**API / 페이지 코드랑 분리해서 작성**

API 가져오는 코드 (매개변수X)

|  |
| --- |
| import axios from 'axios';  export const fetchInfluencerData = async () => {    const response = await axios.get('http://3.34.90.217:8080/api/influencers/ranking');    console.log(response.data);      return response.data;  }; |

위 코드랑 연결되는 페이지 코드

사실 이 코드는 연동하기 전에 제가 페이지를 만들어놓았어서 제가 원래 만들던 변수에 보내주는 데이터를 매칭해서 연결했습니다. 챗봇만들때는 이런식으로 안하고 데이터에 따라 다르게 만들 것 같습니다. (아래참고)

|  |
| --- |
| import { fetchInfluencerData } from "../../api/influencersApi";    useEffect(() => {      const getData = async () => {        try {          const result = await fetchInfluencerData();          const transformed = result.map((item) => {            // 태그 처리: 문자열이면 ','로 나눠 배열로 만들고 trim            let tagList = [];            if (Array.isArray(item.tags)) {              tagList = item.tags.flatMap((tag) =>                tag.split(",").map((t) => t.trim())              );            } else if (typeof item.tags === "string") {              tagList = item.tags.split(",").map((t) => t.trim());            }            return {              name: item.name,              categories: Array.isArray(item.categories)                ? item.categories                : item.categories                  ? [item.categories]                  : [],              description: "-",              tags: tagList,              profileImage: getProfileImage(item.name),              // SNS별 값들              insta\_followers: item.instaFollowers,              insta\_averageLikes: item.instaAverageLikes,              insta\_averageViews: item.instaAverageViews,              insta\_scopeScore: item.instaFss,              insta\_id: item.instaName,              you\_followers: item.youFollowers,              you\_averageLikes: item.youAverageLikes,              you\_averageViews: item.youAverageViews,              you\_scopeScore: item.youFss,              you\_id: item.youName,              tik\_followers: item.tikFollowers,              tik\_averageLikes: item.tikAverageLikes,              tik\_averageViews: item.tikAverageViews,              tik\_scopeScore: item.tikFss,              tik\_id: item.tikName,            };          });          setInfluencers(transformed);          setFilteredInfluencers(transformed);        } catch (error) {          console.error("API 호출 실패:", error);        }      };      getData();    }, []); |

API 가져오는 코드 (id로 매개변수 전달할 때 – 인플루언서별로 상세 분석 페이지 보여줄 때 사용한 코드)

|  |
| --- |
| import axios from 'axios';  export const fetchAccountData = (id) => {    return axios.get(`http://3.34.90.217:8080/api/influencers/detail/${id}/analysis`).then(res => res.data);  }; |

위 코드랑 연결되는 페이지 코드

|  |
| --- |
| import { fetchAccountData } from "../../api/DetailApi.js";    useEffect(() => {      const fetchData = async () => {        try {          const data = await fetchAccountData(id);          // platformScores 변환          const convertedScores = {};          if (data.platformScores) {            for (const platform in data.platformScores) {              convertedScores[platform] = data.platformScores[platform].map(({ date, fss }) => ({                date,                scopeScore: fss,              }));            }          }          setAccountData(data);          setPlatformScores(convertedScores); // 변환된 데이터 저장        } catch (error) {          console.error("계정 분석 데이터 불러오기 실패:", error);        }      };      fetchData();    }, [id]);    if (!accountData) {      return <div>데이터 로딩 중...</div>;    } |