**사진이 있으면 URL을 받아서 nweet에 넣고 업로드하는 챕터**

1. storage 를 쓰기위해 먼저 firebase.storage()를 import 해야함

<https://firebase.google.com/docs/reference/js/firebase.storage.Storage>

* myFirebase.js에서 export하여 받아오는 식으로 하자

//myFirebase.js

import ‘firebase/storage’;

…

export storageService = firebase.storage();

//home.js

import { dbService, storageService } from “myFirebase”;

1. storage의 buket의 이름과 그 안의 폴더 안의 파일을 참조하기 위해 ref()와 ref().child()를 사용

* ref()는 버켓의 상대주소를 반환함

https://firebase.google.com/docs/reference/js/firebase.storage.Storage#ref

* ref().child()는 그 안의 내용의 상대주소를 반환함

<https://firebase.google.com/docs/reference/js/firebase.storage.Reference#child>

* 현재 buket은 하나이니 이름을 넣어줄 필요는 없고, child에는 userObj안의 uid를 폴더로 submit으로 생성하는 것이니, 파일이름은 무작위로 넣어주자.
* 이때 랜덤 파일이름을 생성하는 함수도 있지만, 그냥 npm install uuid 하고

import { v4 as uuidv4 } from ‘uuid’; 하고 밑처럼 사용하면 랜덤 이름을 생성해준다.

//home.js

const onSubmit = async (event) => {

event.preventDefault();

const fileRef = storageService.ref().child(`${userObj.uid}/${uuidv4()}`);

}

1. 업로드를 위해 Reference의 put 또는 putString을 사용하자

* <https://firebase.google.com/docs/reference/js/firebase.storage.Reference#putstring>
* 아무거나 써도 상관없으나, 우린 이미 attachment에 파일 url을 가지고 있으므로 putString을 사용하자

//home.js

const onSubmit = async (event) => {

event.preventDefault();

const fileRef = storageService.ref().child(`${userObj.uid}/${uuidv4()}`);

const response = await fileRef.putString(attahment, ‘data\_url’);

}