

컨트리뷰션 도전기

김재섭

목차

01 참여 동기

- 참여 동기
- 기존 구조에서 발견한 문제점
- 개선 방향성

02 참여 이력

- 패키지 구조 개선
- 코드 구조 개선

03 후기

참여 동기

“아, 이거 내가 직접 고쳐보고 싶다!”

참여 동기

“아, 이거 내가 직접 고쳐보고 싶다!”

오픈소스를 보다보면 “이부분을 내가 조금만 수정 또는 추가를 한다면 더 좋아질것같다”라는 생각이 들때가 있습니다.

이러한 것을 그냥 지나치지 않고 내가 해보고 싶은걸 행동으로 옮겼습니다.

기존 구조에서 발견한 문제점

비표준 패키지 구조

web/service 중심의 단순 구조로 기능별 책임이 분리되지 않음

- 전체 구조 파악이 어려움
- 확장 시 의존성 증가
- 도메인 객체 생성 및 일부 비즈니스 로직이 컨트롤러에 적용
- domain, dto 계층의 부재

Map 기반 응답의 한계

모든 API 및 서비스 응답이 Map<String,Object> 형태로 구성

- API 스펙 불명확
- Swagger 구조 표시 불가
- 데이터 흐름 파악 불가

기존 구조에서 발견한 문제점

비표준 패키지 구조

web/service 중심의 단순 구조로 기능별 책임이 분리되지 않음

- 전체 구조 파악이 어려움
- 확장 시 의존성 증가
- 도메인 객체 생성 및 일부 비즈니스 로직이 컨트롤러에 적용
- domain, dto 계층의 부재

Map 기반 응답의 한계

모든 API 및 서비스 응답이 Map<String,Object> 형태로 구성

- API 스펙 불명확
- Swagger 구조 표시 불가
- 데이터 흐름 파악 불가

가장 먼저 구조적 문제와 문서화 문제, 두 가지가 동시에 해결 되어야 한다고 판단했습니다.

개선 방향성

01

유지보수성 & 확장성 확보

표준적인 패키지 구조로 재정비하여
장기적 관리 용이성 확보

02

명확한 API 스펙 정의

응답/요청 객체를 명시적으로 정의하여
Swagger 문서 품질 향상

03

가이드라인 일관성 유지

표준프레임워크의 정체성을 지키며
점진적으로 개선

개선 방향성

01

유지보수성 & 확장성 확보

표준적인 패키지 구조로 재정비하여
장기적 관리 용이성 확보

02

명확한 API 스펙 정의

응답/요청 객체를 명시적으로 정의하여
Swagger 문서 품질 향상

03

가이드라인 일관성 유지

표준프레임워크의 정체성을 지키며
점진적으로 개선

📌 핵심 원칙: 기존 코드를 깨지 않되, 더 명확하고 표준화된 형태로 발전시키기

패키지 구조 개선

기존 패키지 구조

- ▼ let
 - ▼ cop
 - ▼ bbs
 - ▼ service
 - > impl
 - Board.java
 - BoardMaster.java
 - BoardMasterVO.java
 - BoardVO.java
 - EgovBBSAttributeMana...
 - EgovBBSLoneMasterSer...
 - EgovBBSManageService....
 - ▼ web
 - EgovBBSAttributeMana...
 - EgovBBSManageApiCon...
 - > com

패키지 구조 개선

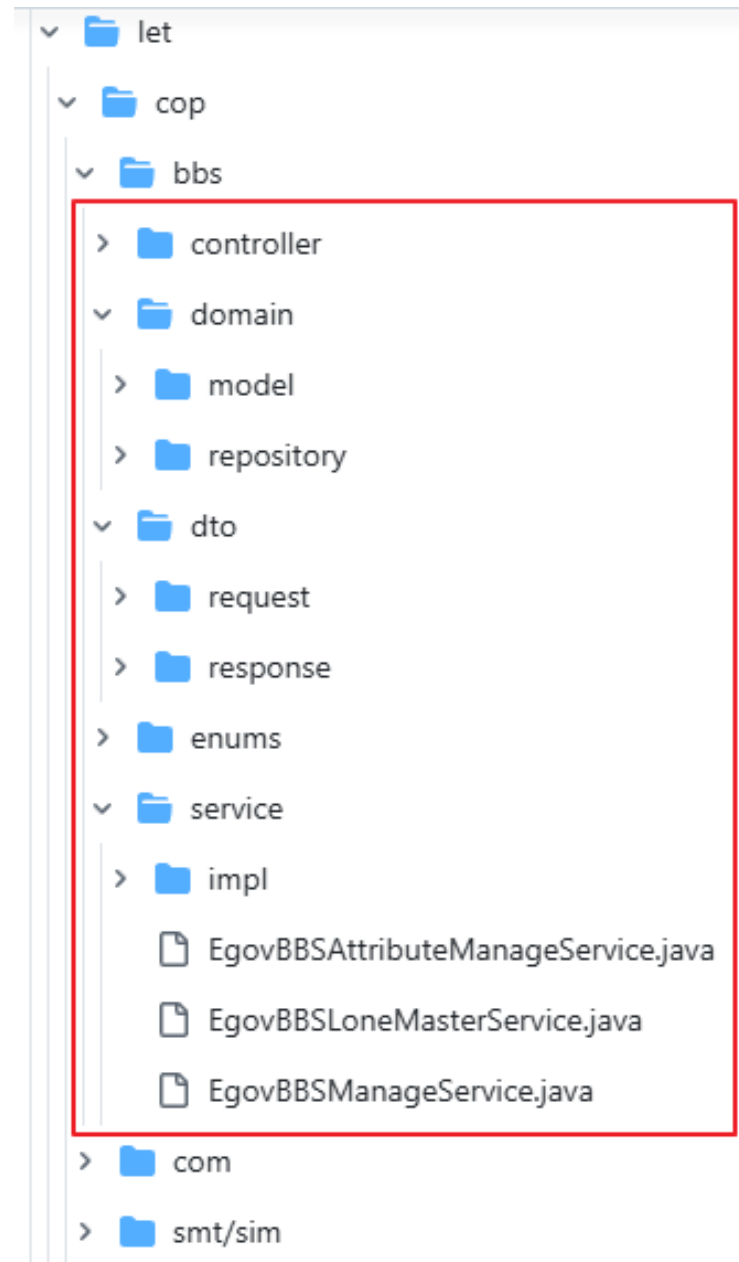
기존 패키지 구조

- ▼ let
 - ▼ cop
 - ▼ bbs
 - ▼ service
 - > impl
 - Board.java
 - BoardMaster.java
 - BoardMasterVO.java
 - BoardVO.java
 - EgovBBSAttributeMana...
 - EgovBBSLoneMasterSer...
 - EgovBBSManageService....
 - ▼ web
 - EgovBBSAttributeMana...
 - EgovBBSManageApiCon...
 - > com

모든 코드가 web과 service에
밀집되어 있어 확장과 유지보수에 제약이 많음

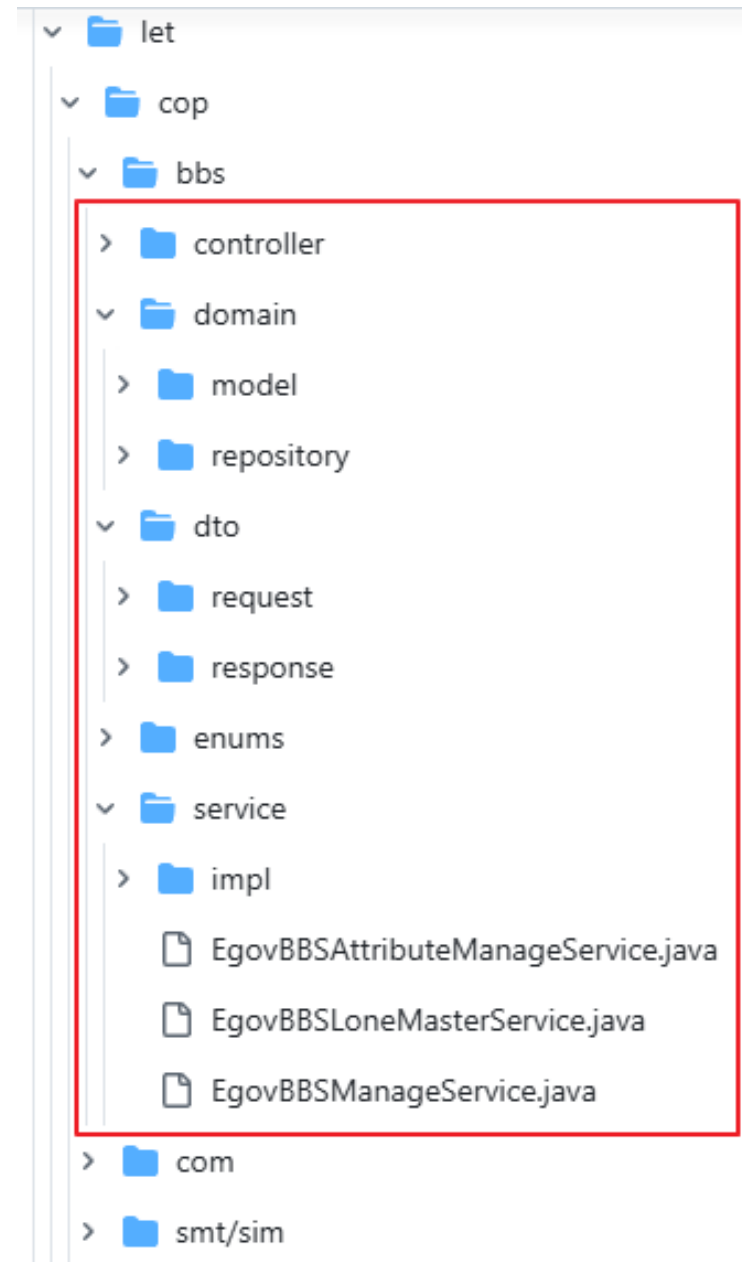
패키지 구조 개선

패키지 개선 후



패키지 구조 개선

패키지 개선 후



계층별 역할이 명확하게 분리되어
구조적 일관성을 확보

코드 구조 개선

기존 코드 - 필드 주입

```
@Autowired
private EgovJwtTokenUtil jwtTokenUtil;
public static final String HEADER_STRING = "Authorization";

@Resource(name = "EgovBBSManageService")
private EgovBBSManageService bbsMngService;

@Resource(name = "EgovBBSAttributeManageService")
private EgovBBSAttributeManageService bbsAttrbService;

@Resource(name = "EgovFileMngService")
private EgovFileMngService fileMngService;

@Resource(name = "EgovFileMngUtil")
private EgovFileMngUtil fileUtil;

@Resource(name = "propertiesService")
protected EgovPropertyService propertyService;

@Resource(name = "egovMessageSource")
EgovMessageSource egovMessageSource;

@Resource(name = "EgovFileMngService")
private EgovFileMngService fileService;
```

코드 구조 개선

기존 코드 - 필드 주입

```
@Autowired
private EgovJwtTokenUtil jwtTokenUtil;
public static final String HEADER_STRING = "Authorization";

@Resource(name = "EgovBBSManageService")
private EgovBBSManageService bbsMngService;

@Resource(name = "EgovBBSAttributeManageService")
private EgovBBSAttributeManageService bbsAttrbService;

@Resource(name = "EgovFileMngService")
private EgovFileMngService fileMngService;

@Resource(name = "EgovFileMngUtil")
private EgovFileMngUtil fileUtil;

@Resource(name = "propertiesService")
protected EgovPropertyService propertyService;

@Resource(name = "egovMessageSource")
EgovMessageSource egovMessageSource;

@Resource(name = "EgovFileMngService")
private EgovFileMngService fileService;
```

필드 주입 방식으로 인해 테스트, 순환 참조 발생 가능성,
불변성 보장 불가 등의 문제가 존재

코드 구조 개선

개선 후 코드 – 생성자 주입

```
@RestController
@RequiredArgsConstructor
@Tag(name="EgovBBSManageApiController",description = "게시물 관리")
public class EgovBBSManageApiController {
    public static final String HEADER_STRING = "Authorization";
    private final EgovJwtTokenUtil jwtTokenUtil;
    private final EgovFileMngUtil fileUtil;
    private final ResultVoHelper resultVoHelper;
    private final EgovBBSManageService bbsMngService;
    private final EgovFileMngService fileMngService;
    private final EgovPropertyService propertyService;
    private final EgovBBSAttributeManageService bbsAttrbService;
    private final DefaultBeanValidator beanValidator;
```

코드 구조 개선

컨트롤러 기존 코드

```
@Operation(
    summary = "게시판 파일 첨부 관련 정보 조회",
    description = "게시판의 파일 첨부가능 여부 및 첨부가능 파일 수 조회",
    tags = {"EgovBBSManageApiController"}
)
@ApiResponses(value = {
    @ApiResponse(responseCode = "200", description = "조회 성공"),
    @ApiResponse(responseCode = "403", description = "인가된 사용자가 아님")
})
@GetMapping(value = "/boardFileAtch/{bbsId}")
public ResultVO selectUserBBSMasterInf(
    @Parameter(name = "bbsId", description = "게시판 Id", in = ParameterIn.PATH, example="BBSMSTR_AAAAAAAAAA")
    @PathVariable("bbsId") String bbsId)
    throws Exception {
    ResultVO resultVO = new ResultVO();
    Map<String, Object> resultMap = new HashMap<String, Object>();

    BoardMasterVO searchVO = new BoardMasterVO();
    searchVO.setBbsId(bbsId);

    BoardMasterVO master = bbsAttrbService.selectBBSMasterInf(searchVO);

    // 파일 첨부 외의 다른 정보를 전달하지 않기 위해 신규 객체 생성
    BoardMasterVO masterFileAtchInfo = new BoardMasterVO();

    masterFileAtchInfo.setFileAtchPosblAt(master.getFileAtchPosblAt());
    masterFileAtchInfo.setPosblAtchFileNumber(master.getPosblAtchFileNumber());
    masterFileAtchInfo.setPosblAtchFileSize(master.getPosblAtchFileSize());

    resultMap.put("brdMstrVO", masterFileAtchInfo);

    resultVO.setResult(resultMap);
    resultVO.setResultCode(ResponseCode.SUCCESS.getCode());
    resultVO.setResultMessage(ResponseCode.SUCCESS.getMessage());

    return resultVO;
}
```


코드 구조 개선

컨트롤러 기존 코드

```
@Operation(
    summary = "게시판 파일 첨부 관련 정보 조회",
    description = "게시판의 파일 첨부가능 여부 및 첨부가능 파일 수 조회",
    tags = {"EgovBBSManageApiController"}
)
@ApiResponses(value = {
    @ApiResponse(responseCode = "200", description = "조회 성공"),
    @ApiResponse(responseCode = "403", description = "인가된 사용자가 아님")
})
@GetMapping(value = "/boardFileAtch/{bbsId}")
public ResultVO selectUserBBSMasterInf(
    @Parameter(name = "bbsId", description = "게시판 Id", in = ParameterIn.PATH, example="BBSMSTR_AAAAAAAAAA")
    @PathVariable("bbsId") String bbsId)
    throws Exception {
    ResultVO resultVO = new ResultVO();
    Map<String, Object> resultMap = new HashMap<String, Object>();

    BoardMasterVO searchVO = new BoardMasterVO();
    searchVO.setBbsId(bbsId);

    BoardMasterVO master = bbsAttrbService.selectBBSMasterInf(searchVO);

    // 파일 첨부 외의 다른 정보를 전달하지 않기 위해 신규 객체 생성
    BoardMasterVO masterFileAtchInfo = new BoardMasterVO();

    masterFileAtchInfo.setFileAtchPosblAt(master.getFileAtchPosblAt());
    masterFileAtchInfo.setPosblAtchFileNumber(master.getPosblAtchFileNumber());
    masterFileAtchInfo.setPosblAtchFileSize(master.getPosblAtchFileSize());

    resultMap.put("brdMstrVO", masterFileAtchInfo);

    resultVO.setResult(resultMap);
    resultVO.setResultCode(ResponseCode.SUCCESS.getCode());
    resultVO.setResultMessage(ResponseCode.SUCCESS.getMessage());

    return resultVO;
}
```

코드 구조 개선

컨트롤러 기존 코드

```
@Operation(
    summary = "게시판 파일 첨부 관련 정보 조회",
    description = "게시판의 파일 첨부가능 여부 및 첨부가능 파일 수 조회",
    tags = {"EgovBBSManageApiController"}
)
@ApiResponses(value = {
    @ApiResponse(responseCode = "200", description = "조회 성공"),
    @ApiResponse(responseCode = "403", description = "인가된 사용자가 아님")
})
@GetMapping(value = "/boardFileAtch/{bbsId}")
public ResultVO selectUserBBSMasterInf(
    @Parameter(name = "bbsId", description = "게시판 Id", in = ParameterIn.PATH, example="BBSMSTR_AAAAAAAAAA")
    @PathVariable("bbsId") String bbsId)
    throws Exception {
    ResultVO resultVO = new ResultVO();
    Map<String, Object> resultMap = new HashMap<String, Object>();

    BoardMasterVO searchVO = new BoardMasterVO();
    searchVO.setBbsId(bbsId);

    BoardMasterVO master = bbsAttrbService.selectBBSMasterInf(searchVO);

    // 파일 첨부 외의 다른 정보를 전달하지 않기 위해 신규 객체 생성
    BoardMasterVO masterFileAtchInfo = new BoardMasterVO();

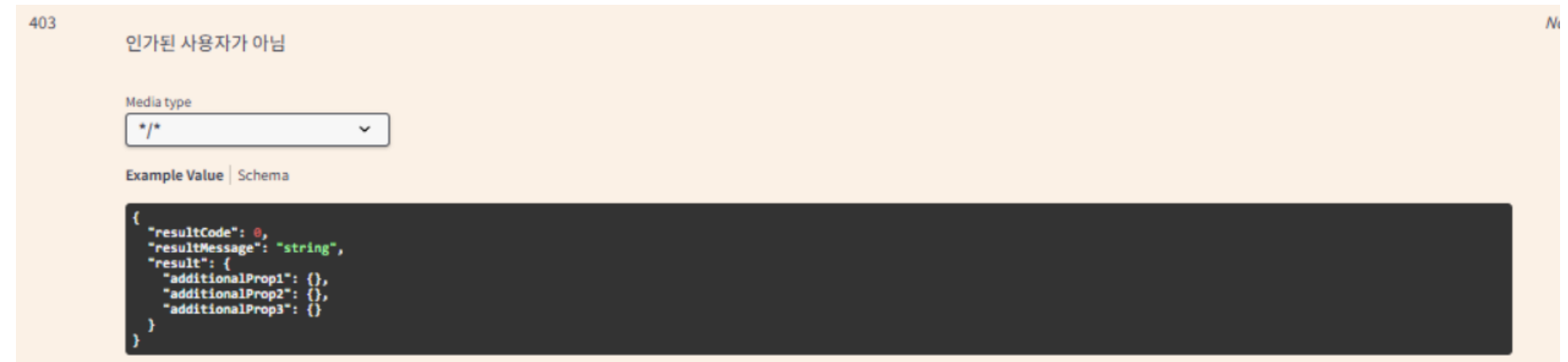
    masterFileAtchInfo.setFileAtchPosblAt(master.getFileAtchPosblAt());
    masterFileAtchInfo.setPosblAtchFileNumber(master.getPosblAtchFileNumber());
    masterFileAtchInfo.setPosblAtchFileSize(master.getPosblAtchFileSize());

    resultMap.put("brdMstrVO", masterFileAtchInfo);

    resultVO.setResult(resultMap);
    resultVO.setResultCode(ResponseCode.SUCCESS.getCode());
    resultVO.setResultMessage(ResponseCode.SUCCESS.getMessage());

    return resultVO;
}
```

스웨거 불명확



코드 구조 개선

컨트롤러 기존 코드

```
@Operation(
    summary = "게시판 파일 첨부 관련 정보 조회",
    description = "게시판의 파일 첨부가능 여부 및 첨부가능 파일 수 조회",
    tags = {"EgovBBSManageApiController"}
)
@ApiResponses(value = {
    @ApiResponse(responseCode = "200", description = "조회 성공"),
    @ApiResponse(responseCode = "403", description = "인가된 사용자가 아님")
})
@GetMapping(value = "/boardFileAtch/{bbsId}")
public ResultVO selectUserBBSMasterInf(
    @Parameter(name = "bbsId", description = "게시판 Id", in = ParameterIn.PATH, example="BBSMSTR_AAAAAAAAAA")
    @PathVariable("bbsId") String bbsId)
    throws Exception {
    ResultVO resultVO = new ResultVO();
    Map<String, Object> resultMap = new HashMap<String, Object>();

    BoardMasterVO searchVO = new BoardMasterVO();
    searchVO.setBbsId(bbsId);

    BoardMasterVO master = bbsAttrbService.selectBBSMasterInf(searchVO);

    // 파일 첨부 외의 다른 정보를 전달하지 않기 위해 신규 객체 생성
    BoardMasterVO masterFileAtchInfo = new BoardMasterVO();

    masterFileAtchInfo.setFileAtchPosblAt(master.getFileAtchPosblAt());
    masterFileAtchInfo.setPosblAtchFileNumber(master.getPosblAtchFileNumber());
    masterFileAtchInfo.setPosblAtchFileSize(master.getPosblAtchFileSize());

    resultMap.put("brdMstrVO", masterFileAtchInfo);

    resultVO.setResult(resultMap);
    resultVO.setResultCode(ResponseCode.SUCCESS.getCode());
    resultVO.setResultMessage(ResponseCode.SUCCESS.getMessage());

    return resultVO;
}
```

코드 구조 개선

컨트롤러 기존 코드

```
@Operation(  
    summary = "게시판 파일 첨부 관련 정보 조회",  
    description = "게시판의 파일 첨부가능 여부 및 첨부가능 파일 수 조회",  
    tags = {"EgovBBSManageApiController"}  
)  
@ApiResponses(value = {  
    @ApiResponse(responseCode = "200", description = "조회 성공"),  
    @ApiResponse(responseCode = "403", description = "인가된 사용자가 아님")  
})  
@GetMapping(value = "/boardFileAtch/{bbsId}")  
public ResultVO selectUserBBSMasterInf(  
    @Parameter(name = "bbsId", description = "게시판 Id", in = ParameterIn.PATH, example="BBSMSTR_AAAAAAAAAA")  
    @PathVariable("bbsId") String bbsId)  
    throws Exception {  
    ResultVO resultVO = new ResultVO();  
    Map<String, Object> resultMap = new HashMap<String, Object>();  
  
    BoardMasterVO searchVO = new BoardMasterVO();  
    searchVO.setBbsId(bbsId);  
  
    BoardMasterVO master = bbsAttrbService.selectBBSMasterInf(searchVO);  
  
    // 파일 첨부 외의 다른 정보를 전달하지 않기 위해 신규 객체 생성  
    BoardMasterVO masterFileAtchInfo = new BoardMasterVO();  
  
    masterFileAtchInfo.setFileAtchPosblAt(master.getFileAtchPosblAt());  
    masterFileAtchInfo.setPosblAtchFileNumber(master.getPosblAtchFileNumber());  
    masterFileAtchInfo.setPosblAtchFileSize(master.getPosblAtchFileSize());  
  
    resultMap.put("brdMstrVO", masterFileAtchInfo);  
  
    resultVO.setResult(resultMap);  
    resultVO.setResultCode(ResponseCode.SUCCESS.getCode());  
    resultVO.setResultMessage(ResponseCode.SUCCESS.getMessage());  
  
    return resultVO;  
}
```

→ 도메인 레벨의 객체를 컨트롤러에서 생성 및 조립

코드 구조 개선

컨트롤러 기존 코드

```
@Operation(  
    summary = "게시판 파일 첨부 관련 정보 조회",  
    description = "게시판의 파일 첨부가능 여부 및 첨부가능 파일 수 조회",  
    tags = {"EgovBBSManageApiController"}  
)  
@ApiResponses(value = {  
    @ApiResponse(responseCode = "200", description = "조회 성공"),  
    @ApiResponse(responseCode = "403", description = "인가된 사용자가 아님")  
})  
@GetMapping(value = "/boardFileAtch/{bbsId}")  
public ResultVO selectUserBBSMasterInf(  
    @Parameter(name = "bbsId", description = "게시판 Id", in = ParameterIn.PATH, example="BBSMSTR_AAAAAAAAAAAAA")  
    @PathVariable("bbsId") String bbsId)  
    throws Exception {  
    ResultVO resultVO = new ResultVO();  
    Map<String, Object> resultMap = new HashMap<String, Object>();  
  
    BoardMasterVO searchVO = new BoardMasterVO();  
    searchVO.setBbsId(bbsId);  
  
    BoardMasterVO master = bbsAttrbService.selectBBSMasterInf(searchVO);  
  
    // 파일 첨부 외의 다른 정보를 전달하지 않기 위해 신규 객체 생성  
    BoardMasterVO masterFileAtchInfo = new BoardMasterVO();  
  
    masterFileAtchInfo.setFileAtchPosblAt(master.getFileAtchPosblAt());  
    masterFileAtchInfo.setPosblAtchFileNumber(master.getPosblAtchFileNumber());  
    masterFileAtchInfo.setPosblAtchFileSize(master.getPosblAtchFileSize());  
  
    resultMap.put("brdMstrVO", masterFileAtchInfo);  
  
    resultVO.setResult(resultMap);  
    resultVO.setResultCode(ResponseCode.SUCCESS.getCode());  
    resultVO.setResultMessage(ResponseCode.SUCCESS.getMessage());  
  
    return resultVO;  
}
```

코드 구조 개선

컨트롤러 기존 코드

```
@Operation(
    summary = "게시판 파일 첨부 관련 정보 조회",
    description = "게시판의 파일 첨부가능 여부 및 첨부가능 파일 수 조회",
    tags = {"EgovBBSManageApiController"}
)
@ApiResponses(value = {
    @ApiResponse(responseCode = "200", description = "조회 성공"),
    @ApiResponse(responseCode = "403", description = "인가된 사용자가 아님")
})
@GetMapping(value = "/boardFileAtch/{bbsId}")
public ResultVO selectUserBBSMasterInf(
    @Parameter(name = "bbsId", description = "게시판 Id", in = ParameterIn.PATH, example="BBSMSTR_AAAAAAAAAA")
    @PathVariable("bbsId") String bbsId)
    throws Exception {
    ResultVO resultVO = new ResultVO();
    Map<String, Object> resultMap = new HashMap<String, Object>();

    BoardMasterVO searchVO = new BoardMasterVO();
    searchVO.setBbsId(bbsId);

    BoardMasterVO master = bbsAttrbService.selectBBSMasterInf(searchVO);

    // 파일 첨부 외의 다른 정보를 전달하지 않기 위해 신규 객체 생성
    BoardMasterVO masterFileAtchInfo = new BoardMasterVO();

    masterFileAtchInfo.setFileAtchPosblAt(master.getFileAtchPosblAt());
    masterFileAtchInfo.setPosblAtchFileNumber(master.getPosblAtchFileNumber());
    masterFileAtchInfo.setPosblAtchFileSize(master.getPosblAtchFileSize());

    resultMap.put("brdMstrVO", masterFileAtchInfo);

    resultVO.setResult(resultMap);
    resultVO.setResultCode(ResponseCode.SUCCESS.getCode());
    resultVO.setResultMessage(ResponseCode.SUCCESS.getMessage());

    return resultVO;
}
```

Map<String, Object> 기반 응답

```
@Schema(description = "응답 객체 VO")
@Getter
@Setter
public class ResultVO {

    @Schema(description = "응답 코드")
    private int resultCode = 0;

    @Schema(description = "응답 메시지")
    private String resultMessage = "OK";
    private Map<String, Object> result = new HashMap<String, Object>();

    public void putResult(String key, Object value) {
        result.put(key, value);
    }

    public Object getResult(String key) {
        return this.result.get(key);
    }
}
```

코드 구조 개선

개선 후 코드

```
@Operation(summary = "게시판 파일 첨부 관련 정보 조회",
description = "게시판의 파일 첨부가능 여부 및 첨부가능 파일 수 조회",
tags = {"EgovBBSManageApiController"})
)
@ApiResponses(value = {
    @ApiResponse(responseCode = "200", description = "조회 성공", content = @Content(
        schema = @Schema(oneOf = {
            BbsFileAtchResponseDTO.class
        })
    )),
    @ApiResponse(
        responseCode = "403",
        description = "인가된 사용자가 아님",
        content = @Content(
            mediaType = "application/json",
            examples = @ExampleObject(
                name = "403 응답 예시",
                summary = "Forbidden",
                value = "{\n" +
                    "  \"resultCode\": 403,\n" +
                    "  \"resultMessage\": \"인가된 사용자가 아님\"\n" +
                    "}"
            )
        )
    )
}),
))
@GetMapping(value = "/boardFileAtch/{bbsId}")
public IntermediateResultVO<BbsFileAtchResponseDTO> selectUserBBSMasterInf(
    @Parameter(name = "bbsId", description = "게시판 Id", in = ParameterIn.PATH, example="BBSMSTR_AAAAAAAAAA")
    @PathVariable("bbsId") String bbsId)
throws Exception {

    BbsFileAtchResponseDTO response = bbsAttrbService.selectBBSMasterInf(bbsId, null, BbsDetailRequestType.FILE_ATCH);

    return IntermediateResultVO.success(response);
}
```


코드 구조 개선

개선 후 코드

```
@Operation(summary = "게시판 파일 첨부 관련 정보 조회",
description = "게시판의 파일 첨부가능 여부 및 첨부가능 파일 수 조회",
tags = {"EgovBBSManageApiController"})
@ApiResponses(value = {
    @ApiResponse(responseCode = "200", description = "조회 성공", content = @Content(
        schema = @Schema(oneOf = {
            BbsFileAtchResponseDTO.class
        })
    )),
    @ApiResponse(
        responseCode = "403",
        description = "인가된 사용자가 아님",
        content = @Content(
            mediaType = "application/json",
            examples = @ExampleObject(
                name = "403 응답 예시",
                summary = "Forbidden",
                value = "{\n" +
                    "  \"resultCode\": 403,\n" +
                    "  \"resultMessage\": \"인가된 사용자가 아님\"\n" +
                    "}"
            )
        )
    )
})
@GetMapping(value = "/boardFileAtch/{bbsId}")
public IntermediateResultVO<BbsFileAtchResponseDTO> selectUserBBSMasterInf(
    @Parameter(name = "bbsId", description = "게시판 Id", in = ParameterIn.PATH, example="BBSMSTR_AAAAAAAAAA")
    @PathVariable("bbsId") String bbsId)
    throws Exception {

    BbsFileAtchResponseDTO response = bbsAttrbService.selectBBSMasterInf(bbsId, null, BbsDetailRequestType.FILE_ATCH);

    return IntermediateResultVO.success(response);
}
```


코드 구조 개선

개선 후 코드

```
@Operation(summary = "게시판 파일 첨부 관련 정보 조회",
description = "게시판의 파일 첨부가능 여부 및 첨부가능 파일 수 조회",
tags = {"EgovBBSManageApiController"})
@ApiResponses(value = {
    @ApiResponse(responseCode = "200", description = "조회 성공", content = @Content(
        schema = @Schema(oneOf = {
            BbsFileAtchResponseDTO.class
        })
    )),
    @ApiResponse(
        responseCode = "403",
        description = "인가된 사용자가 아님",
        content = @Content(
            mediaType = "application/json",
            examples = @ExampleObject(
                name = "403 응답 예시",
                summary = "Forbidden",
                value = "{\n" +
                    "  \"resultCode\": 403,\n" +
                    "  \"resultMessage\": \"인가된 사용자가 아님\"\n" +
                    "}"
            )
        )
    )
})
@GetMapping(value = "/boardFileAtch/{bbsId}")
public IntermediateResultVO<BbsFileAtchResponseDTO> selectUserBBSMasterInf(
    @Parameter(name = "bbsId", description = "게시판 Id", in = ParameterIn.PATH, example="BBSMSTR_AAAAAAAAAA")
    @PathVariable("bbsId") String bbsId)
    throws Exception {

    BbsFileAtchResponseDTO response = bbsAttrbService.selectBBSMasterInf(bbsId, null, BbsDetailRequestType.FILE_ATCH);

    return IntermediateResultVO.success(response);
}
```

코드 구조 개선

개선 후 코드

```
@Operation(
    summary = "게시판 파일 첨부 관련 정보 조회",
    description = "게시판의 파일 첨부가능 여부 및 첨부가능 파일 수 조회",
    tags = {"EgovBBSManageApiController"}
)
@ApiResponses(value = {
    @ApiResponse(responseCode = "200", description = "조회 성공", content = @Content(
        schema = @Schema(oneOf = {
            BbsFileAtchResponseDTO.class
        })
    )),
    @ApiResponse(
        responseCode = "403",
        description = "인가된 사용자가 아님",
        content = @Content(
            mediaType = "application/json",
            examples = @ExampleObject(
                name = "403 응답 예시",
                summary = "Forbidden",
                value = "{\n" +
                    "  \"resultCode\": 403,\n" +
                    "  \"resultMessage\": \"인가된 사용자가 아님\"\n" +
                    "}"
            )
        )
    )
})
@GetMapping(value = "/boardFileAtch/{bbsId}")
public IntermediateResultVO<BbsFileAtchResponseDTO> selectUserBBSMasterInf(
    @Parameter(name = "bbsId", description = "게시판 Id", in = ParameterIn.PATH, example="BBSMSTR_AAAAAAAAAA")
    @PathVariable("bbsId") String bbsId)
    throws Exception {

    BbsFileAtchResponseDTO response = bbsAttrbService.selectBBSMasterInf(bbsId, null, BbsDetailRequestType.FILE_ATCH);

    return IntermediateResultVO.success(response);
}
```

도메인 객체는 서비스 계층에서 생성하도록 역할 분리

```
public BbsFileAtchResponseDTO selectBBSMasterInf(String bbsId, String uniqId, BbsDetailRequestType requestType) throws Exception {
    //-----
    // 2009.06.26 : 2단계 기능 추가
    //-----
    //return attrbMngDAO.selectBBSMasterInf(searchVO);
    BoardMasterVO searchVO = new BoardMasterVO();
    searchVO.setBbsId(bbsId);
    searchVO.setUniqId(uniqId);

    BoardMasterVO result = attrbMngDAO.selectBBSMasterInf(searchVO);

    String flag = propertyService.getString("Globals.addedOptions");
    if (flag != null && flag.trim().equalsIgnoreCase("true")) {
        BoardMasterVO options = addedOptionsDAO.selectAddedOptionsInf(searchVO);

        if (options != null) {
            if (options.getCommentAt().equals("Y")) {
                result.setOption("comment");
            }

            if (options.getStsfdgAt().equals("Y")) {
                result.setOption("stsfdg");
            }
        } else {
            result.setOption("na"); // 미지정 상태로 수정 가능 (이미 지정된 경우는 수정 불가로 처리)
        }
    }

    if (requestType == BbsDetailRequestType.DETAIL || requestType == BbsDetailRequestType.LIST) {
        return BbsAttributeDetailResponseDTO.from(result);
    } else if (requestType == BbsDetailRequestType.FILE_ATCH) {
        return BbsFileAtchResponseDTO.from(result);
    }

    throw new IllegalArgumentException("지원하지 않는 requestType입니다.");
}
```

코드 구조 개선

개선 후 코드

```
@Operation(summary = "게시판 파일 첨부 관련 정보 조회",
description = "게시판의 파일 첨부가능 여부 및 첨부가능 파일 수 조회",
tags = {"EgovBBSManageApiController"})
)
@ApiResponses(value = {
    @ApiResponse(responseCode = "200", description = "조회 성공", content = @Content(
        schema = @Schema(oneOf = {
            BbsFileAtchResponseDTO.class
        })
    )),
    @ApiResponse(
        responseCode = "403",
        description = "인가된 사용자가 아님",
        content = @Content(
            mediaType = "application/json",
            examples = @ExampleObject(
                name = "403 응답 예시",
                summary = "Forbidden",
                value = "{\n" +
                    "  \"resultCode\": 403,\n" +
                    "  \"resultMessage\": \"인가된 사용자가 아님\"\n" +
                    "}"
            )
        )
    )
}),
))
@GetMapping(value = "/boardFileAtch/{bbsId}")
public IntermediateResultVO<BbsFileAtchResponseDTO> selectUserBBSMasterInf(
    @Parameter(name = "bbsId", description = "게시판 Id", in = ParameterIn.PATH, example="BBSMSTR_AAAAAAAAAA")
    @PathVariable("bbsId") String bbsId)
throws Exception {

    BbsFileAtchResponseDTO response = bbsAttrbService.selectBBSMasterInf(bbsId, null, BbsDetailRequestType.FILE_ATCH);

    return IntermediateResultVO.success(response);
}
```

코드 구조 개선

개선 후 코드

```
@Operation(summary = "게시판 파일 첨부 관련 정보 조회",
description = "게시판의 파일 첨부가능 여부 및 첨부가능 파일 수 조회",
tags = {"EgovBBSManageApiController"})
)
@ApiResponses(value = {
    @ApiResponse(responseCode = "200", description = "조회 성공", content = @Content(
        schema = @Schema(oneOf = {
            BbsFileAtchResponseDTO.class
        })
    )),
    @ApiResponse(
        responseCode = "403",
        description = "인가된 사용자가 아님",
        content = @Content(
            mediaType = "application/json",
            examples = @ExampleObject(
                name = "403 응답 예시",
                summary = "Forbidden",
                value = "{\n" +
                    "  \"resultCode\": 403,\n" +
                    "  \"resultMessage\": \"인가된 사용자가 아님\"\n" +
                    "}"
            )
        )
    )
})
@GetMapping(value = "/boardFileAtch/{bbsId}")
public IntermediateResultVO<BbsFileAtchResponseDTO> selectUserBBSMasterInf(
    @Parameter(name = "bbsId", description = "게시판 Id", in = ParameterIn.PATH, example="BBSMSTR_AAAAAAAAAA")
    @PathVariable("bbsId") String bbsId)
    throws Exception {

    BbsFileAtchResponseDTO response = bbsAttrbService.selectBBSMasterInf(bbsId, null, BbsDetailRequestType.FILE_ATCH);

    return IntermediateResultVO.success(response);
}
```

응답 객체 전용 VO 도입

```
@Schema(description = "응답 객체 VO (최종 구조로 가는 중간 단계 VO)")
@Getter
@Setter
public class IntermediateResultVO<T> {

    @Schema(description = "응답 코드", example = "0")
    private int resultCode;

    @Schema(description = "응답 메시지", example = "OK")
    private String resultMessage;

    @JsonInclude(JsonInclude.Include.NON_NULL)
    @Schema(description = "응답 객체")
    private T result;

    public static <T> IntermediateResultVO<T> success(T data) {
        IntermediateResultVO<T> result = new IntermediateResultVO<>();
        result.setResultCode(ResponseCode.SUCCESS.getCode());
        result.setResultMessage(ResponseCode.SUCCESS.getMessage());
        result.setResult(data);
        return result;
    }

    public static <T> IntermediateResultVO<T> inputCheckError(T data) {
        IntermediateResultVO<T> result = new IntermediateResultVO<>();
        result.setResultCode(ResponseCode.INPUT_CHECK_ERROR.getCode());
        result.setResultMessage(ResponseCode.INPUT_CHECK_ERROR.getMessage());
        result.setResult(data);
        return result;
    }
}
```

코드 구조 개선

그 외에도 bbs에 관련된 코드들의 구조 개선을 위주로 진행하였습니다.

코드 구조 개선

개선 전 코드

```
@Operation(
    summary = "게시판 마스터 조회",
    description = "게시판 마스터 목록을 조회",
    security = {@SecurityRequirement(name = "Authorization")},
    tags = {"EgovBBSAttributeManageApiController"}
)
@ApiResponses(value = {
    @ApiResponse(responseCode = "200", description = "조회 성공"),
    @ApiResponse(responseCode = "403", description = "인가된 사용자가 아님")
})
@GetMapping(value = "/bbsMaster")
public ResultVO selectBBSMasterInfs(HttpServletRequest request,
    @Parameter(
        in = ParameterIn.QUERY,
        schema = @Schema(type = "object",
            additionalProperties = Schema.AdditionalPropertiesValue.TRUE,
            ref = "#/components/schemas/searchMap"),
        style = ParameterStyle.FORM,
        explode = Explode.TRUE
    ) @RequestParam Map<String, Object> commandMap)
    throws Exception {

    ResultVO resultVO = new ResultVO();
    BoardMasterVO boardMasterVO = new BoardMasterVO();

    boardMasterVO.setSearchCnd((String)commandMap.get("searchCnd"));
    boardMasterVO.setSearchWrds((String)commandMap.get("searchWrds"));

    boardMasterVO.setPageUnit(propertyService.getInt("Globals.pageUnit"));
    boardMasterVO.setPageSize(propertyService.getInt("Globals.pageSize"));

    PaginationInfo paginationInfo = new PaginationInfo();

    paginationInfo.setCurrentPageNo(boardMasterVO.getPageIndex());
    paginationInfo.setRecordCountPerPage(boardMasterVO.getPageUnit());
    paginationInfo.setPageSize(boardMasterVO.getPageSize());

    boardMasterVO.setFirstIndex(paginationInfo.getFirstRecordIndex());
    boardMasterVO.setLastIndex(paginationInfo.getLastRecordIndex());
    boardMasterVO.setRecordCountPerPage(paginationInfo.getRecordCountPerPage());

    Map<String, Object> resultMap = bbsAttrbService.selectBBSMasterInfs(boardMasterVO);
    int totCnt = Integer.parseInt((String)resultMap.get("resultCnt"));

    paginationInfo.setTotalRecordCount(totCnt);

    resultMap.put("paginationInfo", paginationInfo);

    resultVO.setResult(resultMap);
    resultVO.setResultCode(ResponseCode.SUCCESS.getCode());
    resultVO.setResultMessage(ResponseCode.SUCCESS.getMessage());

    return resultVO;
}
```

개선 후 코드

```
@Operation(
    summary = "게시판 마스터 조회",
    description = "게시판 마스터 목록을 조회",
    security = {@SecurityRequirement(name = "Authorization")},
    tags = {"EgovBBSAttributeManageApiController"}
)
@ApiResponses(value = {
    @ApiResponse(responseCode = "200", description = "조회 성공"),
    @ApiResponse(responseCode = "403", description = "인가된 사용자가 아님")
})
@GetMapping(value = "/bbsMaster")
public IntermediateResultVO<BbsAttributeListResponseDTO> selectBBSMasterInfs(@ModelAttribute BbsSearchRequestDTO bbsSearchRequestDTO)
    throws Exception {
    // 1. 페이지 정보 구성
    int pageUnit = propertyService.getInt("Globals.pageUnit");
    int pageSize = propertyService.getInt("Globals.pageSize");

    PaginationInfo paginationInfo = new PaginationInfo();
    paginationInfo.setCurrentPageNo(bbsSearchRequestDTO.getPageIndex());
    paginationInfo.setRecordCountPerPage(pageUnit);
    paginationInfo.setPageSize(pageSize);

    // 2. 서비스 호출 및 응답 객체 구성
    BbsAttributeListResponseDTO response = bbsAttrbService.selectBBSMasterInfs(bbsSearchRequestDTO, paginationInfo);
    paginationInfo.setTotalRecordCount(response.getResultCnt());
    response.setPaginationInfo(paginationInfo);
    return IntermediateResultVO.success(response);
}
```


코드 구조 개선

개선 전 코드

```
public Map<String, Object> selectBBSMasterInfs(BoardMasterVO searchVO) throws Exception {
    List<BoardMasterVO> result = attrbMngDAO.selectBBSMasterInfs(searchVO);
    int cnt = attrbMngDAO.selectBBSMasterInfsCnt(searchVO);

    Map<String, Object> map = new HashMap<String, Object>();

    map.put("resultList", result);
    map.put("resultCnt", Integer.toString(cnt));

    return map;
}
```



개선 후 코드

```
public BbsAttributeListResponseDTO selectBBSMasterInfs(BbsSearchRequestDTO bbsAttributeSearchRequestDTO, PaginationInfo paginationInfo) throws Exception {
    // 도메인 모델(BoardMasterVO) 구성
    BoardMasterVO boardMasterVO = new BoardMasterVO();
    boardMasterVO.setSearchCnd(bbsAttributeSearchRequestDTO.getSearchCnd());
    boardMasterVO.setSearchWrds(bbsAttributeSearchRequestDTO.getSearchWrds());
    boardMasterVO.setPageUnit(paginationInfo.getRecordCountPerPage());
    boardMasterVO.setPageSize(paginationInfo.getPageSize());
    boardMasterVO.setFirstIndex(paginationInfo.getFirstRecordIndex());
    boardMasterVO.setLastIndex(paginationInfo.getLastRecordIndex());
    boardMasterVO.setRecordCountPerPage(paginationInfo.getRecordCountPerPage());
    boardMasterVO.setUseAt("Y");

    List<BoardMasterVO> voList = attrbMngDAO.selectBBSMasterInfs(boardMasterVO);
    int cnt = attrbMngDAO.selectBBSMasterInfsCnt(boardMasterVO);

    List<BbsAttributeDetailResponseDTO> dtoList = voList.stream()
        .map(BbsAttributeDetailResponseDTO::from)
        .collect(Collectors.toList());

    return BbsAttributeListResponseDTO.builder()
        .resultList(dtoList)
        .resultCnt(cnt)
        .build();
}
```


코드 구조 개선

표준프레임워크에 기여를 하며 겪었던 문제

코드 구조 개선

표준프레임워크에 기여를 하며 겪었던 문제

신규 개발자가 해당 템플릿에 기여를 할 때 DB 스키마를 파악하기가 매우 어려움

코드 구조 개선

표준프레임워크에 기여를 하며 겪었던 문제

신규 개발자가 해당 템플릿에 기여를 할 때 DB 스키마를 파악하기가 매우 어려움

☞ README에 참고 문서 리스트를 정리하고 DB 스키마 가이드를 만들어두자!

코드 구조 개선

README 참고 문서 리스트 추가

참고 문서 리스트

문서	경로	한줄 설명
Configuration 설정과 Bean 등록	Docs/configuration-setting-bean-regist.md	자바 기반 @Configuration/@Bean 규칙과 컴포넌트 메시지소스 등 Bean 등록 요령
Aspect 설정	Docs/context-aspect.md	@EnableAspectJAutoProxy로 프록시 기반 AOP 활용 및 AOP 개념·옵션 요약
conext-aspect 설정 변환	Docs/context-aspect-convert.md	예외 처리 AOP(context-aspect.xml) 를 JavaConfig 환하는 방법(Handler/패턴/매니저 등록)
context-common-convert 설정 변환	Docs/context-common-convert.md	컴포넌트 스캔·메시지소스 등 공통 Bean을 context-common.xml에서 JavaConfig로 이전
context-datasource.xml 설정 변환	Docs/context-datasource-convert.md	HSQL 내장 DB·DBCP BasicDataSource 등 데이터소스를 JavaConfig로 변환
[참고] Context Hierarchy(확인 필요)	Docs/context-hierarchy.md	Root/Servlet WebApplicationContext 계층 구조와 로딩 방식 정리
context-idgen.xml 설정 변환	Docs/context-idgen-convert.md	테이블 기반 ID 생성기(EgovTableIdGnrServiceImpl·blockSize·관리 테이블 설정 변환
context-mapper.xml 설정 변환	Docs/context-mapper-convert.md	MyBatis SqlSessionFactory/매퍼·LobHandler 설정 XML→JavaConfig로 이전
context-properties.xml 설정 변환	Docs/context-properties-convert.md	EgovPropertyService에 전역 프로퍼티 (pageUnit·fileStorePath 등) 등록(JavaConfig)
context-transaction.xml 설정 변환	Docs/context-transaction-convert.md	DataSourceTransactionManager·TransactionInterceptor 기반 트랜잭션 AOP 설정(JavaConfig)
context-validator.xml 설정 변환	Docs/context-validator-convert.md	Commons Validator 룰 파일 로딩과 BeanValidator/Factory 설정을 JavaConfig로 변환
context-whitelist.xml 설정 변환	Docs/context-whitelist-convert.md	페이지 링크 화이트리스트를 util:list → List Bean(JavaConfig)으로 전환
[컨트리뷰터 참고 권장] DB 스키마 가이드	Docs/db-schema-guide.md	shtdb.sql 기반 테이블 용도·컬럼(한글)·제약·기본값 등 DB 스키마 가이드
JavaConfig 변환	Docs/java-config-convert.md	web.xml 및 context-*.xml 전반을 JavaConfig로 옮긴 전체 변환 가이드(연결 문서 포함)
Servlet 개념	Docs/servlet.md	서블릿/CGI 개념과 요청→서비스→응답 라이프사이클 및 원리 요약
[참고] 순수 자바 클래스를 이용한 시동이 가능한 이유	Docs/WebApplicationInitializer.md	Servlet 3.x WebApplicationInitializer와 Spring의 @ServletComponentScan 기반 부트스트랩 원리
WebApplicationInitializer 변환	Docs/WebApplicationInitializer-convert.md	web.xml의 리스너·컨텍스트·디스패처 설정을 EgovWebApplicationInitializer(JavaConfig)로 변환

코드 구조 개선

Docs/db-schema-guide.md 작성

LETTNEMPLYRSCRTYESTBS (사용자-권한 매핑)

용도: 보안결정대상ID ↔ 권한코드 매핑

컬럼	타입	제약	기본값	설명
SCRTY_DTRMN_TRGET_ID	VARCHAR(20)	PK, NN, FK→LETTNEMPLYRINFO(EMPLYR_ID)		보안결정대상ID(보통 사용자ID)
MBER_TY_CODE	VARCHAR(15)			회원유형코드
AUTHOR_CODE	VARCHAR(30)	NN, FK→LETTNAUTHORINFO(AUTHOR_CODE)		권한코드

참고: 본 테이블의 PK는 SCRTY_DTRMN_TRGET_ID 단일 컬럼입니다. (복합 PK 아님)

LETTNGNRLMBER (일반회원)

용도: 일반회원 정보

컬럼	타입	제약	기본값	설명
MBER_ID	VARCHAR(20)	PK, NN		회원ID
PASSWORD	VARCHAR(200)	NN		비밀번호(해시)
PASSWORD_HINT	VARCHAR(100)			비밀번호 힌트
PASSWORD_CNSR	VARCHAR(100)			비밀번호 정답
IHIDNUM	VARCHAR(13)			주민등록번호
MBER_NM	VARCHAR(50)	NN		회원명
ZIP	VARCHAR(6)	NN		우편번호
ADRES	VARCHAR(100)	NN		주소
AREA_NO	VARCHAR(4)	NN		지역번호
MBER_STTUS	VARCHAR(15)			회원상태코드
DETAIL_ADRES	VARCHAR(100)			상세주소
END_TELNO	VARCHAR(4)	NN		전화 끝번호
MBTLNUM	VARCHAR(20)	NN		휴대전화번호
GROUP_ID	CHAR(20)	FK→LETTNAUTHORGROUPINFO(GROUP_ID), ON DELETE CASCADE		권한그룹ID
MBER_FXNUM	VARCHAR(20)			팩스번호
MBER_EMAIL_ADRES	VARCHAR(50)			이메일주소

후기

“오픈소스 기여는 내 시간을 쏟기만 하는 낭비되는 시간이 아니다”

후기

“오픈소스 기여는 내 시간을 쏟기만 하는 낭비되는 시간이 아니다”

☞ 오픈소스 생태계에 직접 참여하고, 내 코드에 대한 피드백을 받는 배움의 시간이었습니다.

eGovFrame

내 손으로 개선하는 전자정부 표준프레임워크!

표준프레임워크를 직접 개발하고 수정해보고 싶으신 분
나의 아이디어를 표준프레임워크에 적용시키고 싶으신 분
표준프레임워크에 관심 있는 누구나 환영합니다!

참여대상

전자정부 표준프레임워크(eGovFrame)에 관심 있는 누구나!



참여기간

2025년 5월 19일(월) ~ 9월 19일(금)



후기

감사합니다.