**업로드 사진의 주소를 다운**

1. putString으로 업로드하면 UploadTask를 반환함.
2. UploadTask는 완료하면 UploadTaskSnapshot을 반환(?) 하고, 그 안에 downloadURL이 있음
3. 하지만 downloadURL은 depreciated 임. reference 안에 getDownloadURL을 사용해야함
4. 따라서 UploadTaskSnapshot안에 ref안에 getDownloadURL로 주소를 받음

const attachmentUrl = await response.ref.getDownloadURL();

**nweet에 같이 넣어주어 firestore에 보내기**

1. 주소를 받았으므로 주소까지 포함하여 nweet를 만들고 nweet등록

const nweetObj = {

text: nweet,

createdAt : Date.now(),

creatorId : userObj.uid,

attachmentUrl,

};

await dbService.collection(‘nweets’).add(nweetObj);

setAttachment(null);

1. 이때 사진을 등록하지 않고 업로드하는 사람도 있으므로, 잘 처리해주자. 다음은 완성된 onSubmit

const onSubmit = async (event) => {

        event.preventDefault();

        let attachmentUrl = '';

        if (attachment !== '') {

            const attachmentRef = storageService

                .ref()

                .child(`${userObj.uid}/${uuidv4()}`);

            const response = await attachmentRef.putString(

                attachment,

                'data\_url'

            );

            attachmentUrl = await response.ref.getDownloadURL();

        }

        const nweetObj = {

            text: nweet,

            createdAt: Date.now(),

            creatorId: userObj.uid,

            attachmentUrl,

        };

        await dbService.collection('nweets').add(nweetObj);

        setNweet('');

        setAttachment(null);

    };

**불러와서 사진까지 보여주기**

1. Home.js 내부 useEffect 에서 불러울 때 attachmentUrl도 nweets에 들어있고, 이걸 Nweet.js 컴포넌트에서 nweetObj로 받음
2. Nweet.js에서 attachmentUrl이 있으면 사진을 보여주는 태그를 적절히 선언

<h4>{nweetObj.text}</h4>

{nweetObj.attachmentUrl && (

         <img

            src={nweetObj.attachmentUrl}

            width="50px"

            height="50px"

            alt="error"

         />

)}