



# cloudinary

서버 사용하지 않고 storage에 따로 이미지 파일을 저장하고자 함  
사용자가 찍어서 업로드한 사진을 서버로 전송해야 하는 경우,  
사용자가 이미지를 업로드하면 URL로 변환해줘서 DB에는 사진의 URL만을 저장하는 방법  
→ 클라우드너리 `자동으로 customizing`



## 개념

‘서버 사용하지 않고 storage에 따로 이미지 파일을 저장’



## Flow



## 클라우드너리를 사용하여 REST API 방식의 이미지 저장하기

- ☐ Cloudinary로 이미지 업로드
- ☐ Cloudinary의 이미지의 path를 프론트에 전달
- ☐ 이미지 이름을 Unique하게 하기

### ▼ 1. 프리셋 생성

- Settings > Upload > Add upload preset

Upload presets:

Unsigned uploading enabled

Name	Mode	Settings
lxqyrq5	Unsigned	Unique filename: <b>true</b> Delivery type: <b>upload</b> Access mode: <b>public</b> Auto tagging: <b>0.7</b> <a href="#">Edit</a> <a href="#">Duplicate</a>
thuz5nnh	Unsigned	Use filename or externally defined Public ID: <b>true</b> Unique filename: <b>true</b> Delivery type: <b>upload</b> Access mode: <b>public</b> Auto tagging: <b>0.7</b> <a href="#">Edit</a> <a href="#">Duplicate</a>
ml_default	Signed	Use filename or externally defined Public ID: <b>true</b> Unique filename: <b>true</b> Delivery type: <b>upload</b> Access mode: <b>public</b> Auto tagging: <b>0.7</b> <a href="#">Edit</a> <a href="#">Duplicate</a>

[Add upload preset](#)

Upload presets allow you to define the default behavior for your uploads, instead of receiving these as parameters during the upload request itself. Parameters can include tags, incoming or on-demand transformations, notification URL, and more. Upload presets have precedence over client-side upload parameters.

- signing Mode를 unsigned로 체크하고 save 클릭

[Save](#)

Upload preset name

epf7zmr

The upload preset name must be unique in your account or sub-account. This value is passed as the 'upload\_preset' parameter when calling the upload API.

Signing Mode:

Set to 'Unsigned' to enable unsigned uploads. Set to 'Signed' to require signed uploads.

Signed  
Signed  
Unsigned

Folder

As outlined in the folder path, this will be appended to the Public ID. This can be useful for creating

- 생성된 프리셋의 이름을 기억

Upload presets:

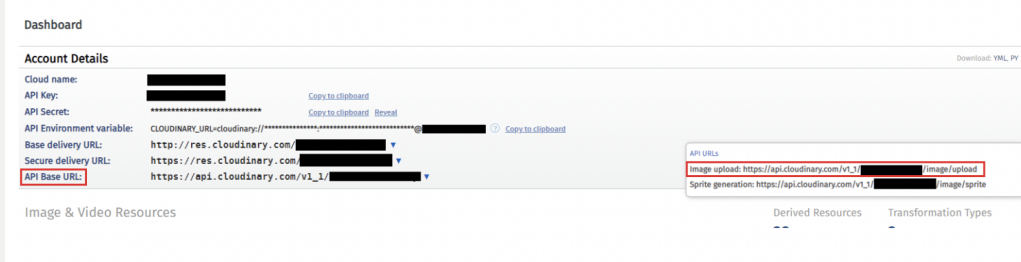
Unsigned uploading enabled

Name	Mode	Settings	
pnzjemgn	Unsigned	Unique filename: true Delivery type: upload Access mode: public Auto tagging: 0.7	<a href="#">Edit</a> <a href="#">Duplicate</a>

## ▼ 2. 클라우드 URL 확인

- 메인으로 가서 Account Details을 확인
- Rest API를 사용하여 이미지를 저장할 것이므로 여기서 API key와 API Base URL을 확인
- 이미지 업로드용 API URL은 아래와 같은 형식으로 되어있다.

[https://api.cloudinary.com/v1\\_1/\[클라우드 이름\]/image/upload](https://api.cloudinary.com/v1_1/[클라우드 이름]/image/upload)



## ▼ 3. 클라우드에 이미지 업로드 하기

☐ 여러 개의 파일을 업로드하는 예제 코드

```
const form = document.querySelector("form");

form.addEventListener("submit", (e) => {
  e.preventDefault();

  const files = document.querySelector("[type=file]").files;
  const formData = new FormData();

  for (let i = 0; i < files.length; i++) {
    let file = files[i];
    formData.append("file", file);
    formData.append("upload_preset", "docs_upload_example_us_preset");
  }

  fetch(url, {
    method: "POST",
    body: formData
  })
    .then((response) => {
      return response.text();
    })
  })
```

```

    .then((data) => {
      document.getElementById("data").innerHTML += data;
    });
  }
});

```

사진 업로드는 **POST** 메서드를 사용

요청 URL 은 `https://api.cloudinary.com/v1_1/<cloud name>/<resource_type>/upload`

- **cloud name**: 대쉬보드에서 확인할 수 있는 내 클라우드 이름
- **resource\_type** : image
- **file** 실제로 업로드할 파일
- **upload\_preset** settings 페이지에서 등록했던 **preset 설정의 이름**
- 쿼리 셀렉터를 이용해서 업로드한 파일을 files 변수에 할당
- 반복문을 통해 `files` 배열의 각각의 요소(각각의 파일) 들을 new 연산자로 생성된 `FormData` 객체 에 `append` 메서드를 통해 `file` 프로퍼티의 값 으로 추가
- **upload\_preset** 도 직접 입력하는 방식으로 추가했고, `file` 과 `upload_preset` 프로퍼티 를 가지고 있는 `FormData` 객체 를 fetch를 이용하여 post 메서드의 `body` 에 넣어 전송 하고 그 결과값인 `response` 값을 출력

## ☐ imageUploader 함수

```

const imageUploader = async (file) => {
  // FormData 생성
  const data = new FormData();

  // FormData 에 file, upload_preset 추가
  data.append("file", file);
  data.append("upload_preset", "내 preset Name");

  // .env 파일에 있는 cloudName 가져오기
  const cloudName = import.meta.env.VITE_CLOUD_NAME;

  // body에 FormData 객체를 넣어 전송
  const res = await fetch(`https://api.cloudinary.com/v1_1/${cloudName}/upload`, {
    method: "POST",
    body: data,
  });

  // fetch로 받아온 값은 .json() 메서드를 이용해야 사용 가능. 프로미스 반환
  return res.json();
};

```

`imageUploader` 함수의 반환값 → image state 값을 url로 변경해 준다.

## ☐ 업로드한 사진을 파라미터로 전달하는 로직

```
// 초기값은 null, input 태그를 연결해야하니 타입은 HTMLInputElement 지정
const inputRef = useRef<HTMLInputElement>(null);

return (
  // input에 ref 로 inputRef 연결, 타입과 업로드할 파일 제한 설정
  <input ref={inputRef} type="file" accept="image/*">
)

```

- useRef로 사진을 업로드할 input 태그를 연결

□ 버튼을 생성해서 버튼을 클릭하면 input이 클릭되는 것과 같은 효과

```
const inputRef = useRef<HTMLInputElement>(null);

// input.current.click = ref로 연결된 요소 클릭 이벤트 실행
const imageUploadBtn = () => {
  inputRef.current.click();
};

return (
  <input ref={inputRef} type="file" accept="image/*">
    // 버튼에 input이 클릭되는 효과를 주는 이벤트 연결
    <button onClick={imageUploadBtn}>이미지 업로드</button>
  )

```

□ input에 업로드 된 파일을 `imageUploader` 로 전달하는 `onChange` 이벤트

```
const inputRef = useRef<HTMLInputElement>(null);

// input.current.click = ref로 연결된 요소 클릭 이벤트 실행
const imageUploadBtn = () => {
  inputRef.current.click();
};

const fileChange = async (e) => {
  const uploaded = await imageUploader(e.target.files[0]);
  setImage(uploaded.url);
};

return (
  <input ref={inputRef} type="file" accept="image/*" onChange={fileChange}>
    // 버튼에 input이 클릭되는 효과를 주는 이벤트 연결
    <button onClick={imageUploadBtn}>이미지 업로드</button>
  )

```

위 코드 결과

image state가 빈 문자열이면 대표 사진을 업로드 해주세요라는 p 태그를 출력하고, image 값이 url로 변경되면 Box의 전달된 url로 바로 이미지를 출력

```
{asset_id: 'bb246671190c96f76ad98ea9aa9dc310', public_id: 'qehgylpmmwbn4uoi1zo', version: 1653644379, version_id: '570b9173e53cc7c90b9a7f2792466ee5', signature: '38e461119495639ddcf042d1194c5177493e7853', ...}
  access_mode: "public"
  asset_id: "bb246671190c96f76ad98ea9aa9dc310"
  bytes: 10163
  created_at: "2022-05-27T09:39:39Z"
  etag: "883c726beb699f98b14aa8c1e63a15f"
  format: "png"
  height: 500
  original_filename: "home_icon"
  placeholder: false
  public_id: "qehgylpmmwbn4uoi1zo"
  resource_type: "image"
  secure_url: "https://res.cloudinary.com/deiyompy0/image/upload/v1653644379/qehgylpmmwbn4uoi1zo/qehgylpmmwbn4uoi1zo.png"
  signature: "38e461119495639ddcf042d1194c5177493e7853"
  tags: []
  type: "upload"
  url: "http://res.cloudinary.com/deiyompy0/image/upload/v1653644379/qehgylpmmwbn4uoi1zo.png"
  version: 1653644379
  version_id: "570b9173e53cc7c90b9a7f2792466ee5"
  width: 500
  [[Prototype]]: Object
```

db에 저장 : public\_id , version 정보

2순위

시간이 남으면 사진 변환

간단하게 파일 경로