

# 승인 상태변경 시 업무영역 처리

## 승인 상태변경 발생 기준

- 전자결재시스템과 연계된 승인항목은 전자결재시스템으로부터 상태변경 API가 호출되어 상태 변경이 진행된다.
- KALWBS 내부용 승인항목은 승인관리 > 내 승인문서의 '승인', '반려' 버튼 클릭 이벤트를 통해 상태 변경이 진행된다.
- 그 외 '인사발령' 유형에 한해서 '자동승인'이라는 기능을 통해 스케줄 작업으로 자동 상태 변경이 진행된다.
- 모든 승인상태 변경은 내부적으로 아래 명시된 API를 호출하여 동작하도록 설계되어 있으므로, 각 상태 변경에 따른 업무 고유영역 처리 기능 부분을 호출하도록 해야 한다.

## 승인 진행상태에 따른 업무 테이블 데이터 상태 관계

- 업무 테이블의 데이터상태(approve\_status\_type\_value)는 'INVALID' → 'VALID', 'VALID' → 'INVALID'로 처리될 수 있다.
- 승인 테이블의 승인진행상태(approve\_process\_stts\_typ\_vlu)에 따라 업무 테이블 데이터상태 표기 내용 링크 참고
- 업무 테이블 형태(승인상태, 등록승인ID, 만료승인ID, 폐기승인ID)에 따라 승인신청 → 승인완료 시 변경 모습은 링크 참고


## API - 승인처리 및 관련 테이블 상태 업데이트

- 승인진행상태를 파라미터로 명시하여 최초 REQUEST 이후, 모든 상태 변경에 사용되는 API이다.
- PUT [/api/v1/system/approve/approve-process](#)
- Parameter 설명

Request 파라미터	필수여부	비고
approveId	필수	승인ID
processEmpNo	필수	상태 변경 처리자 고유번호
approveProcessSttsTypVlu	옵션	APPROVE (Default), REJECT, CANCEL

- Response 설명

```
{
  "approveId": "2023SELXW12210005", // ID
  "resultFlag": "Complete", // "Complete()", "Success" (APPROVE      ), "Fail"()
  "errorMessage": "" //
}
```

- APPROVE 처리
  -  [PRJ KALWBS-13317](#) - (APPROVAL) 승인처리 및 관련 테이블 상태 업데이트 (APPROVE) (API 개발) 완료
  - 승인자가 2명 이상인 경우, 중간 단계 승인 시에는 "Success"로 리턴한다. 이 때 XXWBS\_APPROVE(승인기본정보) 테이블의 승인진행 상태(approve\_process\_stts\_typ\_vlu)는 'STANDBY'로 유지된다.
  - 최종 단계까지 승인되면 XXWBS\_APPROVE(승인기본정보) 테이블의 승인진행상태(approve\_process\_stts\_typ\_vlu)는 'APPROVE'로 변경되고, 승인유형 정의 분기로직에 의해 업무별 처리 부분을 호출하도록 구현한다.

업무별 처리 부분 개발 샘플 : WBS(근태)

- Approve 내에서 xxwbs\_approve\_type\_domain 기준의 승인 Target Domain 분기처리 진행  
<https://itro.koreanair.com/git/projects/PRJSDSAM/repos/kalwbs-be/browse/hr/infrastructure/aggregation/kalwbs-server/src/main/java/com/koreanair/hr/kalwbs/system/approve/ApproveController.java> , 301Line이후 참고
- <https://itro.koreanair.com/git/projects/PRJSDSAM/repos/kalwbs-be/browse/hr/domain/attendance/src/main/java/com/koreanair/hr/attendance/approve/ApproveSystemReactiveServiceImpl.java>(수정일시 - 240610/근태 램다반영이후)

```

private Mono<ApproveProcessResponse> toUpdateResponse(ApproveProcessDomainResponse approved, String processEmpNo) {
    if (approved.resultFlag().equals(CODE_COMPLETE)) {
        // TODO refactor multi target domain 13864
        final var targetList = approved.approveTargets();
        if (!targetList.isEmpty()) {
            List<Map<String, String>> approveParams = new ArrayList<>();
            targetList.forEach( it -> {
                Map<String, String> param = new HashMap<>();
                param.put("targetTable", it.approveTargetDomainName());
                param.put("registerApproveId", approved.approveId());
                param.put("processEmpNo", processEmpNo);
                param.put("approveStatus", approved.approveStatus());
                approveParams.add(param);
            });
            String domainTarget = approved.approveTargets().get(0).approveTargetDomainName();
            String idx = domainTarget.substring(0, 5);

            if (idx.equals("XXATD")) {
                return commonDutyReactiveService.attendanceApprove(approveParams.get(0))
                    .map(it -> toFinalResponse(it, approved))
                    .onErrorResume(error -> {
                        return xxwbsApproveReactiveService.rollbackApproveProcess(approved.approveId(), processEmpNo, STATUS_STANDBY)
                            .map(it -> toFinalResponse(false, it));
                    });
            }
        }
    }
}

```

○ <https://itro.koreanair.com/git/projects/PRJSDSAM/repos/kalwbs-be/browse/hr/infrastructure/domain/attendance-server/src/main/java/com/koreanair/hr/attendance/leave/controller/XxatdLeaveController.java> 102라인 이후 참고

```

@PutMapping("/approve")
public Boolean leaveApprove(@RequestBody Map<String,String> param){
    String targetTable = param.get("targetTable");


    if(param.get("approveStatus").toUpperCase().equals("APPROVE")) {
        if (targetTable.indexOf("XXATD_DUTY") < 0) {
            return xxatdLeaveService.leaveApprove(param);
        } else {
            return attendanceApplyService.apply(param);
        }
    } else {
        return true;
    }
}

```

○ XXATD\_LEAVE\_% 테이블관련 로직 호출 구현 : <https://itro.koreanair.com/git/projects/PRJSDSAM/repos/kalwbs-be/browse/hr/infrastructure/domain/attendance-server/src/main/java/com/koreanair/hr/attendance/leave/persistence/XxatdLeaveSpringDataRepository.java> 526라인 이후 참고


○ XXATD\_DUTY\_% 테이블관련 로직 호출 구현부분 : <https://itro.koreanair.com/git/projects/PRJSDSAM/repos/kalwbs-be/browse/hr/domain/attendance/src/main/java/com/koreanair/hr/attendance/apply/AttendanceApplyServiceImpl.java> 15라인 이후 참고.

#### • REJECT 처리


-  **PRJ KALWBS-13627** - (APPROVAL) REJECT 공통기능 (API 개발) 완료
- REJECT 시, 승인 단계는 더 이상 진행되지 않는다.
- XXWBS\_APPROVE(승인기본정보) 테이블의 승인진행상태(approve\_process\_stts\_typ\_vlu)는 'REJECT'로 변경되고, 승인유형 정의 분기로직에 의해 업무별 처리 부분을 호출하도록 구현한다.
- 기안자는 REJECT 상태의 승인항목을 최종 CANCEL 처리 할 수 있다. 이 때, PUT [/api/v1/system/approve/approve-process-cancel](#)을 사용한다.

#### • CANCEL 처리

- CANCEL 시, 승인 단계는 더 이상 진행되지 않는다.
- 기안자가 아닌 승인자가 CANCEL 처리하면, XXWBS\_APPROVE\_PROCESS(결재선) 테이블의 해당 단계를 'STANDBY'로 변경하여 임시상태로 만든다.  **PRJ KALWBS-13865** - (APPROVAL) CANCEL - Step 1 - 공통 전처리 (API 개발) 완료
- XXWBS\_APPROVE(승인기본정보) 테이블의 승인진행상태(approve\_process\_stts\_typ\_vlu)가 기존 'APPROVE'인 경우, 데이터를 이전상태로 되돌리는 업무별 처리로직을 구현한다.

 **PRJ KALWBS-13864** - (APPROVAL) CANCEL - Step 2 - 전체승인상태 'APPROVE' 시 (CANCEL\_APPROVAL) (API 개발) 완료

- 기안자가 CANCEL 처리하면 기안자 전용 CANCEL\_REQUEST 까지 자동 처리된다.
  - 최종 XXWBS\_APPROVE(승인기본정보) 테이블의 승인진행상태(approve\_process\_stts\_typ\_vlu)는 'CANCEL'로 변경되고, 승인유형 정의 분기로직에 의해 업무별 처리 부분을 호출하도록 구현한다.

 **PRJ KALWBS-13863** - (APPROVAL) CANCEL - Step 3 (단독호출가능) - 기안자 요청 시 (CANCEL\_REQUEST) (API 개발) 완료

- 첨부파일 정보가 존재할 때, 다른 승인항목과 연결되어 있지 않은 경우에만 테이블에서 삭제를 진행한다. (S3 실제 데이터 삭제 처리 여부는 미정)

## API - 기안자 전용 CANCEL\_REQUEST (단독 사용 가능)

- PUT [/api/v1/system/approve/approve-process-cancel](#)
- Parameter 설명

Request 파라미터	필수여부	비고
approveId	필수	승인ID
processEmpNo	필수	상태 변경 처리자 고유번호
approveProcessSttsTypVlu	필수	CANCEL

- Response 설명

```
{
  "approveId": "2023SELXW12210005", // ID
  "resultFlag": "Complete", // "Complete()", "Fail()"
  "errorMessage": "" //
}
```