

[FE] Cropper.js

[FE] Cropper.js

1. Cropper.js
2. 기본 사용법 (깃헙 내용)
3. jubging 내 사용 (프로필 사진 변경)

1. Cropper.js

1. 공식

- 홈 <https://fengyuanchen.github.io/cropperjs/>
- 깃헙 <https://github.com/fengyuanchen/cropperjs/blob/main/README.md>
- 예제 코드 <https://github.com/fengyuanchen/cropperjs/tree/main/docs/examples>

2. 참고

- Cropper.js 사용기 <https://velog.io/@eunoia/Cropper.js-%EC%82%AC%EC%9A%A9%EA%B8%B0>
- cropper.js를 사용해 업로드한 이미지 자르기 <https://soa-memo.tistory.com/22>

2. 기본 사용법 (깃헙 내용)

1. Installation

```
// npm 설치
$ npm install cropperjs

// import 후 사용, cdn 링크는 https://cdnjs.com/libraries/cropperjs 이쪽에
<link href="/path/to/cropper.css" rel="stylesheet">
<script src="/path/to/cropper.js"></script>
```

2. Usage

```
<div>
  
</div>
```

```
/* Ensure the size of the image fit the container perfectly */
img {
  display: block;

  /* This rule is very important, please don't ignore this */
  max-width: 100%;
}
```

```
// import 'cropperjs/dist/cropper.css';
import Cropper from 'cropperjs';

const image = document.getElementById('image');
```

```
const cropper = new Cropper(image, {
  aspectRatio: 16 / 9,
  crop(event) {
    console.log(event.detail.x);
    console.log(event.detail.y);
    console.log(event.detail.width);
    console.log(event.detail.height);
    console.log(event.detail.rotate);
    console.log(event.detail.scaleX);
    console.log(event.detail.scaleY);
  },
});
```

3. Usage (getCroppedCanvas): 크롭된 이미지를 가져오는 메서드

- <https://github.com/fengyuanchen/cropperjs/blob/main/README.md#getcroppedcanvasoptions>

```
// 1. 기본 형태
cropper.getCroppedCanvas();

// 2. 옵션을 준 형태
cropper.getCroppedCanvas({
  width: 160,
  height: 90,
  minWidth: 256,
  minHeight: 256,
  maxWidth: 4096,
  maxHeight: 4096,
  fillColor: '#fff',
  imageSmoothingEnabled: false,
  imageSmoothingQuality: 'high',
});

// 3. 크롭된 이미지를 얻어서 jQuery.ajax 보내는 형태
// 브라우저가 HTMLCanvasElement.toBlob를 지원하면 크롭된 이미지를 서버에 업로드할 수 있다.
// toBlob의 두번째 parameter(default)는 image/png 이고, 필요하면 바꾸어라.
cropper.getCroppedCanvas().toBlob((blob) => {
  const formData = new FormData();

  // 세번째 parameter로 파일 이름을 지정할 수 있다.
  formData.append('croppedImage', blob/*, 'example.png' */);

  // 예제에서는 jQuery.ajax를 사용한다.
  $.ajax('/path/to/upload', {
    method: 'POST',
    data: formData,
    processData: false,
    contentType: false,
    success() {
      console.log('Upload success');
    },
    error() {
      console.log('Upload error');
    },
  });
});
```

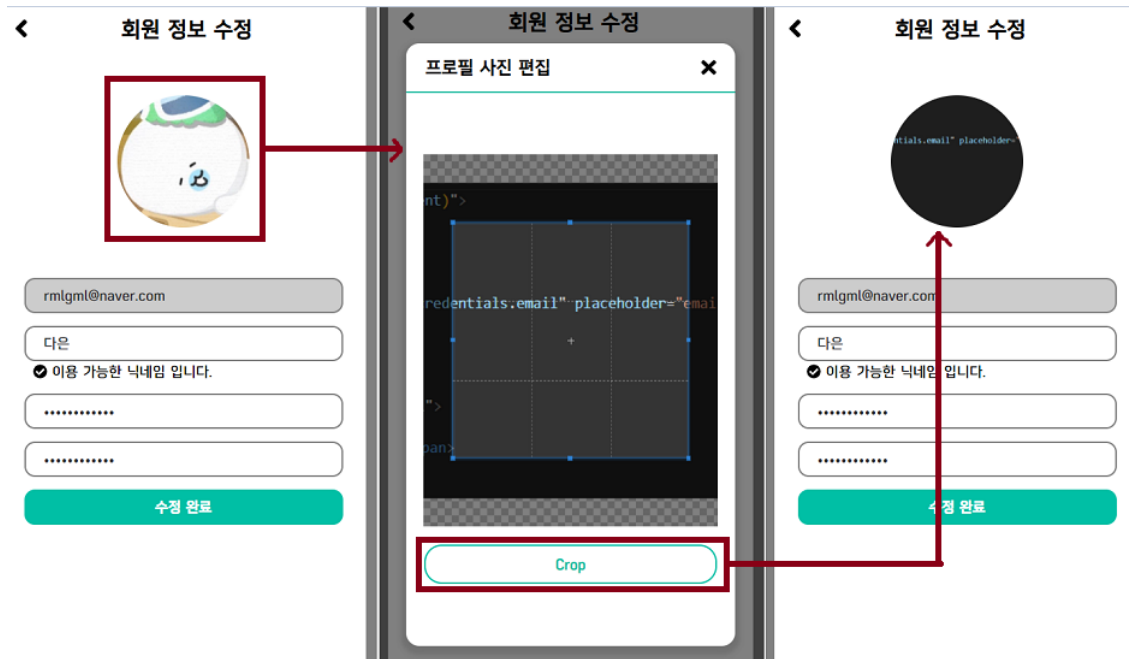
```
}/*, 'image/png' */);
```

3. jubging 내 사용 (프로필 사진 변경)

1. 사용 예제 코드

- crop-a-round-image.html <https://github.com/fengyuanchen/cropperjs/blob/main/docs/examples/crop-a-round-image.html>
- 2-3의 getCroppedCanvas <https://github.com/fengyuanchen/cropperjs/blob/main/README.md#getcroppedcanvasoptions>

2. 결과



3. jubging/user/ChangeSetting.vue

```
// template input 부분
<label class="label">
  
  <input type="file" class="sr-only" id="input" name="image"
  accept="image/*" @change="inputChange">
</label>

// template modal 부분
<ModalView v-show="isModalViewed" @close-modal="modalOff" :modalTitle="'프로
필 사진 편집'">
  <div class="img-container">
    
    <button type="button" id="button" @click="crop"
    class="btn">Crop</button>
  </div>
</ModalView>

// flow 요약
(1단계) template input에 입력이 들어오면 @change="inputChange"가 실행
→ (2단계) inputChange 내에서 modalOn 실행 → modalOn 내에서 cropper 객체 생성
```

→ (3단계) 원하는 대로 영역 잡은 후 `template modal`의 `button`을 눌러 `@click="crop"`가 실행

→ (4단계) `getCroppedCanvas`로 크롭된 이미지를 얻을 수 있음

(→ (5단계) 이후 회원정보 변경 버튼을 통해 `axios` 요청 + `s3` 저장)

4. (1단계) `@change="inputChange"`

```
inputChange(event) {
  var image = document.getElementById('image'); // id=image 인 element (3-2의
  // 모달 내 이미지)
  var input = document.getElementById('input'); // id=input 인 element (3-2의
  // input 태그)
  var files = event.target.files; // input으로 들어온 파일

  if (files && files.length > 0) {
    var file = files[0]; // input 으로 들어온 파일이 리스트 형태라 [0]

    // input.value = '' : input의 값 초기화
    // image.src = ~ : image의 src 값 변경 (모달 내 이미지 변경)
    // this.modalOn() : modalOn 함수 호출
    if (URL) {
      input.value = '';
      image.src = URL.createObjectURL(file);
      this.modalOn()
    } else if (FileReader) {
      var reader = new FileReader();
      reader.onload = function (event) {
        input.value = '';
        image.src = event.target.result;
        this.modalOn()
      };
      reader.readAsDataURL(file);
    }
  }
},

// else if (FileReader) 코드는 이미지 업로드 미리보기 구현 이런 식으로 검색하면 더
// 나옴
// https://taewooblog.tistory.com/6
// http://yoonbumtae.com/?p=3304
```

5. (2단계) modal on/off

```
modalOn() {
  // 모달 켜기
  this.isModalViewed = true

  // input 태그로 받은 image를 가져와서 cropper 첫번째 인자로 넣고 객체를 만듦
  var image = document.getElementById('image');
  this.cropper = new Cropper(image, {
    aspectRatio: 1,
    viewMode: 1,
  });
},

modalOff() {
```

```
// 모달 끄기
this.isModalViewed = false

// 기존에 만든 객체가(이미지나 크롭하던 내역이) 남아 있으면 안되니까 destroy
this.cropper.destroy();
this.cropper = null;
},
```

6. (3단계) @click="crop" + (4단계) getCroppedCanvas

```
crop() {
  // id=avatar 인 element (3-2의 프로필 사진 img 태그)
  var avatar = document.getElementById('avatar');

  // this.croppedCanvas에 getCroppedCanvas를 이용하여 크롭된 이미지 저장
  if (this.cropper) {
    this.croppedCanvas = this.cropper.getCroppedCanvas({
      width: 150,
      height: 150,
    });
    // toDataURL로 src 변경 (폴필 사진 img 태그가 크롭된 이미지로 변경됨)
    avatar.src = this.croppedCanvas.toDataURL();
  }

  // modalOff 함수 호출
  this.modalOff()
},
```

7. (5단계) axios 요청

```
changeSettingAndPic() {
  // toBlob(function (blob) {...}) 때문에 {} 내에서 this가 가리키는 객체가 달라
  self로 저장해 둬
  let self = this

  // changeSettingAndPic 함수가 호출됐을 때 this.croppedCanvas(초기값 '')가 있으면 실행
  if (this.croppedCanvas) {
    // 2-3의 예제 코드 참고 blob 형식으로 변환 후 axios
    this.croppedCanvas.toBlob(function (blob) {
      var form = new FormData();

      // 세번째 인자 파일이름.png를 써야 png 형식으로 저장됨. 안 쓰면 알 수 없는 확장자 + 여러장 보낼 땐 아래 참고
      // 첫번째 인자는 백에서 요청받는 key, 두번째는 value. 이름 맞춰줘야 함.
      form.append('file', blob, 'profile.png');

      // 헤더는 예제 코드 참고대로 작성했음
      // 백에 /images 경로로 요청 보내면 s3에 저장 후 url 링크를 반환해준다.
      HTTP.post('/images', form, {
        header: {
          'processData' : false,
          'Content-Type' : false,
        },
      })
    }).then(res => {
```

```

        // 반환받은 s3 url 경로를 self.credentials(유저)의 프로필 사진 경로에 입력
        self.credentials.profilePath = res.data.data
    })
    .then(() => {
        // 이미지 변경 외에 다른 닉네임, 비번 변경하는 함수
        self.$store.dispatch('changeSetting', self.credentials)
    })
    .catch(err => {
        console.log(err)
    })
});
} else {
    // 여기는 this.croppedCanvas가 없을 때 (폴필 사진 변경x 일 때) 실행
    // 이미지 변경 외에 다른 닉네임, 비번 변경하는 함수
    this.$store.dispatch('changeSetting', this.credentials)
}
},

```

```

// views/jubging/NewArticle.vue
for (let i=0; i<this.canvasList.length; i++){
    this.canvasList[i].toBlob(function (blob) {
        var form = new FormData();

        // for문으로 여러장 axios보낼 때는 각각 파일이름을 다르게 해줘야한다.
        form.append('file', blob, i+".png");

        HTTP.post('/images', form)
        .then(res => {
            ...
        })
        .catch(err => {
            ...
        })
    })
}
}

```