

## S3

---

- 로그인 후 Amazon S3 > 버킷 > gumi-d209에서 진행
- 객체
  - 사용할 파일을 업로드
  - 음성 파일 및 음악(.mp3), 포스터(.jpg), 가사(.lrc) 저장하고 스프링에서링크로 불러와서 사용
- 권한
  - 버킷 정책

```
{
  "Version": "2012-10-17",
  "Id": "Policy1667203862309",
  "Statement": [
    {
      "Sid": "Stmt1667203510202",
      "Effect": "Allow",
      "Principal": "*",
      "Action": "s3:*",
      "Resource": "arn:aws:s3:::gumi-d209"
    }
  ]
}
```

- ACL(엑세스 제어 목록)
  - 버킷 소유자(AWS 계정)
    - 객체 : 나열, 쓰기 허용
    - 버킷 ACL : 읽기, 쓰기 허용
  - 외부 계정
    - 객체 : 나열, 쓰기 허용
    - 버킷 ACL : 읽기, 쓰기 허용
  - 모든 사람(퍼블릭 액세스)
    - 객체 : 나열 허용
    - 버킷 ACL : 읽기 허용
- CORS 설정

```
[
  {
    "AllowedHeaders": [
      "*"
    ]
  }
]
```

```

    ],
    "AllowedMethods": [
      "HEAD",
      "GET",
      "PUT",
      "POST",
      "DELETE"
    ],
    "AllowedOrigins": [
      "https://k7d209.p.ssafy.io"
    ],
    "ExposeHeaders": [
      "ETag",
      "x-amz-meta-custom-header"
    ]
  }
]

```

## 소셜 로그인(KaKao)

---

- 카카오 디벨로퍼 설정
  - <https://developers.kakao.com/> 에서 로그인
  - 내 애플리케이션에서 서비스 추가
  - 발급된 REST API키 사용
  - 카카오 로그인으로 들어가는 Redirect URI 설정(로컬, 서버 전부 등록)
  - 로그인 시 동의항목에 들어가는 동의 내역 설정(필수 동의/선택동의/사용 안함)
  - Web에 사이트 도메인 등록

## 날씨

---

- OpenWeather API 이용(<https://openweathermap.org/>)

```

/**
 * AxiosInstance 캡슐화
 * @author 복성범
 */

import axios from "axios";
import { useEffect, useState } from "react"
import "./Nowlocation.scss";

const Nowlocation = (props) => {

  const [today, setToday] = useState("");

  const currentTimer = () => {
    const date = new Date();

```

```

const years = String(date.getFullYear()).padStart(2, "0");
const monthes = date.getMonth() + 1;
const daies = String(date.getDate()).padStart(2, "0");
const week = [
  'Sunday, ',
  'Monday, ',
  'TuesDay, ',
  'WendsDay, ',
  'ThursDay, ',
  'FriDay, ',
  'SaturDay, '
];
const dayOfWeek = week[date.getDay()];

setToday(`${dayOfWeek}${daies} ${monthes} ${years}`)
// props.setToday(`${years}-${monthes}-${daies}`)
}

const startTimer = () => {
  setInterval(currentTimer, 1000)
}

startTimer()

const [geoReady, setGeoReady] = useState(false)
const [tempCelcius, setTempCelcius] = useState()
const [city, setCity] = useState("")
const [description, setDescription] = useState("")
const [icon, setIcon] = useState("")

const [state, setState] = useState({
  center: {
    lat: 33.450701,
    lng: 126.570667,
  },
  errMsg: null,
  isLoading: true,
})

const _callApi = async () => {
  try {
    let res = await axios
      .get(
        `https://api.openweathermap.org/data/2.5/weather?
lat=${state.center.lat}&lon=${state.center.lng}&appid=${process.env.REACT_APP_WEAT
ER_API}` ,
        {},
      )
    .then(response => {
      // console.log(response.data)
      // console.log(response.data.name)
      // console.log(response.data.main.temp)
      // console.log(response.data.weather[0].icon)
      // console.log(response.data.weather[0].description)

```

```

        setCity(response.data.name)
        setTempCelcius((response.data.main.temp - 273.15).toFixed(1))
        setDescription(response.data.weather[0].description)
        props.setDiaryWeather(response.data.weather[0].description)
        setIcon(response.data.weather[0].icon)
      });
    } catch (error) {
      console.log(error.message);
    }
  };

  useEffect(() => {
    if (geoReady) {
      _callApi();
    }
  }, [state])

  useEffect(() => {
    if (navigator.geolocation) {
      // GeoLocation을 이용해서 접속 위치를 얻어옵니다
      navigator.geolocation.getCurrentPosition(
        (position) => {
          setState((prev) => ({
            ...prev,
            center: {
              lat: position.coords.latitude, // 위도
              lng: position.coords.longitude, // 경도
            },
            isLoading: false,
          }))
          setGeoReady(true);
        },
        (err) => {
          setState((prev) => ({
            ...prev,
            errMsg: err.message,
            isLoading: false,
          }))
        }
      )
    } else {
      // HTML5의 GeoLocation을 사용할 수 없을때 마커 표시 위치와 인포윈도우 내용을 설정합니다
      setState((prev) => ({
        ...prev,
        errMsg: "geolocation을 사용할수 없어요..",
        isLoading: false,
      }))
    }
  }, [])
  return (
    <div className="now-inform">
      <div className="date">
        {today}

```

```
        { /* Monday, 20 Aug 2018 */ }
    </div>

    <div className="city">
        {city}
    </div>
    <div className="temp">
        {tempCelcius}&deg;
    </div>

    <div className="right-panel panel">
        <img src={`https://openweathermap.org/img/w/${icon}.png`} alt=""
width="160" />
    </div>

    </div>
    );
}

export default Nowlocation;
```

## 클로바(<https://auth.ncloud.com/>)

---

- 네이버 클로바 서비스로 프로젝트 질문 리스트들 음성으로 변환
  - 클로바 로그인
  - 콘솔 → CLOVA Dubbing → Project Group → 실행
  - 프로젝트 추가 → 2022\_2학기\_D209 생성
  - 보이스 고음 설정 → 필요한 질문 TTS로 변환