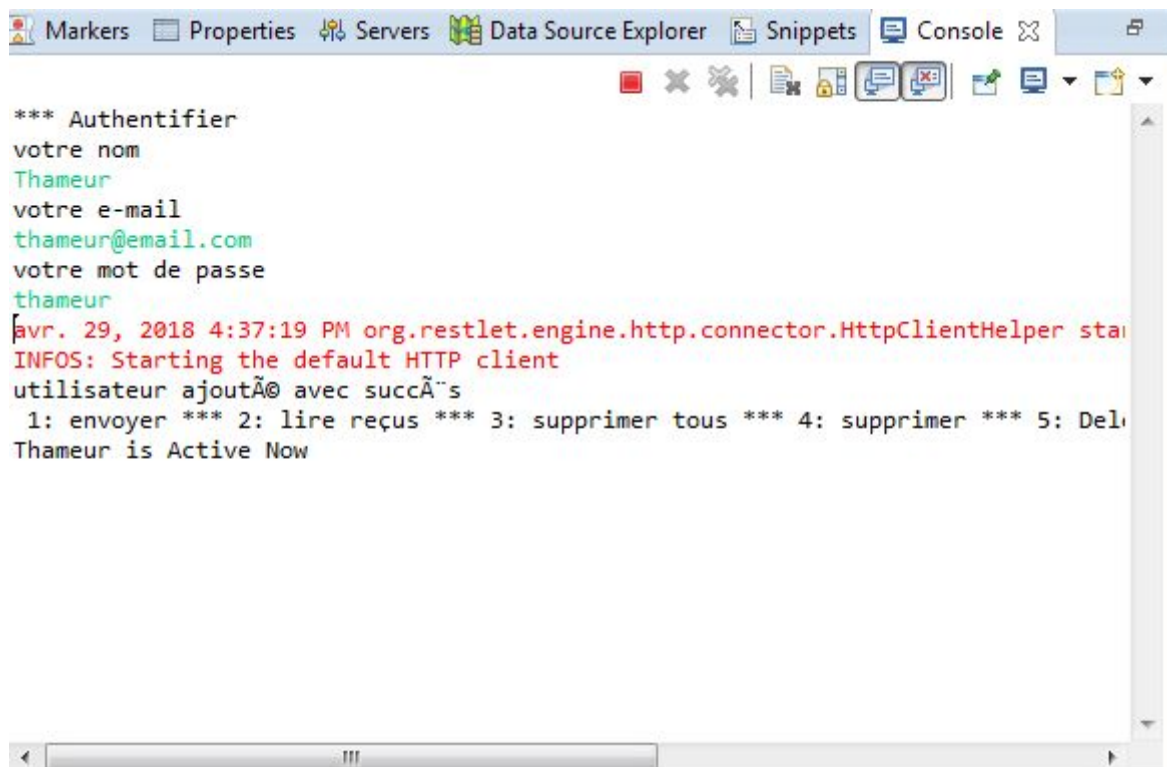
Le fichier WSDL correspondant:

```
http://localhost:8080/messagerieWS/services/Composite?wsdl  
  
<?xml version="1.0" encoding="UTF-8"?>  
- <wsl:definitions targetNamespace="http://mail.services" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"  
  xmlns:ns1="http://org.apache.axis2/xsd" xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"  
  xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"  
  xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/" xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"  
  xmlns:ns="http://mail.services" xmlns:wsl="http://schemas.xmlsoap.org/wsdl/">  
  <wsl:documentation> Please Type your service description here </wsl:documentation>  
  - <wsl:types>  
    - <xs:schema targetNamespace="http://mail.services" elementFormDefault="qualified"  
      attributeFormDefault="qualified">  
      - <xs:element name="ChangePerior">  
        - <xs:complexType>  
          - <xs:sequence>  
            <xs:element name="nom" type="xs:string" nillable="true" minOccurs="0"/>  
          </xs:sequence>  
        </xs:complexType>  
      </xs:element>  
      - <xs:element name="ChangePeriorResponse">  
        - <xs:complexType>  
          - <xs:sequence>  
            <xs:element name="return" type="xs:string" nillable="true" minOccurs="0"/>  
          </xs:sequence>  
        </xs:complexType>  
      </xs:element>  
      - <xs:element name="accuseBienRecu">  
        - <xs:complexType>  
          - <xs:sequence>  
            <xs:element name="exp" type="xs:string" nillable="true" minOccurs="0"/>  
            <xs:element name="nom" type="xs:string" nillable="true" minOccurs="0"/>  
          </xs:sequence>  
        </xs:complexType>  
      </xs:element>  
      - <xs:element name="accuseBienRecuResponse">  
        - <xs:complexType>
```

## Exemples d'exécution:

### REST:

#### Authentication:



```
*** Authentifier
votre nom
Thameur
votre e-mail
thameur@email.com
votre mot de passe
thameur
[avr. 29, 2018 4:37:19 PM org.restlet.engine.http.connector.HttpClientHelper start]
INFOS: Starting the default HTTP client
utilisateur ajoutÃ© avec succÃ¨s
1: envoyer *** 2: lire reÃ§us *** 3: supprimer tous *** 4: supprimer *** 5: Del
Thameur is Active Now
```

### Envoyer un message :

```
1: envoyer *** 2: lire reçus *** 3: supprimer tous *** 4: supprimer *** 5: Delete your msg with spam *** 6: Log out **** 7: QuitterProg
Thameur is Active Now
1
*** send Message
to
Jasser
ecrit un message
Hi Jasser
avr. 29, 2018 4:41:06 PM org.restlet.engine.http.connector.HttpClientHelper start
INFOS: Starting the default HTTP client
Message sent successfully
1: envoyer *** 2: lire reçus *** 3: supprimer tous *** 4: supprimer *** 5: Delete your msg with spam *** 6: Log out **** 7: QuitterProg
Thameur is Active Now
```

### Authentification par un autre utilisateur:

```
1: envoyer *** 2: lire reçus *** 3: supprimer tous *** 4: supprimer *** 5: Delete your msg with spam *** 6: Log out **** 7: QuitterProg
Thameur is Active Now
6
*** Authentifier
votre nom
Jasser
votre e-mail
jasser@email.com
votre mot de passe
jasser
avr. 29, 2018 4:45:58 PM org.restlet.engine.http.connector.HttpClientHelper start
INFOS: Starting the default HTTP client
utilisateur ajoutÃ© avec succÃ©s
1: envoyer *** 2: lire reçus *** 3: supprimer tous *** 4: supprimer *** 5: Delete your msg with spam *** 6: Log out **** 7: QuitterProg
Jasser is Active Now
```



### Lire les messages reçus:

```
1: envoyer *** 2: lire reçus *** 3: supprimer tous *** 4: supprimer *** 5: Delete your msg with spam *** 6: Log out ***** 7:QuitterProg
Jasser is Active Now
2
*** get messages
avr. 29, 2018 4:48:44 PM org.restlet.engine.http.connector.HttpClientHelper start
INFOS: Starting the default HTTP client
avr. 29, 2018 4:48:44 PM org.restlet.engine.http.connector.HttpClientHelper stop
INFOS: Stopping the default HTTP client

Liste des messages de Jasser
From ***** Thameur*****
Hi Jasser
From ***** Admin*****
Bienvenue Ã notre service :)

1: envoyer *** 2: lire reçus *** 3: supprimer tous *** 4: supprimer *** 5: Delete your msg with spam *** 6: Log out ***** 7:QuitterProg
Jasser is Active Now
```

### Vider la boîte aux lettres:

```
1: envoyer *** 2: lire reçus *** 3: supprimer tous *** 4: supprimer *** 5: Delete your msg with spam *** 6: Log out ***** 7:QuitterProg
Jasser is Active Now
avr. 29, 2018 4:48:45 PM org.restlet.engine.http.connector.HttpClientHelper stop
INFOS: Stopping the default HTTP client
avr. 29, 2018 4:48:45 PM org.restlet.engine.http.connector.HttpClientHelper stop
INFOS: Stopping the default HTTP client
3
avr. 29, 2018 4:49:45 PM org.restlet.engine.http.connector.HttpClientHelper start
INFOS: Starting the default HTTP client
old messages removed successfully
1: envoyer *** 2: lire reçus *** 3: supprimer tous *** 4: supprimer *** 5: Delete your msg with spam *** 6: Log out ***** 7:QuitterProg
Jasser is Active Now
2
avr. 29, 2018 4:49:57 PM org.restlet.engine.http.connector.HttpClientHelper start
INFOS: Starting the default HTTP client
*** get messages

Liste des messages de Jasser

1: envoyer *** 2: lire reçus *** 3: supprimer tous *** 4: supprimer *** 5: Delete your msg with spam *** 6: Log out ***** 7:QuitterProg
Jasser is Active Now
```

### Réauthentification:

```
6
*** Authentifier
votre nom
Thameur
votre e-mail
thameur@email.com
votre mot de passe
thameur
avr. 29, 2018 9:12:00 PM org.restlet.engine.http.connector.HttpClientHelper start
INFOS: Starting the default HTTP client
Bienvenue Thameur
1: envoyer *** 2: lire reçus *** 3: supprimer tous *** 4: supprimer *** 5: Delete your msg with spam *** 6: Log out ***** 7:QuitterProg
Thameur is Active Now
```

## Envoyer deux messages, l'une qu'il contient un spam:

```
1: envoyer *** 2: lire reçus *** 3: supprimer tous *** 4: supprimer *** 5: Delete your msg with spam *** 6: Log out **** 7:QuitterProg
Thameur is Active Now
1
*** send Message
to
Dali
ecrit un message
Hi Dali
avr. 29, 2018 9:13:55 PM org.restlet.engine.http.connector.HttpClientHelper start
INFOS: Starting the default HTTP client
Message sent successfully
1: envoyer *** 2: lire reçus *** 3: supprimer tous *** 4: supprimer *** 5: Delete your msg with spam *** 6: Log out **** 7:QuitterProg
Thameur is Active Now
1
*** send Message
to
Dali
ecrit un message
Hi i am thameur
avr. 29, 2018 9:14:14 PM org.restlet.engine.http.connector.HttpClientHelper start
INFOS: Starting the default HTTP client
Message sent successfully
1: envoyer *** 2: lire reçus *** 3: supprimer tous *** 4: supprimer *** 5: Delete your msg with spam *** 6: Log out **** 7:QuitterProg
Thameur is Active Now
```

## Supprimer les spam:

```
6
*** Authentifier
votre nom
Dali
votre e-mail
dali@email.com
votre mot de passe
dali
avr. 29, 2018 9:15:19 PM org.restlet.engine.http.connector.HttpClientHelper start
INFOS: Starting the default HTTP client
utilisateur ajoutÃ© avec succÃ©s
1: envoyer *** 2: lire reçus *** 3: supprimer tous *** 4: supprimer *** 5: Delete your msg with spam *** 6: Log out **** 7:QuitterProg
Dali is Active Now
5
avr. 29, 2018 9:15:34 PM org.restlet.engine.http.connector.HttpClientHelper start
INFOS: Starting the default HTTP client
Spam delete succesfully
1: envoyer *** 2: lire reçus *** 3: supprimer tous *** 4: supprimer *** 5: Delete your msg with spam *** 6: Log out **** 7:QuitterProg
Dali is Active Now
2
f*** get messages
avr. 29, 2018 9:15:38 PM org.restlet.engine.http.connector.HttpClientHelper start
INFOS: Starting the default HTTP client
avr. 29, 2018 9:15:38 PM org.restlet.engine.http.connector.HttpClientHelper stop
INFOS: Stopping the default HTTP client

Liste des messages de Dali
From ***** Thameur*****
Hi Dali
From ***** Admin*****
Bienvenue Ã notre service :)

1: envoyer *** 2: lire reçus *** 3: supprimer tous *** 4: supprimer *** 5: Delete your msg with spam *** 6: Log out **** 7:QuitterProg
Dali is Active Now
```

# SOAP:

## Exemple d'exécutions:

De même, mais on exécute le fichier (MessagerieClt/src/ClientSoapWithComposite/client.java) les deux méthode: changerPriorite( ) et accuse( ):

### Authentification:

```
*** Authentifier
votre nom
thameur
votre e-mail
thameur@email.com
votre mot de passe
thameur
Bienvenue thameur
1: envoyer *** 2: lire reçus *** 3: supprimer tous *** 4: supprimer ***5: Log out ***6: Fin Prog
thameur is Active Now
```

---

### Envoyer un message :

```
1: envoyer *** 2: lire reçus *** 3: supprimer tous *** 4: supprimer ***5: Log out ***6: Fin Prog
thameur is Active Now
1
*** Send message
destinataire:
Dali
message:
Hi Dali
Message sent successfully
1: envoyer *** 2: lire reçus *** 3: supprimer tous *** 4: supprimer ***5: Log out ***6: Fin Prog
thameur is Active Now
```

### Accuse et mettre à jour la priorité :

