

Hugo HERAULT
Nicolas MAERTEN
Nathanaël LE GALL
Hélori GOUIN

Jeudi 16 Mai 2019

A3 Dev
Concepteur Développeur d'Applications

UTweet.io

Spécialisation Technique

Sommaire

Introduction.....	1
I.Contexte, Demandes et Contraintes.....	2
1.Contexte.....	2
2.Demande du client.....	2
3.Contraintes.....	2
II.Description des fonctionnalités.....	3
1.Inscription & Connexion.....	3
2.Vue Principale.....	4
3.Création & Modification d'un tweet.....	5
4.Spinner Content.....	6
5.Analytics.....	6
6.Assistance à l'utilisation.....	7
7.Paramètres.....	8
8.Gestion de l'abonnement.....	9
9.Nouvel abonnement.....	10
III.Arborescence.....	11
IV.Amélioration supplémentaires.....	12
1.Version Web.....	12
2.Langages supplémentaires	12
3.Recherche de Tweets.....	12
4.Renvoyer un tweet précédemment publié.....	12
5.Publié un tweet sur plusieurs comptes.....	12
6.Automatisation des paiement dans l'application.....	12

Introduction

Ce document est destiné à la réalisation de l'application Utweet.io. Cette application a pour objectif de répondre au besoin de la communication ainsi que de l'exposition de masse sur les réseaux sociaux.

I. Fonctionnalités

1. Authentification d'un compte Twitter

Les différentes façon d'autoriser son application à accéder à twitter, et d'y connecter un utilisateur est spécifié dans la documentation officielle de l'API twitter (lien ci-dessous).

<https://developer.twitter.com/en/docs/basics/authentication/overview/oauth>

2. Déconnexion

La déconnexion de l'utilisateur sera fera par la suppression du token de l'utilisateur.

3. Envoyer un tweet sans spinning content

L'envoi d'un tweet se fait par la requête POST statuses/update, avec un objet json contenant le message et le nom de l'auteur et un id. Comme précisé dans la documentation (lien ci-dessous).

Verbe http et url requis :

<https://developer.twitter.com/en/docs/tweets/post-and-engage/overview>

Format de l'objet JSON:

<https://developer.twitter.com/en/docs/tweets/data-dictionary/overview/intro-to-tweet-json.html>

4. Créer un tweet avec spinning content

L'utilisateur devra choisir la période d'envoi du tweet dans un input date, ainsi que la fréquence d'envoi grâce un un selector variant de 1/min a 1/jours.

Le tweet est enregistré en base de données par une requête POST /message, avec un objet identique à un tweet normal avec un timestamp permettant à un script qui vérifie toute les minutes si les tweets programmés ont un timestamp identique au timestamp actuelle, et qui l'envoie si la condition est vérifié. Si le tweet est envoyé, le champ de timestamp en base est mis à jour en ajoutant au timestamp l'interval choisi.

5. Modifié un tweet programmé

Sur le clique du bouton "valider" de la vue "Modify tweet", une requête PUT /message/{messageId} permettant de modifier un tweet sera envoyée avec un objet JSON créé avec le texte (*text* string), la date (*date* string), le décompte (*countdown* string), le temps (*timer* string), et éventuellement des images (*images* array of string) ainsi que l'identifiant de l'utilisateur (*user_id* integer) et le compte twitter concerné (*twitter_account* array of integer)

6. Ajout d'un compte

Sur le clique du bouton “ajouter un autre compte” de la modal “Main” l’application envoie une requête GET à Twitter comme décrit dans la documentation officielle de l’API twitter (lien ci-dessous).

<https://developer.twitter.com/en/docs/basics/authentication/overview/oauth>

7. Confirmation de suppression de compte

Sur le clique du bouton “supprimer mon compte” de la vue “confirmation de suppression du compte” on envoi une requête DELETE /user avec un token contenant l’id de l’user permettant de supprimer ce même user et donc son compte.

8. Analytics

L’intégration de l’api google analytics se fait comme la documentation le précise (lien ci-dessous).

IOS:

<https://developers.google.com/analytics/devguides/collection/firebase/ios/>

Android:

<https://developers.google.com/analytics/devguides/collection/firebase/android/>

Web App:

<https://developers.google.com/analytics/devguides/collection/analyticsjs/>

9. Changer d'abonnement

Le changement d’abonnement se fait lorsque l’utilisateur, depuis le menu, clique sur le bouton changer d’abonnement, qui redirige vers une vue de paiement stripe pour confirmer son changement d’abonnement. Si l’utilisateur passe d’un compte payant à gratuit. Pas de vue de paiement.

Une requête PUT /subscription , est envoyée en base avec la modification apportée. L’objet envoyé est un json contenant le nom de l’abonnement, le nombre maximum de tweet qu’il pourra envoyer et la date de l’historique maximal.

II. Technologies abordées

Ci-dessous, la liste des technologies utilisées dans notre environnement de développement :

- **Repository** GitLab
- **Software quality** GitLab
- **CI/CD** GitLab
- **Front** Ionic 4
- **Back** Node.JS 11 (Express 4)
- **ORM** Sequelize
- **Routing** Express
- **Database** SQL
- **IDE** PhpStorm
- **Database client** PhpStorm Plugin
- **FTP client** PhpStorm Plugin

III. Tests

1. Tests unitaires

Pour chaque requête, il faut tester les dépassements de valeur, mauvais type, valeur erroné et ajout partiel de données (ex : attendu A,B,C envoyé : B,C).

2. Tests de performance

Une fois la première version de l'api développé, des tests de performances seront effectué en requêtant l'api jusqu'au crash.

Ils permettront de connaître la charge possible à chaque routes ou chaque globale en cas de requête simultanée.