

2019

# Info 0503 : Modélisation client-serveur et programmation Web avancé

Halal Rossie Juliete  
Loiseau Pintaux Antinéa

Intervenant :  
Rabat Cyril



## I- Introduction :

Nous souhaitons réaliser une application de sondages. Cette application permet à un utilisateur de créer un sondage avec des questions qui peuvent avoir plusieurs choix, ou de répondre à un sondage créé par un autre utilisateur. Pour pouvoir faire cela, les utilisateurs doivent d'abord se connecter, s'il n'a pas de compte, il doit s'en créer un. L'utilisateur qui a créé un sondage peut consulter les réponses des autres utilisateurs.

## II- Travail effectué :

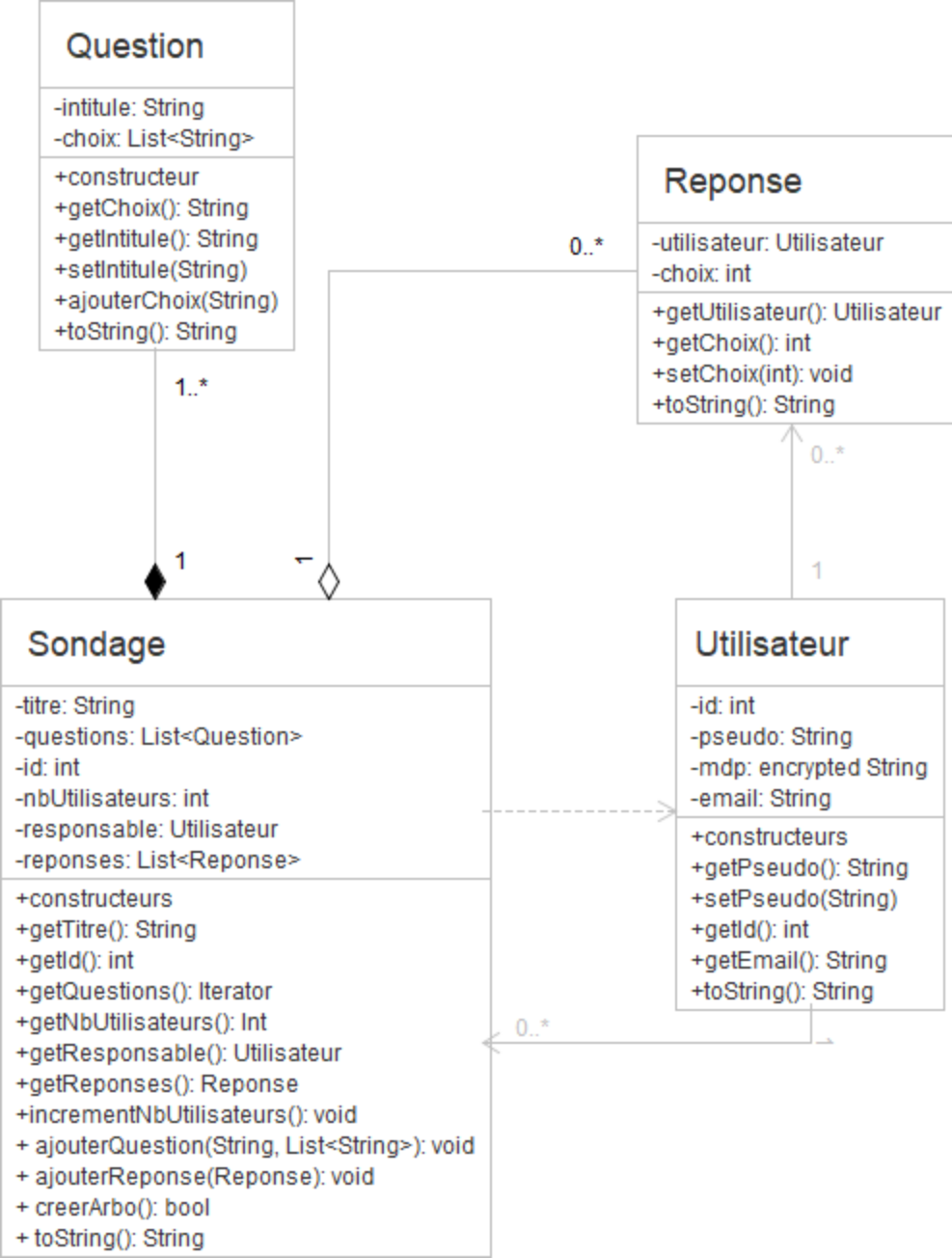
Tout d'abord on a réalisé notre application de sondage en php, et on a vérifié le bon fonctionnement de toutes nos méthodes (les tests existent encode dans le fichier sondage/index.php). Par la suite on a refait les même classes mais en java, et on a testé le bon fonctionnement de ces classes (dans sondage/java/Test.java).

La partie suivante était la réalisation d'un serveur d'authentification en java. Pour cela on a les classes ServeurAuthentification qui permet de créer les contextes dont on a besoin et de démarrer le serveur. Ce serveur utilise 4 handler, LoginHandler et CreateHandler qui permettent de, respectivement, se connecter et créer un compte. La seule utilité des 2 autre handlers, IndexHandler et CreateFormHandler, est l'affichage des formulaires, ils ne seront pas utiles pour notre application finale. Le ServeurAuthentification a une instance du GestionnaireUtilisateur ce qui nous permet d'avoir une HashMap de tout les utilisateur et permettre la gestion des connexions et créations de compte.

Pour la partie serveur de sondage, les seuls fichier à réaliser étaient les fichier php car le serveur c'est un serveur uwamp. Les deux fichiers connexion.php et creerCompte.php utilisent le serveur d'authentification pour connecter les utilisateurs. Le fichier lister.php permet de lister tous les sondages. mesSondages.php permet d'afficher tous les sondages créés par l'utilisateur connecté. recupSondage.php permet de récupérer les réponses d'un sondage et repondreSondage.php permet de répondre à un sondage. Finalement on a écrit la classe Client.java qui est connecté seulement au serveur de sondages. Cette classe permet de se connecter, de créer un compte de répondre à un sondage, voir les liste des sondages, et récupérer les réponses à un sondage.

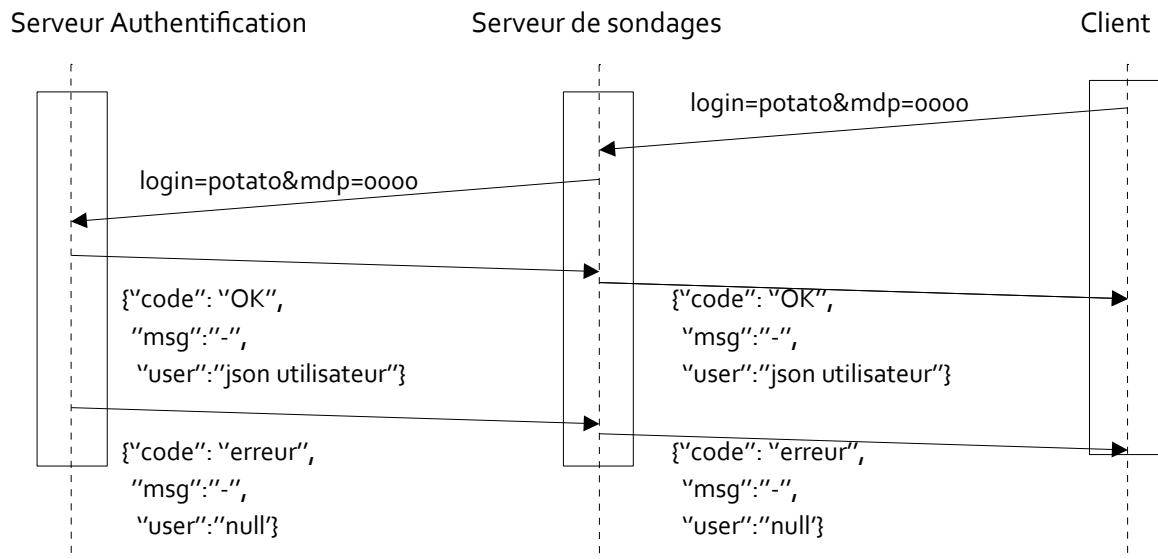
## III- Modélisation :

Diagramme des classes principales de l'application Sondage :



## IV- Diagrammes des communications :

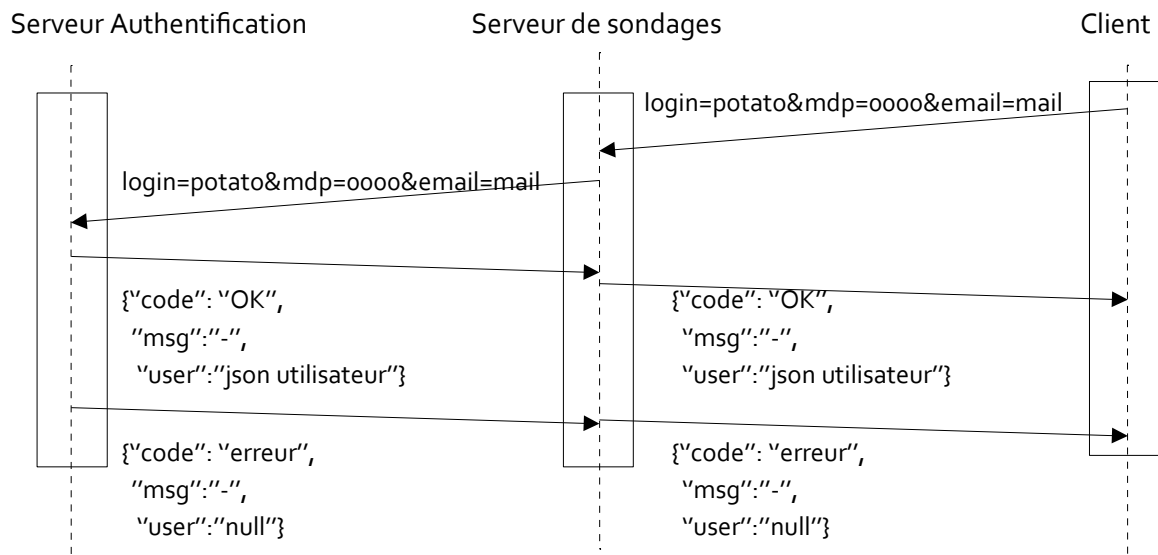
### 1- Connexion :



Erreurs possibles :

- Pas de données dans le post
- Mauvais login et/ou mauvais mot de passe

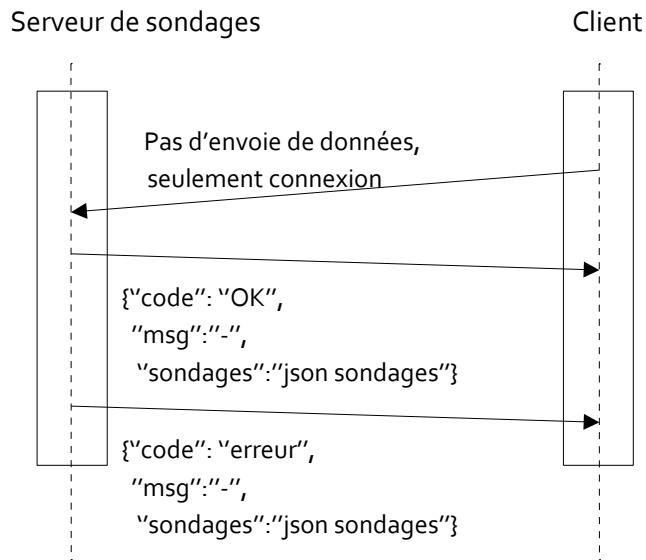
### 2- Création de compte :



Erreurs possibles :

- Pas de données dans le post
- login existe déjà

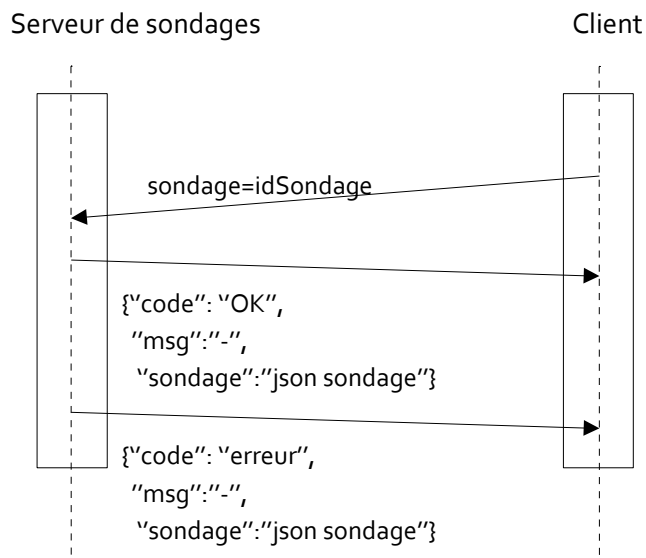
### 3- Lister les sondages :



Erreurs possibles :

- Il n'y a aucun sondage pour le moment

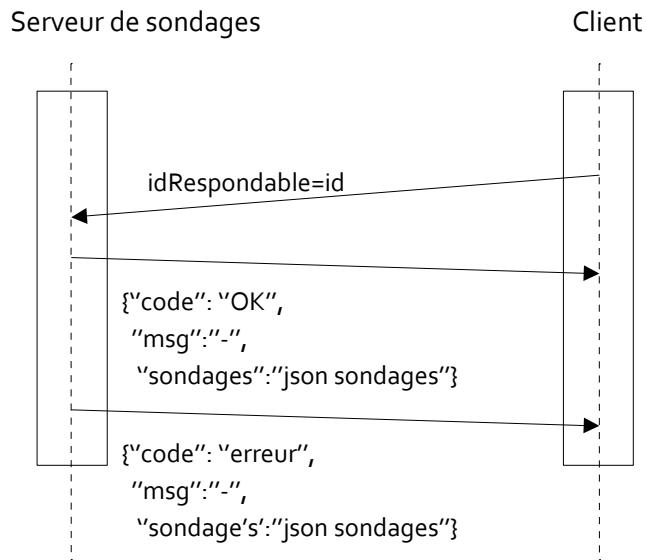
### 4- Répondre à un sondage :



Erreurs possibles :

- Mauvais id

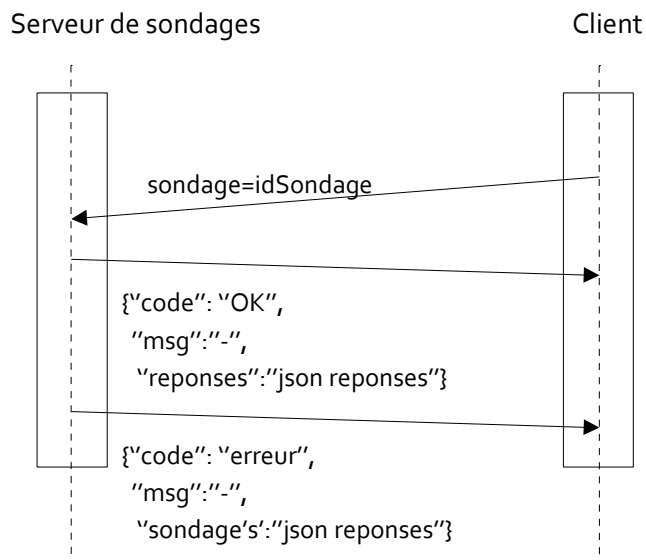
## 5- Afficher les sondages dont l'utilisateur est responsable :



Erreurs possibles :

- Mauvais id
- L'utilisateur n'a aucun sondage

## 6- Récupérer les réponses d'un sondage :



Erreurs possibles :

- Mauvais id
- Le sondage n'a aucune réponse

## V- Format des fichiers JSON :

On a choisit de séparer les fichiers de la manière suivante :

JSON

```
|_ Sondages
    |_ Sondage#id1
        |_ Reponses
            idUser1.json
            idUser2.json
            idUser3.json
        sondage.json
    |_ Sondage#id2
    |_ Sondage#id3
|_ Users
    users.json
```

Le dossier Sondages contient un dossier pour chaque sondage avec le fichier sondage.json qui contient les questions du sondage et les choix possibles pour chaque question, et un dossier Reponses contenant un fichier pour chaque utilisateur qui a répondu au sondage. Cette structure permet d'alléger les fichiers JSON, si par exemple des milliers d'utilisateurs répondent au sondage, et un accès plus rapide et plus facile aux données qui nous intéressent. Cependant, nous avons décidé d'enregistrer tous les utilisateurs dans un même fichier pour éviter d'ouvrir et fermer plusieurs fichiers lors de l'authentification, ce qui peut être long si on a beaucoup d'utilisateurs et que celui qui se connecte est le dernier. C'est la classe GestionnaireUtilisateur (qui sera instancié dans le serveur d'authentification) qui va ouvrir ce fichier et charger tous les utilisateur du fichier dans son attribut qui est sous forme de HashMap, où on classe les utilisateurs selon leur pseudo pour simplifier les recherches.

## VI- Execution :

1. Mettre le dossier sondage dans le dossier www de uwamp (ou équivalent)
2. Démarrer le serveur uwamp
3. Se placer dans le dossier sondage/java
4. Compiler les classes : `javac -cp «../json.jar» *.java`
5. Se placer dans le dossier sondage/serveurs
6. Compiler les classes : `javac -cp «../json.jar;../java» *.java`
7. Exécuter le serveur : `java -cp «../json.jar;../java» ServeurAuthentification`
8. Dans un autre terminal, se placer dans le dossier sondage/serveurs, puis exécuter le serveur :  
`java -cp «../json.jar;../java» Client`

## VII- Arborescence des Fichiers :

Sondage

## |\_ JAVA

- Client.java
- GestionnaireUtilisateurs.java
- Question.java
- Reponse.java
- Sondage.java
- Test.java
- Utilisateur.java

## |\_ JSON

## |\_ PHP

### |\_ Classes

- Question.php
- Reponse.php
- Sondage.php
- Utilisateur.php

### |\_ HTTP

- connexion.php
- creerSondage.php
- repondreSondage.php

## |\_ Serveurs

- Client.java
- CreateFormHandler.java
- CreateHandler.java
- IndexHandler.java
- LoginHandler.java
- ServeurAuthentification.java