Discover New Streams - Streamforge

Discover new streams by watching top clips related to what the user is currently streaming.

Table of Contents

- Table of Contents
  - 1️⃣ 🏁 Quick start
    - Running locally for devolopment
      - 1. Adding Your API Keys.
      - 2. Starting Up The Server
      - 3. Build React Application
  - 2️⃣ Problem
  - 3️⃣ Solution
  - 4️⃣ Technical
  - 5️⃣ API Calls.
    - Twitch API Calls.
      - Get Streams
      - Get Clips

1️⃣ 🏁 Quick start

Running locally for devolopment

1. Adding Your API Keys.

Make a copy of the .env_example file, name it .env, and place your API keys inside.

```
# Make a copy of this file and name it .env
# Place your API keys in the placeholder.
TWITCH_CLIENT_ID = 'Place key here'
TWITCH_CLIENT_SECRET = 'Place key here'
```

2. Starting Up The Server

```
# Run command in the services directory.
npm run watch
```

3. Build React Application

```
# Run command in the client directory.
npm run watch
```

2 Problem

```
Problem: Finding a new Twitch streamer to watch is very difficult.
- Viewers rely on game categories, a thumbnail and the viewer count of a stream to determine if they want to watc
- Due to its nature, it is a very slow process of watching a streamer for 5-10 minutes to determine if they are v
- Overall this results in most viewers resorting to watching the very top streamers (something like top 1% have 9
```

3 Solution

```
Solution: Implement successful discovery features from social media platforms.
- Random clips are shown of streamers that are streaming right now
- If the viewer enjoys it, they can choose to be taken to the stream
- Otherwise they skip to the next clip and continue the discovery process
```

4 Technical

```
As for the technical side of things:
- Fetch a certain amount of live streams
- Fetch top clips from each channel (filter clips by clips of game they are streaming at the moment)
- Shuffle the list of clips and display them visually one by one, have link to take viewer to their twitch page
```

5 API Calls.

Twitch API Calls.

Making calls to the Twitch API.

Get Streams

Fetch current live streams.

```
export async function getStreams(req: Request, res: Response) {
  const { limit, pagination, cursor } = req.query;

  let url = `https://api.twitch.tv/helix/streams?first=${limit ? limit : 20}`;

  pagination && cursor ? (url += `&${pagination}=${cursor}`) : null;

  const options: AxiosRequestConfig = {
    url,
    method: "GET",
    headers: {
      Authorization: `Bearer ${accessEnv("APP_ACCESS_TOKEN")}`,
      "Client-ID": accessEnv("TWITCH_CLIENT_ID"),
      Accept: "application/vnd.twitchtv.v5+json",
    },
  };

  return axios(options)
    .then((response) => {
      res.json(response.data);
    })
    .catch((error) => {
      console.log(error);
    });
}
```

Get Clips

Fetch clips based on broadcaster ID.

```
export async function getClipsFromUser(req: Request, res: Response) {
  const { broadcasterID } = req.query;

  const url = `https://api.twitch.tv/helix/clips?broadcaster_id=${broadcasterID}`;

  const options: AxiosRequestConfig = {
    url,
    method: "GET",
    headers: {
      Authorization: `Bearer ${accessEnv("APP_ACCESS_TOKEN")}`,
      "Client-ID": accessEnv("TWITCH_CLIENT_ID"),
      Accept: "application/vnd.twitchtv.v5+json",
    },
  };

  return axios(options)
    .then((response) => {
      res.json(response.data);
    })
    .catch((error) => {
      console.log(error);
    });
}
```