



Hackathon Ocupa2 (Big Data & Social Dev)

Indice

1. API de Instagram
 - a. Búsqueda por Hashtag
 - b. Extracción de Posts de un Hashtag
 - c. Obtener metadatos de un Post
 - d. Obtener metadatos de un Usuario
 - e. Dar / quitar like a un post
 - f. Dar / quitar follow a un usuario
2. API de Twitter
 - a. Búsqueda por Hashtag
 - b. Obtener metadatos de un tweet
 - c. Dar / Quitar like a un tweet
 - d. Dar / Quitar retweet a un tweet
 - e. Dar / Quitar follow a un usuario

Consideraciones

Las ids de los hashtags son únicas, es decir, cada hashtag tiene un campo id único que podría no ser el mismo entre APIs

API de Instagram

Primeros pasos

El primer paso es obtener una apikey para la api, para ello, se deberá realizar una consulta a la url:

Hackathon.ocupa2.com/instagram/get/key?email={email}

Donde {email} será el correo electrónico con el que os registrasteis para la hackathon

La respuesta tendrá el siguiente formato:

```
{
  status: CREATED/RETRIEVED
  key : apikey
  userid: userid
}
```

Búsqueda por hashtag

Para obtener la id de un hashtag (necesario para obtener los posts del hashtag) deberemos realizar una consulta a la url:

Hackathon.ocupa2.com/instagram/ig_hashtag_search?q={hashtag}&user_id={user_id}

Donde {hashtag} será el hashtag que queremos consultar (sin #) y user_id será nuestra id de desarrollador

En caso de respuesta positiva, la consulta nos devolverá la siguiente respuesta en formato JSON:

```
{
  "id": 1
}
```

Donde 1 será la id del hashtag encontrado.

Extracción de Posts de un hashtag

Para obtener posts de un hashtag, se han habilitado dos endpoints:

Recent_media: nos devolverá los últimos posts de un hashtag, deberemos realizar una consulta a la url:

`Hackathon.ocupa2.com/instagram/{hashtag_id}/recent_media?user_id={user_id}`

Donde {hashtag_id} será la id del hashtag a consultar y {user_id} será nuestra id de desarrollador

En caso de respuesta positiva, la consulta nos devolverá la siguiente respuesta en formato JSON:

```
{
  "data": [
    {
      "id": 64,
      "commentsCount": 0,
      "likeCount": 0,
      "mediaType": "CAROUSEL"
    },
    .
    .
    .
  ]
}
```

Este metodo devolverá todos los posts de un hashtag, por lo que "data" puede contener mas de una entrada.

Top_media: nos devolverá los posts mas vistos de un hashtag, deberemos realizar una consulta a la url:

Hackathon.ocupa2.com/instagram/{hashtag_id}/top_media?user_id={user_id}

Donde {hashtag_id} será la id del hashtag a consultar y {user_id} será nuestra id de desarrollador

En caso de respuesta positiva, la consulta nos devolverá la siguiente respuesta en formato JSON:

```
{
  "data": [
    {
      "id": 64,
      "commentsCount": 0,
      "likeCount": 0,
      "mediaType": "CAROUSEL"
    },
    .
    .
    .
  ]
}
```

Este metodo devolverá todos los posts de un hashtag, por lo que "data" puede contener mas de una entrada.

Obtener metadatos de un post

Para obtener los datos de un post, tendremos que realizar una consulta a la url:

Hackathon.ocupa2.com/instagram/media/{media}?fields={fields}

Donde {media} será el parámetro "id" de un post y {fields} será un string de los valores que queremos obtener, separado por comas.

En concreto, {fields} se podrá formar concatenando uno o mas de los siguientes términos:

- id
- comments_count

- like_count
- media_type
- username

Por ejemplo, para la consulta:

Hackathon.ocupa2.com/Instagram/media/6514?fields=id,media_type,like_count

Obtendremos la siguiente respuesta:

```
[
  {
    "id": 6514,
    "mediaType": "VIDEO",
    "likeCount": 0
  }
]
```

Obtener metadatos de un Usuario

Para obtener los datos de un usuario, tendremos que realizar una consulta a la url:

Hackathon.ocupa2.com/instagram/{user}?fields={fields}

Donde {user} será el parámetro "id" de un usuario y {fields} será un string de los valores que queremos obtener, separado por comas.

En concreto, {fields} se podrá formar concatenando uno o mas de los siguientes términos:

- id
- follower_count
- media_count
- username

Por ejemplo, para la consulta:

Hackathon.ocupa2.com/instagram/635?fields=id,follower_count,media_count,username

Obtendremos la siguiente respuesta:

```
[
  {
    "id": 35,
    "followerCount": 0,
    "mediaCount": 0,
    "username": "Chanel CLAY COMBS"
  }
]
```

Dar / Quitar like a un post

Para dar o quitar nuestro like a un post determinado, deberemos realizar una consulta a la siguiente url:

Hackathon.ocupa2.com/instagram/media/{media}/like?user_id{user_id}&action={action}

Donde:

- {media} será la id del post con el que queremos interactuar
- {user_id} será nuestra id de desarrollador
- {action} será la acción que queremos tomar sobre el post, puede ser like o unlike

Si enviamos la petición usando action=like, la API interpretará la consulta como una petición de dar like al post indicado (lo dará únicamente si el post no tiene ya uno desde nuestra cuenta). Si por el contrario usamos action=dislike, la API buscará si hemos dado like al post previamente y, en caso afirmativo, lo eliminará

En ambos casos, la respuesta que obtendremos tendrá la siguiente estructura:

```
{
  "status": "OK",
  "code": "CREATED"/"DELETED" (en función de action)
}
```

Dar / Quitar follow a un usuario

Para dar o quitar nuestro follow a un usuario determinado, deberemos realizar una consulta a la siguiente url:

Hackathon.ocupa2.com/instagram/{user}/follow?user_id{user_id}&action={action}

Donde:

- {user} será la id del user con el que queremos interactuar
- {user_id} será nuestra id de desarrollador
- {action} será la acción que queremos tomar sobre el user, puede ser follow o unfollow

Si enviamos la petición usando action=follow, la API interpretará la consulta como una petición de dar follow al usuario indicado (lo dará únicamente si el usuario no tiene ya uno desde nuestra cuenta). Si por el contrario usamos action=unfollow, la API buscará si hemos dado follow al usuario previamente y, en caso afirmativo, lo eliminará

En ambos casos, la respuesta que obtendremos tendrá la siguiente estructura:

```
{
  "status": "OK",
  "code": "CREATED"/"DELETED" (en función de action)
}
```

API de Twitter

Primeros pasos

En primer lugar, es necesario solicitar una apikey para interactuar con la api. Para ello, será necesario realizar una consulta a la siguiente url:

Hackathon.ocupa2.com/twitter/1.1/get/key?email={email}

Donde {email} será el correo electrónico con el que os registrasteis para la hackathon

La respuesta tendrá el siguiente formato:

```
{
  status: CREATED/RETRIEVED
  key : apikey
  userid: userid
}
```

Búsqueda por Hashtag

Para realizar una búsqueda por hashtag, será necesario realizar una consulta a la url:

`Hackathon.ocupa2.com/twitter/1.1/search/tweets.json?q={hashtag}`

Siendo {hashtag} el hashtag (sin #) que se quiere consultar.

Un ejemplo de respuesta sería:

```
[
  {
    "tweetId":8,
    "name":"Joy BENTON WAGNER",
    "screenName":"Joy BENTON WAGNER",
    "userid":1
  },
  .
  .
  .
]
```

Este metodo devolverá todos los posts de un hashtag, por lo que la respuesta puede contener mas de una entrada.

Obtener metadatos de un tweet

Para obtener todos los metadatos de un tweet, debemos realizar una consulta a la url:

`Hackathon.ocupa2.com/twitter/1.1/statuses/retweets/{tweet_id}.json`

Donde {tweet_id} será la id de un tweet obtenida previamente

Por ejemplo, para la consulta:

`http://hackathon.ocupa2.com/twitter/1.1/statuses/retweets/35.json`

obtenemos el resultado:

```
[{"id":35,"idStr":"35","likeCount":3,"retweetCount":3,"replyCount":0,"user":"1"}]
```


Dar / Quitar like a un tweet

Para dar like a un tweet, deberemos realizar una consulta a la siguiente url:

`http://hackathon.ocupa2.com/twitter/1.1/favorites/create.json?id={tweet_id}&bearer={apikey}`

donde {tweet_id} es el id del tweet al que queremos darle like y {apikey} es el parámetro apikey que recibimos del primer paso

por ejemplo, la consulta:

`http://hackathon.ocupa2.com/twitter/1.1/favorites/create.json?id=35&bearer=jc0Nrc1KesaOAqvEAifZ`

nos devolverá el siguiente resultado:

```
{"status":200,"msg":"CREATED"}
```

Para retirar un like, realizaremos una consulta a la siguiente url:

`http://hackathon.ocupa2.com/twitter/1.1/favorites/destroy.json?id={tweet_id}&bearer={apikey}`

donde {tweet_id} es el id del tweet al que queremos quitar el like y {apikey} es el parámetro apikey que recibimos del primer paso

nos devolverá el siguiente resultado:

```
{"status":200,"msg":"DELETED"}
```

Dar / Quitar retweet a un tweet

Para dar retweet a un tweet, realizaremos una consulta a la url:

`http://hackathon.ocupa2.com/twitter/1.1/statuses/retweet/{tweet_id}.json?bearer={apikey}`

donde {tweet_id} es el id del tweet al que queremos dar retweet y {apikey} es el parámetro apikey que recibimos del primer paso

por ejemplo, la consulta:

```
http://hackathon.ocupa2.com/twitter/1.1/statuses/retweet/35.json?bearer=jc0Nrc1KesaOAqvEAifZ
```

nos devolverá:

```
{"status":200,"msg":"CREATED"}
```

Para quitar un retweet, deberemos realizar una consulta a la url:

```
http://hackathon.ocupa2.com/twitter/1.1/statuses/unretweet/{tweet_id}.json?bearer={apikey}
```

donde {tweet_id} es el id del tweet al que queremos quitar el retweet y {apikey} es el parámetro apikey que recibimos del primer paso

por ejemplo, la consulta:

```
http://hackathon.ocupa2.com/twitter/1.1/statuses/unretweet/35.json?bearer=jc0Nrc1KesaOAqvEAifZ
```

nos devolverá:

```
{"status":200,"msg":"DELETED"}
```

Dar / Quitar follow a un usuario

Para dar follow a un usuario, realizaremos una consulta a la url:

```
http://hackathon.ocupa2.com/twitter/1.1/friendships/create.json?user_id={user_id}&bearer={apikey}
```

donde {user_id} es el id del usuario al que queremos seguir y {apikey} es el parámetro apikey que recibimos del primer paso

por ejemplo, la consulta:

```
http://hackathon.ocupa2.com/twitter/1.1/friendships/create.json?user_id=1&bearer=jc0Nrc1KesaOAqvEAifZ
```

OCUPA2

nos devolverá:

```
{"status":200,"msg":"CREATED"}
```

Para quitar un follow, deberemos realizar una consulta a la url:

```
http://hackathon.ocupa2.com/twitter/1.1/friendships/destroy.json?user_id={user_id}&bearer={apikey}
```

donde {user_id} es el id del usuario al que queremos quitar el follow y {apikey} es el parámetro apikey que recibimos del primer paso

por ejemplo, la consulta:

```
http://hackathon.ocupa2.com/twitter/1.1/friendships/create.json?user_id=1&bearer=jcONrc1KesaOAqvEAifZ
```

nos devolverá:

```
{"status":200,"msg":"DELETED"}
```