Ckeditor5

- description : Ckediotr5 관련 내용 기술
- author : slaptear
- email : sgjang@repia.com
- lastupdate : 2023-03-02

적용 방법

- CKEditor5는 5가지의 형태가 있지만 Classic editor와 Document editor 2가지의 사용 방법이 다릅니다.
- 사용할 editor를 선택하고 그 후에 적용할 Ckeditor5 CDN Script를 head 태그 안에 삽입
- 또는 CKEditor5 온라인빌드사이트에서 다운로드하여 활용 (아래 Ref 참고)

```
<script
src="https://cdn.ckeditor.com/ckeditor5/36.0.1/decoupled-document/ckeditor.j
s"></script>
```

Classic editor 적용 예시

- editor는 div 또는 textarea 사용

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>CKEditor 5 – Classic editor</title>
    <script
src="https://cdn.ckeditor.com/ckeditor5/36.0.1/classic/ckeditor.js"></script
>
</head>
<body>
    <h1>Classic editor</h1>
    <div id="editor">  //
        <p>This is some sample content.</p>
    </div>
    <script>
        ClassicEditor
            .create( document.querySelector( '#editor' ) )
            .catch( error => {
                console.error( error );
            } );
    </script>
</body>
</html>
```

Document editor 적용 예시

- toolbar 부분과 editor 부분이 나뉨
- editor와 toolbar 부분은 div 사용

```
 <!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>CKEditor 5 – Document editor</title>
    <script
src="https://cdn.ckeditor.com/ckeditor5/36.0.1/decoupled-document/ckeditor.j
s"></script>
</head>
<body>
    <h1>Document editor</h1>
    <!-- editor의 toolbar container-->
    <div id="toolbar-container"></div>

    <!-- editor 부분-->
    <div id="editor">
        <p>This is the initial editor content.</p>
    </div>

    <script>
        DecoupledEditor
            .create( document.querySelector( '#editor' ) )
            .then( editor => {
                const toolbarContainer = document.querySelector( '#toolbar-
container' );

                toolbarContainer.appendChild( editor.ui.view.toolbar.element
);
            } )
            .catch( error => {
                console.error( error );
            } );
    </script>
</body>
</html>
```

Tip

- get,setData()
  - CKEditor5는 getData()와 setData()로 값을 가져오거나 값을 삽입할 수 있다.
  - 여기서 주의 사항은 아래를 참고
  - <예시>

```
 <script>
```

```
  let testEditor;                          => 사용할 editor의 변수를 선언
 // 생성
  ClassicEditor
        .create(document.querySelector( '#editor' ))
        .then( editor => {
              testEditor= editor;      => 해당 editor를 변수에 적용
         })
        .catch( error => {
          console.error( error );
         });
// 값 가져오기
 testEditor.getData();                    => 변수에 적용 후 사용해야한다.
// 값 세팅
 testEditor.setData('<p>example</p>'); => 변수에 적용 후 사용해야한다.
</script>
```

이미지 업로드

- CKEditor5는 이미지 업로드 다양한 플러그인이 존재하지만 대부분 유료 플러그인이다.
- 그래서 Adapter부분을 새로 만들어야 한다.

UploadAdapter 예시

- Custom image upload adapter 공식 문서
- 아래 사이트를 참고
  - CKEditor5 이미지 업로드
  - CKEditor5 이미지 업로드 기능 정리
- 주의사항
  - 이미지 및 file의 데이터는 FormData 변경 후 controller에 보내야 함.

Controller 예시

- url로 addObject 하는 이유
  - 이미지가 업로드가 되고 저장 경로가 base64로 인코딩되어 img 태그 안의 src부분에 들어간다.
  - editor를 활용하여 등록할 경우 데이터베이스에 p 태그와 img 태그가 함께 저장이 되어 불러올 때 로딩시간이 길어진다.

```
@PostMapping(value = "url부분")
public ModelAndView imageUpload (MultipartHttpServletRequest request) throws
Exception {

    String currContextPath = request.getContextPath();
    ModelAndView mv = new ModelAndView("jsonView");
    String savePath= null;         //저장경로
    String originalImagename= null;    //원본이름
    String imageName= null;         //저장본이름
    String extension= null;         //확장자
```

```
    String realPathtoUploads =
EgovProperties.getProperty("globals.properties에 생성하여 삽입");
    List<MultipartFile> imageList = request.getFiles("upload"); //
UploadAdapter에서 _sendRequest() 사용했을 때
    for (MultipartFile mf : imageList) {
        if (imageList.get(0).getSize() > 0) {
            originalImagename = mf.getOriginalFilename();              // 원
본 파일 명
            extension = FilenameUtils.getExtension(originalImagename); // 확
장자
            imageName = "img_" + UUID.randomUUID() + "." + extension;  // 이
미지가 업로드 되면서 저장되는 이름
            savePath = realPathtoUploads +"/"+imageName;               => 이
미지 업로드 경로
            File imageUpload = new File(savePath);
            try {
                mf.transferTo(imageUpload);                          => 실질적
으로 이미지 업로드하는 부분
            } catch (IllegalStateException | IOException e) {
                e.printStackTrace();
            }
        }
    }
    savePath = "이미지의 src 부분에 들어갈 저장 경로를 적어준다.";
    log.debug("savePath=[{}]", savePath);
    mv.addObject("url", savePath);
    return mv;
}
```

이미지 업로드 후 처리

- Jenkins로 배포한 경우
  - globals.propertie 안에 저장경로를 다시 설정 해준다.
- <예시>
- APPS Portal의 경우
- CMD로 해당 서버로 들어가서
- '/WAS_APPS/tomcat-instance/apps.repia.com/deploy/' 안에 'globals.properties' 부분에 이미지가 저장되는 경로를 설정
- 저장 경로는 '/WAS_APPS/tomcat-instance/apps.repia.com/webapps/ROOT/resource/(저장 경로)'로 설정함

Ref

- CKEditor5 공식 문서
- CKEditor5 설정 방법

slaptear, ckeditor5

From:
http://125.132.25.164/dokuwiki/ - **문제를 잘 정의하는 것은 문제를 절반 해결한 것이다. - 2023.12**

Permanent link:
http://125.132.25.164/dokuwiki/doku.php?id=wiki:editor:ckeditor5&rev=1677727102

Last update: **2023/03/02 12:18**