

# Développement d'un robot pour extraire des données à partir de Twitter

TwittyBot



S3B

FUCKS Ariirau

SIMON Thomas

NIGRO Antony

MALLERET Maxence

# Table des matières

## Table des matières

Présentation du projet.....	3
Liste des fonctionnalités du système.....	4
Diagramme des cas d'utilisation.....	5
Diagramme de classe .....	6
Diagramme d'activité .....	7
Diagramme de séquence .....	8
Recensement et évaluation des risques.....	9
Maquette.....	10

# Présentation du projet

Notre projet consiste à réaliser un robot capable d'extraire des données à partir de Twitter, des données telles que les « followers », « following », « biographie » ainsi que les tweets partagés d'une personne souhaitée.

Nous avons ressorti 4 missions :

- Extraction des données
- Conversion des données
- Traitement des données
- Création de l'interface utilisateur

Nous allons également réaliser une interface permettant d'afficher toutes les données acquises par le robot et les restituer graphiquement (voir Maquette).

Nous pouvons détailler les 4 points cités précédemment :

- L'extraction des données se fera grâce à l'API Sélénium, en réalisant un programme en Java capable de faire un « scrapping » d'une page Twitter.
- La conversion des données se fera également grâce à l'API Sélénium, nous allons récupérer de nombreuses informations pour ensuite les restituer dans une base de données, créée au préalable (voir Diagramme de Classe)
- Le traitement des données se fera quant à elle directement sur la base de données, grâce aux requêtes SQL.
- Concernant la création de l'interface utilisateur, nous la réaliserons en Java.

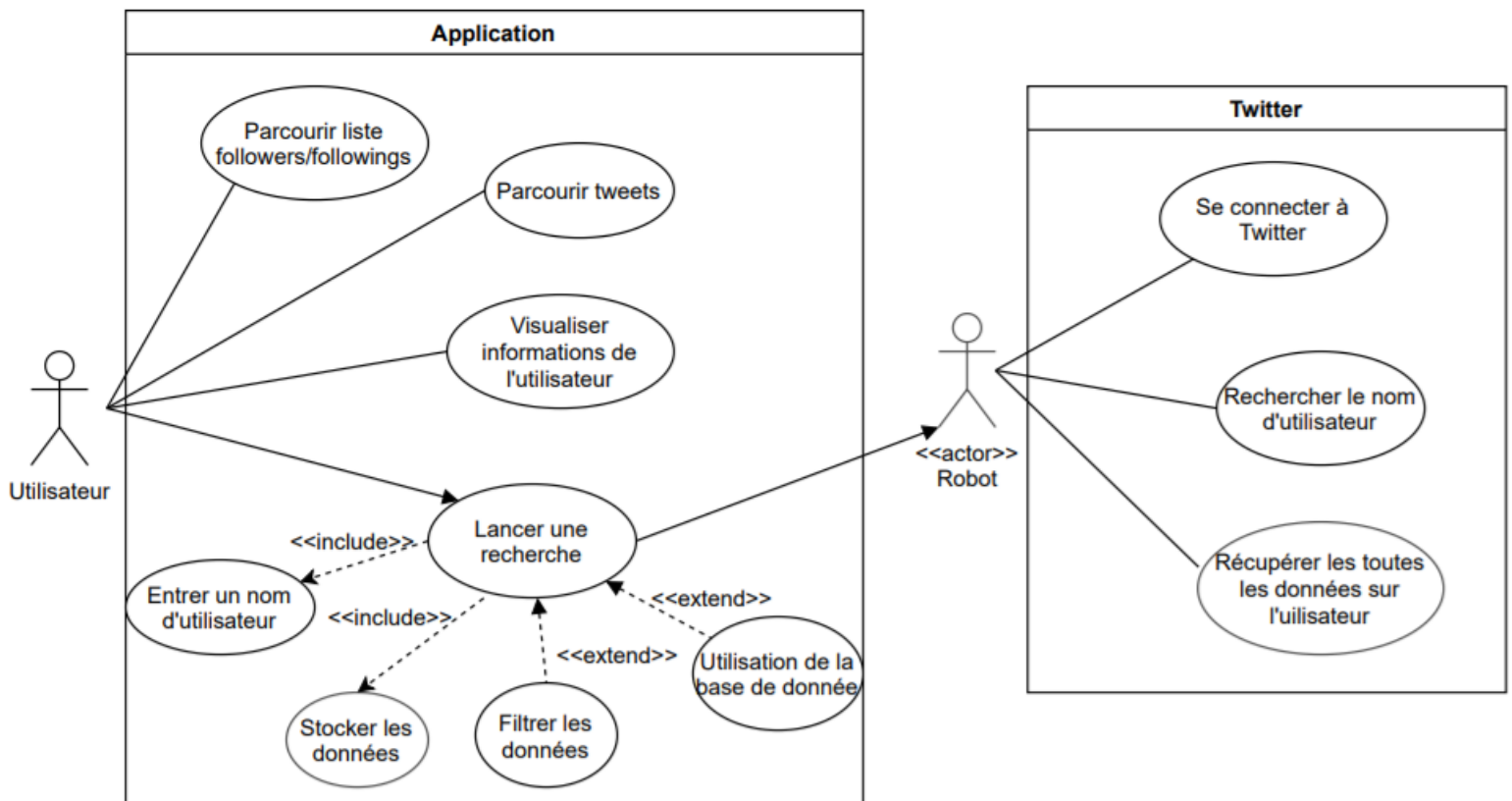
# Liste des fonctionnalités du système

Ci-dessous la liste des fonctionnalités de notre système :

- Rechercher un utilisateur
- Parcourir la liste des followers/followings d'un utilisateur
- Parcourir les tweets d'un utilisateur
- Visualiser les informations d'un utilisateur (biographie, pseudonyme...)
- Filtrer les informations d'un utilisateur
- Affichage d'un tweet en particulier

Le peu de fonctionnalités citées ci-dessus paraissent assez simple en raison de l'objectif premier du projet plutôt basique.

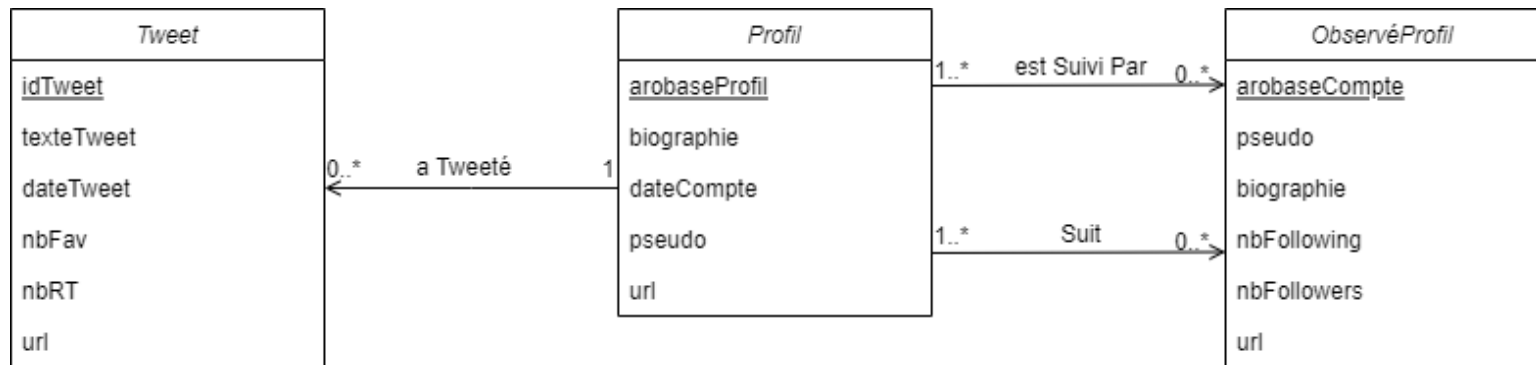
# Diagramme des cas d'utilisation



## Description textuelle :

Dans l'application, l'utilisateur peut parcourir la liste des followers/followings d'un utilisateur que celui-ci aurait fourni au préalable. L'utilisateur émet une recherche, les données de l'utilisateur recherchées sont stockées mais peuvent également être filtrées (par date, par popularité...) grâce à l'utilisation de la base de données (avec des requêtes SQL). Lancer une recherche est réalisée par le robot que nous allons produire. A partir de ce robot, nous devons lui fournir un accès à Twitter avec un compte (que nous avons créé) mais celui-ci devra uniquement récupérer toutes les données (HTML) du profil Twitter de la personne recherchée.

# Diagramme de classe



## Description textuelle :

La base de données sera composée de 5 tables, dont 2 étant des tables de liaison. Les 3 autres sont les suivantes :

**Tweet** : chaque tweet enregistré sera stocké ici avec toutes les informations qu'il contient et le « propriétaire » de ce tweet.

**Le Profil** est la table représentant les différents profils de compte twitter recherchés directement par l'utilisateur. Le compte enregistré dans cette table est celui tapé par l'utilisateur dans la barre de recherche.

**ObservéProfil** est la table contenant des informations (moins complète puisque nous n'avons que le nombre de follower et de following et non le détail de chacun) sur les profils qui sont en relation avec le profil recherché par l'utilisateur, en d'autres termes ce sont dans cette table tous les profils qui follow ou sont follow par le profil recherché.

Les 2 dernières tables permettent de savoir quelle est la relation entre le profil recherché et ObservéProfil.

## Table relationnelle :

**Tweet** (idTweet, texteTweet, dateTweet, nbFav, nbRT, url, #arobaseProfil)

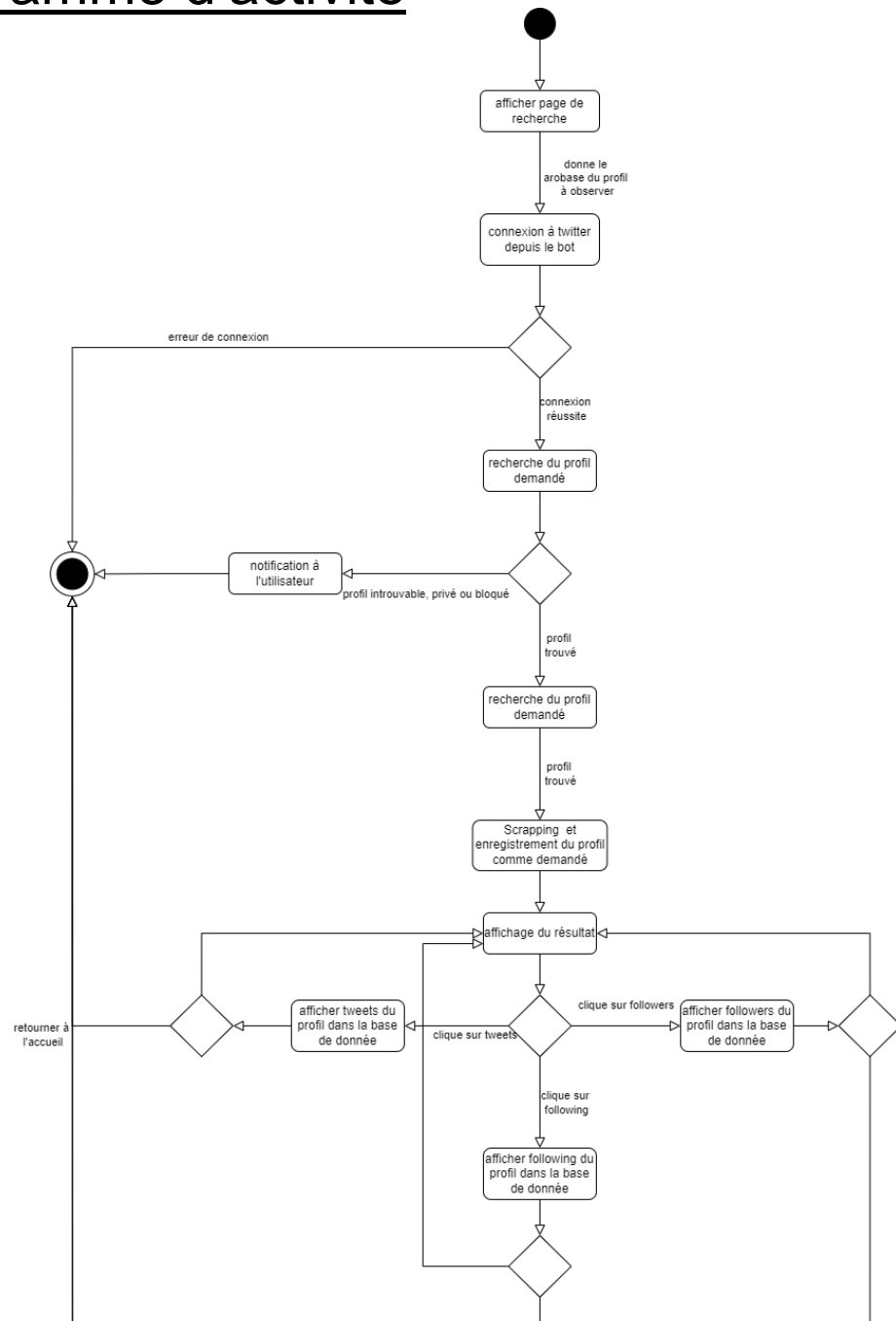
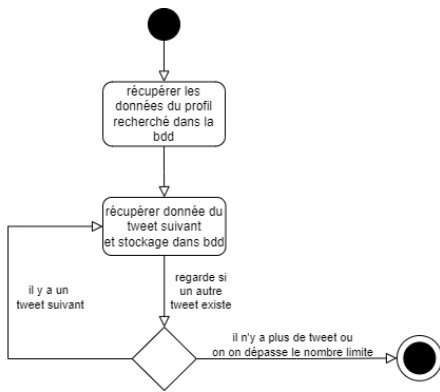
**Profil** (arobaseProfil, biographie, dateCompte, pseudo, url)

**estSuiviPar** (idSuiviPar, #arobaseProfil, #arobaseCompte)

**Suit**(idSuit, #arobaseProfil, #arobaseCompte)

**ObservéProfil** (arobaseCompte, pseudo, biographie, nbFollowing, nbFollowers, url)

# Diagramme d'activité

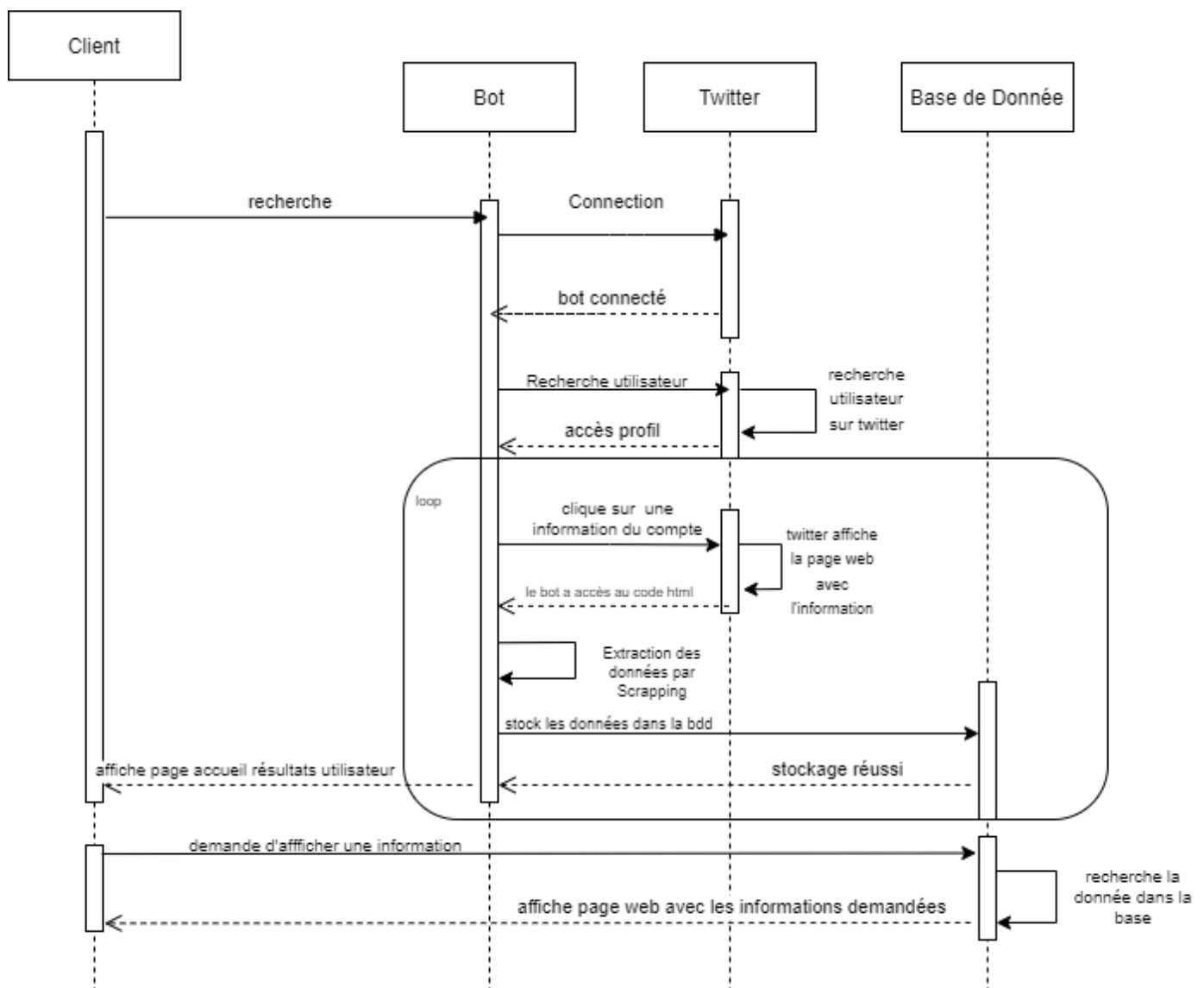


## Description textuelle :

Le diagramme de droite représente le fonctionnement général de l'application pour sa recherche de profil par nom d'utilisateur (ou arobase). Ce diagramme général est très bien représenté par la partie maquette faite sous balsamiq.

Le diagramme de gauche représente quant à lui la partie sur fonctionnement du « scrapping » lors de la recherche par le robot. Le robot commence par récupérer les données de la page du profil recherché, puis parcourt les tweets pour les récupérer jusqu'à atteindre une limite (soit lorsqu'il n'y a plus de tweet, soit lorsque l'on dépasse la date paramétrée).

# Diagramme de séquence



## Description textuelle :

Voici le diagramme séquence Système représentant un cas concret de recherche de profil. Le client donne le nom du profil à chercher puis le robot prend la main. Dans ce diagramme tout est fonctionnel donc l'accès à twitter et au compte est possible. Ensuite une boucle a lieu lors du scrapping pour étudier tous les tweets un par un du profil puis les stocker avant de passer aux suivants. Lorsque tout cela est fini, on affiche toutes ces différentes informations relevées en fonction des demandes de l'utilisateur.



# Recensement et évaluation des risques

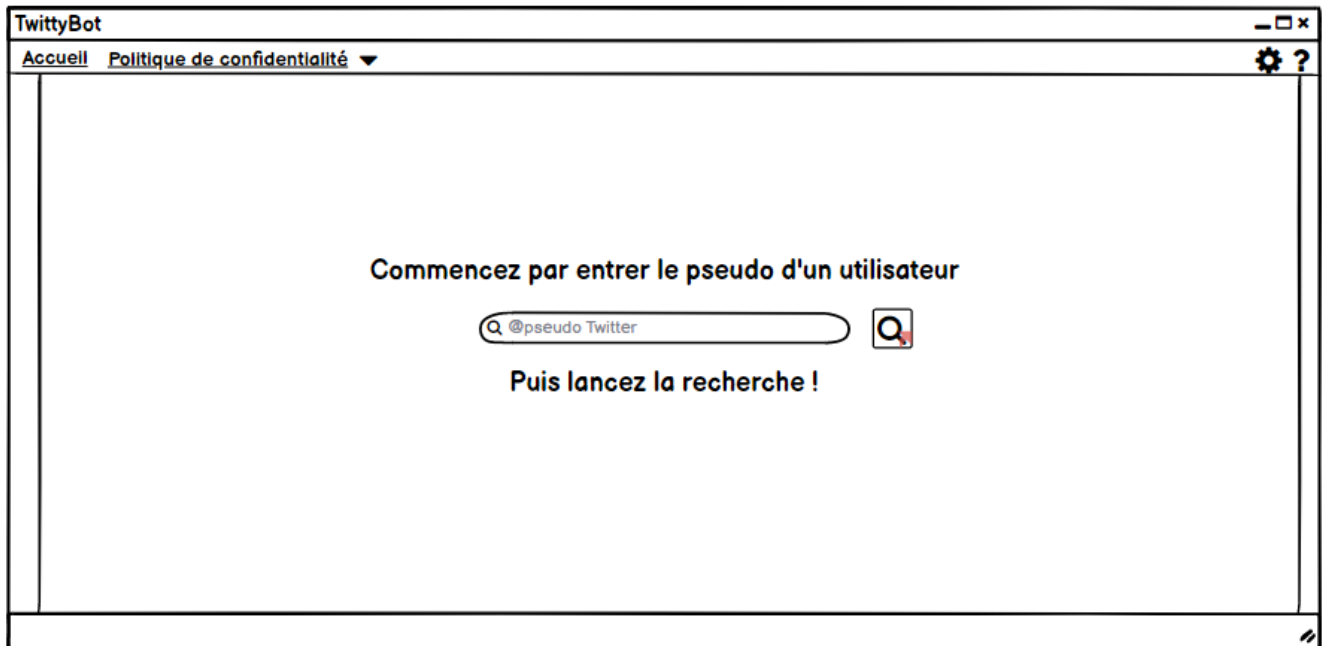
Concernant les risques, nous en avons recensé quelques-uns :

- Durée de téléchargement des informations
  - Gestion de la base de données
  - Le compte de l'utilisateur est privé
- 
- La durée de téléchargement des informations n'est pas compliquée à régler. Twitter peut bloquer notre programme en remarquant que nous récupérons des données d'un utilisateur grâce à un robot, pour régler ce problème nous devons ajouter des délais entre les requêtes d'extraction de ses informations, de cette façon Twitter pensera que notre robot est « humain ».
  - Pour la gestion de la base de données, elle est en partie résolue. Si nous partons du principe où nous utilisons notre robot uniquement sur des personnes qui ne sont pas populaire mais seulement entre les membres du groupe, ce problème est à exclure. Exemple : *nous n'allons pas utiliser notre robot pour extraire les données Twitter de Emmanuel MACRON*
  - Si l'utilisateur a le compte en privé, il n'y a **aucune** alternative et nous ne pourrons rien y faire. Le robot n'est donc pas fonctionnel si le compte de l'utilisateur souhaité est en privé.

# Maquette

Concernant la maquette, nous avons utilisé Balsamiq pour sa production :

Page d'accueil :



Page de résumé :



## Page des followers :



## Page des followings :



## Page des tweets :

TwittyBot

[Accueil](#) [Politique de confidentialité](#)

Liste des Tweets de @pseudo

Filtrer les Tweets

Du plus au moins récent

Du plus vieux au plus récent

Followers

Followings

Aujourd'hui à 13h02 :

20

02

LOREM ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure

Aujourd'hui à 8h46 :

23

12

LOREM ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Médias associés:  
[Image1](#)  
[Vidéo2](#)

Hier à 17h58 :

23

LOREM ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

Le 11/12/2021 à 15h37 :

04

LOREM ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.