

WebParts Réseaux sociaux

Spécifications techniques



Damien BRIOL-DUHALDE

Magellan Partners

26/07/2022

● Introduction : Contexte

Le rapport a été réalisé dans le cadre d'un stage sous le tutorat de Monsieur Olivier HANNOIS, à l'agence Exakis CSV de Bidart. Le projet a duré 2 mois (mi-mai/mi-juillet 2022).

Pour un complément d'informations, vous pourrez consulter le Call #63 des Points Dev d'Olivier (avec le compte Microsoft exakisCSV) : [Call #63 Web Part Instagram \(sharepoint.com\)](#)

● Objectif du document

Ce document a pour objectif de définir le projet « WebParts Réseaux Sociaux ». La communication avec les API des réseaux sociaux étant complexe à mettre en place, ce rapport vous permettra de mieux comprendre le fonctionnement des WebParts.

● Documents de Référence

Nom du document	Description
Spécifications Techniques – Rapport WebParts Réseaux Sociaux	Rapport de projet de Damien BRIOL--DUHALDE

Sommaire

Chapitre 1 : Instagram

Page 4

Partie 1 : Fonctionnement de la WebPart

I - API Instagram et Fonction Azure

II - Fonction React et WebPart

Partie 2 : Utilisation de la WebPart

I - Utilisation de la WebPart existante

II - Application de la WebPart sur un autre compte Instagram

III - Particularités des requêtes de l'API Graph d'Instagram

Chapitre 2 : Facebook

Page 19

Partie 1 : Fonctionnement de la WebPart

I - API Facebook et Fonction Azure

II - Fonction React et WebPart

Partie 2 : Utilisation de la WebPart

I - Utilisation de la WebPart existante

II - Application de la WebPart sur un autre compte Facebook

III - Particularités des requêtes de l'API Graph de Facebook

Chapitre 3 : TikTok

Page 21

Partie 1 : Fonctionnement de la WebPart

I - Service React

II - Fonction React et WebPart

Partie 2 : Utilisation de la WebPart

I - Utilisation de la WebPart existante

Chapitre 4 : LinkedIn

Page 22

Partie 1 : Avancées et problèmes rencontrés

Spécifications techniques

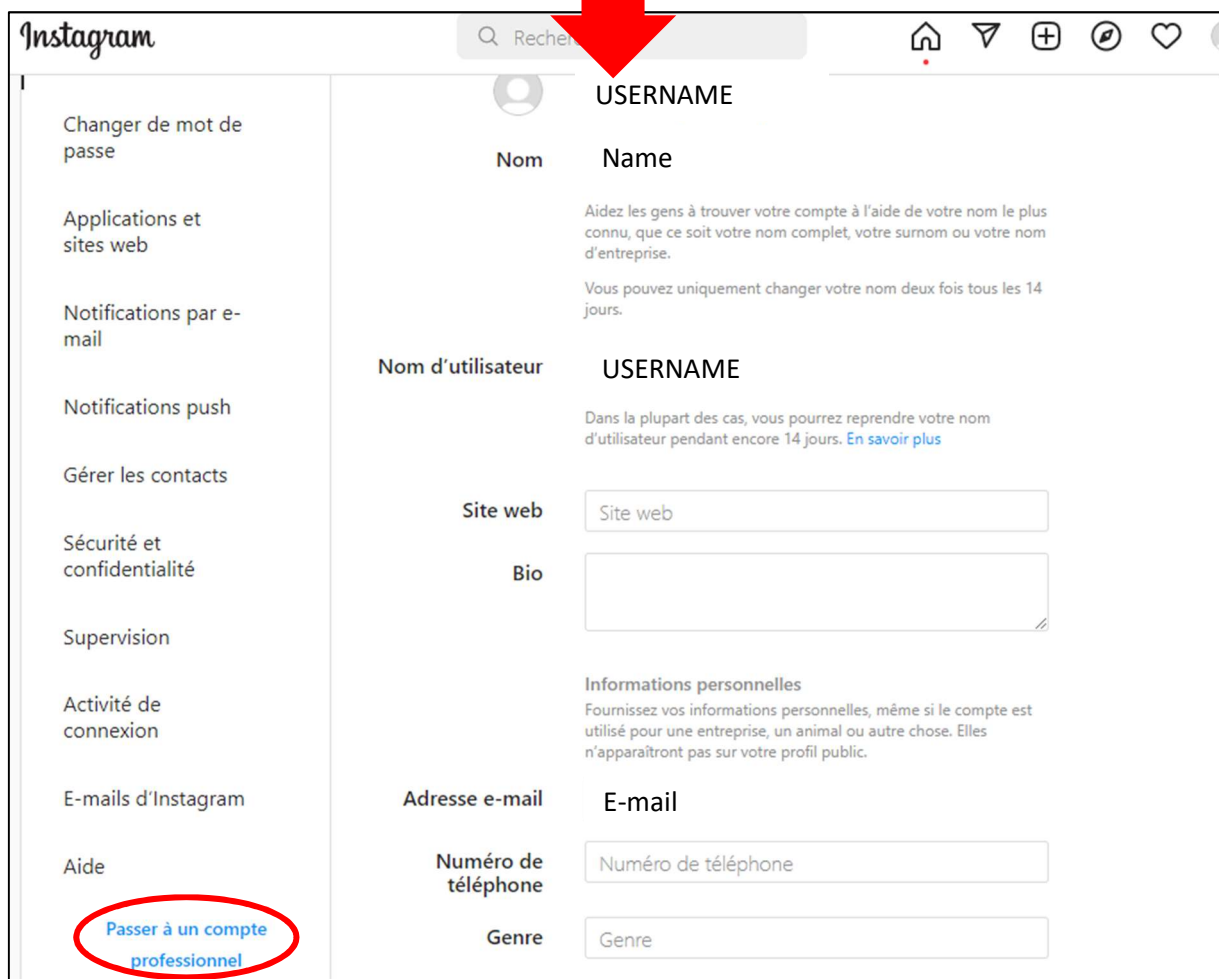
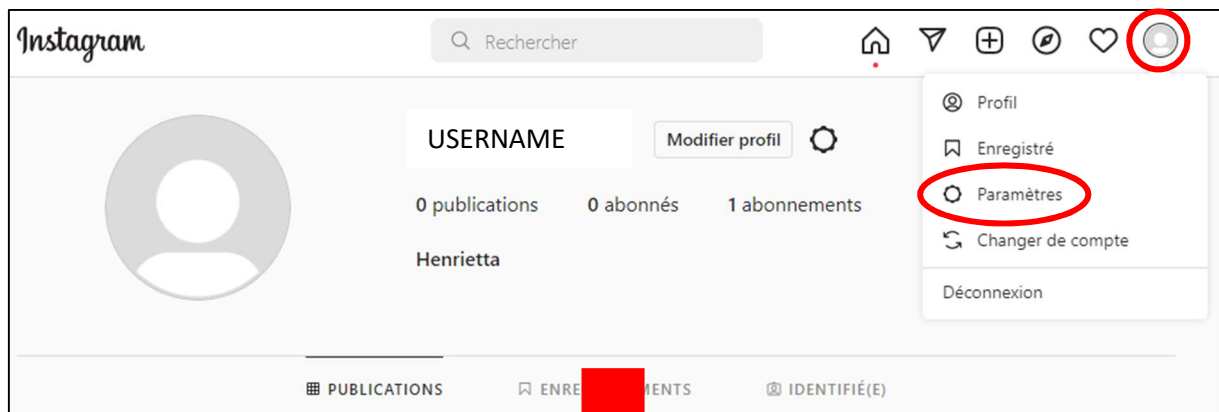
Chapitre 1 : Instagram

Partie 1 : Fonctionnement de la WebPart

I - API Instagram et Fonction Azure

L'accès à l'API Graph d'Instagram est assez restreint et nous contraint à réaliser un certain nombre de tâches avant de pouvoir communiquer avec l'API :

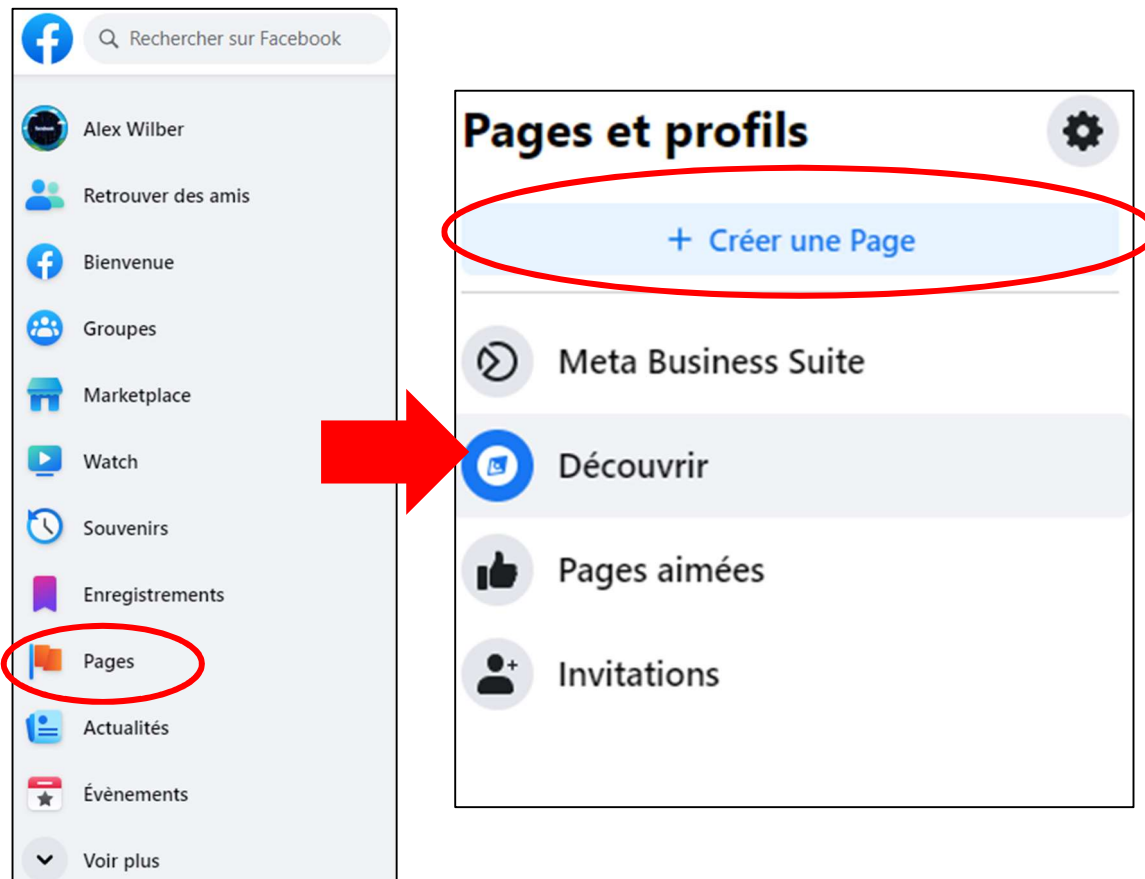
- 1) Tout d'abord, il faut convertir son [compte Instagram](#) en un compte Instagram Business (Voir captures d'écran ci-dessous).



- 2) Ensuite, il faudra créer, avec le même login que pour Instagram, une Page Facebook. Cette dernière devra être reliée au compte Instagram. Ainsi, un espace Compte Meta a été créé pour lier nos comptes Instagram et Facebook (voir [Espace Comptes Meta](#) dans Instagram)

Aide	Numéro de téléphone	Numéro de téléphone
Passer à un compte professionnel	Genre	Genre
Meta Espace Comptes	Comptes similaires suggérés	<input checked="" type="checkbox"/> Incluez votre compte lorsque vous recommandez des comptes similaires que les personnes pourraient suivre. [?]
Réglez les paramètres des		Désactiver temporairement mon

- 3) Une fois votre espace compte configuré, il faut créer une page [Facebook](#) :

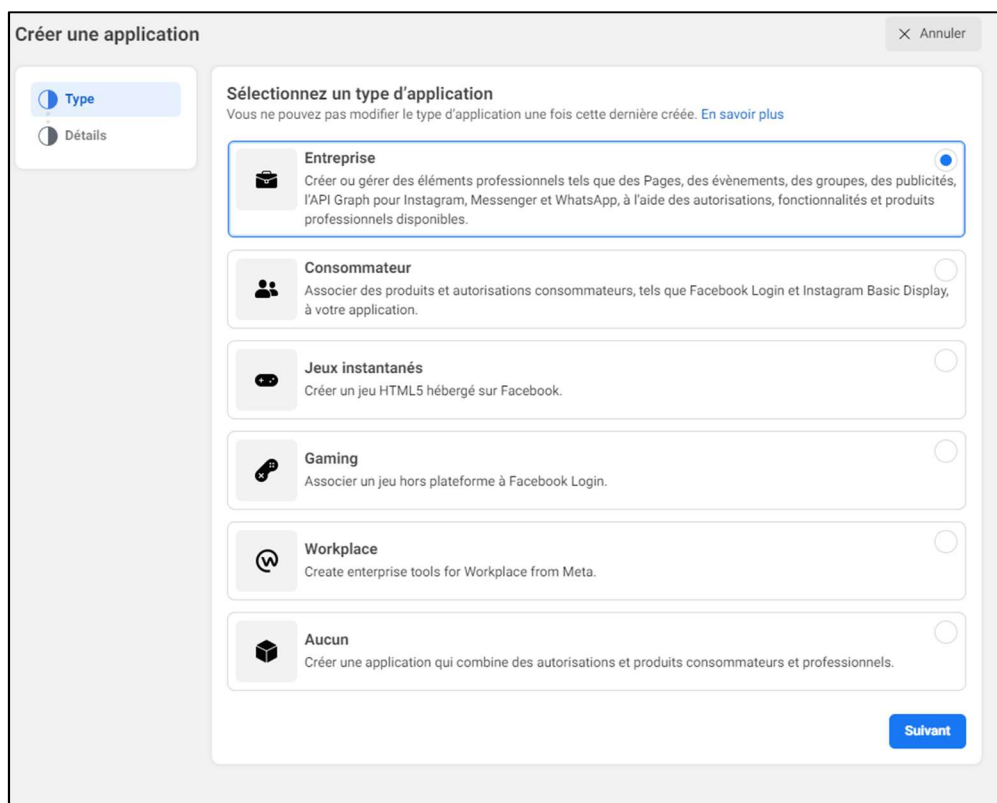
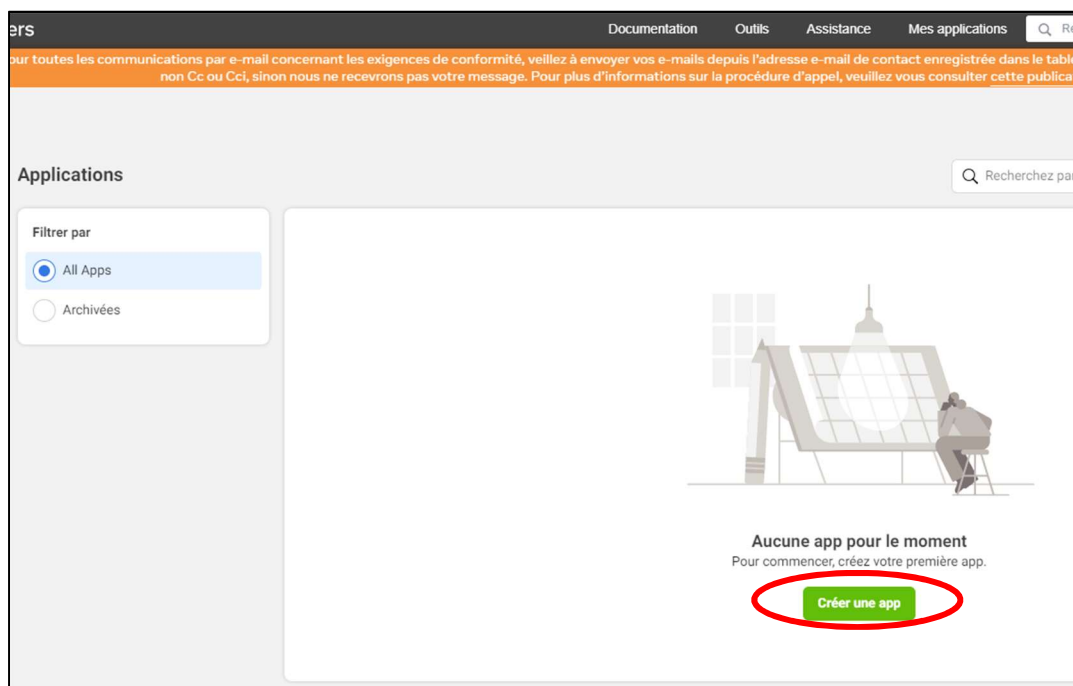


- 4) Enfin, nous devons synchroniser notre page facebook avec notre compte Instagram. En allant dans les paramètres de la Page :

The image shows the Facebook Page settings interface. On the left is a sidebar menu titled "Paramètres de la Page" with a search bar. The menu items are: Général, Messagerie, Infos sur la Page, Modèles et onglets, Notifications, Messagerie avancée, Rôles de la Page, Personnes et autres Pages, Audience de Page préférée, Publicités portant sur un enjeu social, électoral ou politique, Limites de publicités, Contenu de marque, and Instagram. The "Instagram" item at the bottom is circled in red. The main content area is titled "Associer à Instagram" and includes the Instagram logo. It contains the text "Associez votre Page Facebook à un compte Instagram. En savoir plus" and a blue button labeled "Associer le compte", which is also circled in red. Below this, it explains that others can help manage elements on both platforms, listing: Contenu, publicités et statistiques; Messages et commentaires; and Paramètres et autorisations. A note at the bottom states: "Vous pourrez vérifier les rôles plus tard dans les paramètres de la Page Facebook."

Concernant notre environnement de développeur META, nous devons créer un compte [Facebook Développeur](#) ayant les mêmes logins que le compte Facebook :

- Dans ce compte, nous devons créer une application avec le produit « API Graph pour Instagram ».



Créer une application

✕ Annuler

✓ Type

🕒 Détails

Fournir des informations de base

Nom à l'écran

Il s'agit du nom de l'application qui est associée à votre ID d'app. Vous pouvez le modifier plus tard.

App de test API Graph

21/32

E-mail du contact de l'application

Cette adresse e-mail servira à vous contacter au sujet d'infractions potentielles au règlement, de restrictions de l'application ou des étapes à suivre pour restaurer l'application si elle a été supprimée ou compromise.

alexw@y5g8v.onmicrosoft.com

Compte business · Facultatif

Pour accéder à certaines autorisations ou fonctionnalités, les applications doivent être associées à un compte business.

Aucun compte Business Manager sélectionné

En continuant, vous acceptez les [Conditions de la plateforme Facebook](#) et les [Politiques développeur](#).

Précédent

Créer une application

Ajouter des produits à votre app

Nous avons simplifié le processus de création d'app en montrant les autorisations et produits requis pour créer le type d'app sélectionné.



App Events

Comprenez comment les gens interagissent avec votre entreprise à travers les applications, les appareils, les plateformes et les sites web.

[Lire la documentation](#)

Configurer



Audience Network

Monétisez votre application et générez des revenus avec des publicités d'annonceurs Meta.

[Lire la documentation](#)

Configurer



Facebook Login

Le produit de connexion social numéro un au monde.

[Lire la documentation](#)

Configurer



Fundraisers

Create and manage fundraisers for charities.

[Lire la documentation](#)

Configurer



API Graph pour Instagram

Intégrez l'API Instagram à votre app pour permettre aux entreprises de l'utiliser avec leurs comptes Instagram.

[Lire la documentation](#)

Configurer



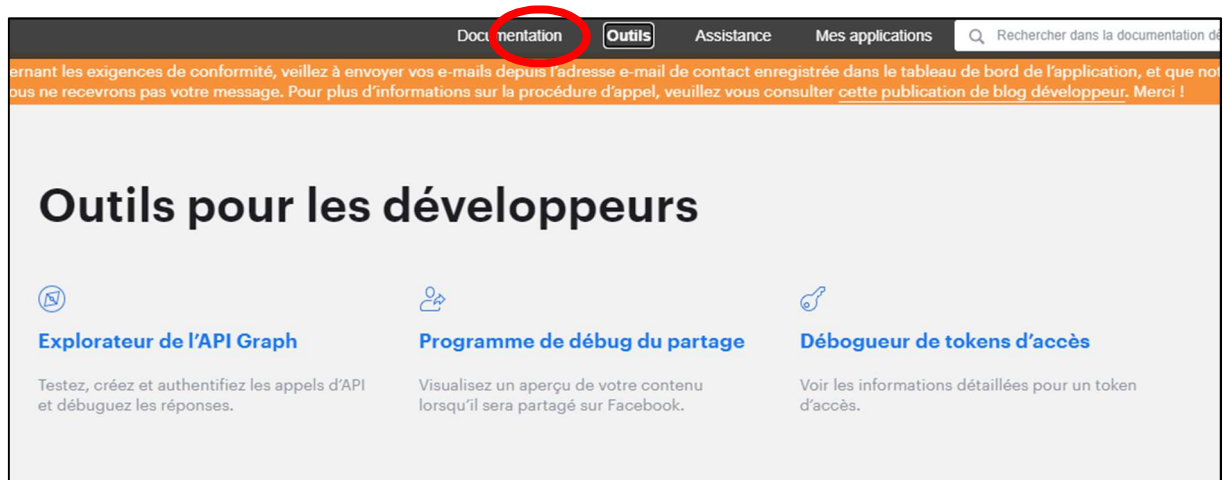
Jobs

Post jobs to the Facebook platform and receive applications from users.

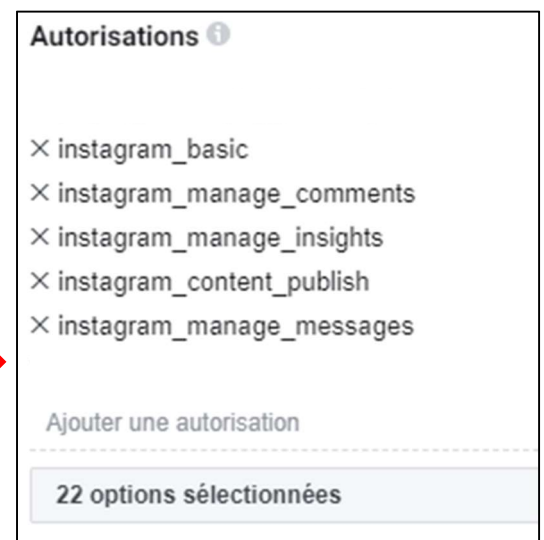
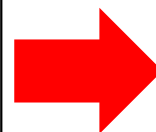
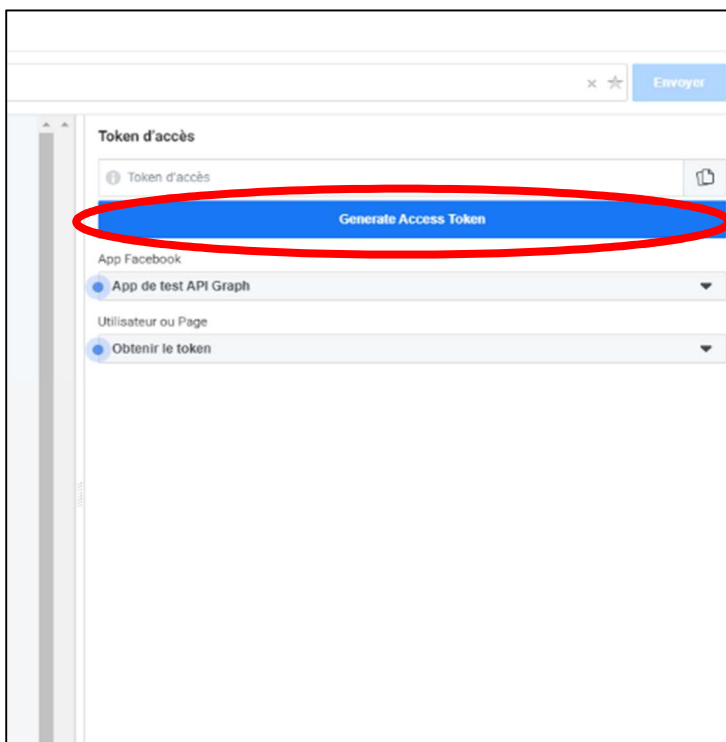
[Lire la documentation](#)

Configurer

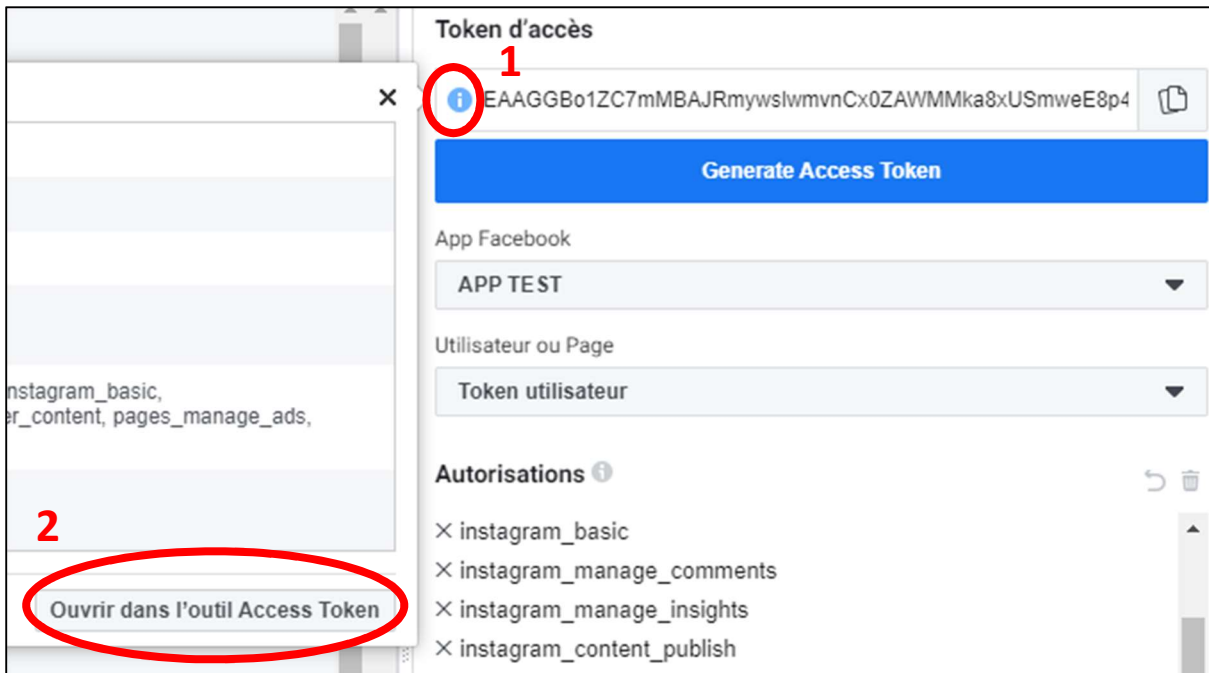
- Pour obtenir notre Access token (= clé privée permettant d'accéder aux données d'Instagram), nous allons, dans l'outil d'exploration de l'API Graph, sélectionner les autorisations nécessaires et étendre la durée de validité de notre accès (initialement 1h) :



- Se placer dans l'App Facebook développer fraîchement créée, sélectionner le token utilisateur et ajouter toutes les autorisations liées à Instagram

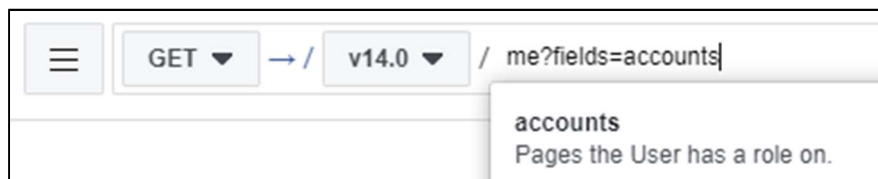


- Étendre la durée de validité pour avoir un token à durée Infinie

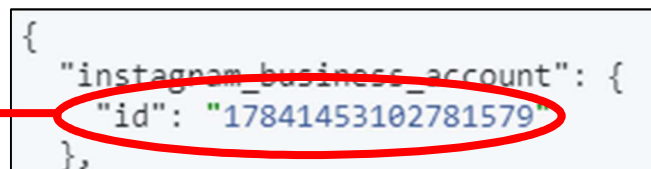
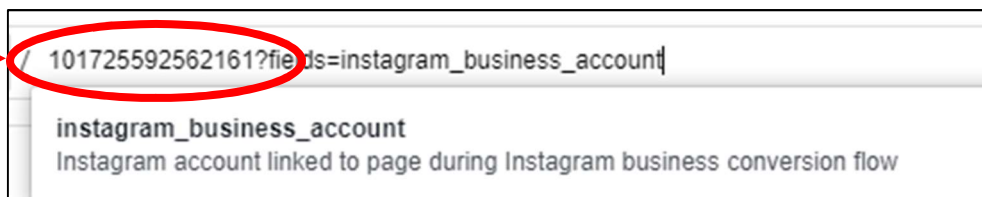


Étendre le token d'accès

- Revenir dans l'explorateur de l'API et saisir les requêtes suivantes :
 - « /me ?fields=accounts » permet d'obtenir l'ID de notre page Facebook



- « /*ID_PAGE_FACEBOOK*/?fields=instagram_business_account » permet d'obtenir l'ID de notre compte Instagram à requêter.



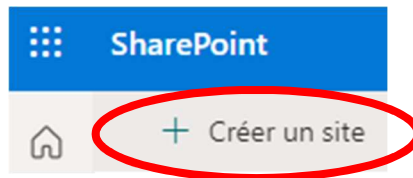
- Maintenant, nous sommes prêts à requêter Instagram (exemple : récupérer le lien de chacun des post Instagram du compte : « /*ID_COMPTE_INSTAGRAM*/media ?fields=media_url »)



Maintenant que nous sommes en mesure de communiquer avec l'API Graph d'Instagram, nous allons créer un espace de stockage pour sauvegarder les données après leur requête. Instagram et Facebook nous retournant des données au format .Json, nous les stockerons dans des cellules de plusieurs lignes de texte. Quant à TikTok, son fonctionnement ne nécessite pas de stocker les données.

Pour notre solution, nous avons décidé d'utiliser la liste SharePoint :

- Tout d'abord, nous allons créer la liste :
 - o Sur l'écran d'accueil SharePoint de votre tenant, nous allons créer un site pour héberger notre liste :



- o Ensuite, nous suivons les étapes de création d'un site SharePoint :



Nom du site *

Le nom du site est disponible.

Adresse du site *

L'adresse du site est disponible.
<https://y5g8v.sharepoint.com/sites/Sitededevreseauxsociaux>

Description du site

Sélectionnez une langue

Anglais

Sélectionnez la langue par défaut pour votre site. Vous ne pourrez plus la modifier.

- Notre site créé, nous allons y ajouter une liste :



Nom *



Liste_de_dev

Description

Récupération des données d'API (Instagram, Facebook, TikTok)

Navigation sur le site

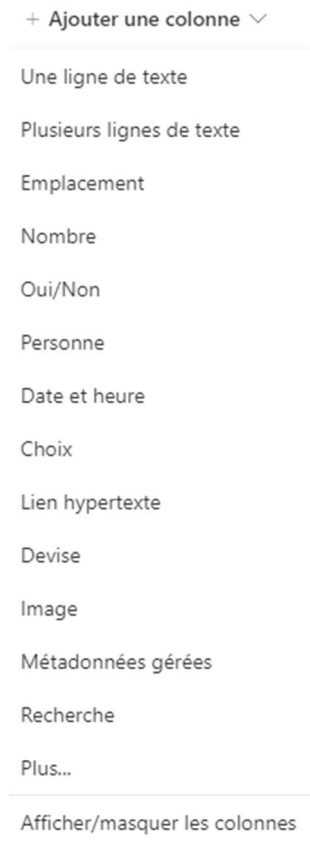
☒ Afficher dans la navigation du site

< Précédent

Créer

Annuler

- Maintenant que notre liste est créée, il nous reste à ajouter des colonnes



<input checked="" type="checkbox"/>	ID	A sélectionner
<input checked="" type="checkbox"/>	Titre	A sélectionner
<input checked="" type="checkbox"/>	Type	A sélectionner
<input checked="" type="checkbox"/>	DataAccount	A créer : plusieurs lignes de texte
<input checked="" type="checkbox"/>	DataMediaLastPublication	A créer : plusieurs lignes de texte
<input checked="" type="checkbox"/>	DataMediaCollection	A créer : plusieurs lignes de texte
<input checked="" type="checkbox"/>	Comptes_Business	A créer : une ligne de texte
<input checked="" type="checkbox"/>	Resultat_requete_business	A créer : plusieurs lignes de texte
<input checked="" type="checkbox"/>	DataStory	A créer : plusieurs lignes de texte
<input checked="" type="checkbox"/>	DataStory2	A créer : plusieurs lignes de texte
<input checked="" type="checkbox"/>	DataStory3	A créer : plusieurs lignes de texte
<input checked="" type="checkbox"/>	DataStory4	A créer : plusieurs lignes de texte
<input checked="" type="checkbox"/>	DataStory5	A créer : plusieurs lignes de texte
<input checked="" type="checkbox"/>	DataFacebook	A créer : plusieurs lignes de texte
<input checked="" type="checkbox"/>	Modified	A sélectionner
<input checked="" type="checkbox"/>	Created	A sélectionner
<input checked="" type="checkbox"/>	Created By	A sélectionner
<input checked="" type="checkbox"/>	Modified By	A sélectionner

- Maintenant que toutes les colonnes sont créées et affichées, notre espace de stockage peut être utilisé :
 - o Pour les WebParts

Afin d'avoir une WebPart à jour, nous devons requêter Instagram régulièrement.

- Pour fournir aux 4 WebParts suffisamment d'information d'Instagram, nous avons besoin d'effectuer 5 requêtes :
- Récupérer le dernier post Instagram du compte :
`https://graph.facebook.com/v14.0/{Votre_ID_Instagram}/media?fields=id%2Ccaption%2Cmedia_type%2Cmedia_product_type%2Cmedia_url%2Cpermalink%2Clike_count%2Ctimestamp%2Cchildren%7Bmedia_url%7D&access_token={Votre_Access_Token}&limit=1`
- Récupérer les données du compte Instagram :
`https://graph.facebook.com/v14.0/{Votre_ID_Instagram}?fields=username%2Cname%2Cbiography%2Cid%2Cfollows_count%2Cfollowers_count%2Cmedia_count%2Cprofile_picture_url&access_token={Votre_Access_Token}`
- Récupérer les publications du compte Instagram (nombre à saisir) :
`https://graph.facebook.com/v14.0/{Votre_ID_Instagram}/media?fields=media_type%2Cmedia_url%2Cthumbnail_url%2Cchildren%7Bmedia_type%2Cmedia_url%7D%2Cpermalink%2Ccaption%2Clike_count%2Ctimestamp%2Ccomments_count&limit={Nombre_Publications}&access_token={Votre_Access_Token}`
- Récupérer les stories du compte Instagram :
`https://graph.facebook.com/v14.0/{Votre_ID_Instagram}/stories?fields=media_type%2Cmedia_url%2Cthumbnail_url%2Cpermalink%2Ccaption%2Ctimestamp%2Cusername&limit=5&access_token={Votre_Access_Token}`
- Récupérer les données d'un quelconque compte Instagram Business :
`https://graph.facebook.com/v14.0/{Votre_ID_Instagram}?fields=business_discovery.username({Compte_A_Requeter})%7Busername%2Cwebsite%2Cname%2Cig_id%2Cid%2Cprofile_picture_url%2Cbiography%2Cfollows_count%2Cfollowers_count%2Cmedia_count%2Cmedia%7Bid%2Ccaption%2Clike_count%2Ccomments_count%2Ctimestamp%2Cusername%2Cmedia_product_type%2Cmedia_type%2Cowner%2Cpermalink%2Cmedia_url%2Cchildren%7Bmedia_url%7D%7D%7D&limit=25&access_token={Votre_Access_Token}`

Ainsi, pour automatiser ces requêtes et par la même occasion sécuriser notre code (éviter d'afficher notre Access Token dans la console F12 de notre navigateur), nous avons utilisé une fonction Azure :

- 1) Pour créer cette fonction, nous sommes partis de la template générée par Azure : [comment créer une template d'Azure function](#).
- 2) Une fois la template générée, nous avons défini, selon notre besoin, le cron (=fréquence d'exécution de la fonction). Dans notre cas, la fonction de la dernière publication et du compte Instagram s'exécutent toutes les heures (cron : 0 0 * * * *) alors que la fonction des stories s'exécute une fois par jour (cron : 0 0 0 * * *).
- 3) Concernant le code, nous voulions requêter Instagram puis stocker nos données dans une liste SharePoint.
Pour stocker des données, il faut tout d'abord créer une Liste SharePoint : Nous avons donc utilisé Axios pour exécuter nos requêtes URL puis PNP pour stocker nos données dans une liste SharePoint.
[Voir code](#)

II - Fonction React et WebPart

Concernant le développement Frontend, nous avons utilisé [Yeoman](#) pour créer une template de WebPart React.

Dans le développement, nous avons séparé les tâches en 2 parties :

- Une partie permettant de récupérer les données de la liste SharePoint
- Une autre partie permettant d'utiliser les données reçues pour créer la WebPart.

Pour récupérer les données de la liste SharePoint, nous avons utilisé PNP. Une fois connecté à la liste, nous devons chercher la ligne avec le titre correspondant à notre besoin :

- « Requête API Story Instagram » : ligne où sont stockées les données nécessaires à la WebPart Stories
- « Requête API Instagram » : ligne où sont stockées les données pour les WebParts InstaAccount, LastPublication et prochainement InstaDiscovery.

Pour créer nos WebParts, nous avons utilisé plusieurs outils mis à disposition par Microsoft et par Node JS :

- [Les Shimmers](#) : importés depuis « office-ui-fabric-react/lib/Shimmer » pour créer des éléments de chargement pendant que les données de la WebPart s'affichent
- [Le Stack](#) : importé depuis « office-ui-fabric-react » pour créer une structure en grille pour notre WebPart de compte Instagram.
- [Modal](#) : importé depuis « office-ui-fabric-react/lib/Modal » pour créer des fenêtres modales (= pop-up)
- [Les Icons](#) : importées depuis « office-ui-fabric-react/lib/Icon » pour ajouter des icônes reprenant le style de Microsoft
- [Le Slider](#) : importé depuis « react_slick » pour créer des carrousels d'images
- [Les Stories](#) : importées depuis « react-insta-stories » pour créer des stories reprenant le style d'Instagram

[Voir code](#)

[Voir résultat](#) (compte Microsoft exakisCSV nécessaire)

Partie 2 : Utilisation des WebParts

I - Utilisation des WebParts

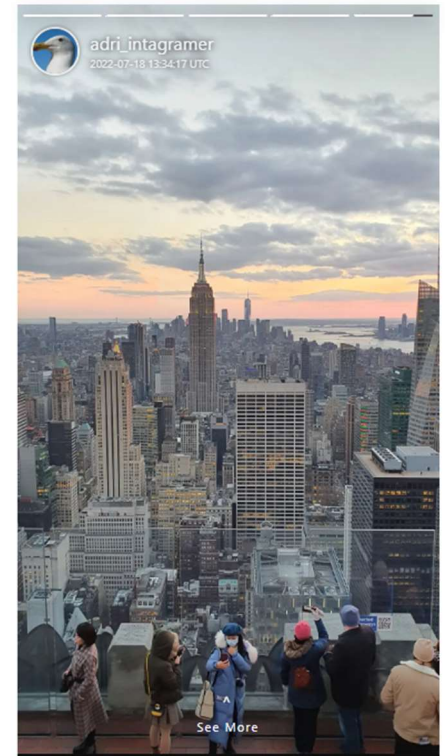
Actuellement, 4 WebParts Instagram sont disponibles : InstaAccount, instaPublication, StoryInsta et InstaDiscovery (interface similaire à la WebPart InstaAccount). Une dernière pouvant afficher n'importe quel compte Instagram Business avec un seul Access token est en cours de développement.



InstaAccount



InstaPublication



StoryInsta

Les 4 WebParts publiées reprennent 3 aspects d'Instagram : l'affichage du compte Instagram d'un utilisateur, l'affichage de la dernière publication d'un compte Instagram et l'affichage des 5 dernières stories du compte Instagram.

Parmi elles, certaines proposent quelques options de modification :

- InstaAccount, StoryInsta et InstaDiscovery : un slider dans la property pane permet de modifier le nombre d'éléments à afficher.
- InstaDiscovery : un champ de texte permet de modifier le nom du compte Instagram à requêter.

II - Application de la WebPart sur un autre compte Facebook

Pour utiliser cette WebPart avec un autre compte Instagram que celui de test, il suffit de modifier la partie Backend de notre WebPart : la fonction Azure.

En effectuant les étapes précédemment décrites, vous devriez obtenir un Access Token étendu avec les autorisations Instagram et vous devriez aussi avoir l'ID de votre compte Instagram.

Ainsi, les seules modifications à prévoir pour le code se situent dans les requêtes Axios (voir image ci-dessous pour les champs à modifier)

Modifier dans son environnement local (dans le repos) :

```
{ } local.settings.json
```

```
"instagramID" : "NOUVEL ID"
```

Modifier dans l'environnement AZURE :



Quant à la WebPart InstaDiscovery, elle permet d'afficher le contenu de différents comptes Instagram Business en modifiant uniquement l'username dans la property pane de la WebPart.

III - Particularités des requêtes de l'API Graph d'Instagram

Si nous « requêtons » un compte avec un token possédant les autorisations associées :

- Requêtes à partir d'un Access token spécifique au compte Instagram (obtenable grâce à la manipulation décrite précédemment)

POSSIBLE	IMPOSSIBLE
Les stories du jour	Les stories à la une
Les informations du compte Instagram	Les réels
Les publications du compte (Images, Vidéos, Carrousel)	

Si nous « requêtons » un compte à partir de la Business View :

- Requêtes à partir d'un nom de compte BUSINESS

POSSIBLE	IMPOSSIBLE
Les informations du compte Instagram Business	Les réels (sauf certains mais la condition nous est inconnue)
Les publications du compte (Images, Vidéos, Carrousel)	Les stories à la une
	Les stories

Chapitre 2 : Facebook

Partie 1 : Fonctionnement de la WebPart

I - API Facebook et Fonction Azure

Le fonctionnement de la WebPart Facebook est très similaire au fonctionnement des WebParts Instagram.

Depuis 2012, Instagram fait partie du groupe META. Grâce à cette acquisition, META a décidé d'utiliser une seule et même API pour les 2 services : l'API Graph.

Précédemment, nous vous avons décrit la démarche à suivre pour obtenir un Access token pour Instagram. Pour Facebook, il suffit de reprendre ces étapes en modifiant les autorisations au moment de générer l'Access token. (C'est-à-dire, mettre toutes les autorisations liées aux Pages Facebook : voir ci-dessous)



Concernant la requête permettant d'obtenir les données de la Page Facebook, nous avons récupéré l'ID de la Page Facebook (voir la partie Instagram).

Ainsi, grâce à l'ID de notre Page et à l'Access token paramétré avec les bonnes autorisations, nous pouvons effectuer la requête pour récupérer les données de la Page Facebook :

https://graph.facebook.com/v14.0/{VOTRE_ID_DE_PAGE_FACEBOOK}/feed?fields=id%2Cmessage%2Cstory%2Cfull_picture%2Cpermalink_url%2Ccreated_time%2Cfrom%7Bid%2Cname%2Cpicture%7D&access_token={VOTRE_ACCESS_TOKEN}

Ensuite, nous avons automatisé notre requête avec Azure (exécution toutes les heures, cron : 0 0 * * *) ; puis stocké nos données dans une liste SharePoint avec PNP.

II - Fonction React et WebPart

En reprenant la méthode utilisée pour Instagram, nous avons généré un « template Yeoman » et divisé notre code en 2 parties :

- 1) La récupération des données d'un côté
- 2) Et l'utilisation des données de l'autre.

Concernant la récupération des données, le principe est exactement le même que celui d'Instagram, nous utilisons les données stockées dans la liste SharePoint sur la ligne « Requête API Facebook »

Quant à l'utilisation des données, le seul élément qui diffère est la mise en forme de la WebPart.

Pour compléter cette WebPart, nous avons utilisé le [DefaultButton](#) d'office-ui-fabric-react pour rediriger l'utilisateur vers la page Facebook du post.

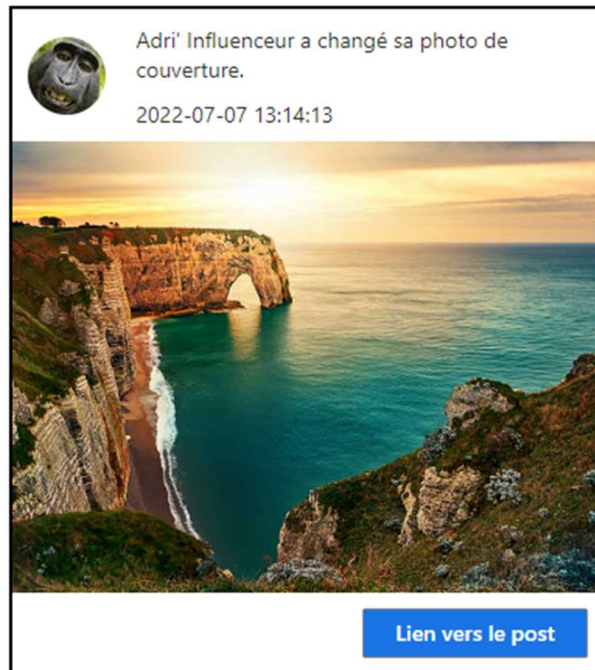
[Voir code.](#)

[Voir résultat.](#)

Partie 2 : Utilisation de la WebPart

I - Utilisation de la WebPart existante

La WebPart existante de Facebook permet d'afficher le contenu d'une Page Facebook :



Etant donné que l'API n'apporte pas beaucoup de données sur les pages Facebook, nous avons implémenté un DefaultButton pour rediriger l'utilisateur vers le post sur Facebook.

II - Application de la WebPart sur un autre compte Facebook

Pour afficher la page d'un autre compte Facebook, nous devons générer un nouvel Access token dans une application Facebook développer et reprendre les étapes de création décrites dans le chapitre sur Instagram.

A ce jour (API Graph V14), il n'existe pas de solution de Business View équivalente à celle d'Instagram pour les Pages Facebook.

III - Particularités des requêtes de l'API Graph de Facebook

Si nous « requêtons » un compte avec un token possédant les autorisations associées :

- Requêtes à partir d'un Access token spécifique au compte Facebook (obtenable grâce à la manipulation décrite précédemment)

POSSIBLE

Les publications du compte (photos, miniatures de vidéo, état)
Les informations du compte Facebook (Photo de profil, username)

IMPOSSIBLE

Les publications vidéo
Les informations de compte sur les utilisateurs externes ayant publiés sur notre page

Chapitre 3 : TikTok

Contrairement à META, TikTok ne propose pas d'API permettant de récupérer des données librement. En effet, TikTok nous renvoie un code HTML et un script Javascript. (Proche d'un comportement IFrame)

Partie 1 : Fonctionnement de la WebPart

I - Service React

Le service React de la WebPart TikTok nous fournit, à l'aide de Axios, différentes données telles que le code HTML ou encore le lien vers le script Javascript en ligne de TikTok. En effectuant une requête avec un nom d'utilisateur TikTok, nous récupérons le code HTML et JS reprenant les caractéristiques du compte demandé. Ces données sont ensuite mises en forme dans une interface puis retournées.

II - Fonction React et WebPart

L'API de TikTok nous permet de récupérer que 2 types de données : les comptes TikTok et les publications TikTok. Pour utiliser le Javascript fourni par TikTok, nous devons passer par la fonction : « SPComponentLoader.loadScript(<https://www.tiktok.com/embed.js>) ». Également, pour rendre le HTML fourni par TikTok (au format string) reconnaissable en tant que code, nous devons utiliser la fonction :

« <div dangerouslySetInnerHTML={{__html : *(*Votre constante avec le code html de TikTok*)* }}/> »

Partie 2 : Utilisation de la WebPart

I - Utilisation de la WebPart existante

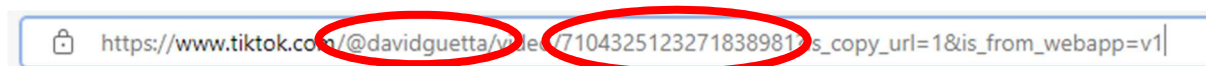
Les WebParts actuelles de TikTok (Compte et Publication) permettent de visualiser le compte et une publication du compte de test.

Chacune des WebPart possède des paramètres modifiables dans le property pane :

- Compte TikTok : en changeant de nom d'utilisateur, vous pourrez afficher n'importe quel compte TikTok
- Publication TikTok : en changeant l'ID du média (à trouver dans l'URL lorsque l'on navigue sur le site de TikTok), vous pourrez afficher n'importe quelle publication TikTok.

Pour changer les paramètres des WebParts, il faudra se rendre sur la version Web de TikTok :

- Pour changer de compte à afficher : aller sur la version web de TikTok et copier-coller le nom d'un compte
- Pour changer de publication TikTok : aller sur TikTok, copier le nom du compte propriétaire du post et copier le code du post dans L'url :



Chapitre 4 : LinkedIn

Partie 1 : Avancées et problèmes rencontrés

Parmi les réseaux sociaux abordés lors de ce projet, LinkedIn est le seul à nous avoir refusé l'accès à leur API.

Ainsi, les étapes pour obtenir un Access token et pour demander l'autorisation d'utiliser leurs produits sont les suivantes :

- Créer un compte LinkedIn
- Créer/Faire partie d'un Entreprise reconnue par LinkedIn
- Créer un compte et une application dans [LinkedIn Developers](#)
- Récupérer le Client ID et le Client Secret de votre application dans « Auth »



App de Test

Client ID: 78sszr1ftxjfbt | Created: Jul 5, 2022

Settings

Auth

Products

Analytics

Team members

Application credentials

Authentication keys

Client ID:

78sszr1ftxjfbt

Client Secret:

.....



- Toujours dans « Auth », ajouter dans « Authorized redirects URLs for your app » les URLs :
 - o <https://oauth.pstmn.io/v1/callback>
 - o <https://www.linkedin.com/developers/tools/oauth/redirect>

OAuth 2.0 settings

Token time to live duration

Access token: **2 months** (5184000 seconds)

Authorized redirect URLs for your app





[+ Add redirect URL](#)

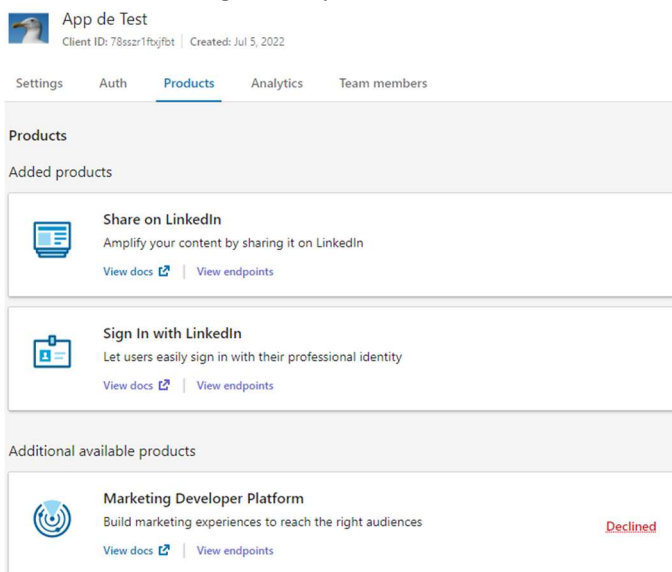
Cancel

Update

- Coller dans la barre URL de votre navigateur :
https://www.linkedin.com/oauth/v2/authorization?response_type=code&client_id={VOTRE_CLIENT_ID}&redirect_uri=https://oauth.pstmn.io/v1/callback&state=poiuytreza&scope=r_liteprofile%20r_emailaddress%20w_member_social
- Copier l'Access token dans l'URL :

https://oauth.pstmn.io/v1/callback?code=AQQtSEWzfZ6-0GTZczmoe0JORFTMqo8MacSjIEqzSXPEz7tEAbn7_FUgt0blw_4nA2ZI888MQnpWuGrMQi18Qssdlp498vLglYtip_mK5-WT8rs_gaPWTvdV5-uwAkOYPEAAUlc8P-HUk

- Dans l'onglet « product » de LinkedIn Developer, ajouter tous les produits et demander l'accès au Marketing Developer Platform



Si vous arrivez à obtenir l'autorisation de tous les produits, je vous invite à compléter cette documentation.