

# EV FIRE CMS 공통 파일 업로드 모듈

## 패키지 구조

```
com.nexsol.evfire.cms
├─ controller
│   └─ CmsFileController.java
│
├─ dto
│   ├── CmsFileUploadResponse.java
│   └─ CmsFileResponse.java
│
├─ service
│   └─ CmsFileService.java
│
├─ service.impl
│   └─ CmsFileServiceImpl.java
│
└─ repository
    └─ CmsFileMasterRepository.java
```

## CmsFileUploadResponse.java

```
package com.nexsol.evfire.cms.dto;

import io.swagger.v3.oas.annotations.media.Schema;
import lombok.Builder;

@Builder
@Schema(description = "CMS 파일 업로드 응답")
public record CmsFileUploadResponse(

    @Schema(description = "파일 PK", example = "1")
    Long id,

    @Schema(description = "파일 그룹 ID", example = "CMS_BANNER")
    String fileId,

    @Schema(description = "원본 파일명", example = "banner.png")
    String originalFileName,

    @Schema(description = "저장 파일명", example = "e4d2f6f1-1111-2222-3333.png")
    String fileName
)
```

```

String storedFileName,

@Schema(description = "파일 URL", example = "/uploads/ev-fire-cms/banner/e4d2f6f1.png")
String fileUrl,

@Schema(description = "파일 크기(byte)", example = "102400")
Long fileSize,

@Schema(description = "MIME 타입", example = "image/png")
String mimeType
){
}

```

## CmsFileResponse.java

```

package com.nexsol.evfire.cms.dto;

import io.swagger.v3.oas.annotations.media.Schema;
import lombok.Builder;

@Builder
@Schema(description = "CMS 파일 조회 응답")
public record CmsFileResponse(

    @Schema(description = "파일 PK", example = "1")
    Long id,

    @Schema(description = "파일 그룹 ID", example = "CMS_POST")
    String fileId,

    @Schema(description = "원본 파일명", example = "manual.pdf")
    String originalFileName,

    @Schema(description = "저장 파일명", example = "f1f2f3f4.pdf")
    String storedFileName,

    @Schema(description = "확장자", example = "pdf")
    String fileExt,

    @Schema(description = "MIME 타입", example = "application/pdf")
    String mimeType,

    @Schema(description = "파일 크기(byte)", example = "204800")
    Long fileSize,

```

```
@Schema(description = "파일 URL", example = "/uploads/ev-fire-cms/post/f1f2f3f4.pdf")
String fileUrl
){
}
```

---

## CmsFileService.java

```
package com.nexsol.evfire.cms.service;

import org.springframework.web.multipart.MultipartFile;

import com.nexsol.evfire.cms.dto.CmsFileResponse;
import com.nexsol.evfire.cms.dto.CmsFileUploadResponse;

public interface CmsFileService {

    CmsFileUploadResponse upload(
        String fileGroupId,
        MultipartFile file,
        Long employeeId
    );

    CmsFileResponse findById(Long id);
}
```

---

## CmsFileServiceImpl.java

```
package com.nexsol.evfire.cms.service.impl;

import java.io.IOException;
import java.nio.file.Files;
import java.nio.file.Path;
import java.nio.file.Paths;
import java.nio.file.StandardCopyOption;
import java.util.UUID;

import org.springframework.beans.factory.annotation.Value;
import org.springframework.stereotype.Service;
import org.springframework.transaction.annotation.Transactional;
import org.springframework.util.StringUtils;
import org.springframework.web.multipart.MultipartFile;
```

```

import com.nexsol.evfire.cms.dto.CmsFileResponse;
import com.nexsol.evfire.cms.dto.CmsFileUploadResponse;
import com.nexsol.evfire.cms.entity.CmsFileMasterEntity;
import com.nexsol.evfire.cms.repository.CmsFileMasterRepository;
import com.nexsol.evfire.cms.service.CmsFileService;

import lombok.RequiredArgsConstructor;

@Service
@RequiredArgsConstructor
@Transactional(readOnly = true)
public class CmsFileServiceImpl implements CmsFileService {

    private final CmsFileMasterRepository cmsFileMasterRepository;

    @Value("${app.file.upload-dir}")
    private String uploadDir;

    @Value("${app.file.public-base-url}")
    private String publicBaseUrl;

    @Override
    @Transactional
    public CmsFileUploadResponse upload(
        String fileId,
        MultipartFile file,
        Long employeeId
    ) {

        if (file == null || file.isEmpty()) {
            throw new IllegalArgumentException("업로드 파일이 없습니다.");
        }

        try {

            String originalFileName = file.getOriginalFilename();
            String ext = StringUtils.getFilenameExtension(originalFileName);

            String storedFileName = UUID.randomUUID() +
                (ext != null ? "." + ext : "");

            String lowerGroup = fileId.toLowerCase();

            Path targetDir = Paths.get(uploadDir, lowerGroup);

            if (!Files.exists(targetDir)) {
                Files.createDirectories(targetDir);
            }
        }
    }
}

```

```

    }

    Path targetPath = targetDir.resolve(storedFileName);

    Files.copy(
        file.getInputStream(),
        targetPath,
        StandardCopyOption.REPLACE_EXISTING
    );

    String fileUrl = publicBaseUrl + "/" + lowerGroup + "/" + storedFileName;

    CmsFileMasterEntity entity = CmsFileMasterEntity.builder()
        .fileGroupId(fileGroupId)
        .originalFileName(originalFileName)
        .storedFileName(storedFileName)
        .fileExt(ext)
        .mimeType(file.getContentType())
        .fileSize(file.getSize())
        .storageType("LOCAL")
        .filePath(targetPath.toString())
        .fileUrl(fileUrl)
        .downloadCount(0)
        .useYn("Y")
        .deleteYn("N")
        .createdBy(employeeId)
        .updatedBy(employeeId)
        .build();

    cmsFileMasterRepository.save(entity);

    return CmsFileUploadResponse.builder()
        .id(entity.getId())
        .fileGroupId(entity.getFileGroupId())
        .originalFileName(entity.getOriginalFileName())
        .storedFileName(entity.getStoredFileName())
        .fileUrl(entity.getFileUrl())
        .fileSize(entity.getFileSize())
        .mimeType(entity.getMimeType())
        .build();

    } catch (IOException e) {
        throw new RuntimeException("파일 업로드 실패", e);
    }
}

@Override
public CmsFileResponse findById(Long id) {

```

```

CmsFileMasterEntity entity = cmsFileMasterRepository.findById(id)
    .orElseThrow(() -> new IllegalArgumentException("파일이 존재하지 않습니다."));

return CmsFileResponse.builder()
    .id(entity.getId())
    .fileGroupId(entity.getFileGroupId())
    .originalFileName(entity.getOriginalFileName())
    .storedFileName(entity.getStoredFileName())
    .fileExt(entity.getFileExt())
    .mimeType(entity.getMimeType())
    .fileSize(entity.getFileSize())
    .fileUrl(entity.getFileUrl())
    .build();
}
}

```

## CmsFileController.java

```

package com.nexsol.evfire.cms.controller;

import org.springframework.http.MediaType;
import org.springframework.security.core.Authentication;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.bind.annotation.RestController;
import org.springframework.web.multipart.MultipartFile;

import com.nexsol.evfire.cms.dto.CmsFileResponse;
import com.nexsol.evfire.cms.dto.CmsFileUploadResponse;
import com.nexsol.evfire.cms.service.CmsFileService;
import com.nexsol.evfire.common.security.auth.AuthPrincipal;

import io.swagger.v3.oas.annotations.Operation;
import io.swagger.v3.oas.annotations.Parameter;
import io.swagger.v3.oas.annotations.media.Content;
import io.swagger.v3.oas.annotations.media.Schema;
import io.swagger.v3.oas.annotations.responses.ApiResponse;
import io.swagger.v3.oas.annotations.tags.Tag;
import lombok.RequiredArgsConstructor;

@RestController

```

```

@RequiredArgsConstructor
@RequestMapping("/ev-fire/api/private/cms/files")
@Tag(name = "CMS 파일", description = "CMS 공통 파일 업로드 API")
public class CmsFileController {

    private final CmsFileService cmsFileService;

    @PostMapping(
        consumes = MediaType.MULTIPART_FORM_DATA_VALUE
    )
    @Operation(
        summary = "CMS 파일 업로드",
        description = "배너/게시판/자료실 등 CMS 공통 파일 업로드"
    )
    @ApiResponse(
        responseCode = "200",
        description = "업로드 성공",
        content = @Content(schema = @Schema(implementation = CmsFileUploadResponse.class))
    )
    public CmsFileUploadResponse upload(

        @Parameter(
            description = "파일 그룹 ID",
            example = "BANNER"
        )
        @RequestParam String fileGroupId,

        @Parameter(
            description = "업로드 파일"
        )
        @RequestParam MultipartFile file,

        Authentication authentication
    ){

        AuthPrincipal principal =
            (AuthPrincipal) authentication.getPrincipal();

        return cmsFileService.upload(
            fileGroupId,
            file,
            principal.employeeld()
        );
    }

    @GetMapping("/{id}")
    @Operation(
        summary = "CMS 파일 조회",

```

```
        description = "파일 단건 조회"  
    )  
    @ApiResponse(  
        responseCode = "200",  
        description = "조회 성공",  
        content = @Content(schema = @Schema(implementation = CmsFileResponse.class))  
    )  
    public CmsFileResponse findById(  
        @PathVariable Long id  
    ){  
        return cmsFileService.findById(id);  
    }  
}
```

---

## Swagger 확인 URL

<http://localhost:8081/swagger-ui/index.html>

---

## 테스트 엔드포인트

POST /ev-fire/api/private/cms/files

form-data:

fileGroupId = BANNER  
file = 이미지파일

Authorization:

Bearer {new-boon access token}