

## 02. 개발 템플릿 [오더AA]

(team) ICIS-TR

7 July, 2025

# Table of Contents

1	api (RestController).....	4
2	controller .....	5
3	payload (VO).....	6
4	service .....	7
5	repository.....	8
6	dto .....	9
7	Entity .....	10
8	01. JAVA 주식 템플릿 [오더AA].....	11
9	02. Api 개발 템플릿 [오더AA].....	15
10	03. Controller 개발 템플릿 [오더AA].....	21
11	04. Service 개발 템플릿 [오더AA].....	27
12	05. Repository 개발 템플릿 [오더AA] .....	31
12.1	사용규칙.....	31
12.2	Mybatis Provider .....	31
12.3	01. Repository Inteface 개발 템플릿 [오더AA] .....	32
12.4	02. Mybatis Repository Interface 개발 템플릿[오더AA].....	35
12.4.1	SQL 클래스 제공 메소드 .....	35
12.4.2	참고 Mybatis Provider Code 생성 [오더AA] 에서 Mybatis Provider Code 생성 가이드 .....	42
12.5	03. Bulk Insert 개발 템플릿[오더AA] .....	42
12.6	04. Repository Metrix [오더AA].....	46
13	06. Entity개발 템플릿 [오더AA].....	50
14	07. OpenFeign 개발 템플릿 [오더AA] .....	57
15	08. Sqls.java 개발 템플릿[오더AA].....	60
16	09. DTO개발 템플릿 [오더AA].....	64
16.1	Payload 탑재 DTO Date 포맷 [오더AA] .....	67
16.1.1	1. ICIS-TR 내부 통신 - String / java.sql.Timestamp.....	67
16.1.2	2. 외부 연동 - RestGateway - String / java.sql.Timestamp (Date, Dt, Time으로 종료되는 컬럼 한정).....	67
16.1.3	3. 외부 연동 - RestGateway 외 - String .....	67
16.2	serialVersionUID 생성 [오더AA].....	67

17	10. Payload 개발 템플릿 [오더AA].....	70
18	11. Kafka 호출 개발 템플릿 [오더AA].....	74
18.1	Kafka 호출 Dto KafkaSendInDto 생성 가이드 - KafkaProducerCntr 모듈 사용 .....	74
18.2	Kafka 호출 + ITG Gateway 연동 Dto KafkaSendInDto 생성 가이드 - ItggwCntr모듈 사용 .....	76
19	12. ITG Gateway 연동 Factory 개발 템플릿 .....	81
19.1	1. KOS 연동 Factory 템플릿 .....	81
19.2	2. AnyLink 연동 Factory 템플릿 .....	83
19.3	3. InfiniLink 연동 Factory 템플릿 .....	84
19.4	4. API Link 연동 Factory 템플릿.....	85
20	13. Redis 연동 개발 템플릿.....	88

## 1 api (RestController)

- 사용자 요청을 수신하여 이를 서비스단에 전달한다. 가급적 비즈니스 로직은 처리하지 않아야 하며 요청의 변환, 제어 등의 역할을 수행한다. 직접 Repository에 접근하지 않는다.

## 2 controller

- 비즈니스 분기처리 및 Transaction 처리를 담당한다. 다른 controller, service, openfeign 을 호출하여 서비스 분기처리를 할 수 있다.

### 3 payload (VO)

- 사용자로부터 받은 요청이 도메인 정보와 다를 경우 이를 전달하기 위한 api로 부터 이를 받기 위한 Value Object.

## 4 service

- 컨트롤러로부터 받은 사용자 요청을 다양한 쿼리를 수행하여(Repository 호출) 그 결과를 생성한다. (repository와 1:1 처리 하여 다른 repository 호출 금지)

## 5 repository

- DB에 접근하여 쿼리를 실행하고 결과를 수신한다.



## 6 dto

- 내부적으로 도메인 관련 정보를 전달하거나 변경해야 할 필요가 있을 경우 이를 전달하기 위한 Value Object.

## 7 Entity

- DB와 직접 연결하여 데이터를 주고받을 수 있는 Entity 클래스.
- 01. JAVA 주석 템플릿 [오더AA] (see page 11)
- 02. Api 개발 템플릿 [오더AA] (see page 15)
- 03. Controller 개발 템플릿 [오더AA] (see page 21)
- 04. Service 개발 템플릿 [오더AA] (see page 27)
- 05. Repository 개발 템플릿 [오더AA] (see page 31)
  - 01. Repository Interface 개발 템플릿 [오더AA] (see page 32)
  - 02. Mybatis Repository Interface 개발 템플릿[오더AA] (see page 35)
  - 03. Bulk Insert 개발 템플릿[오더AA] (see page 42)
  - 04. Repository Metrix [오더AA] (see page 46)
- 06. Entity개발 템플릿 [오더AA] (see page 50)
- 07. OpenFeign 개발 템플릿 [오더AA] (see page 57)
- 08. Sqls.java 개발 템플릿[오더AA] (see page 60)
- 09. DTO개발 템플릿 [오더AA] (see page 64)
  - Payload 탑재 DTO Date 포맷 [오더AA] (see page 67)
  - serialVersionUID 생성 [오더AA] (see page 67)
- 10. Payload 개발 템플릿 [오더AA] (see page 70)
- 11. Kafka 호출 개발 템플릿 [오더AA] (see page 74)
- 12. ITG Gateway 연동 Factory 개발 템플릿 (see page 81)
- 13. Redis 연동 개발 템플릿 (see page 88)

## 8 01. JAVA 주석 템플릿 [오더AA]

에러 트레이스 와 소스 검증을 위해 Controller, Api 클래스 주석과 Sqls의 변수 주석에는

@tuxedo tuxedo 모듈 ID 입력한다.

Controller, Api 에는 2023년 5월 25일 이후 생산된 소스는 필수, 그 이전 생산된 소스는 권고

Sqls의 변수 주석은 권고한다.

### Class 라이선스 주석

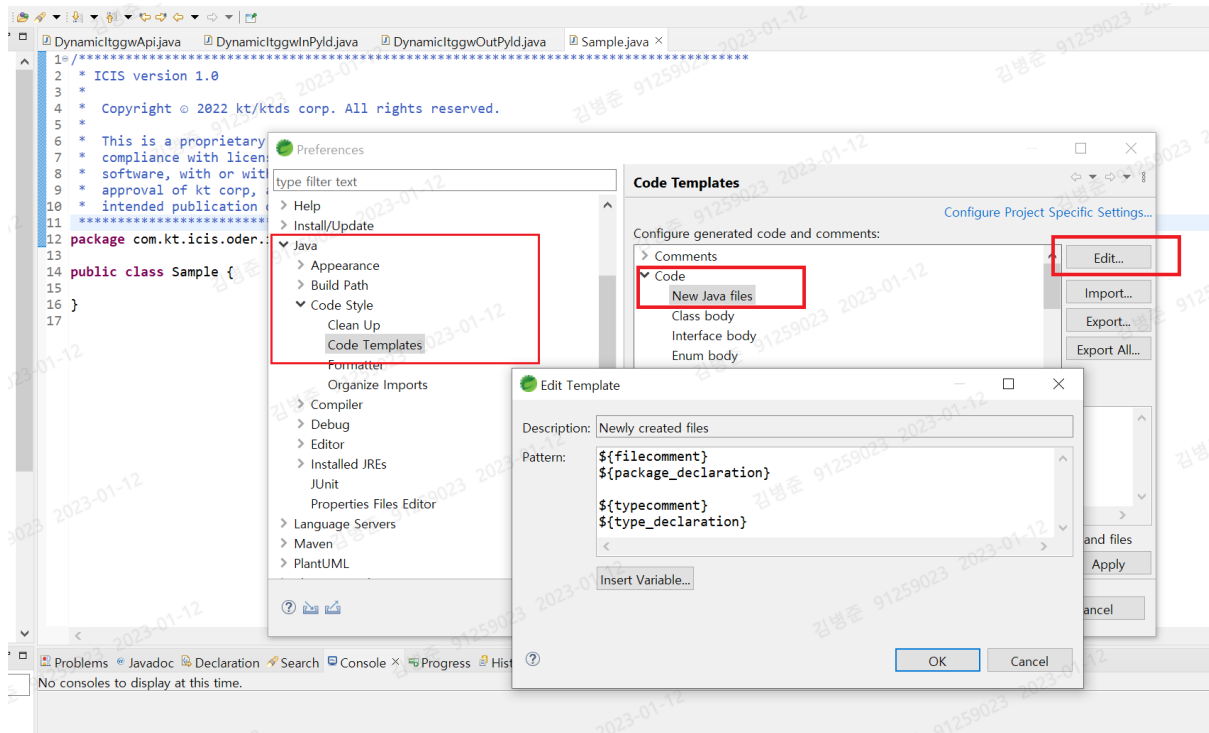
```

/*****
 * ICIS version 1.0
 *
 * Copyright © 2022 kt/ktids corp. All rights reserved.
 *
 * This is a proprietary software of kt corp, and you may not use this file except in
 * compliance with license agreement with kt corp. Any redistribution or use of this
 * software, with or without modification shall be strictly prohibited without prior
written
 * approval of kt corp, and the copyright notice above does not evidence any actual or
 * intended publication of such software.
 *****/

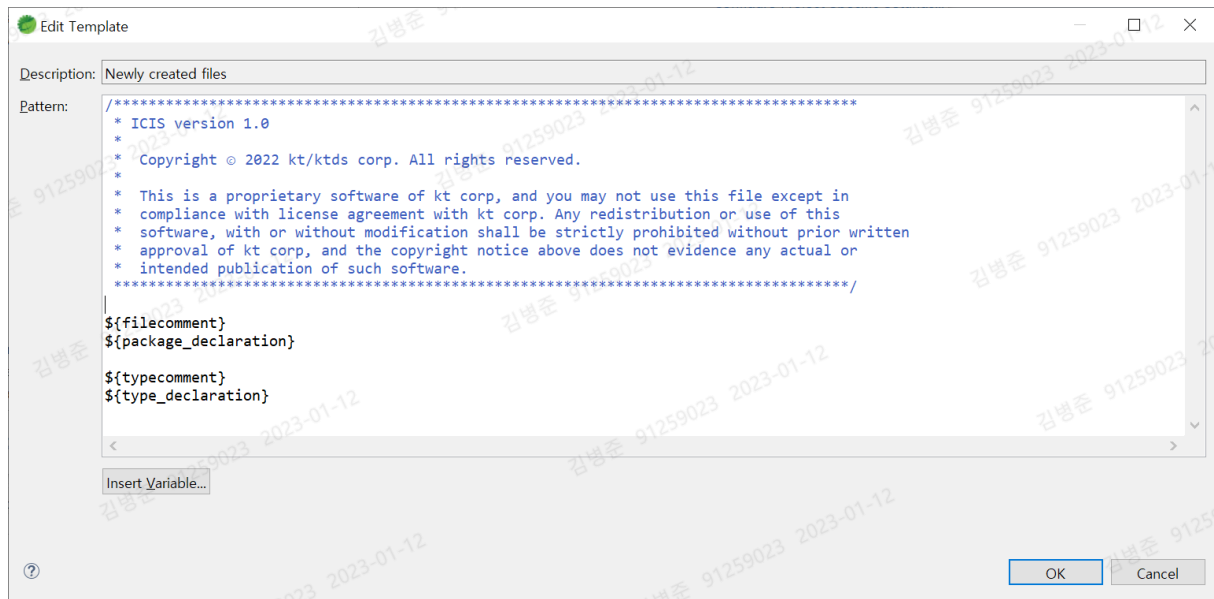
```

Class 라이선스 주석 자동 생성 설정

STS > Window > Preference > Java > Code Style > Code Templates > Code > New Java Files > Edit 진행  
하면 Edit Template 창이 오픈됨



기본값을 삭제하지 않고 상단에 Class 라이선스 주석 덧붙이고 저장



## Class 주석

[형식]

/\*\*

\* 클래스 설명

```
* @tuxedo tuxedo 프로그램id
*/
```

#### [예시]

```
/**
 * 애니팩스 조회 Controller
 * @tuxedo cbor402sf
 */
```

### Method 주석

#### [형식]

```
/**
 * 메소드 설명
 * ${tags} ${return_type}
 * @tuxedo
 */
```

#### [예시]

```
/**
 * 애니팩스 CustID 목록 조회
 * @param queryBsreanyfaxrfndInDs
 * @param request
 * @tuxedo cbor402sf
 * @return ResponseEntity<Object>
 */
```

### Overriding Method 주석

#### [형식]

```
/**
 * 메소드 설명
 * ${tags}
 */
```

#### [예시]

```
/**
 * 애니팩스 CustID 목록 조회
 * @param queryBsreanyfaxrfndInDs
 * @param request
 */
```

## 변수 주석

### [형식]

```
/**  
 *  
 */
```

### [예시]

```
/**  
 * 결과 값 체크  
 * @tuxedo  
 */  
boolean resultChk = true;
```

## 9 02. Api 개발 템플릿 [오더AA]

- Query, Command Api 2개 템플릿
- RequestMapping + PostMapping 의 값이 API의 URL
- RequestMapping 에는 class에서 공통으로 사용하는 URL 정의
- PostMapping 은 method 별 구분되는 URL 정의
- PostMapping 외는 사용금지
- Uri에 @PathVariable 사용 금지, LOB\_CD는 예외

### Query Api 개발 템플릿

```

1  /
   *****
   *****
2  * ICIS version 1.0
3  *
4  * Copyright © 2022 kt/kt ds corp. All rights reserved.
5  *
6  * This is a proprietary software of kt corp, and you may not use this file
   except in
7  * compliance with license agreement with kt corp. Any redistribution or
   use of this
8  * software, with or without modification shall be strictly prohibited
   without prior written
9  * approval of kt corp, and the copyright notice above does not evidence
   any actual or
10 * intended publication of such software.
11
   *****
   *****/
12
13 package com.kt.icis.oder.sample.common.query.api;
14
15 import java.util.List;
16
17 import org.springframework.web.bind.annotation.PostMapping;
18 import org.springframework.web.bind.annotation.RequestBody;
19 import org.springframework.web.bind.annotation.RequestMapping;
20 import org.springframework.web.bind.annotation.RestController;
21
22 import com.kt.icis.oder.common.utils.StringUtils;
23 import com.kt.icis.oder.sample.common.query.controller.CsyscdQryCntr;
24 import com.kt.icis.oder.sample.common.query.payload.in.CsysInfoInQryPyld;
25 import
   com.kt.icis.oder.sample.common.query.payload.in.CsyscdByGrpListInQryPyld;
26 import com.kt.icis.oder.sample.common.query.payload.in.CsyscdByIdInQryPyld;
27 import com.kt.icis.oder.sample.common.query.payload.out.CsyscdOutQryPyld;
28 import
   com.kt.icis.oder.sample.common.query.payload.out.CsyscdOutListQryPyld;

```

```

29 import com.kt.icis.oder.sample.common.query.payload.out.MultiOutListQryPyld;
30 import com.kt.icis.oder.sample.common.query.repository.dto.CsyscdQryDto;
31
32 import io.swagger.v3.oas.annotations.tags.Tag;
33 import lombok.RequiredArgsConstructor;
34
35 /**
36  * Repository Metrix - Sample
37  * 복합키, SELECT
38  */
39 @RestController
40 @Tag(name = "[샘플 : 기준정보 코드 조회]", description = "기준정보 샘플 코드 조회 API 목록")
41 @RequestMapping("/common/query/csycsd")
42 @RequiredArgsConstructor
43 public class CsyscdQryApi {
44     private final CsyscdQryCntr queryCsyscdController;
45
46     /**
47      * 복합키, ID로 조회
48      * @param inPyld
49      * @return
50      */
51     @Operation(summary = "bi_csycsd 테이블 key 조회", description = "bi_csycsd 테이블 key 조회 복합키, ID로 조회")
52     @PostMapping(path = "/retvById")
53     public CsyscdOutQryPyld retvById(@RequestBody CsyscdByIdInQryPyld inPyld) {
54         CsyscdQryDto outDto =
55         queryCsyscdController.findByGrpIdAndCd(inPyld.getCsyscdByIdInQryDs());
56         return CsyscdOutQryPyld.builder().csycsdOutQryDs(outDto).build();
57     }
58
59     /**
60      * 복합키, 조건 1개로 목록 조회
61      * @param inPyld
62      * @return
63      */
64     @Operation(summary = "bi_csycsd 테이블 그룹ID 조회", description = "bi_csycsd 테이블 그룹 ID 조회 복합키, 조건 1개로 목록 조회")
65     @PostMapping(path = "/retvByGrpList")
66     public CsyscdOutListQryPyld retvByGrpList(@RequestBody CsyscdByGrpListInQryPyld inPyld) {
67         List<CsyscdQryDto> outDtoList =
68         queryCsyscdController.findByGrpId(inPyld.getCsyscdByGrpListInQryDs());
69         return
70         CsyscdOutListQryPyld.builder().csycsdOutListQryDs(outDtoList).build();
71     }
72
73     /**
74      * 복합키, Repository query로 조회
75      * @param inPyld
76      * @return
77      */
78     @Operation(summary = "복합키, Repository query로 조회", description = "복합키 , Repository query로 조회")
79     @PostMapping(path = "/retvByCsycsd")

```



```

77     public CsyscdOutListQryPyld retvByCsyscd(@RequestBody
CsyscdByGrpListInQryPyld inPyld) {
78         List<CsyscdQryDto> outDtoList =
queryCsyscdController.selectCsyscd(inPyld.getCsyscdByGrpListInQryDs().getGrp
Id());
79         return
CsyscdOutListQryPyld.builder().csyscdOutListQryDs(outDtoList).build();
80     }
81
82     /**
83      * jdbcTemplate.query 를 사용한 List 조회 샘플
84      * @param inPyld
85      * @return
86      */
87     @Operation(summary = "jdbcTemplate.query 를 사용한 List 조회 샘플",
description = "jdbcTemplate.query 를 사용한 List 조회 샘플")
88     @PostMapping(path = "/retvByCsyscdQueryList")
89     public CsyscdOutListQryPyld retvByCsyscdQueryList(@RequestBody
CsyscdByGrpListInQryPyld inPyld) {
90         CsyscdQryDto inDto = new CsyscdQryDto();
91         inDto.setGrpId(inPyld.getCsyscdByGrpListInQryDs().getGrpId());
92         List<CsyscdQryDto> outDtoList =
queryCsyscdController.selectJoin2(inDto);
93         return
CsyscdOutListQryPyld.builder().csyscdOutListQryDs(outDtoList).build();
94     }
95
96     /**
97      * 멀티 dto 전송 샘플
98      * @param inDs
99      * @return QueryMultiOutListDs
100    */
101    @Operation(summary = "멀티 dto 전송 샘플", description = "멀티 dto 전송 샘플")
102    @PostMapping(path = "/retvbymultitest")
103    public MultiOutListQryPyld retvByMultiTest(@RequestBody
CsysInfoInQryPyld inPyld) {
104        CsyscdQryDto inDto = new CsyscdQryDto();
105
inDto.setGrpId(StringUtils.convertNull2Nbsp(inPyld.getCsysInfoInQryDs().getG
rpId()));
106        return queryCsyscdController.selectMulti(inDto);
107    }
108 }

```

### Command Api 개발 템플릿

```

1  /
*****
*****
2  * ICIS version 1.0
3  *
4  * Copyright © 2022 kt/kt ds corp. All rights reserved.
5  *

```

```

6      * This is a proprietary software of kt corp, and you may not use this file
      except in
7      * compliance with license agreement with kt corp. Any redistribution or
      use of this
8      * software, with or without modification shall be strictly prohibited
      without prior written
9      * approval of kt corp, and the copyright notice above does not evidence
      any actual or
10     * intended publication of such software.
11
12     *****
13     *****/
14
15     package com.kt.icis.oder.sample.common.command.api;
16
17     import org.springframework.web.bind.annotation.PostMapping;
18     import org.springframework.web.bind.annotation.RequestBody;
19     import org.springframework.web.bind.annotation.RequestMapping;
20     import org.springframework.web.bind.annotation.RestController;
21
22     import com.kt.icis.oder.sample.common.command.controller.CsyscdCmdCntr;
23     import com.kt.icis.oder.sample.common.command.payload.in.CsyscdAddInCmdPyld;
24     import com.kt.icis.oder.sample.common.command.payload.in.CsyscdRemoveInCmdPyld;
25     import com.kt.icis.oder.sample.common.command.payload.in.CsyscdSaveInCmdPyld;
26     import com.kt.icis.oder.sample.common.command.payload.out.CsyscdOutCmdPyld;
27     import com.kt.icis.oder.sample.common.command.repository.dto.CsyscdCmdDto;
28
29     import io.swagger.v3.oas.annotations.tags.Tag;
30     import lombok.RequiredArgsConstructor;
31
32     /**
33      * Repository Metrix - Sample
34      * 복합키, CUD
35      */
36     @RestController
37     @Tag(name = "[샘플 : 기준정보 코드 처리]", description = "기준정보 샘플 코드 처리 API 목록")
38     @RequestMapping("/common/command/csyscd")
39     @RequiredArgsConstructor
40     public class CsyscdCmdApi {
41         private final CsyscdCmdCntr csyscdCmdCntr;
42
43         @Operation(summary = "복합키, insert", description = "샘플 API 복합키, insert ")
44         @PostMapping(path = "/addCsyscd")
45         public CsyscdOutCmdPyld addCsyscd(@RequestBody CsyscdAddInCmdPyld inDs)
46         {
47             CsyscdCmdDto outDto =
48             csyscdCmdCntr.insert(inDs.getCsyscdAddInCmdDs());
49             return CsyscdOutCmdPyld.builder().csyscdOutCmdDs(outDto).build();
50         }
51
52         @Operation(summary = "복합키, update", description = "샘플 API 복합키, update")

```

```

50     @PostMapping(path = "/saveCsyscd")
51     public CsyscdOutCmdPyld saveCsyscd(@RequestBody CsyscdSaveInCmdPyld
inDs) {
52         CsyscdCmdDto outDto =
csyscdCmdCntr.update(inDs.getCsyscdSaveInCmdDs());
53         return CsyscdOutCmdPyld.builder().csyscdOutCmdDs(outDto).build();
54     }
55
56
57     @Operation(summary = "복합키, delete", description = "샘플 API 복합키,
delete")
58     @PostMapping(path = "/removeCsyscd")
59     public CsyscdOutCmdPyld removeCsyscd(@RequestBody CsyscdRemoveInCmdPyld
inDs) {
60         int result =
csyscdCmdCntr.delete(inDs.getCsyscdRemoveInCmdDs().getGrpId(),
inDs.getCsyscdRemoveInCmdDs().getCd());
61         return CsyscdOutCmdPyld.builder().deleteResult(result).build();
62     }
63 }

```

신규API 개발이 완료될 경우 Devpilot을 통해 API관리 목록에 추가하여야 한다.

## API관리 - 등록 클릭

The screenshot shows the DevPilot API Management interface. The left sidebar contains navigation links for various management tasks. The main area displays a table of registered APIs with columns for domain, app name, service name, API name, registration status, usage status, start date, expiration date, user, and registration time.

도메인	APP명	서비스명	API명	등록필요 여부	사용 여부	시작일	종료일	등록자	등록일시	수정자	수정일시
icis.oder	NBSS_TORD	icis-oder-trmncust	/trmncust/trmncust	N	Y	2025.02.07	2099.12.31	김상준	2025.02.07 16:27:42	-	-
icis.oder	NBSS_TORD	icis-oder-sample	/sample/conncst	N	Y	2025.01.13	2026.01.31	박동준	2025.01.13 16:46:40	-	-
icis.oder	NBSS_TORD	icis-oder-baseinfo	/baseinfo/conncx	N	Y	2025.01.03	2025.01.04	김병준	2025.01.03 15:37:55	-	-
icis.oder	NBSS_TORD	icis-oder-etcterr-daen	/etcterr-daen	N	Y	2024.12.11	2099.12.31	김상준	2024.12.11 09:00:00	김상준	2024.12.11 09:00:00
icis.oder	NBSS_TORD	icis-oder-baseinfo	/baseinfo/conncx	N	Y	2024.12.11	2099.12.31	김상준	2024.12.11 09:00:00	김상준	2024.12.11 09:00:00
icis.oder	NBSS_TORD	icis-oder-infocomm	/infocomm/conncx	N	Y	2024.12.11	2099.12.31	김상준	2024.12.11 09:00:00	김상준	2024.12.11 09:00:00
icis.oder	NBSS_TORD	icis-oder-wrlincomn-d	/wrlincomn-dae	N	Y	2024.12.11	2099.12.31	김상준	2024.12.11 09:00:00	김상준	2024.12.11 09:00:00
icis.oder	NBSS_TORD	icis-oder-infocomm	/infocomm/conncx	N	Y	2024.12.11	2099.12.31	김상준	2024.12.11 09:00:00	김상준	2024.12.11 09:00:00
icis.oder	NBSS_TORD	icis-oder-infocomm	/infocomm/conncx	N	Y	2024.12.11	2099.12.31	김상준	2024.12.11 09:00:00	김상준	2024.12.11 09:00:00
icis.oder	NBSS_TORD	icis-oder-baseinfo-dai	/baseinfo-daem	N	Y	2024.12.11	2099.12.31	김상준	2024.12.11 09:00:00	김상준	2024.12.11 09:00:00
icis.oder	NBSS_TORD	icis-oder-baseinfo	/baseinfo/conncx	N	Y	2024.12.11	2099.12.31	김상준	2024.12.11 09:00:00	김상준	2024.12.11 09:00:00
icis.oder	NBSS_TORD	icis-oder-baseinfo-dai	/baseinfo-daem	N	Y	2024.12.11	2099.12.31	김상준	2024.12.11 09:00:00	김상준	2024.12.11 09:00:00
icis.oder	NBSS_TORD	icis-oder-baseinfo-dai	/baseinfo-daem	N	Y	2024.12.11	2099.12.31	김상준	2024.12.11 09:00:00	김상준	2024.12.11 09:00:00
icis.oder	NBSS_TORD	icis-oder-baseinfo-dai	/baseinfo-daem	N	Y	2024.12.11	2099.12.31	김상준	2024.12.11 09:00:00	김상준	2024.12.11 09:00:00

- 24 x 7 사용 여부는 24 x 7 절차 시 허용을 의미한다. 기본적으로 조회성의 경우 24 x 7 진입 시 모드 사용으로 처리 된다.

- Skip 사용 여부는 처리성에 한해서 사용일 경우 해당 API는 즉시 처리되지 않고 Async를 통해서 후처리 된다.

**DEV 환경 API 등록**

Type \*  
☒ 단건 ☐ 일괄

도메인 \*  
icis.oder

APP명(일반API) \*  
☒ NBSS\_TORD ☐ NBSS\_TORD\_CO

프로젝트 타입 \*  
Online

서비스명 \*  
icis-oder-sample

API명 \*  
/sample/test

치러성 여부 \*  
☒ 치러성 ☐ 조희성

24x7 사용 여부 \*  
☐ 사용 ☒ 사용 안함

스칼 사용 여부 \*  
☐ 사용 ☒ 사용 안함

사용 시작일 \*  
2025.02.21

사용 종료일 \*  
2040.02.25

설명  
샘플용 API

사용 여부 \*  
☒

취소 확인

## 10 03. Controller 개발 템플릿 [오더AA]

- Query, Command Controller 2개 템플릿
- Command 의 트랜잭션 처리 메소드에는 @Transactional 어노테이션을 정의한다.
- 비즈니스 처리는 Service가 아닌 Controller에서 구현한다.
- @Component Annotation 으로 Class 를 선언한다.
- Controller는 Api, Controller에서만 호출 가능하다.

### Query Controller 호출 예제

```

1  /
   *****
   *****
2  * ICIS version 1.0
3  *
4  * Copyright © 2022 kt/ktids corp. All rights reserved.
5  *
6  * This is a proprietary software of kt corp, and you may not use this file
   except in
7  * compliance with license agreement with kt corp. Any redistribution or
   use of this
8  * software, with or without modification shall be strictly prohibited
   without prior written
9  * approval of kt corp, and the copyright notice above does not evidence
   any actual or
10 * intended publication of such software.
11 *****
12 *****/
13 package com.kt.icis.oder.sample.common.query.controller;
14
15 import java.util.ArrayList;
16 import java.util.List;
17
18 import org.springframework.stereotype.Component;
19
20 import com.kt.icis.oder.common.utils.ConvertUtil;
21 import com.kt.icis.oder.common.utils.StringUtils;
22 import
23
24 com.kt.icis.oder.sample.common.query.payload.in.dto.CsyscdByGrpListInQryDto;
25 import
26     com.kt.icis.oder.sample.common.query.payload.in.dto.CsyscdByIdInQryDto;
27 import com.kt.icis.oder.sample.common.query.payload.out.MultiOutListQryPyld;
28 import
29
30 com.kt.icis.oder.sample.common.query.repository.dto.CcissecdownstatusQryDto;
31 import com.kt.icis.oder.sample.common.query.repository.dto.CsyscdQryDto;

```

```

27  import

com.kt.icis.oder.sample.common.query.repository.entity.CcissecdownstatusQryEntt;
28  import com.kt.icis.oder.sample.common.query.repository.entity.CsyscdQryEntt;
29  import com.kt.icis.oder.sample.common.query.service.CcissecdownstatusQrySvc;
30  import com.kt.icis.oder.sample.common.query.service.CsyscdQrySvc;
31
32  import lombok.RequiredArgsConstructor;
33
34  /**
35   * 복합키 Query Csyscd Controller
36   */
37  @RequiredArgsConstructor
38  @Component
39  public class CsyscdQryCntr {
40      private final CsyscdQrySvc csyscdQrySvc;
41
42      private final CcissecdownstatusQrySvc ccissecdownstatusQrySvc;
43
44      /**
45       * grpId + cd PK로 조회
46       * @param inDs
47       * @return
48       */
49      public CsyscdQryDto findByGrpIdAndCd(CsyscdByIdInQryDto inDs) {
50          CsyscdQryDto outDto = new CsyscdQryDto();
51          CsyscdQryEntt outEntt =
52          csyscdQrySvc.findByGrpIdAndCd(inDs.getGrpId(), inDs.getCd());
53          ConvertUtil.objectToObject(outEntt, outDto);
54          return outDto;
55      }
56
57      /**
58       * grpId로 조회
59       * @param inDs
60       * @return
61       */
62      public List<CsyscdQryDto> findByGrpId(CsyscdByGrpListInQryDto inDs) {
63          List<CsyscdQryDto> outDtoList = new ArrayList<CsyscdQryDto>();
64          List<CsyscdQryEntt> outEnttList =
65          csyscdQrySvc.findByGrpId(inDs.getGrpId());
66          ConvertUtil.objectListToObjectList(outEnttList, outDtoList,
67          CsyscdQryDto.class);
68          return outDtoList;
69      }
70
71      /**
72       * 전체건수 조회
73       * @param inDto
74       * @return
75       */
76      public long count(CsyscdQryDto inDto) {
77          long cnt = 0;
78          if(StringUtils.isEmpty(inDto.getGrpId())) {
79              cnt = csyscdQrySvc.countByGrpId(inDto.getGrpId());
80          }
81      }

```

```

78         else {
79             cnt = csyscdQrySvc.countAll();
80         }
81
82         return cnt;
83     }
84
85     /**
86     * jdbc template를 사용한 join 조회 방법
87     * @param inDto
88     * @return
89     */
90     public List<CsyscdQryDto> selectJoin2(CsyscdQryDto inDto) {
91         List<CsyscdQryDto> outDtoList = new ArrayList<CsyscdQryDto>();
92         List<CsyscdQryEntt> outDomainList = csyscdQrySvc.selectJoin2(inDto);
93         ConvertUtil.objectListToObjectList(outDomainList, outDtoList,
CsyscdQryDto.class);
94         return outDtoList;
95     }
96
97     /**
98     * repository anotation을 사용한 join 조회 방법
99     * @param inDto
100    * @return
101    */
102    public List<CsyscdQryDto> selectJoin(CsyscdQryDto inDto) {
103        List<CsyscdQryDto> outDtoList =
csyscdQrySvc.selectJoin(inDto.getGrpId());
104        return outDtoList;
105    }
106
107    /**
108    * repository anotation을 사용한 조회 방법
109    * @param inDto
110    * @return
111    */
112    public List<CsyscdQryDto> selectCsyscd(String grpId) {
113        List<CsyscdQryDto> outDtoList = csyscdQrySvc.selectCsyscd(grpId);
114        return outDtoList;
115    }
116
117    /**
118    * selectMulti, 멀티 outDS 사용법
119    * @param inDto
120    * @return
121    */
122    public MultiOutListQryPyld selectMulti(CsyscdQryDto inDto) {
123        MultiOutListQryPyld outListDs = new MultiOutListQryPyld();
124
125        List<CsyscdQryEntt> csyscdOutDomainList =
csyscdQrySvc.findByGrpId(inDto.getGrpId());
126        List<CsyscdQryDto> csyscdQryDtoList = new ArrayList<CsyscdQryDto>();
127        ConvertUtil.objectListToObjectList(csyscdOutDomainList,
csyscdQryDtoList, CsyscdQryDto.class);
128        outListDs.setCsyscdOutListQryDs(csyscdQryDtoList);
129

```

```

130         List<CcissecdownstatusQryEntt> ccissecdownstatusQryEnttList =
ccissecdownstatusQrySvc.findAll();
131         List<CcissecdownstatusQryDto> ccissecdownDtoList = new
ArrayList<CcissecdownstatusQryDto>();
132         ConvertUtil.objectListToObjectList(ccissecdownstatusQryEnttList,
ccissecdownDtoList, CcissecdownstatusQryDto.class);
133         outListDs.setCcissecdownstatusOutListQryDs(ccissecdownDtoList);
134
135         return outListDs;
136     }
137 }

```

### Command Controller 호출 예제

```

1      /
*****
*****
2      * ICIS version 1.0
3      *
4      * Copyright © 2022 kt/kt ds corp. All rights reserved.
5      *
6      * This is a proprietary software of kt corp, and you may not use this file
except in
7      * compliance with license agreement with kt corp. Any redistribution or
use of this
8      * software, with or without modification shall be strictly prohibited
without prior written
9      * approval of kt corp, and the copyright notice above does not evidence
any actual or
10     * intended publication of such software.
11
*****
*****/
12
13     package com.kt.icis.oder.sample.common.command.controller;
14
15     import org.springframework.stereotype.Component;
16     import org.springframework.transaction.annotation.Transactional;
17
18     import com.kt.icis.cmmnfrwk.utils.LogUtil;
19     import com.kt.icis.oder.common.utils.ConvertUtil;
20     import
com.kt.icis.oder.sample.common.command.payload.in.dto.CsyscdAddInCmdDto;
21     import
com.kt.icis.oder.sample.common.command.payload.in.dto.CsyscdSaveInCmdDto;
22     import com.kt.icis.oder.sample.common.command.repository.dto.CsyscdCmdDto;
23     import
com.kt.icis.oder.sample.common.command.repository.entity.CsyscdCmdEntt;
24     import com.kt.icis.oder.sample.common.command.service.CsyscdCmdSvc;
25
26     import lombok.RequiredArgsConstructor;
27
28     /**

```



```

29      * 복합키, CUD
30      */
31      @RequiredArgsConstructor
32      @Component
33      public class CsyscdCmdCntr {
34          private final CsyscdCmdSvc csyscdCmdSvc;
35
36          /**
37           * insert
38           * @param inDs
39           * @return
40           */
41          @Transactional
42          public CsyscdCmdDto insert(CsyscdAddInCmdDto inDs) {
43              //builder로 domain 생성
44              CsyscdCmdEntt insertDomain = CsyscdCmdEntt.builder()
45                  .grpId(inDs.getGrpId())
46                  .cd(inDs.getCd())
47                  .name(inDs.getName()).build();
48              int result = csyscdCmdSvc.insertCsyscd(insertDomain);
49              LogUtil.info("update resut="+result);
50              CsyscdCmdEntt outEntt =
51              csyscdCmdSvc.findByGrpIdAndCd(inDs.getGrpId(), inDs.getCd());
52              CsyscdCmdDto outDto = new CsyscdCmdDto();
53              ConvertUtil.objectToObject(outEntt, outDto);
54              return outDto;
55          }
56
57          /**
58           * update
59           * @param inDs
60           * @return
61           */
62          @Transactional
63          public CsyscdCmdDto update(CsyscdSaveInCmdDto inDs) {
64              CsyscdCmdEntt csyscdCmdEntt =
65              csyscdCmdSvc.findByGrpIdAndCd(inDs.getGrpId(), inDs.getCd());
66              csyscdCmdEntt.setName(inDs.getName());
67              int result = csyscdCmdSvc.updateCsyscd(csyscdCmdEntt);
68              LogUtil.info("update resut="+result);
69              CsyscdCmdDto outDto = new CsyscdCmdDto();
70              ConvertUtil.objectToObject(csyscdCmdEntt, outDto);
71              return outDto;
72          }
73
74          /**
75           * delete
76           * @param grpId
77           * @param cd
78           * @return
79           */
80          @Transactional
81          public int delete(String grpId, String cd) {
82              //domain 조회해서 삭제
83              csyscdCmdSvc.deleteCsyscd(csyscdCmdSvc.findByGrpIdAndCd(grpId, cd));
84              return 1;
85          }
86      }

```

84	}
----	---

## 11 04. Service 개발 템플릿 [오더AA]

- Query, Command Service 2개 템플릿
- 비즈니스 처리는 Service가 아닌 Controller에서 구현한다. Service에 비즈니스 처리 및 로직을 구현하지 않는다.
- Service에서는 Validation 체크를 한다.
- @Service Annotation 으로 Class 를 선언한다.

### Query Service 개발 템플릿

```

1  /
   *****
   *****
2      * ICIS version 1.0
3      *
4      * Copyright © 2022 kt/kt ds corp. All rights reserved.
5      *
6      * This is a proprietary software of kt corp, and you may not use this file
   except in
7      * compliance with license agreement with kt corp. Any redistribution or
   use of this
8      * software, with or without modification shall be strictly prohibited
   without prior written
9      * approval of kt corp, and the copyright notice above does not evidence
   any actual or
10     * intended publication of such software.
11
   *****
   *****/
12
13     package com.kt.icis.oder.sample.common.query.service;
14
15     import java.util.List;
16
17     import org.springframework.stereotype.Service;
18
19     import com.kt.icis.oder.sample.common.query.repository.CsyscdQryRepo;
20     import com.kt.icis.oder.sample.common.query.repository.dto.CsyscdQryDto;
21     import com.kt.icis.oder.sample.common.query.repository.entity.CsyscdQryEntt;
22
23     import lombok.RequiredArgsConstructor;
24
25
26     /**
27      * 복합키 Query Csyscd Service
28      */
29     @RequiredArgsConstructor
30     @Service
31     public class CsyscdQrySvc {
32         private final CsyscdQryRepo csyscdQryRepo;

```

```

33 private final CsyscdQryMbtsRepo csyscdQryMbtsRepo;
34
35 /**
36  * pk 조회, id 조합해서 조회
37  * @param grpId
38  * @param cd
39  * @return
40  */
41 public CsyscdQryEntt findByGrpIdAndCd(String grpId, String cd) {
42     return csyscdQryRepo.findByGrpIdAndCd(grpId, cd);
43 }
44
45 /**
46  * column 조회
47  * @param grpId
48  * @return
49  */
50 public List<CsyscdQryEntt> findByGrpId(String grpId) {
51     return csyscdQryRepo.findByGrpId(grpId);
52 }
53
54 /**
55  * 전체건수
56  * @return
57  */
58 public long countAll() {
59     return csyscdQryRepo.count();
60 }
61
62 /**
63  * grpId매핑 raw건수
64  * @param grpId
65  * @return
66  */
67 public long countByGrpId(String grpId) {
68     return csyscdQryRepo.countByGrpId(grpId);
69 }
70
71 /**
72  * jdbc templete, select
73  * @param dto
74  * @return
75  */
76 public List<CsyscdQryEntt> selectJoin2(CsyscdQryDto dto) {
77     return csyscdQryRepo.selectJoin2(dto);
78 }
79
80 /**
81  * repository query,
82  * @param grpId
83  * @return
84  */
85 public List<CsyscdQryDto> selectJoin(String grpId) {
86     return csyscdQryRepo.selectJoin(grpId);
87 }
88
89 /**

```

```

90         * repository query,
91         * @param grpId
92         * @return
93         */
94         public List<CsyscdQryDto> selectCsyscd(String grpId) {
95             return csyscdQryRepo.selectCsyscd(grpId);
96         }
97     }
98

```

### Command Service 개발 템플릿

```

1  /
   *****
   *****
2      * ICIS version 1.0
3      *
4      * Copyright © 2022 kt/ktDs corp. All rights reserved.
5      *
6      * This is a proprietary software of kt corp, and you may not use this file
   except in
7      * compliance with license agreement with kt corp. Any redistribution or
   use of this
8      * software, with or without modification shall be strictly prohibited
   without prior written
9      * approval of kt corp, and the copyright notice above does not evidence
   any actual or
10     * intended publication of such software.
11
   *****
   *****/
12
13     package com.kt.icis.oder.sample.common.command.service;
14
15     import org.springframework.stereotype.Service;
16
17     import com.kt.icis.oder.sample.common.command.repository.CsyscdCmdRepo;
18     import
   com.kt.icis.oder.sample.common.command.repository.entity.CsyscdCmdEntt;
19
20     import lombok.RequiredArgsConstructor;
21
22     /**
23      * 복합키 Command Csyscd Service
24      */
25     @RequiredArgsConstructor
26     @Service
27     public class CsyscdCmdSvc {
28         private final CsyscdCmdRepo csyscdCmdRepo;
29
30         /**

```

```

31      * findById
32      * @param grpId
33      * @param cd
34      * @return
35      */
36      public CsyscdCmdEntt findByIdAndCd(String grpId, String cd) {
37          return csyscdCmdRepo.findByIdAndCd(grpId, cd);
38      }
39
40      /**
41      * insert
42      * @param csyscdCmdEntt
43      * @return
44      */
45      public int insertCsyscd(CsyscdCmdEntt csyscdCmdEntt) {
46          return csyscdCmdRepo.insertCsyscd(csyscdCmdEntt);
47      }
48
49      /**
50      * update
51      * @param csyscdCmdEntt
52      * @return
53      */
54      public int updateCsyscd(CsyscdCmdEntt csyscdCmdEntt) {
55          return csyscdCmdRepo.updateCsyscd(csyscdCmdEntt);
56      }
57
58      /**
59      * delete
60      * @param csyscdCmdEntt
61      * @return
62      */
63      public int deleteCsyscd(CsyscdCmdEntt csyscdCmdEntt) {
64          return csyscdCmdRepo.deleteCsyscd(csyscdCmdEntt);
65      }
66  }
67

```

## 12 05. Repository 개발 템플릿 [오더AA]

### 12.1 사용규칙

1. 01. 프로젝트 구조 및 패키지 명명규칙[오더AA]<sup>1</sup> 로 패키징 한다.
2. 02. Java 파일 명명규칙 [오더AA]<sup>2</sup> 으로 명명한다.
3. Mybatis Repository는 DynamicSQL 사용으로 한다.
4. DynamicSQL 이외의 Repository --> PagingAndSortingRepository / CrudRepository를 상속 받아 처리 한다.
5. Function은 대표 테이블로 Repository 한다.
6. Insert/Update/Delete는 Repository의 method에는 @Modifying annotation을 선언한다.

01. Repository Interface 개발 템플릿 [오더AA] (see page 32)

02. Mybatis Repository Interface 개발 템플릿[오더AA] (see page 35)

### 12.2 Mybatis Provider

동적쿼리를 적용 할 수 있도록 Mybatis 3이후 지원 하는 Provider를 사용하여 다음과 같이 개발을 진행 한다.

어노테이션	기능	예제
@InsertProvider	Insert	<pre> public String insertPersonSql() {     String sql = new SQL()         .INSERT_INTO("PERSON");         .VALUES("ID, FIRST_NAME", "#{id}, #{firstName}");         .VALUES("LAST_NAME", "#{lastName}")         .toString();     return sql; } </pre>

1 <https://wiki.dspace.kt.co.kr/pages/viewpage.action?pagelId=74342435>

2 <https://wiki.dspace.kt.co.kr/pages/viewpage.action?pagelId=74350113>

어노테이션	기능	예제
@UpdateProvider	update	<pre> public String updatePersonSql() {     return new SQL() {{         UPDATE("PERSON");         SET("FIRST_NAME = #{firstName}");         WHERE("ID = #{id}");     }}.toString(); } </pre>
@DeleteProvider	delete	<pre> public String deletePersonSql() {     return new SQL() {{         DELETE_FROM("PERSON");         WHERE("ID = #{id}");     }}.toString(); } </pre>
@SelectProvider	select	<pre> public String selectPersonLike(final String id, final String firstName, final String lastName) {     return new SQL() {{         SELECT("P.ID, P.USERNAME, P.PASSWORD, P.FIRST_NAME, P.LAST_NAME");         FROM("PERSON P");         if (id != null) {             WHERE("P.ID like #{id}");         }     }}.toString(); } </pre>

## 12.3 01. Repository Interface 개발 템플릿 [오더AA]

- Query, Command Repository Interface 2개 템플릿
- Query는 PagingAndSortingRepository 를 상속 받고 Command는 CrudRepository 상속 받아 interface 로 구현한다. 이때 Entity만 사용한다.
- interface에는 @Repository annotation을 선언한다.
- Command의 method 가 Insert/Update/Delete 을 실행하면 @Modifying annotation을 선언한다.
- Query annotation을 사용하면 SQL을 직접 작성 가능하다.



## Query Repository Interface 개발 템플릿

```

1  /
   *****
   *****
2  * ICIS version 1.0
3  *
4  * Copyright © 2022 kt/kt ds corp. All rights reserved.
5  *
6  * This is a proprietary software of kt corp, and you may not use this file
   except in
7  * compliance with license agreement with kt corp. Any redistribution or
   use of this
8  * software, with or without modification shall be strictly prohibited
   without prior written
9  * approval of kt corp, and the copyright notice above does not evidence
   any actual or
10 * intended publication of such software.
11
   *****
   *****/
12
13 package com.kt.icis.oder.sample.common.query.repository;
14
15 import java.util.List;
16
17 import org.springframework.data.domain.Pageable;
18 import org.springframework.data.jdbc.repository.query.Query;
19 import org.springframework.data.repository.PagingAndSortingRepository;
20 import org.springframework.stereotype.Repository;
21
22 import
23
24 com.kt.icis.oder.sample.common.query.payload.in.dto.CsyscdByGrpListInQryDto;
25 import com.kt.icis.oder.sample.common.query.repository.dto.CsyscdQryDto;
26 import com.kt.icis.oder.sample.common.query.repository.entity.CsyscdQryEntt;
27 import com.kt.icis.oder.sample.common.query.repository.sql.CsyscdQrySql;
28
29 /**
30  * Query Csyscd Jdbc Repository interface
31  *
32  */
33 @Repository
34 public interface CsyscdQryRepo extends
35     PagingAndSortingRepository<CsyscdQryEntt, String> {
36
37     CsyscdQryEntt findByGrpIdAndCd(String grpId, String cd);
38
39     List<CsyscdQryEntt> findByGrpId(String grpId);
40
41     List<CsyscdQryEntt> findByGrpId(String grpId, Pageable pageable);

```

```

40         int countByGrpId(String grpId);
41
42         @Query(CsyscdQrySql.SELECT_CSYS CDGRP_JOIN_CSYS CD)
43         List<CsyscdQryDto> selectJoin(String grpId);
44
45         @Query(CsyscdQrySql.SELECT_CSYS CDGRP_JOIN_CSYS CD)
46         List<CsyscdQryEntt> selectJoin2(CsyscdQryDto dto);
47
48         @Query(CsyscdQrySql.SELECT_CSYS CD)
49         List<CsyscdQryDto> selectCsyscd(CsyscdByGrpListInQryDto inDs);
50     }

```

### Command Repository Interface 개발 템플릿

```

1      /
      *****
      *****
2      * ICIS version 1.0
3      *
4      * Copyright © 2022 kt/kt ds corp. All rights reserved.
5      *
6      * This is a proprietary software of kt corp, and you may not use this file
      except in
7      * compliance with license agreement with kt corp. Any redistribution or
      use of this
8      * software, with or without modification shall be strictly prohibited
      without prior written
9      * approval of kt corp, and the copyright notice above does not evidence
      any actual or
10     * intended publication of such software.
11
      *****
      *****/
12
13     package com.kt.icis.oder.sample.common.command.repository;
14
15     import org.springframework.data.jdbc.repository.query.Modifying;
16     import org.springframework.data.jdbc.repository.query.Query;
17     import org.springframework.data.repository.CrudRepository;
18     import org.springframework.stereotype.Repository;
19
20     import
      com.kt.icis.oder.sample.common.command.repository.entity.CsyscdCmdEntt;
21     import com.kt.icis.oder.sample.common.command.repository.sql.CsyscdCmdSql;
22
23     /**
24      * Command Csyscd Jdbc Repository interface
25      */
26     @Repository
27     public interface CsyscdCmdRepo extends CrudRepository<CsyscdCmdEntt, String>
      {
28         CsyscdCmdEntt findByGrpIdAndCd(String grpId, String cd);
29     }

```

```

30      @Modifying
31      @Query(CsyscdCmdSql.INSERT_CSYS_CD)
32      int insertCsyscd(CsyscdCmdEntt inDs);
33
34      @Modifying
35      @Query(CsyscdCmdSql.UPDATE_CSYS_CD)
36      int updateCsyscd(CsyscdCmdEntt inDs);
37
38      @Modifying
39      @Query(CsyscdCmdSql.DELETE_CSYS_CD)
40      int deleteCsyscd(CsyscdCmdEntt inDs);
41  }
42

```

## 12.4 02. Mybatis Repository Interface 개발 템플릿[오더AA]

### 12.4.1 SQL 클래스 제공 메소드

메소드	설명
<ul style="list-style-type: none"> <li>• <code>SELECT(String)</code></li> <li>• <code>SELECT(String...)</code></li> </ul>	<p><code>SELECT</code> 절로 시작하거나 덧붙이기. 한번 이상 호출할 수 있고 파라미터는 <code>SELECT</code> 절에 덧붙일것이다. 파라미터는 칼럼과 별칭의 목록을 콤마를 구분자로 나열하지만 드라이버에 따라 처리가 안될수도 있다.</p>
<ul style="list-style-type: none"> <li>• <code>SELECT_DISTINCT(String)</code></li> <li>• <code>SELECT_DISTINCT(String...)</code></li> </ul>	<p><code>SELECT</code> 절로 시작하거나 덧붙이기. 생성된 쿼리에 <code>DISTINCT</code> 를 추가한다. 한번 이상 호출할 수 있고 파라미터는 <code>SELECT</code> 절에 덧붙일것이다. 파라미터는 칼럼과 별칭의 목록을 콤마를 구분자로 나열하지만 드라이버에 따라 처리가 안될수도 있다.</p>
<ul style="list-style-type: none"> <li>• <code>FROM(String)</code></li> <li>• <code>FROM(String...)</code></li> </ul>	<p><code>FROM</code> 절로 시작하거나 덧붙이기. 한번 이상 호출할 수 있고 파라미터는 <code>FROM</code> 절에 덧붙일것이다. 파라미터는 테이블명과 별칭이지만 드라이버에 따라 처리가 안될수도 있다.</p>

메소드	설명
<ul style="list-style-type: none"> <li>• <code>JOIN(String )</code></li> <li>• <code>JOIN(String ...)</code></li> <li>• <code>INNER_JOIN(String)</code></li> <li>• <code>INNER_JOIN(String...)</code></li> <li>• <code>LEFT_OUTER_JOIN(String )</code></li> <li>• <code>LEFT_OUTER_JOIN(String ...)</code></li> <li>• <code>RIGHT_OUTER_JOIN(Strin g)</code></li> <li>• <code>RIGHT_OUTER_JOIN(Strin g...)</code></li> </ul>	<p>새로운 적절한 타입의 <code>JOIN</code> 절을 추가. 타입은 호출하는 메소드에 따라 다르다. 파라미터는 조인을 구성하는 칼럼과 조건이다.</p>
<ul style="list-style-type: none"> <li>• <code>WHERE(Strin g)</code></li> <li>• <code>WHERE(Strin g...)</code></li> </ul>	<p>새로운 <code>WHERE</code> 절과 <code>AND</code> 로 합친 조건을 덧붙인다. 여러번 호출할 수 있고 <code>AND</code> 를 사용할때 마다 새로운 조건을 합친다. <code>OR</code> 로 분리하려면 <code>OR()</code> 를 사용하자.</p>
<code>OR()</code>	<p><code>OR</code> 를 사용해서 <code>WHERE</code> 조건절을 분리한다. 여러번 호출할 수 있지만 한개의 로우에 여러번 호출할 경우 에러를 가진 <code>SQL</code> 을 만들수도 있다.</p>
<code>AND()</code>	<p><code>AND</code> 를 가진 <code>WHERE</code> 절을 분리한다. 여러번 호출할 수 있지만 여러번 호출할 경우 에러를 가 진 <code>SQL</code> 을 만들수도 있다. <code>WHERE</code> 와 <code>HAVING</code> 두가지는 <code>AND</code> 를 자동으로 붙이기 때문에 흔하게 사용하지 않고 필요할때만 사용한다.</p>

메소드	설명
<ul style="list-style-type: none"> <li>GROUP_BY(String)</li> <li>GROUP_BY(String...)</li> </ul>	새로운 GROUP BY 절을 콤마를 더해서 덧붙인다. 여러번 호출할 수 있고 개별 조건을 콤마를 붙여서 합칠 수 있다.
<ul style="list-style-type: none"> <li>HAVING(String)</li> <li>HAVING(String...)</li> </ul>	새로운 HAVING 절을 AND를 더해서 덧붙인다. 여러번 호출할 수 있고 개별 조건을 AND 를 붙여서 합칠 수 있다. OR 로 분리하기 위해서는 OR 를 사용하자.
<ul style="list-style-type: none"> <li>ORDER_BY(String)</li> <li>ORDER_BY(String...)</li> </ul>	새로운 ORDER BY 절을 콤마를 더해서 덧붙인다. 여러번 호출할 수 있고 개별조건을 콤마를 붙여서 합칠수 있다.
<ul style="list-style-type: none"> <li>LIMIT(String)</li> <li>LIMIT(int)</li> </ul>	Appends a LIMIT clause. This method valid when use together with SELECT(), UPDATE() and DELETE(). And this method is designed to use together with OFFSET() when use SELECT(). (Available since 3.5.2)
<ul style="list-style-type: none"> <li>OFFSET(String)</li> <li>OFFSET(long)</li> </ul>	Appends a OFFSET clause. This method valid when use together with SELECT(). And this method is designed to use together with LIMIT(). (Available since 3.5.2)
<ul style="list-style-type: none"> <li>OFFSET_ROWS(String)</li> <li>OFFSET_ROWS(long)</li> </ul>	Appends a OFFSET n ROWS clause. This method valid when use together with SELECT(). And this method is designed to use together with FETCH_FIRST_ROWS_ONLY(). (Available since 3.5.2)
<ul style="list-style-type: none"> <li>FETCH_FIRST_ROWS_ONLY(String)</li> <li>FETCH_FIRST_ROWS_ONLY(int)</li> </ul>	Appends a FETCH FIRST n ROWS ONLY clause. This method valid when use together with SELECT(). And this method is designed to use together with OFFSET_ROWS(). (Available since 3.5.2)

메소드	설명
<code>DELETE_FROM(String)</code>	delete구문을 시작하고 삭제할 테이블을 명시한다. 대개는 WHERE구문이 뒤에 붙여서 삭제한 대상 조건을 명시한다.
<code>INSERT_INTO(String)</code>	insert구문을 시작하고 입력한 테이블을 명시한다. <code>VALUES()</code> or <code>INTO_COLUMNS()</code> and <code>INTO_VALUES()</code> 메소드 호출은 여러번 해서 입력한 칼럼과 값을 명시한다.
<ul style="list-style-type: none"> <li><code>SET(String)</code></li> <li><code>SET(String..)</code></li> </ul>	update구문에서 "set" 대상 목록을 덧붙인다.
<code>UPDATE(String)</code>	update구문을 시작하고 update수정할 테이블을 명시한다. 수정할 칼럼과 값을 명시하기 위해 <code>SET()</code> 메소드 호출을 여러번 할수 있고 대개 수정할 대상 데이터를 한정하기 위해 <code>WHERE()</code> 메소드도 호출한다.
<code>VALUES(String, String)</code>	insert구문에 덧붙인다. 첫번째 파라미터는 입력한 칼럼이고 두번째 파라미터는 입력할 값이다.
<code>INTO_COLUMNS(String...)</code>	insert 구문에 columns 절을 추가한다. 반드시 <code>INTO_VALUES()</code> 와 함께 호출되어야 한다.
<code>INTO_VALUES(String...)</code>	insert 구문에 values 절을 추가한다. 반드시 <code>INTO_COLUMNS()</code> 와 함께 호출되어야 한다.
<code>ADD_ROW()</code>	Add new row for bulk insert. (Available since 3.5.2)

`LIMIT` SQL 클래스는, `OFFSET` 및 절을 생성된 명령문에 있는 그대로 작성한다는 점 `OFFSET n ROWS`에 유의해야 합니다. `FETCH FIRST n ROWS ONLY` 즉, 라이브러리는 이러한 절을 직접 지원하지 않는 데이터베이스에 대해 해당 값을 정규화하려고 시도하지 않습니다. 따라서 사용자가 대상 데이터베이스가 이러한 절을 지원하는지 여부를 이해하는 것이 매우 중요합니다. 대상 데이터베이스가 이러한 절을 지원하지 않는 경우 이 지원을 사용하면 런타임 오류가 있는 SQL이 생성될 수 있습니다.

#### CbovrfresultQrySql.java 예제

```

1 package com.kt.icis.oder.sample.common.query.repository.sql;
2

```

```

3  import org.apache.ibatis.jdbc.SQL;
4
5  import
    com.kt.icis.oder.sample.common.query.payload.in.dto.CbovrfresultInQryDto;
6
7  public class CbovrfresultQrySql {
8      public static final String SELECT_WORK_DATE = "WORK_DATE as workDate";
9      public static final String SELECT_VRF_ID = "VRF_ID as vrfId";
10     public static final String SELECT_TARGET_CNT = "TARGET_CNT as targetCnt";
11     public static final String SELECT_ERR_CNT = "ERR_CNT as errCnt";
12     public static final String SELECT_REG_DATE = "REG_DATE as regDate";
13     public static final String WHERE_REG_DATE = "WORK_DATE = #{workDate}";
14     public static final String WHERE_VRF_ID = "VRF_ID = #{vrfId}";
15     public static final String JOIN_BI_CBOVRFRESULT = "BI_CBOVRFRESULT";
16
17     public static final String SELECT_CBOVRFRESULT = ""
18         SELECT
19             WORK_DATE,
20             VRF_ID,
21             TARGET_CNT,
22             ERR_CNT,
23             REG_DATE
24         FROM BI_CBOVRFRESULT
25         WHERE WORK_DATE = :workDate // String param 매핑
26         AND VRF_ID = :vrfId // String param 매핑
27         "";
28
29     public static final String SELECT_CBOVRFRESULT_OBJECT = ""
30         SELECT
31             WORK_DATE,
32             VRF_ID,
33             TARGET_CNT,
34             ERR_CNT,
35             REG_DATE
36         FROM BI_CBOVRFRESULT
37         WHERE WORK_DATE = :#{inDs.workDate} // inDs Object 매핑
38         AND VRF_ID = :#{inDs.vrfId} // inDs Object 매핑
39         "";
40
41     public String selectCbovrfresultMbtsSql(final CbovrfresultInQryDto
inDs) {
42         return new SQL() {{
43             SELECT("WORK_DATE as workDate");
44             SELECT("VRF_ID as vrfId");
45             SELECT("TARGET_CNT as targetCnt");
46             SELECT("ERR_CNT as errCnt");
47             SELECT("REG_DATE as regDate");
48             FROM("BI_CBOVRFRESULT");
49             if (inDs.getVrfId() != null) {
50                 WHERE("WORK_DATE = #{workDate} "); // inDs Object 매핑
51             }
52             if (inDs.getWorkDate() != null) {
53                 WHERE("VRF_ID = #{vrfId}"); // inDs Object 매핑
54             }
55         }}.toString();
56     }
57

```

```

58     public String selectCbovrresultMbtsFinalSql(final CbovrresultInQryDto
inDs) {
59         return new SQL() {{
60             SELECT(SELECT_WORK_DATE);
61             SELECT(SELECT_VRF_ID);
62             SELECT(SELECT_TARGET_CNT);
63             SELECT(SELECT_ERR_CNT);
64             SELECT(SELECT_REG_DATE);
65             FROM(JOIN_BI_CBOVRFRESULT);
66             if (inDs.getVrfId() != null) {
67                 WHERE(WHERE_REG_DATE);
68             }
69             if (inDs.getWorkDate() != null) {
70                 WHERE(WHERE_VRF_ID);
71             }
72         }}.toString();
73     }
74 }

```

#### CbovrresultQryMbtsRepo.java 예제

```

1     package com.kt.icis.oder.sample.common.query.repository;
2
3     import java.util.List;
4
5     import org.apache.ibatis.annotations.Mapper;
6     import org.apache.ibatis.annotations.SelectProvider;
7
8     import
9         com.kt.icis.oder.sample.common.query.payload.in.dto.CbovrresultInQryDto;
10        com.kt.icis.oder.sample.common.query.repository.dto.CbovrresultQryDto;
11    import
12        com.kt.icis.oder.sample.common.query.repository.sql.CbovrresultQrySql;
13
14    /**
15     * Query AcItgContEqpAdrDtl Repository Impl
16     */
17    @Mapper
18    public interface CbovrresultQryMbtsRepo {
19        /**
20         * Mybatis Provider 사용을 위한 샘플 method
21         * @param inDs
22         * @return
23         */
24        @SelectProvider(type = CbovrresultQrySql.class, method =
25            "selectCbovrresultMbtsSql")
26        List<CbovrresultQryDto> selectCbovrresultMbts(CbovrresultInQryDto
27            inDs);
28
29        /**
30         * Mybatis Provider 사용을 위한 샘플 final method
31         * @param inDs

```



```

28         * @return
29         */
30         @SelectProvider(type = CbovrresultQrySql.class, method =
"selectCbovrresultMbtsFinalSql") // type = sql.class 선언 , method = sql
method 선언
31         List<CbovrresultQryDto>
selectCbovrresultMbtsFinal(CbovrresultInQryDto inDs);
32     }

```

#### insert SQL.java 예제

```

public String insertPersonSql() {
    String sql = new SQL()
        .INSERT_INTO("PERSON")
        .VALUES("ID, FIRST_NAME", "#{id}, #{firstName}")
        .VALUES("LAST_NAME", "#{lastName}")
        .toString();
    return sql;
}

```

#### update SQL.java 예제

```

public String updatePersonSql() {
    return new SQL() {{
        UPDATE("PERSON");
        SET("FIRST_NAME = #{firstName}");
        WHERE("ID = #{id}");
    }}.toString();
}

```

#### delete SQL.java 예제

```

public String deletePersonSql() {
    return new SQL() {{
        DELETE_FROM("PERSON");
        WHERE("ID = #{id}");
    }}.toString();
}

```

**select SQL.java 예제**

```

public String selectPersonLike(final String id, final String firstName, final String
lastName) {
    return new SQL() {{
        SELECT("P.ID, P.USERNAME, P.PASSWORD, P.FIRST_NAME, P.LAST_NAME");
        FROM("PERSON P");
        if (id != null) {
            WHERE("P.ID like #{id}");
        }
    }}.toString();
}

```

### 12.4.2 참고 [Mybatis Provider Code 생성 \[오더AA\]](#)<sup>3</sup> 에서 Mybatis Provider Code 생성 가이드

## 12.5 03. Bulk Insert 개발 템플릿[오더AA]

- 처리 건수 별 소요시간 결과 Mybatis Provider 사용 시간이 단축 되었음을 확인 할 수 있습니다. (단위 ms)

처리 방법	1천건	1만건	10만건
for 문으로 Insert 단건 호출	13,659	125,879	1,268,208
Mybatis Provider를 통한 다량호출	168	1,017	20,382

<sup>3</sup> <https://wiki.dspace.kt.co.kr/pages/viewpage.action?pagelId=106795425>

	for 문으로 건수 만큼 실행시	mybatis로 실행시
Repo.java	<p>갯수만큼 반복</p> <div> <b>Service 파일</b> <pre> for (int i=0; i&lt;1000;i++) {     InsertTestDto     dto = new InsertTestDto();      dto.setGrpId("ITQ1"); //     insert test query 1     dto.setCd("G"+i+" "); //test      dto.setName("TEST"); //test      repo.insertTest(dto); } </pre> </div>	<p>여기까진 가이드와 동일하나 입력 파라미터가 list</p> <div> <b>Repository 파일</b> <pre> @Mapper public interface InsertTestMbtsRepo {     @InsertProvider(type = InsertTestSql.class, method = "insertTestMbts")     public void     insertTest(ArrayList&lt;InsertTestDto&gt; list); } </pre> </div>

	for 문으로 건수 만큼 실행시	mybatis로 실행시
SQL. java	<p>기존 mbts 가이드 와 동일</p> <div> <p><b>sql 파일</b></p> <pre> public static final String TEST_BULKINSERT=""          INSERT INTO         BI_CSYS_CD         (GRP_ID, CD, NAME)         VALUES          (:#{#dto.grpId}, :#{#dto.cd}         , :#{#dto.name})         """; </pre> </div>	<p>리스트 1건일때 대비</p> <div> <p><b>sql파일</b></p> <pre> public class InsertTestSql {      public String     insertTestMbts(List&lt;InsertTestDto&gt; list)     {         InsertTestDto dto = list.get(0);         SQL sql = new SQL()             .INSERT_INTO("BI_CSYS_CD")             .INTO_COLUMNS("GRP_ID",             "CD", "NAME")             .INTO_VALUES(                 new                 StringBuilder("\'").append(StringUtils.c onvertNull2Nbsp(dto.getGrpId())).append("\'")                 .toString(),                 new                 StringBuilder("\'").append(StringUtils.c onvertNull2Nbsp(dto.getCd())).append("\'")                 .toString(),                 new                 StringBuilder("\'").append(StringUtils.c onvertNull2Nbsp(dto.getName())).append("\'")                 .toString()             );          if (list.size()&gt;1) {             for (int i =1; i&lt;list.size() ; i++) {                 dto = list.get(i);                 sql.ADD_ROW()                 .INTO_VALUES(                     new                     StringBuilder("\'").append(StringUtils.c onvertNull2Nbsp(dto.getGrpId())).append("\'")                     .toString(),                     new                     StringBuilder("\'").append(StringUtils.c onvertNull2Nbsp(dto.getCd())).append("\'")                     .toString(),                     new                     StringBuilder("\'").append(StringUtils.c onvertNull2Nbsp(dto.getName())).append("\'")                     .toString()                 );             }         }     } } </pre> </div>

	for 문으로 건수 만큼 실행시	mybatis로 실행시
		<pre>        }         return sql.toString();     } }</pre>

---

---

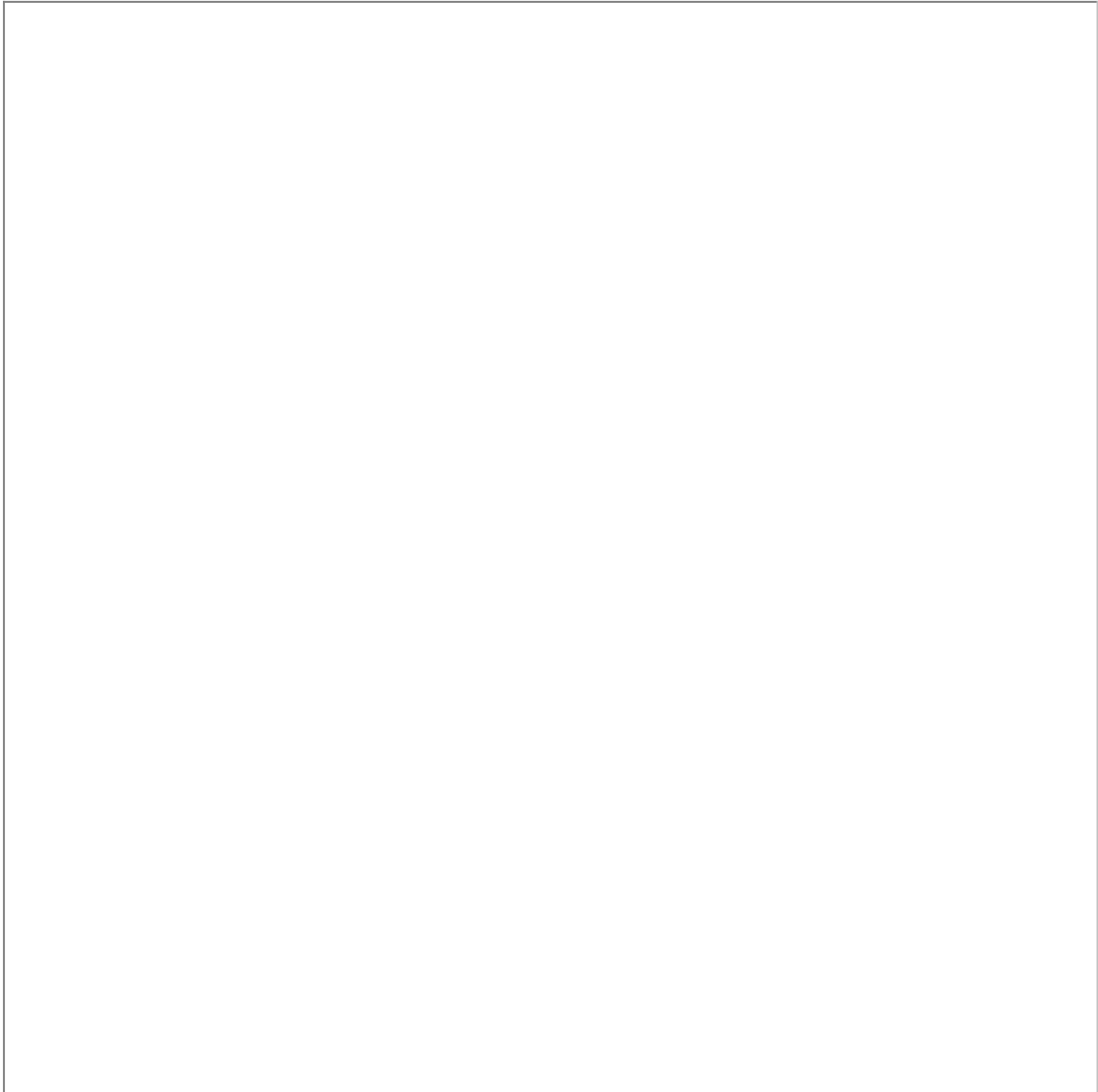
---

---

---

기타 이슈 사항

1. SYSDATE 항목이 에러 발생 이슈 제기(LOB3)



→ 테스트 결과 이슈 없음 확인.

## 12.6 04. Repository Metrix [오더AA]

Repository 사용법 Metrix를 정리한다.

# 샘플 소스 GIT : <https://gitlab.dspace.kt.co.kr/icis-tr/order/sample/icis-oder-sample>

샘플소스는 기준정보DB에서 실행하니 기준정보 DB에서 테스트 데이터 확보

git에서 chekout 한 이후 springboot 기동하면 아래의 swagger ui로 확인 가능

<http://localhost:8088/oder/sample/swagger-ui/index.html#/>

README.md 파일에 샘플 테스트 데이터 있으니 참고해서 테스트 할 수 있음

#### # CRUD Metrix

CRUD 유형	개발 패턴	select	insert	update	delete
PK 유형		샘플	샘플	샘플	샘플
PK 없음	Repository	CbovrfresultQryApi - retvByIndex	X	X	X
	Repository @Query("")	CbovrfresultQryApi - retvCbovrfresult	X	X	X
	jdbe template	CsyscdQryApi -retvByCsyscdQueryList	GbovrfresultCmd Api - -addCbovrfresult	CbovrfresultCmd Api - -saveCbovrfresult	CbovrfresultCmd Api - -removeCbovrfresult
단일키 (PK 1개)	Repository	CcissecdownstatusQry Api - retvByld - retvAll ClocareafstQryApi - retvByFstAreaCd - retvByCityTypeCd	Ccissecdownsta tusCmdApi - addCcissecDow nStatus	Ccissecdownsta tusCmdApi - saveCcissecDow nStatus	Ccissecdownstat usCmdApi - removeCcissecD ownStatus
	Repository @Query("")	ClocareafstQryApi - retvClocareafst	CsyscdCmdApi - addCsyscd	CsyscdCmdApi - saveCsyscd	CsyscdCmdApi - removeCsyscd

	jdbe template	CsyscdQryApi - retvByCsyscdQueryList	X	X	X
복합키 (PK 2개 이상)	Repository	CsyscdQryApi - retvById - retvByGrpList	X	X	X
	Repository @Query("")	CsyscdQryApi - retvByCsyscd	X	X	X
	jdbe template	CsyscdQryApi - retvByCsyscdQueryList	CsyscdCmdApi - addCsyscd	CsyscdCmdApi - saveCsyscd	CsyscdCmdApi - removeCsyscd
Join select	Repository @Query("")	CsyscdQryApi - selectJoin	해당사항 없음	해당사항 없음	해당사항 없음

## # Select Matrix

개발패턴	페이징	멀티 ROW	싱글 ROW	정렬	건수
Repository	넥사 페이징 처리 확인 필요	Ccissecd ownstatusQryApi - retvAll	Ccissecd ownstatusQryA pi - retvById		CsyscdQryCntr - count
Repository @Query("")	넥사 페이징 처리 확인 필요	CsyscdQr yApi - retvByCs yscd			
jdbe template	넥사 페이징 처리 확인 필요				

#처리성 서비스 개발 중 조회를 해야하는 경우

만약 조회 해야하는 테이블이 처리성을 하고 있는 테이블 이라면 동일 Repository 안에 findBy 함수 구현 가능



만약 조회 해야하는 테이블이 처리성을 하고 있지 않는 테이블 이라면 query 패키지 내에 해당 테이블 조회 패키지를 만들고 그 안에서 구현.

but 다른 어느곳에라도 해당 테이블을 조회하는 서비스가 만들어져 있다면 해당 서비스 사용해도 무방.(중복되서 만들어도 되지만 명명규칙이나 패키지 구조로 인해서 중복 문제가 발생할수도 있으니 주의.)

---

# 에러 발생 케이스 - 히스토리 관리 목적

PK 없음 - Repository @Query("") 로 Update시 DB 저장은 되나 ResultSet 매핑 중 에러 발생함, PK없음 CUD는 사용 불가

복합 Key, 단일 Key - jdbcTemplate.queryForList 조회시 오류 발생

[org.springframework.jdbc.IncorrectResultSetColumnCountException: Incorrect column count: expected 1, actual 6]

## 13 06. Entity개발 템플릿 [오더AA]

- Query, Command Entity 2개 플릿
- Command 의 isNew 는 파라미터 전달 받아 처리, Query의 isNew 는 false 로 고정한다.
- Repository와의 목적으로 사용하지 않는다.
- 실제하는 테이블로 @Table Annotation에 선언한다.
- [Entity / DTO Generate 소스 \[오더AA\]](#)<sup>4</sup> 의 자동 생성된 Entity를 사용한다.

### Query Entity 개발 템플릿

```

1  /
   *****
   *****
2  * ICIS version 1.0
3  *
4  * Copyright © 2022 kt/kt ds corp. All rights reserved.
5  *
6  * This is a proprietary software of kt corp, and you may not use this file
   except in
7  * compliance with license agreement with kt corp. Any redistribution or
   use of this
8  * software, with or without modification shall be strictly prohibited
   without prior written
9  * approval of kt corp, and the copyright notice above does not evidence
   any actual or
10 * intended publication of such software.
11
   *****
   *****/
12
13 package com.kt.icis.oder.sample.common.query.repository.entity;
14
15 import java.math.BigDecimal;
16 import java.sql.Timestamp;
17
18 import org.springframework.data.domain.Persistable;
19 import org.springframework.data.relational.core.mapping.Column;
20 import org.springframework.data.relational.core.mapping.Table;
21
22 import com.fasterxml.jackson.annotation.JsonFormat;
23 import com.kt.icis.oder.sample.common.query.repository.dto.CsyscdQryDto;
24
25 import lombok.AllArgsConstructor;
26 import lombok.Builder;
27 import lombok.Data;
28 import lombok.NoArgsConstructor;
29

```

4 <https://wiki.dspace.kt.co.kr/pages/viewpage.action?pagelId=85365411>

```

30  /**
31   * 코드 db table entity
32   */
33  @AllArgsConstructor
34  @Builder
35  @Data
36  @NoArgsConstructor
37  @Table("BI_CSYS_CD")
38  public class CsyscdQryEntt implements Persistable<String> {
39
40      // description = "그룹아이디", nullable = false, maxLength = 4
41      // @Id 복합키 사용불가
42      @Column("GRP_ID")
43      private String grpId;
44
45      // description = "등록국코드", nullable = true, maxLength = 6
46      @Column("REG_OFC_CD")
47      private String regOfcCd;
48
49      // description = "등록자사번", nullable = true, maxLength = 9
50      @Column("REGER_EMP_NO")
51      private String regerEmpNo;
52
53      // description = "등록자명", nullable = true, maxLength = 10
54      @Column("REGER_EMP_NAME")
55      private String regerEmpName;
56
57      // description = "정의명", nullable = true, maxLength = 80
58      @Column("DEFINE_NAME")
59      private String defineName;
60
61      // description = "종료처리자사번", nullable = true, maxLength = 9
62      @Column("END_EMP_NO")
63      private String endEmpNo;
64
65      // description = "종료처리자명", nullable = true, maxLength = 10
66      @Column("END_EMP_NAME")
67      private String endEmpName;
68
69      // description = "만료일자", nullable = true, maxLength = 8
70      @Column("END_DATE")
71      private String endDate;
72
73      // description = "시작일자", nullable = true, maxLength = 8
74      @Column("START_DATE")
75      private String startDate;
76
77      // description = "참조4내용", nullable = true, maxLength = 20
78      @Column("REF4")
79      private String ref4;
80
81      // description = "참조5내용", nullable = true, maxLength = 20
82      @Column("REF5")
83      private String ref5;
84
85      // description = "코드", nullable = false, maxLength = 7
86      // @Id 복합키 사용불가

```

```

87     @Column("CD")
88     private String cd;
89
90     // description = "명", nullable = true, maxLength = 50
91     @Column("NAME")
92     private String name;
93
94     // description = "출력순서", nullable = true, maxLength = 22
95     @Column("OUTPUT_SEQ")
96     private BigDecimal outputSeq;
97
98     // description = "참조1내용", nullable = true, maxLength = 10
99     @Column("REF1")
100    private String ref1;
101
102    // description = "참조2내용", nullable = true, maxLength = 20
103    @Column("REF2")
104    private String ref2;
105
106    // description = "참조3내용", nullable = true, maxLength = 22
107    @Column("REF3")
108    private BigDecimal ref3;
109
110    // description = "비고내용", nullable = true, maxLength = 120
111    @Column("REMARK")
112    private String remark;
113
114    // description = "등록일시", example = "YYYYMMDDHHMISS", nullable = true
115    @Column("REG_DATE")
116    @JsonFormat(shape = JsonFormat.Shape.STRING, pattern = "yyyy-MM-dd
HH:mm:ss", timezone = "Asia/Seoul")
117    private Timestamp regDate;
118
119    @Override
120    public boolean isNew() {
121        return false;
122    }
123
124    @Override
125    public String getId() {
126        return null;
127    }
128
129    public static CsyscdQryEntt of(CsyscdQryDto dto) {
130        return CsyscdQryEntt.builder()
131            .grpId(dto.getGrpId())
132            .regOfcCd(dto.getRegOfcCd())
133            .regerEmpNo(dto.getRegerEmpNo())
134            .regerEmpName(dto.getRegerEmpName())
135            .defineName(dto.getDefineName())
136            .endEmpNo(dto.getEndEmpNo())
137            .endEmpName(dto.getEndEmpName())
138            .endDate(dto.getEndDate())
139            .startDate(dto.getStartDate())
140            .ref4(dto.getRef4())
141            .ref5(dto.getRef5())
142            .cd(dto.getCd())

```

```

143         .name(dto.getName())
144         .outputSeq(dto.getOutputSeq())
145         .ref1(dto.getRef1())
146         .ref2(dto.getRef2())
147         .ref3(dto.getRef3())
148         .remark(dto.getRemark())
149         .regDate(dto.getRegDate())
150         .build();
151     }
152 }

```

### Command Entity 개발 템플릿

```

1  /
   *****
   *****
2  * ICIS version 1.0
3  *
4  * Copyright © 2022 kt/kt ds corp. All rights reserved.
5  *
6  * This is a proprietary software of kt corp, and you may not use this file
   except in
7  * compliance with license agreement with kt corp. Any redistribution or
   use of this
8  * software, with or without modification shall be strictly prohibited
   without prior written
9  * approval of kt corp, and the copyright notice above does not evidence
   any actual or
10  * intended publication of such software.
11
   *****
   *****/
12
13  package com.kt.icis.oder.sample.common.command.repository.entity;
14
15  import java.math.BigDecimal;
16  import java.sql.Timestamp;
17
18  import org.springframework.data.annotation.Transient;
19  import org.springframework.data.domain.Persistable;
20  import org.springframework.data.relational.core.mapping.Column;
21  import org.springframework.data.relational.core.mapping.Table;
22
23  import com.fasterxml.jackson.annotation.JsonFormat;
24  import com.kt.icis.oder.sample.common.command.repository.dto.CsyscdCmdDto;
25
26  import lombok.AllArgsConstructor;
27  import lombok.Builder;
28  import lombok.Data;
29  import lombok.NoArgsConstructor;
30

```

```

31  /**
32   * 코드 db table entity
33   */
34  @AllArgsConstructor
35  @Builder
36  @Data
37  @NoArgsConstructor
38  @Table("BI_CSYS_CD")
39  public class CsyscdCmdEntt implements Persistable<String> {
40
41      // description = "그룹아이디", nullable = false, maxLength = 4
42      // @Id 복합키 사용불가
43      @Column("GRP_ID")
44      private String grpId;
45
46      // description = "등록국코드", nullable = true, maxLength = 6
47      @Column("REG_OFCD_CD")
48      private String regOfcdCd;
49
50      // description = "등록자사번", nullable = true, maxLength = 9
51      @Column("REGER_EMP_NO")
52      private String regerEmpNo;
53
54      // description = "등록자명", nullable = true, maxLength = 10
55      @Column("REGER_EMP_NAME")
56      private String regerEmpName;
57
58      // description = "정의명", nullable = true, maxLength = 80
59      @Column("DEFINE_NAME")
60      private String defineName;
61
62      // description = "종료처리자사번", nullable = true, maxLength = 9
63      @Column("END_EMP_NO")
64      private String endEmpNo;
65
66      // description = "종료처리자명", nullable = true, maxLength = 10
67      @Column("END_EMP_NAME")
68      private String endEmpName;
69
70      // description = "만료일자", nullable = true, maxLength = 8
71      @Column("END_DATE")
72      private String endDate;
73
74      // description = "시작일자", nullable = true, maxLength = 8
75      @Column("START_DATE")
76      private String startDate;
77
78      // description = "참조4내용", nullable = true, maxLength = 20
79      @Column("REF4")
80      private String ref4;
81
82      // description = "참조5내용", nullable = true, maxLength = 20
83      @Column("REF5")
84      private String ref5;
85
86      // description = "코드", nullable = false, maxLength = 7
87      // @Id 복합키 사용불가

```

```

88     @Column("CD")
89     private String cd;
90
91     // description = "명", nullable = true, maxLength = 50
92     @Column("NAME")
93     private String name;
94
95     // description = "출력순서", nullable = true, maxLength = 22
96     @Column("OUTPUT_SEQ")
97     private BigDecimal outputSeq;
98
99     // description = "참조1내용", nullable = true, maxLength = 10
100    @Column("REF1")
101    private String ref1;
102
103    // description = "참조2내용", nullable = true, maxLength = 20
104    @Column("REF2")
105    private String ref2;
106
107    // description = "참조3내용", nullable = true, maxLength = 22
108    @Column("REF3")
109    private BigDecimal ref3;
110
111    // description = "비고내용", nullable = true, maxLength = 120
112    @Column("REMARK")
113    private String remark;
114
115    // description = "등록일시", example = "YYYYMMDDHHMISS", nullable = true
116    @Column("REG_DATE")
117    @JsonFormat(shape = JsonFormat.Shape.STRING, pattern = "yyyy-MM-dd
HH:mm:ss", timezone = "Asia/Seoul")
118    private Timestamp regDate;
119
120    @Transient
121    public boolean isNew = false;
122
123    @Override
124    public boolean isNew() {
125        return isNew;
126    }
127
128    @Override
129    public String getId() {
130        return null;
131    }
132
133    public static CsyscdCmdEntt of(CsyscdCmdDto dto) {
134        return CsyscdCmdEntt.builder()
135            .grpId(dto.getGrpId())
136            .regOfcCd(dto.getRegOfcCd())
137            .regerEmpNo(dto.getRegerEmpNo())
138            .regerEmpName(dto.getRegerEmpName())
139            .defineName(dto.getDefineName())
140            .endEmpNo(dto.getEndEmpNo())
141            .endEmpName(dto.getEndEmpName())
142            .endDate(dto.getEndDate())
143            .startDate(dto.getStartDate())

```

```
144         .ref4(dto.getRef4())
145         .ref5(dto.getRef5())
146         .cd(dto.getCd())
147         .name(dto.getName())
148         .outputSeq(dto.getOutputSeq())
149         .ref1(dto.getRef1())
150         .ref2(dto.getRef2())
151         .ref3(dto.getRef3())
152         .remark(dto.getRemark())
153         .regDate(dto.getRegDate())
154         .build();
155     }
156 }
```



## 14 07. OpenFeign 개발 템플릿 [오더AA]

- OpenFeign은 MS내에서 타 MS의 API를 쉽게 호출할 수 있도록 도와주는 오픈소스이다.
- API를 호출하는 OpenFeign Client와 호출한 API가 에러 시 작동하는 FallbackFactory 2개 Java로 구성된다.
- OpenFeign Client에는 @FeignClient Annotation을 정의하고 name, url, FallbackFactory Java를 선언한다.
- 호출하고자 하는 API 1개와 Method 1개가 매핑되며 1개 이상의 API를 OpenFeign Client Java에 선언할 수 있다.
- @FeignClient 의 url과 Method에 정의된 @PostMapping Annotaion의 값이 조합되어 API를 호출한다.
- Method는 호출하고자 하는 API에 전달하고자 하는 Payload와 Header 정보를 담고 있는 feignHeader 가 파라미터로 전달되어야 한다.
- [99. 샘플 소스 \[오더AA\]<sup>5</sup>](#)에서 샘플을 참조 한다.

### Openfeign Client 호출

```
openfeignSampleSendCnt.openfeignSend(CommonUtil.getFeignHeader(), inPyld);
```

### Openfeign Client

```

1  package com.kt.icis.oder.sample.openfeign.controller.client;
2
3  import org.springframework.cloud.openfeign.FeignClient;
4  import org.springframework.web.bind.annotation.PostMapping;
5  import org.springframework.web.bind.annotation.RequestBody;
6  import org.springframework.web.bind.annotation.RequestHeader;
7
8  import com.kt.icis.cmmnfrwk.constant.Constant;
9  import com.kt.icis.oder.sample.openfeign.payload.in.OpenfeignSampleInPyld;
10 import com.kt.icis.oder.sample.openfeign.payload.out.OpenfeignSampleOutPyld;
11
12 @FeignClient(name = "SP.OpenfeignSampleSendCnt", url = "${api.sample.url}",
13             fallbackFactory = OpenfeignSampleSendCntFallbackFactory.class)
14 public interface OpenfeignSampleSendCnt {
15     @PostMapping("/openfeign/receive")
16     OpenfeignSampleOutPyld openfeignSend(@RequestHeader(Constant.FEIGN_HEADE
17 R) String feignHeader, @RequestBody OpenfeignSampleInPyld inPyld);
18 }
```

<sup>5</sup> <https://wiki.dspace.kt.co.kr/pages/viewpage.action?pagelId=74346523>

- OpenFeign Client로 API를 호출했으나 에러가 발생하는 경우 Out Payload는 없이 Header만 반환한다.
- 에러인 경우 프레임워크에서 FallbackFactory를 호출한다.
- FallbackFactory에서는 Header에 선언된 에러코드, 에러메시지를 보고 Out Payload를 생성하여 OpenFeign Client를 호출했던 코드로 반환한다.
- Header의 에러메시지 정보는 ExceptionUtil를 사용하여 확인한다.

### FallbackFactory

```

1  package com.kt.icis.oder.sample.openfeign.controller.client;
2
3  import org.springframework.cloud.openfeign.FallbackFactory;
4  import org.springframework.stereotype.Component;
5  import org.springframework.web.bind.annotation.RequestHeader;
6
7  import com.kt.icis.cmmnfrwk.constant.Constant;
8  import com.kt.icis.cmmnfrwk.payload.ResponseMessage;
9  import com.kt.icis.cmmnfrwk.utils.ExceptionUtil;
10 import com.kt.icis.cmmnfrwk.utils.LogUtil;
11 import com.kt.icis.oder.sample.openfeign.payload.in.OpenfeignSampleInPyld;
12 import com.kt.icis.oder.sample.openfeign.payload.out.OpenfeignSampleOutPyld;
13 import
    com.kt.icis.oder.sample.openfeign.payload.out.dto.OpenfeignSampleOutDto;
14
15 @Component
16 public class OpenfeignSampleSendClntFallbackFactory implements
    FallbackFactory<OpenfeignSampleSendClnt> {
17
18     @Override
19     public OpenfeignSampleSendClnt create(Throwable cause) {
20         ResponseMessage responseMessage =
21             ExceptionUtil.getResponseMessage(cause);
22         LogUtil.info("***** ResponseMessage : " +
23             responseMessage.toString());
24
25         return new OpenfeignSampleSendClnt() {
26             @Override
27             public OpenfeignSampleOutPyld openfeignSend(@RequestHeader(Const
28                 ant.FEIGN_HEADER) String feignHeader, OpenfeignSampleInPyld inPyld) {
29                 LogUtil.info("***** inPyld : " + inPyld.toString());
30                 LogUtil.info("***** ResponseCode : " +
31                     responseMessage.getResponseCode());
32                 LogUtil.info("***** ResponseType : " +
33                     responseMessage.getResponseType());
34                 LogUtil.info("***** ResponseTitle : " +
35                     responseMessage.getResponseTitle());

```

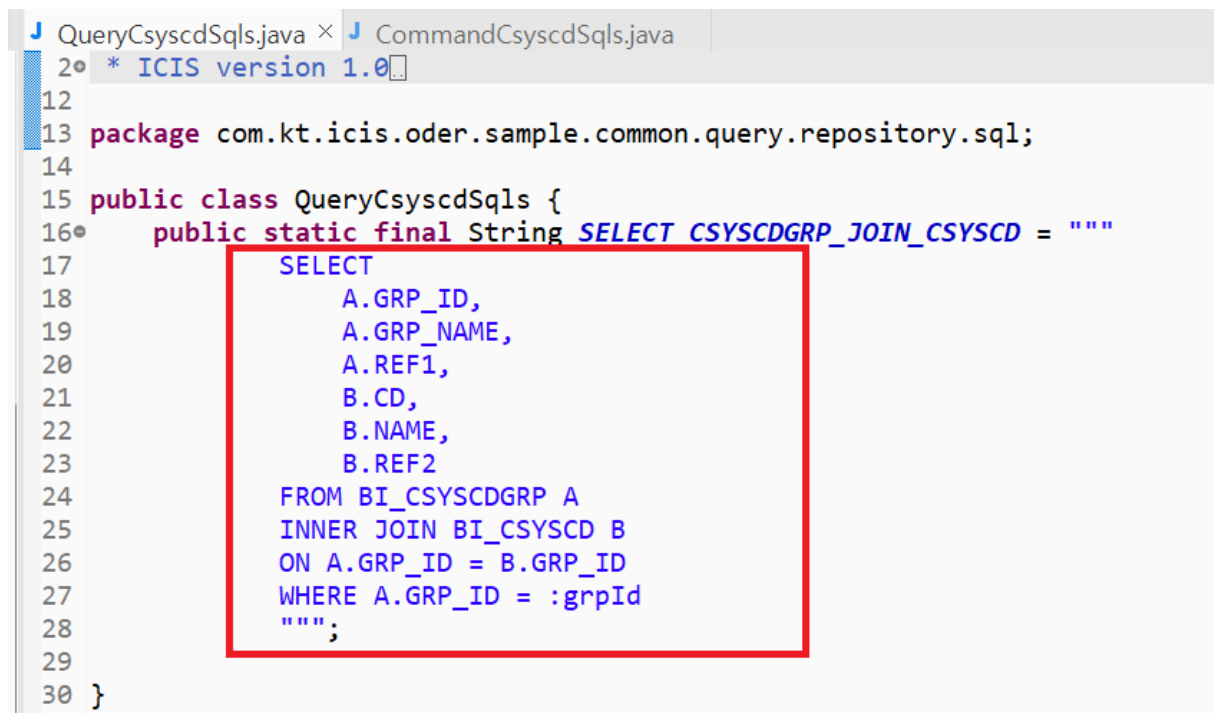
```
30         LogUtil.info("***** ResponseBasc : " +
responseMessage.getResponseBasc());
31         LogUtil.info("***** ResponseDtal : " +
responseMessage.getResponseDtal());
32         return

OpenfeignSampleOutPyld.builder().openfeignSampleOutDto(OpenfeignSampleOutDto
.builder().outVal("비즈니스에러").build()).build();
33     }
34 };
35 }
36 }
```

## 15 08. Sqls.java 개발 템플릿[오더AA]

- SQL 은 Sql.java에만 정의한다.
- Sql.java내 SQL은 파라미터를 제외하고 대문자로 작성한다.
- 줄바꿈 시 빈 문자열이 존재 하면 안된다.
- table 명은 파라미터로 전달해서 사용하면 안된다. 이는 KT내 보안 금지 사항이다.
- 하나의 Sql에서 Select 후 Insert/Update/Delete 은 사용하면 안된다.

SELECT 예시)



```

J QueryCsyscdSqls.java × J CommandCsyscdSqls.java
20 * ICIS version 1.0
12
13 package com.kt.icis.oder.sample.common.query.repository.sql;
14
15 public class QueryCsyscdSqls {
16     public static final String SELECT_CSYS CDGRP_JOIN_CSYS CD = ""
17         SELECT
18             A.GRP_ID,
19             A.GRP_NAME,
20             A.REF1,
21             B.CD,
22             B.NAME,
23             B.REF2
24         FROM BI_CSYS CDGRP A
25         INNER JOIN BI_CSYS CD B
26         ON A.GRP_ID = B.GRP_ID
27         WHERE A.GRP_ID = :grpId
28         "";
29
30 }
  
```

CUD 예시)

```

J QueryCsyscdSqls.java J CommandCsyscdSqls.java ×
20 * ICIS version 1.0
12
13 package com.kt.icis.oder.sample.common.command.repository.sql;
14
15 public class CommandCsyscdSqls {
16     public static final String INSERT_CSYS_CD = ""
17         INSERT INTO BI_CSYS_CD(GRP_ID, CD, NAME) VALUES
18         (:grpId, :cd, :name)
19         "";
20
21     public static final String UPDATE_CSYS_CD = ""
22         UPDATE BI_CSYS_CD
23         SET NAME = :name
24         WHERE GRP_ID = :grpId
25         AND CD = :cd
26         "";
27
28     public static final String DELETE_CSYS_CD = ""
29         DELETE FROM BI_CSYS_CD
30         WHERE GRP_ID = :grpId
31         AND CD = :cd
32         "";
33
34 }

```

### DynamicSql 예제

```

1 package com.kt.icis.oder.sample.common.query.repository.sql;
2
3 public class CsyscdDynamicQrySql {
4     public static final String WHERE_AND = " AND ";
5     public static final String ALIAS_A = " A.";
6     public static final String ALIAS_B = " B.";
7     public static final String CSYS_CD_MAIN_SQL = "SELECT GRP_ID, CD, NAME,
8     REF1, REF2, REF3, REF4, REF5, DEFINE_NAME, OUTPUT_SEQ, START_DATE, REMARK
9     FROM BI_CSYS_CD WHERE 1 = 1";
10
11     public static final String CSYS_CD_WHERE_GRPID = "GRP_ID = :grpId";
12     public static final String CSYS_CD_WHERE_GRPID_IN = "GRP_ID IN (:grpIds)";
13
14     public static final String CSYS_CD_WHERE_CD = "CD = :cd";
15     public static final String CSYS_CD_WHERE_CD_IN = "CD IN (:cds)";
16     public static final String CSYS_CD_WHERE_NAME = "NAME = :name";
17     public static final String CSYS_CD_WHERE_REF1 = "REF1 = :ref1";
18     public static final String CSYS_CD_WHERE_REF2 = "REF2 = :ref2";
19     public static final String CSYS_CD_WHERE_REF3 = "REF3 = :ref3";
20     public static final String CSYS_CD_WHERE_REF4 = "REF4 = :ref4";
21     public static final String CSYS_CD_WHERE_REF5 = "REF5 = :ref5";
22     public static final String CSYS_CD_WHERE_DEFINE_NAME = "DEFINE_NAME
23     = :defineName";

```

```

19     public static final String CSYSCD_WHERE_OUTPUTSEQ = "OUTPUT_SEQ
= :outputSeq";
20     public static final String CSYSCD_WHERE_STARTDATE = "START_DATE
= :startDate";
21     public static final String CSYSCD_WHERE_REMARK = "REMARK = :remark";
22     public static final String CSYSORDTYPE_WHERE_SACD = "SA_CD = :saCd";
23     public static final String CSYSORDTYPE_WHERE_SACD_IN = "SA_CD IN
(:saCds)";
24     public static final String CSYSORDTYPE_WHERE_ORDCOMPLETEDTYPE =
"ORD_COMPLETED_TYPE = :ordCompletedType";
25     public static final String CSYSORDTYPE_WHERE_ORDMNGFLAG = "ORD_MNG_FLAG
= :ordMngFlag";
26     public static final String CSYSORDTYPE_WHERE_USERORDSEQ = "USER_ORD_SEQ
= :userOrdSeq";
27     public static final String CSYSORDTYPE_WHERE_ORDTYPECD = "ORD_TYPE_CD
= :ordTypeCd";
28     public static final String CSYSORDTYPE_WHERE_ORDTYPECD_IN = "ORD_TYPE_CD
IN (:ordTypeCds)";
29     public static final String CSYSCD_CSYSORDTYPE_MAIN_SQL = "SELECT
A.GRP_ID, B.SA_CD, A.CD, B.ORD_COMPLETED_TYPE, B.ORD_MNG_FLAG,
B.USER_ORD_SEQ, B.ORD_TYPE_CD FROM BI_CSYSYCD A, BI_CSYSORDTYPE B WHERE A.CD
= B.ORD_TYPE_CD";
30     public static final String CSYSCD_CSYSRODVCSCD_MAIN_SQL = "SELECT A.CD,
B.SA_CD, B.SO_ITEM_ID, B.FCTR_1, B.FCTR_2, B.FCTR_3, B.FCTR_4, B.FCTR_5,
A.REF1, A.REF2, A.GRP_ID FROM BI_CSYSYCD A, BI_CSYSRODVCSCD B WHERE A.GRP_ID
= B.SO_ITEM_ID";
31     public static final String CSYSRODVCSCD_WHERE_SOITEMID = "SO_ITEM_ID
= :soItemId";
32     public static final String CSYSRODVCSCD_WHERE_FCTR1 = "FCTR_1 = :fctr1";
33     public static final String CSYSRODVCSCD_WHERE_FCTR2 = "FCTR_2 = :fctr2";
34     public static final String CSYSRODVCSCD_WHERE_FCTR3 = "FCTR_3 = :fctr3";
35     public static final String CSYSRODVCSCD_WHERE_FCTR4 = "FCTR_4 = :fctr4";
36     public static final String CSYSRODVCSCD_WHERE_FCTR5 = "FCTR_5 = :fctr5";
37     public static final String CSYSCD_MPRDCD_MAIN_SQL = "SELECT A.GRP_ID,
B.SA_CD FROM BI_CSYSYCD A, BI_MPRDCD B WHERE A.CD = B.SA_CD";
38
39     public static String findCsyscdSql(String grpId, String cd, String name,
String ref1, String ref2, Integer ref3, String ref4, String ref5
40         ,String defineName, Integer outputSeq, String startDate, String
remark) {
41
42         StringBuffer queryBuffer = new StringBuffer();
43         queryBuffer.append(CsyscdDynamicQrySql.CSYSCD_MAIN_SQL);
44
45         if(grpId != null && !"".equals(grpId)){
46             queryBuffer.append(CsyscdDynamicQrySql.WHERE_AND);
47
48             if(grpId.indexOf("||") != -1){
49
50                 queryBuffer.append(CsyscdDynamicQrySql.CSYSCD_WHERE_GRPID_IN);
51             }else {
52                 queryBuffer.append(CsyscdDynamicQrySql.CSYSCD_WHERE_GRPID);
53             }
54         }
55         if(cd != null && !"".equals(cd)){
56             queryBuffer.append(CsyscdDynamicQrySql.WHERE_AND);

```

```

57         if(cd.indexOf("|") != -1){
58             queryBuffer.append(CsyscdDynamicQrySql.CSYSCD_WHERE_CD_IN);
59         }else {
60             queryBuffer.append(CsyscdDynamicQrySql.CSYSCD_WHERE_CD);
61         }
62     }
63     if(name != null && !"".equals(name)){
64         queryBuffer.append(CsyscdDynamicQrySql.WHERE_AND);
65         queryBuffer.append(CsyscdDynamicQrySql.CSYSCD_WHERE_NAME);
66     }
67     if(ref1 != null && !"".equals(ref1)){
68         queryBuffer.append(CsyscdDynamicQrySql.WHERE_AND);
69         queryBuffer.append(CsyscdDynamicQrySql.CSYSCD_WHERE_REF1);
70     }
71     if(ref2 != null && !"".equals(ref2)){
72         queryBuffer.append(CsyscdDynamicQrySql.WHERE_AND);
73         queryBuffer.append(CsyscdDynamicQrySql.CSYSCD_WHERE_REF2);
74     }
75     if(ref3 != null){
76         queryBuffer.append(CsyscdDynamicQrySql.WHERE_AND);
77         queryBuffer.append(CsyscdDynamicQrySql.CSYSCD_WHERE_REF3);
78     }
79     if(ref4 != null && !"".equals(ref4)){
80         queryBuffer.append(CsyscdDynamicQrySql.WHERE_AND);
81         queryBuffer.append(CsyscdDynamicQrySql.CSYSCD_WHERE_REF4);
82     }
83     if(ref5 != null && !"".equals(ref5)){
84         queryBuffer.append(CsyscdDynamicQrySql.WHERE_AND);
85         queryBuffer.append(CsyscdDynamicQrySql.CSYSCD_WHERE_REF5);
86     }
87     if(defineName != null && !"".equals(defineName)){
88         queryBuffer.append(CsyscdDynamicQrySql.WHERE_AND);
89         queryBuffer.append(CsyscdDynamicQrySql.CSYSCD_WHERE_DEFINENAME);
90     }
91     if(outputSeq != null){
92         queryBuffer.append(CsyscdDynamicQrySql.WHERE_AND);
93         queryBuffer.append(CsyscdDynamicQrySql.CSYSCD_WHERE_OUTPUTSEQ);
94     }
95     if(startDate != null && !"".equals(startDate)){
96         queryBuffer.append(CsyscdDynamicQrySql.WHERE_AND);
97         queryBuffer.append(CsyscdDynamicQrySql.CSYSCD_WHERE_STARTDATE);
98     }
99     if(remark != null && !"".equals(remark)){
100         queryBuffer.append(CsyscdDynamicQrySql.WHERE_AND);
101         queryBuffer.append(CsyscdDynamicQrySql.CSYSCD_WHERE_REMARK);
102     }
103
104     return queryBuffer.toString();
105 }
106 }

```

## 16 09. DTO개발 템플릿 [오더AA]

- In, Out DTO 2개 템플릿
- In Dto는 필요한 파라미터만 정의해서 사용하며 Swagger Schema 어노테이션 required를 정의한다.
- Out Dto가 Repository의 DTO와 동일하다면 Repository의 DTO를 사용해도 된다. Repository의 DTO와 별개로 구성한다면 Payload.out.dto에 새로 정의하여 사용한다.
- [Entity / DTO Generate 소스 \[오더AA\]](#)<sup>6</sup>의 자동 생성된 DTO 사용한다.

### In DTO 개발 템플릿

```

1  /
   *****
   *****
2  * ICIS version 1.0
3  *
4  * Copyright © 2022 kt/kt ds corp. All rights reserved.
5  *
6  * This is a proprietary software of kt corp, and you may not use this file
   except in
7  * compliance with license agreement with kt corp. Any redistribution or
   use of this
8  * software, with or without modification shall be strictly prohibited
   without prior written
9  * approval of kt corp, and the copyright notice above does not evidence
   any actual or
10 * intended publication of such software.
11
   *****
   *****/
12
13 package com.kt.icis.oder.sample.common.query.payload.in.dto;
14
15 import io.swagger.v3.oas.annotations.media.Schema;
16 import lombok.AllArgsConstructor;
17 import lombok.Builder;
18 import lombok.Data;
19 import lombok.NoArgsConstructor;
20
21 /**
22  * CsyscdQryApi In Dto
23  */
24 @Data
25 @NoArgsConstructor
26 @AllArgsConstructor
27 @Builder
28 public class CsyscdByIdInQryDto {
29
30     @Schema(description = "그룹아이디", nullable = false, maxLength = 4)

```

<sup>6</sup> <https://wiki.dspace.kt.co.kr/pages/viewpage.action?pagelId=85365411>



```

31     private String grpId;
32
33
34     @Schema(description = "코드", nullable = false, maxLength = 7)
35     private String cd;
36 }

```

### Out DTO 개발 템플릿

```

1  /
   *****
   *****
2      * ICIS version 1.0
3      *
4      * Copyright © 2022 kt/kt ds corp. All rights reserved.
5      *
6      * This is a proprietary software of kt corp, and you may not use this file
   except in
7      * compliance with license agreement with kt corp. Any redistribution or
   use of this
8      * software, with or without modification shall be strictly prohibited
   without prior written
9      * approval of kt corp, and the copyright notice above does not evidence
   any actual or
10     * intended publication of such software.
11
   *****
   *****/
12
13     package com.kt.icis.oder.sample.common.query.repository.dto;
14
15     import java.io.Serializable;
16     import java.math.BigDecimal;
17     import java.sql.Timestamp;
18
19     import io.swagger.v3.oas.annotations.media.Schema;
20     import lombok.AllArgsConstructor;
21     import lombok.Builder;
22     import lombok.Data;
23     import lombok.NoArgsConstructor;
24
25     /**
26      * 코드 db table dto
27      */
28     @AllArgsConstructor
29     @Builder
30     @Data
31     @NoArgsConstructor
32     @Schema(description = "코드")
33     public class CsyscdQryDto implements Serializable {
34         private static final long serialVersionUID = 1672038271634837413L;
35
36         @Schema(description = "그룹아이디", nullable = false, maxLength = 4)

```

```

37     private String grpId;
38
39     @Schema(description = "등록국코드", nullable = true, maxLength = 6)
40     private String regOfcCd;
41
42     @Schema(description = "등록자사번", nullable = true, maxLength = 9)
43     private String regerEmpNo;
44
45     @Schema(description = "등록자명", nullable = true, maxLength = 10)
46     private String regerEmpName;
47
48     @Schema(description = "정의명", nullable = true, maxLength = 80)
49     private String defineName;
50
51     @Schema(description = "종료처리자사번", nullable = true, maxLength = 9)
52     private String endEmpNo;
53
54     @Schema(description = "종료처리자명", nullable = true, maxLength = 10)
55     private String endEmpName;
56
57     @Schema(description = "만료일자", nullable = true, maxLength = 8)
58     private String endDate;
59
60     @Schema(description = "시작일자", nullable = true, maxLength = 8)
61     private String startDate;
62
63     @Schema(description = "참조4내용", nullable = true, maxLength = 20)
64     private String ref4;
65
66     @Schema(description = "참조5내용", nullable = true, maxLength = 20)
67     private String ref5;
68
69     @Schema(description = "코드", nullable = false, maxLength = 7)
70     private String cd;
71
72     @Schema(description = "명", nullable = true, maxLength = 50)
73     private String name;
74
75     @Schema(description = "출력순서", nullable = true, maxLength = 22)
76     private BigDecimal outputSeq;
77
78     @Schema(description = "참조1내용", nullable = true, maxLength = 10)
79     private String ref1;
80
81     @Schema(description = "참조2내용", nullable = true, maxLength = 20)
82     private String ref2;
83
84     @Schema(description = "참조3내용", nullable = true, maxLength = 22)
85     private BigDecimal ref3;
86
87     @Schema(description = "비고내용", nullable = true, maxLength = 120)
88     private String remark;
89
90     @Schema(description = "등록일시", nullable = true, maxLength = 8)
91     private Timestamp regDate;
92

```

93 }

## 16.1 Payload 탑재 DTO Date 포맷 [오더AA]

### 16.1.1 1. ICIS-TR 내부 통신 - String / java.sql.Timestamp

Payload 탑재 DTO에서는 DB의 날짜형 컬럼을 String 또는 java.sql.Timestamp로만 선언해서 사용한다.

[Entity / DTO Generate 소스 \[오더AA\]](#)<sup>7</sup>에서는 java.sql.Timestamp 로 날짜형 컬럼을 생성한다.

### 16.1.2 2. 외부 연동 - RestGateway - String / java.sql.Timestamp (Date, Dt, Time으로 종료되는 컬럼 한정)

RestGateway 와 연계될 때 변수명이 Date, Dt, Time으로 끝나고 데이터형이 Timestamp 인 경우는 SA 공통프레임워크에서 "yyyyMMddHHmmss" 문자열로 변환하여 응답한다.

Date, Dt, Time으로 변수명이 종료되지 않고 데이터형이 java.sql.Timestamp인 경우는 변수명을 변경한다.

### 16.1.3 3. 외부 연동 - RestGateway 외 - String

날짜형은 String 만 사용한다.

java.sql.Timestamp은 지양한다.

[프레임워크 제공 공통 유틸 가이드 \[오더AA\]](#)<sup>8</sup>의 DateUtil.timestampToString(), DateUtil.stringToTimestamp() 를 사용하여 변환처리 후 연동한다.

## 16.2 serialVersionUID 생성 [오더AA]

Serializable 를 implements 하는 DTO는 serialVersionUID를 선언해야 한다.

DTO 생성하고 serialVersionUID를 선언하지 않으면 다음과 같이 !가 표시된다.

<sup>7</sup> <https://wiki.dspace.kt.co.kr/pages/viewpage.action?pagelId=85365411>

<sup>8</sup> <https://wiki.dspace.kt.co.kr/pages/viewpage.action?pagelId=74342848>

```

24
25= /**
26  * 고객오더 db table dto
27  */
28 @AllArgsConstructor
29 @Builder
30 @Data
31 @NoArgsConstructor
32 @Schema(description = "고객오더")
33 public class LobCordordinfoQryDto implements Serializable {
34
35     @Schema(description = "오더번호", nullable = false, maxLength = 15)
36     private String ordNo;
37
38     @Schema(description = "교환국코드", nullable = true, maxLength = 6)
39     private String exchOfcCd;
40
41     @Schema(description = "설치관리국코드", nullable = true, maxLength = 6)
42     private String instlMngOfcCd;
43
44     @Schema(description = "고객아이디", nullable = true, maxLength = 11)
45     private String custId;

```

!를 클릭하면 상세창이 열린다. 여기서 Add generated serial version ID 클릭

```

25 @/**
26 * 고객오더 db table dto
27 */
28 @AllArgsConstructor
29 @Builder
30 @Data
31 @NoArgsConstructor
32 @Schema(description = "고객오더")
33 public class LobCordordinfoQryDto implements Serializable {
34     // Add default serial version ID
35     private static final long serialVersionUID = 1L;
36     // Add generated serial version ID
37     // Create new JUnit test case for 'LobCordordinfoQryDto.java'
38     @Schema(description = "고객오더 정보")
39     private String info;
40     // Rename in file (Ctrl+2, R)
41     // Rename in workspace
42     @Schema(description = "Add @SuppressWarnings 'serial' to 'LobCordordinfoQryDto'")
43     private String custNoType;
44     // Configure problem severity
45     @Schema(description = "고객식별번호유형코드", nullable = true, maxLength = 10)
46     private String custNoType;
47     @Schema(description = "고객식별번호")
48     private String custNo;
49     @Schema(description = "고객명")
50     private String custName;
51     @Schema(description = "고객전화번호")

```

serialVersionUID 가 STS에서 자동생성된다.

```

24
25 /**
26  * 고객오더 db table dto
27  */
28 @AllArgsConstructor
29 @Builder
30 @Data
31 @NoArgsConstructor
32 @Schema(description = "고객오더")
33 public class LobCordordinfoQryDto implements Serializable {
34
35     /**
36      *
37      */
38     private static final long serialVersionUID = 8898825092491848721L;
39
40     @Schema(description = "오더번호", nullable = false, maxLength = 15)
41     private String ordNo;

```

## 17 10. Payload 개발 템플릿 [오더AA]

- In, Out, Out List, Multi Out Payload 4개 템플릿
- Dto 선언시 ~Ds 명으로 종료 한다.

### In Payload 개발 템플릿

```

1  /
   *****
   *****
2  * ICIS version 1.0
3  *
4  * Copyright © 2022 kt/kt ds corp. All rights reserved.
5  *
6  * This is a proprietary software of kt corp, and you may not use this file
   except in
7  * compliance with license agreement with kt corp. Any redistribution or
   use of this
8  * software, with or without modification shall be strictly prohibited
   without prior written
9  * approval of kt corp, and the copyright notice above does not evidence
   any actual or
10 * intended publication of such software.
11
   *****
   *****/
12
13 package com.kt.icis.oder.sample.common.query.payload.in;
14
15 import
   com.kt.icis.oder.sample.common.query.payload.in.dto.CsyscdByIdInQryDto;
16
17 import lombok.AllArgsConstructor;
18 import lombok.Builder;
19 import lombok.Data;
20 import lombok.NoArgsConstructor;
21
22 /**
23  * CsyscdQryApi In Payload
24  */
25 @Data
26 @NoArgsConstructor
27 @AllArgsConstructor
28 @Builder
29 public class CsyscdByIdInQryPyld {
30     @JsonFormat(with = Feature.ACCEPT_SINGLE_VALUE_AS_ARRAY) // Single row
   Array 변환
31     private List<CcpiptstnCmdDto> ccpiptstnCmdDto; // in payload List 수신 시
   @JsonFormat 설정 필요
32
33     private CsyscdByIdInQryDto csyscdByIdInQryDs;

```

34	}
----	---

### Out Payload 개발 템플릿

```

1  /
   *****
   *****
2  * ICIS version 1.0
3  *
4  * Copyright © 2022 kt/ktds corp. All rights reserved.
5  *
6  * This is a proprietary software of kt corp, and you may not use this file
   except in
7  * compliance with license agreement with kt corp. Any redistribution or
   use of this
8  * software, with or without modification shall be strictly prohibited
   without prior written
9  * approval of kt corp, and the copyright notice above does not evidence
   any actual or
10 * intended publication of such software.
11
   *****
   *****/
12
13 package com.kt.icis.oder.sample.common.query.payload.out;
14
15 import com.kt.icis.oder.sample.common.query.repository.dto.CsyscdQryDto;
16
17 import lombok.AllArgsConstructor;
18 import lombok.Builder;
19 import lombok.Data;
20 import lombok.NoArgsConstructor;
21
22 /**
23  * CsyscdQryApi Out Payload
24  */
25 @Data
26 @NoArgsConstructor
27 @AllArgsConstructor
28 @Builder
29 public class CsyscdOutQryPyld {
30     private CsyscdQryDto csyscdOutQryDs;
31 }

```

## Out List Payload 개발 템플릿

```

1  /
   *****
   *****
2  * ICIS version 1.0
3  *
4  * Copyright © 2022 kt/kt ds corp. All rights reserved.
5  *
6  * This is a proprietary software of kt corp, and you may not use this file
   except in
7  * compliance with license agreement with kt corp. Any redistribution or
   use of this
8  * software, with or without modification shall be strictly prohibited
   without prior written
9  * approval of kt corp, and the copyright notice above does not evidence
   any actual or
10 * intended publication of such software.
11
   *****
   *****/
12
13 package com.kt.icis.oder.sample.common.query.payload.out;
14
15 import java.util.List;
16
17 import com.kt.icis.oder.sample.common.query.repository.dto.CsyscdQryDto;
18
19 import lombok.AllArgsConstructor;
20 import lombok.Builder;
21 import lombok.Data;
22 import lombok.NoArgsConstructor;
23
24 /**
25  * CsyscdQryApi Out List Payload
26  */
27 @Data
28 @NoArgsConstructor
29 @AllArgsConstructor
30 @Builder
31 public class CsyscdOutListQryPyld {
32     private List<CsyscdQryDto> csyscdOutListQryDs;
33 }

```

## Multi Out Payload 개발 템플릿

```

1  /
   *****
   *****

```



```

2      * ICIS version 1.0
3      *
4      * Copyright © 2022 kt/kt ds corp. All rights reserved.
5      *
6      * This is a proprietary software of kt corp, and you may not use this file
    except in
7      * compliance with license agreement with kt corp. Any redistribution or
    use of this
8      * software, with or without modification shall be strictly prohibited
    without prior written
9      * approval of kt corp, and the copyright notice above does not evidence
    any actual or
10     * intended publication of such software.
11
12     *****
13     *****/
14
15     package com.kt.icis.oder.sample.common.query.payload.out;
16
17     import java.util.List;
18
19     import
20
21     com.kt.icis.oder.sample.common.query.repository.dto.CcisecdownstatusQryDto;
22     import com.kt.icis.oder.sample.common.query.repository.dto.CsyscdQryDto;
23
24     import lombok.AllArgsConstructor;
25     import lombok.Builder;
26     import lombok.Data;
27     import lombok.NoArgsConstructor;
28
29     /**
30      * CsyscdQryApi Multi Out Payload
31      */
32     @Data
33     @NoArgsConstructor
34     @AllArgsConstructor
35     @Builder
36     public class MultiOutListQryPyld {
37         private List<CsyscdQryDto> csyscdOutListQryDs;
38         private List<CcisecdownstatusQryDto> ccisecdownstatusOutListQryDs;
39     }

```

## 18 11. Kafka 호출 개발 템플릿 [오더AA]

- Controller에서 Kafka 호출 공통 모듈인 KafkaProducerCntr를 사용하여 호출
- KafkaProducerCntr.getKafkaSendInDto를 호출하여 Kafka 전송 KafkaSendInDto로 전환하고 KafkaProducerCntr.send 를 사용하여 Kafka Producer 한다
- 동일한 수신 API에 멀티 데이터를 전송할 때는 Payload에 List<Dto> 한개만 선언해서 send로 전송하고 서로 다른 수신 API를 비즈니스 처리 최종 단계에서 전송할 때는 List<KafkaSendInDto> 선언해서 sendAll로 전송한다.
- 수신 API를 호출한 결과를 전달 받고 싶은 경우 Callback API를 통해 전달 가능하며 callbackLob와 callbackApi에 값을 set 하면 된다.  
Callback API의 응답 Payload는 com.kt.icis.oder.common.kafka.payload.KafkaCallbackOutPyld 를 사용한다.
- Kafka 호출 + ITG Gateway 연동시 ItggwInDto 를 먼저 생성하고 이를 파라미터로 ItggwCntr.getKafkaSendCallItggwInDto 호출하여 Kafka 전송 KafkaSendInDto로 전환하고 KafkaProducerCntr.send 를 사용하여 Kafka Producer 한다

### 18.1 Kafka 호출 Dto KafkaSendInDto 생성 가이드 - KafkaProducerCntr 모듈 사용

getKafkaSendInDto method는 orderNo, svcNo 를 파라미터로 전달 받는 케이스와 전달 받지 않는 케이스 2개가 있다.

orderNo, svcNo 가 파라미터에 없는 Method는 sendMsg 안에서 ordNo, SvcNo를 추출하여 orderNo, svcNo에 매핑시키고

orderNo, svcNo 가 파라미터에 있는 Method는 파라미터로 전달 받은 orderNo, svcNo 를 사용한다.

메소드	파라미터	내용	선택 / 필수	비고
getKafkaSendInDt o	KafkaServiceConstants	KafkaServiceConstants에서 사용하고자 Kafka Topic, 서비스 정보가 정의된 상수를 선택하여 입력 비즈니스에 따라 Kafka 호출 서비스가 달라진다면 KafkaProducerCntr 모듈내 getKafkaServiceConstants 메소드에 KafkaServiceConstants 명을 전달하면 해당 KafkaServiceConstants를 반환한다.  예시) 비즈니스에서 호출하는 Kafka 호출 정보가 단 건인 경우 KafkaServiceConstants.KAFKA_BI_SAVE_CSYSYCD  조건에서 따라 Kafka 호출 정보가 다른 경우 kafkaProducerCntr.getKafkaServiceConstants("KAFKA_BI_SAVE_CSYSYCD")	필수	sendMsg를 스캔 하여 ordNo, svcNo를 추출하여 전달한다.
	sendSvc	this.getClass().getSimpleName() 으로 고정	필수	
	sendMsg	JsonUtil.toJson를 사용하여 Payload를 String으로 변환하여 설정	필수	
getKafkaSendInDt o (method overloading)	orderNo	오더번호를 전달하면 Kafka 처리 이력 테이블 WC_KAFKA_LOG 에 ORDER_NO 컬럼으로 INSERT 되어 조회할 수 있다.	선택	sendMsg를 스캔 하지 않는다.
	svcNo	서비스계약번호를 전달하면 Kafka 처리 이력 테이블 WC_KAFKA_LOG 에 SVC_NO 컬럼으로 INSERT 되어 조회할 수 있다.	선택	
setModKey	modKey	특정키로 순차 처리가 필요할 때 특정키 필드명	선택	예시) said
setModValue	modValue	특정키가 순차 처리가 필요할 때 특정키 필드값	선택	
setCallbackLob	callbackLob	Callback API 를 제공하는 LOB코드로 설정, LobAllConstants에서 온라인 LOB_?? 와 데몬 LOB_??_DAEMON 중에서 선택	선택	

메소드	파라미터	내용	선택 / 필수	비고
setCallbackApi	callbackApi	Callback API URL	선택	

## 18.2 Kafka 호출 + ITG Gateway 연동 Dto KafkaSendInDto 생성 가이드 - ItggwCntr모듈 사용

메소드	파라미터	내용	선택 / 필수	비고
getKafkaSendCallItggwInDto	ItggwInDto	ITG Gateway 연동 Dto, <a href="#">11. ITG Gateway KOS 연동 - 개발 가이드 [오더AA]</a> <sup>9</sup> 를 참고하여 생성	필수	sendMsg를 스캔 하여 ordNo, svcNo를 추출하여 전달한다.
	KafkaServiceConstants	KafkaServiceConstants에서 사용하고자 Kafka Topic, 서비스 정보가 정의된 상수를 선택하여 입력	필수	
	sendSvc	this.getClass().getSimpleName() 으로 고정	필수	
getKafkaSendCallItggwInDto (method overloading)	orderNo	오더번호를 전달하면 Kafka 처리 이력 테이블 WC_KAFKA_LOG 에 ORDER_NO 컬럼으로 INSERT 되어 조회할 수 있다.	선택	sendMsg를 스캔 하지 않는다.
	svcNo	서비스계약번호를 전달하면 Kafka 처리 이력 테이블 WC_KAFKA_LOG 에 SVC_NO 컬럼으로 INSERT 되어 조회할 수 있다.	선택	
setModKey	modValue	특정키로 순차 처리가 필요할 때 특정키 필드명	선택	예시) said
setModValue	modValue	특정키가 순차 처리가 필요할 때 특정키 필드값	선택	
setCallbackLob	callbackLob	Callback API 를 제공하는 LOB코드로 설정, LobAllConstants에서 온라인 LOB_?? 와 데몬 LOB_??_DAEMON 중에서 선택	선택	
setCallbackApi	callbackApi	Callback API URL	선택	

<sup>9</sup> <https://wiki.dspace.kt.co.kr/pages/viewpage.action?pagelId=95455249>

**Producer(Publish), 개발 템플릿**

```

1  /
   *****
   *****
2  * ICIS version 1.0
3  *
4  * Copyright © 2022 kt/kt ds corp. All rights reserved.
5  *
6  * This is a proprietary software of kt corp, and you may not use this file
   except in
7  * compliance with license agreement with kt corp. Any redistribution or
   use of this
8  * software, with or without modification shall be strictly prohibited
   without prior written
9  * approval of kt corp, and the copyright notice above does not evidence
   any actual or
10 * intended publication of such software.
11
   *****
   *****/
12
13 package com.kt.icis.oder.sample.kafka.controller;
14
15 import org.springframework.stereotype.Component;
16
17 import com.kt.icis.cmmnfrwk.utils.JsonUtil;
18 import com.kt.icis.cmmnfrwk.utils.LogUtil;
19 import
   com.kt.icis.oder.baseinfo.common.comcd.query.payload.in.BiMdbMsysofcInQryPyl
   d;
20 import com.kt.icis.oder.common.kafka.controller.KafkaProducerCntr;
21 import com.kt.icis.oder.common.kafka.dto.KafkaSendInDto;
22 import com.kt.icis.oder.lob.itggw.common.controller.ItggwCntr;
23 import
   com.kt.icis.oder.lob.itggw.common.controller.client.in.dto.ItggwInDto;
24 import
   com.kt.icis.oder.lob.itggw.cst.pleqprgnadrcret.PleqRgnAdrCretFactory;
25 import
   com.kt.icis.oder.lob.itggw.cst.pleqprgnadrcret.payload.in.dto.CustCntplcInfo
   Dto;
26 import
   com.kt.icis.oder.sample.common.command.payload.in.CsyscdSaveInCmdPyld;
27
28 import lombok.RequiredArgsConstructor;
29
30 @RequiredArgsConstructor
31 @Component
32 public class KafkaSampleCntr {
33     private final ItggwCntr iggwCntr;

```

```

34     private final KafkaProducerCntr kafkaProducerCntr;
35
36     /**
37      * Kafka 호출
38      * @param inPyld
39      */
40     public void call(CsyscdSaveInCmdPyld inPyld, boolean callBack) {
41         KafkaSendInDto kafkaSendInDto =
42         kafkaProducerCntr.getKafkaSendInDto(kafkaProducerCntr.getKafkaServiceConstants("연동하고자 하는 Kafka 상수명 입력"))
43
44         , this.getClass().getSimpleName()
45
46         , JsonUtil.toJson(inPyld));
47
48         /**
49          * Callback API 설정
50          * 선택사항
51          */
52         if(callBack) {
53             kafkaSendInDto.setCallbackLob("SAMPLE");
54             kafkaSendInDto.setCallbackApi("/kafkaSample/kafkaCallback");
55         }
56
57         /**
58          * Kafka 호출
59          */
60         kafkaProducerCntr.send(kafkaSendInDto);
61     }
62
63     /**
64      * Kafka + ITG-GW 호출
65      * @param inDto
66      */
67     public void itggwCall(CustCntplcInfoDto inDto) {
68         try {
69             ItggwInDto itggwInDto =
70             PLEqpRgnAdrCretFactory.getItggwInDto(inDto);
71             KafkaSendInDto kafkaSendInDto =
72             iggwCntr.getKafkaSendCallItggwInDto(itggwInDto
73
74             , kafkaProducerCntr.getKafkaServiceConstants("연동하고자 하는 Kafka 상수명
75             입력"))
76
77             , this.getClass().getSimpleName());
78
79             /**
80              * Callback API 설정
81              * 선택사항
82              */
83             kafkaSendInDto.setCallbackLob("SAMPLE");
84             kafkaSendInDto.setCallbackApi("/kafkaSample/kafkaItggwCallback")
85
86             ;
87
88             /**
89              * Kafka 호출
90              */
91             kafkaProducerCntr.send(kafkaSendInDto);

```

```

81         }
82         catch (Exception e) {
83             LogUtil.error(e.getMessage(), e);
84         }
85     }
86
87     /**
88      * 대량 List Dto Kafka 호출
89      */
90     public void listDto(BiMdbMsysOfcInQryPyld inPyld) {
91         KafkaSendInDto kafkaSendInDto = kafkaProducerCntr.getKafkaSendInDto(
92             kafkaProducerCntr.getKafkaServiceConstants("연동하고자 하는 KAFKA 상수명 입력")
93                 , this.getClass().getSimpleName(
94                     )
95                 , JsonUtil.toJson(inPyld));
96         kafkaProducerCntr.send(kafkaSendInDto);
97     }
98 }

```

### Callback API 개발 템플릿

```

/*****
 * ICIS version 1.0
 *
 * Copyright © 2022 kt/ktids corp. All rights reserved.
 *
 * This is a proprietary software of kt corp, and you may not use this file except in
 * compliance with license agreement with kt corp. Any redistribution or use of this
 * software, with or without modification shall be strictly prohibited without prior
written
 * approval of kt corp, and the copyright notice above does not evidence any actual or
 * intended publication of such software.
 *****/

package com.kt.icis.oder.sample.kafka.api;

import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

import com.kt.icis.cmmnfrwk.utils.JsonUtil;
import com.kt.icis.cmmnfrwk.utils.LogUtil;
import com.kt.icis.oder.common.kafka.controller.KafkaProducerCntr;
import com.kt.icis.oder.common.kafka.dto.KafkaCallbackOutDto;
import com.kt.icis.oder.common.kafka.payload.KafkaCallbackOutPyld;
import com.kt.icis.oder.lob.itgw.common.controller.client.out.ItggwOutPyld;
import com.kt.icis.oder.lob.itgw.cst.pleqprgnadrcrct.PleqRgnAdrCretFactory;

```

```

import com.kt.icis.oder.lob.itggw.cst.pleqprgnadrcret.payload.in.dto.CustCntplcInfoDto;
import com.kt.icis.oder.sample.common.command.payload.in.CsyscdSaveInCmdPyld;
import com.kt.icis.oder.sample.common.command.payload.in.dto.CsyscdSaveInCmdDto;
import com.kt.icis.oder.sample.common.command.payload.out.CsyscdOutCmdPyld;
import com.kt.icis.oder.sample.kafka.controller.KafkaSampleCntr;

import io.swagger.v3.oas.annotations.Operation;
import io.swagger.v3.oas.annotations.tags.Tag;
import lombok.RequiredArgsConstructor;

@Tag(name = "kafka 샘플")
@RestController
@RequestMapping("/kafkaSample")
@RequiredArgsConstructor
public class KafkaSampleApi {
    @Operation(summary = "Kafka 호출의 Callback API")
    @PostMapping(path = "/kafkaCallback")
    public KafkaCallbackOutPyld kafkaCallback(@RequestBody CsyscdOutCmdPyld inPyld) {
        LogUtil.info("[Kafka Callback][Payload]" + JsonUtil.toJson(inPyld));
        return

KafkaCallbackOutPyld.builder().kafkaCallbackOutDs(KafkaCallbackOutDto.builder().resultY
n("Y").build()).build();
    }

    @Operation(summary = "Kafka + ITG-GW 호출의 Callback API")
    @PostMapping(path = "/kafkaItggwCallback")
    public KafkaCallbackOutPyld kafkaItggwCallback(@RequestBody ItggwOutPyld inPyld) {
        LogUtil.info("[Kafka + ITGGW Callback][Payload]" + JsonUtil.toJson(inPyld));
        CustCntplcInfoDto outDto =
PlEqpRgnAdrCretFactory.getOutDto(inPyld.getItggwOutDs());
        LogUtil.info("[Kafka + ITGGW Callback][custCntplcSeq="+outDto.getCustCntplcSeq(
)+"]");
        return

KafkaCallbackOutPyld.builder().kafkaCallbackOutDs(KafkaCallbackOutDto.builder().resultY
n("Y").build()).build();
    }
}

```



## 19 12. ITG Gateway 연동 Factory 개발 템플릿

- ITG Gateway 연동 서비스 정보, MS에서 개발된 IN / OUT을 ITG Gateway 연동 IN / OUT으로 변환
- ITG Gateway 연동 서비스 정보 상수로 선언
- ITG Gateway 연동 서비스 정보 주석으로 기술
- MS에서 개발된 IN Dto 또는 Payload 파라미터로 전달하여 ItggwInDto 를 리턴 받는 getItggwInDto 메소드 생성
- ITG Gateway 연동 후 응답 받은 ItggwOutDto 를 전달하여 MS에서 개발된 OUT Dto를 리턴 받는 getOutDto 메소드 생성
- SO명, Pipeline의 Function명, 인터페이스ID 등 서비스명 또는 Function명에 Factory 를 suffix로 더해 카멜케이스로 명명한다.

### 19.1 1. KOS 연동 Factory 템플릿

#### KOS 연동 Factory 템플릿

```

/*****
 * ICIS version 1.0
 *
 * Copyright © 2022 kt/ktids corp. All rights reserved.
 *
 * This is a proprietary software of kt corp, and you may not use this file except in
 * compliance with license agreement with kt corp. Any redistribution or use of this
 * software, with or without modification shall be strictly prohibited without prior
 * written
 * approval of kt corp, and the copyright notice above does not evidence any actual or
 * intended publication of such software.
 *****/

package com.kt.icis.oder.lob.itggw.aro.syscdlistmultiretvso;

import com.fasterxml.jackson.core.JsonProcessingException;
import com.kt.icis.cmmnfrwk.payload.BizHeader;
import com.kt.icis.cmmnfrwk.utils.JsonUtil;
import

com.kt.icis.oder.lob.itggw.aro.syscdlistmultiretvso.payload.in.SysCdListMultiRetvSoReq;
import com.kt.icis.oder.lob.itggw.aro.syscdlistmultiretvso.payload.in.dto.SysCdInDto;
import

com.kt.icis.oder.lob.itggw.aro.syscdlistmultiretvso.payload.out.SysCdListMultiRetvSoRes
Body;
import
    com.kt.icis.oder.lob.itggw.aro.syscdlistmultiretvso.payload.out.dto.SysCdListDto;
import com.kt.icis.oder.lob.itggw.common.controller.ItggwJsonUtil;
import com.kt.icis.oder.lob.itggw.common.controller.client.in.dto.ItggwInDto;
import com.kt.icis.oder.lob.itggw.common.controller.client.out.dto.ItggwOutDto;

```

```

/**
 * @bizname 시스템코드목록다건조회SO
 * @sit_esb_meta 2023-07-14
 * @prd_esb_meta
 * @chnlType T0
 *
 * 최초 개발시 sit_esb_meta와 prd_esb_meta 는 반드시 공백으로 commit 하세요
 */
public class SysCdListMultiRetvSoFactory {
    private final static String APP_NAME = "NBSS_ARO";
    private final static String SVC_NAME = "SysCdListMultiRetvSO";
    private final static String FN_NAME = "service";
    private final static String PIPELINE_YN = "N";
    private final static String DYNAMIC_YN = "Y";
    private final static String REQ_ROOT_NM = "service_request"; // 기본값이
    service_request 니 필요한 경우 외에 설정하지 말 것 (여기는 가이드 목적으로 설정했음)

    /**
     * ITG Gateway 연동을 위한 ItggwInDto 생성 후 반환
     * orderId가 없는 경우
     * @param dto
     * @return
     */
    public static ItggwInDto getItggwInDto(SysCdInDto dto) throws
    JsonProcessingException {
        return getItggwInDto(dto, "");
    }

    /**
     * ITG Gateway 연동을 위한 ItggwInDto 생성 후 반환
     * @param dto
     * @param orderId
     * @return
     */
    public static ItggwInDto getItggwInDto(SysCdInDto dto, String orderId) throws
    JsonProcessingException {
        String requestJson =
        JsonUtil.getMapperNonEmpty().writeValueAsString(SysCdListMultiRetvSoReq.builder()
        .bizHeader(BizHeader.builder()
            .cbSvcName(SVC_NAME)
            .cbFnName(FN_NAME)
            .orderId(orderId).build()).sysCdInDto(dto).build());

        return ItggwInDto.builder()
            .appName(APP_NAME)
            .svcName(SVC_NAME)
            .fnName(FN_NAME)
            .pipelineYn(PIPELINE_YN)
            .dynamicYn(DYNAMIC_YN)
            .reqRootNm(REQ_ROOT_NM)
            .requestJson(requestJson).build();
    }

    /**
     * ITG Gateway 연동후 Json to Object 하고 OutDto만 반환

```

```

    * @param itgOutDto
    * @return
    */
    public static SysCdListDto getOutDto(ItggwOutDto itgOutDto) {
        SysCdListMultiRetvSoResBody outRes =
        ItggwJsonUtil.readJson(itgOutDto.getResponseJson(), SysCdListMultiRetvSoResBody.class);
        return outRes.getServiceResponse().getSysCdListDto();
    }
}

```

## 19.2 2. AnyLink 연동 Factory 템플릿

### AnyLink 연동 Factory 템플릿

```

package com.kt.icis.oder.lob.itggw.anylink.lneo.trif_i_tord_lneo_e_0956_00;

import java.util.List;

import com.fasterxml.jackson.core.JsonProcessingException;
import com.kt.icis.cmmnfrwk.utils.JsonUtil;
import

com.kt.icis.oder.lob.itggw.anylink.lneo.trif_i_tord_lneo_e_0956_00.payload.in.KonetPreO
rderingInQryPyld;
import

com.kt.icis.oder.lob.itggw.anylink.lneo.trif_i_tord_lneo_e_0956_00.payload.out.KonetPre
OrderingOutQryPyld;
import

com.kt.icis.oder.lob.itggw.anylink.lneo.trif_i_tord_lneo_e_0956_00.payload.out.dto.Kone
tPreOrderingOutQryDto;
import com.kt.icis.oder.lob.itggw.common.controller.ItggwJsonUtil;
import com.kt.icis.oder.lob.itggw.common.controller.client.in.dto.ItggwInDto;
import com.kt.icis.oder.lob.itggw.common.controller.client.out.dto.ItggwOutDto;

/**
 * @ifid TRIF_I_TORD_LNEO_E_0956_00
 * @bizname 코넷오피스 FES시설정보 프리오더링 조회
 */
public class TrifITordLneoE095600Factory {

    private final static String SVC_NAME = "TORD/TRIF_I_TORD_LNEO_E_0956_00";

    /**
     * ITG Gateway 연동을 위한 ItggwInDto 생성 후 반환
     * @param inPyld
     * @return

```

```

    * @throws JsonProcessingException
    */
    public static ItggwInDto getItggwInDto(KonetPreOrderingInQryPyld inPyld) throws
    JsonProcessingException {
        return ItggwInDto.builder()
            .svcName(SVC_NAME)
            .requestJson(JsonUtil.toJson(inPyld)).build();
    }

    /**
     * ITG Gateway 연동후 Json to Object 하고 OutDto만 반환
     * @param itgOutDto
     * @return
     */
    public static List<KonetPreOrderingOutQryDto> getOutDto(ItggwOutDto itgOutDto) {
        KonetPreOrderingOutQryPyld outPyld =
        ItggwJsonUtil.readJson(itgOutDto.getResponseJson(), KonetPreOrderingOutQryPyld.class);
        return outPyld.getKonetPreOrderingOutQryDto();
    }
}

```

## 19.3 3. InfiniLink 연동 Factory 템플릿

### InfiniLink 연동 Factory 템플릿

```

/*****
 * ICIS version 1.0
 *
 * Copyright © 2022 kt/kt ds corp. All rights reserved.
 *
 * This is a proprietary software of kt corp, and you may not use this file except in
 * compliance with license agreement with kt corp. Any redistribution or use of this
 * software, with or without modification shall be strictly prohibited without prior
 * written
 * approval of kt corp, and the copyright notice above does not evidence any actual or
 * intended publication of such software.
 *****/

package com.kt.icis.oder.lob.itggw.infinilink.ekcs.trif_e_tord_ekcs_x_2819_00;

import com.fasterxml.jackson.core.JsonProcessingException;
import com.kt.icis.cmmnfrwk.utils.JsonUtil;
import com.kt.icis.oder.lob.itggw.common.controller.ItggwJsonUtil;
import com.kt.icis.oder.lob.itggw.common.controller.client.in.dto.ItggwInDto;
import com.kt.icis.oder.lob.itggw.common.controller.client.out.dto.ItggwOutDto;
import

com.kt.icis.oder.lob.itggw.infinilink.ekcs.trif_e_tord_ekcs_x_2819_00.payload.in.Aprgsd
sDelTrtSocketInCmdPyld;
import

```

```

com.kt.icis.oder.lob.itggw.infinilink.ekcs.trif_e_tord_ekcs_x_2819_00.payload.out.AprgsdsDelTrtSocketOutCmdPyld;
import

com.kt.icis.oder.lob.itggw.infinilink.ekcs.trif_e_tord_ekcs_x_2819_00.payload.out.dto.AprgsdsDelTrtSocketOutCmdDto;

/**
 * @ifid TRIF_E_TORD_EKCS_X_2819_00
 * @bizname 사은품 삭제처리 (소켓연동)
 */
public class TrifETordEkcsX281900Factory {

    private final static String SVC_NAME = "http/TORD/TRIF_E_TORD_EKCS_X_2819_00";

    /**
     * ITG Gateway 연동을 위한 ItggwInDto 생성 후 반환
     * @param inPyld
     * @return
     * @throws JsonProcessingException
     */
    public static ItggwInDto getItggwInDto(AprgsdsDelTrtSocketInCmdPyld inPyld) throws
    JsonProcessingException {
        return ItggwInDto.builder()
            .svcName(SVC_NAME)
            .requestJson(JsonUtil.toJson(inPyld)).build();
    }

    /**
     * ITG Gateway 연동후 Json to Object 하고 OutDto만 반환
     * @param itgOutDto
     * @return
     */
    public static AprgsdsDelTrtSocketOutCmdDto getOutDto(ItggwOutDto itgOutDto) {
        AprgsdsDelTrtSocketOutCmdPyld outPyld =
        ItggwJsonUtil.readJson(itgOutDto.getResponseJson(), AprgsdsDelTrtSocketOutCmdPyld.class)
        ;
        return outPyld.getAprgsdsDelTrtSocketOutCmdDto();
    }
}

```

## 19.4 4. API Link 연동 Factory 템플릿

### API Link 연동 Factory 템플릿

```

/*****
 * ICIS version 1.0
 *
 * Copyright © 2022 kt/kt ds corp. All rights reserved.
 *
 * This is a proprietary software of kt corp, and you may not use this file except in

```

```

* compliance with license agreement with kt corp. Any redistribution or use of this
* software, with or without modification shall be strictly prohibited without prior
written
* approval of kt corp, and the copyright notice above does not evidence any actual or
* intended publication of such software.
*****/

package com.kt.icis.oder.lob.itggw.apilink.oif_50293;

import com.fasterxml.jackson.core.JsonProcessingException;
import com.kt.icis.cmmnfrwk.utils.JsonUtil;
import

com.kt.icis.oder.lob.itggw.apilink.oif_50293.payload.in.CheckDupActiveIdRestJsonInPyld;
import

com.kt.icis.oder.lob.itggw.apilink.oif_50293.payload.out.CheckDupActiveIdRestJsonOutPyld;
import

com.kt.icis.oder.lob.itggw.apilink.oif_50293.payload.out.dto.CheckDupActiveIdRestJsonOutDto;
import com.kt.icis.oder.lob.itggw.common.controller.ItggwJsonUtil;
import com.kt.icis.oder.lob.itggw.common.controller.client.in.dto.ItggwInDto;
import com.kt.icis.oder.lob.itggw.common.controller.client.out.dto.ItggwOutDto;

/**
 * @bizname 회선계약의 서비스ID를 신규로 생성하는 경우 호출, 웹계정ID와 회선계약의 서비스ID를 확인해 중복여부를 확인
, ID예약을 수행
 */
public class CheckDupActiveIdRestJsonFactory {

    private final static String APP_NAME = "cus-in";
    private final static String SVC_NAME = "checkDupActiveIdRestJson";
    private final static String FN_NAME = "OIF_50293";
    private final static String PIPELINE_YN = "JSON";

    /**
     * ITG Gateway 연동을 위한 ItggwInDto 생성 후 반환
     * @param inDto
     * @return
     * @throws JsonProcessingException
     */
    public static ItggwInDto getItggwInDto(CheckDupActiveIdRestJsonInPyld inPyld)
throws JsonProcessingException {
        return ItggwInDto.builder()
            .appName(APP_NAME)
            .svcName(SVC_NAME)
            .fnName(FN_NAME)
            .pipelineYn(PIPELINE_YN)
            .requestJson(JsonUtil.toJson(inPyld)).build();
    }

    /**
     * ITG Gateway 연동후 Json to Object 하고 OutPyld 반환
     * @param itggwOutDto

```

```

    * @return
    */
    public static CheckDupActiveIdRestJsonOutPyld getOutPyld(ItggwOutDto itggwOutDto) {
        return ItggwJsonUtil.readJson(itggwOutDto.getResponseJson(),
        CheckDupActiveIdRestJsonOutPyld.class);
    }

    /**
     * ITG Gateway 연동후 Json to Object 하고 OutDto 반환
     * @param itggwOutDto
     * @return
     */
    public static CheckDupActiveIdRestJsonOutDto getOutDto(ItggwOutDto itggwOutDto) {
        CheckDupActiveIdRestJsonOutPyld outPyld = getOutPyld(itggwOutDto);
        return outPyld.getCheckDupActiveIdRestJsonOutDto();
    }
}

```

## 20 13. Redis 연동 개발 템플릿

- Service에서 com.kt.icis.oder.common.redis.service.RedisAndQrySvc를 호출하여 Redis 연동
- com.kt.icis.oder.common.constants.RedisKeyConstants 사용 가능한 Redis 상수가 정의되어 있고 그 목록은 다음의 페이지에서 제공 [01. Redis Key 목록 \[오더AA\]](#)<sup>10</sup>
- RedisKeyConstants.getRedisKey 를 호출하여 Redis Key로 변환
- RedisAndQrySvc.findListValue(RedisKeyConstants redisConstants, String key, Function function, Object dto) 로 Redis 연동
- RedisAndQrySvc.findListValue는 Redis를 조회하여 값이 있으면 반환하고 없으면 DB를 조회하여 Redis에 Set 하고 반환
- RedisAndQrySvc.findListValue 에서 Function function은 DB 조회 lambda 코드이다.

### Redis 연동 개발 템플릿

```

/*****
 * ICIS version 1.0
 *
 * Copyright © 2022 kt/kt ds corp. All rights reserved.
 *
 * This is a proprietary software of kt corp, and you may not use this file except in
 * compliance with license agreement with kt corp. Any redistribution or use of this
 * software, with or without modification shall be strictly prohibited without prior
 * written
 * approval of kt corp, and the copyright notice above does not evidence any actual or
 * intended publication of such software.
 *****/

package com.kt.icis.oder.baseinfo.common.query.service.csys;

import java.util.ArrayList;
import java.util.List;
import java.util.function.Function;

import org.springframework.stereotype.Service;

import com.kt.icis.cmmnfrwk.utils.LogUtil;
import com.kt.icis.oder.baseinfo.common.comcd.query.payload.in.dto.BiCsyscdInQryDto;
import com.kt.icis.oder.baseinfo.common.comcd.query.payload.out.dto.BiCsyscdOutQryDto;
import com.kt.icis.oder.baseinfo.common.constants.BiqueryConstants;
import com.kt.icis.oder.baseinfo.common.query.repository.csys.BiCsyscdQryMbtsRepo;
import com.kt.icis.oder.common.constants.RedisKeyConstants;
import com.kt.icis.oder.common.exception.OrderException;
import com.kt.icis.oder.common.redis.service.RedisAndQrySvc;
import com.kt.icis.oder.common.utils.DtoUtil;
import com.kt.icis.oder.common.utils.StringUtils;

import lombok.RequiredArgsConstructor;

```

<sup>10</sup> <https://wiki.dspace.kt.co.kr/pages/viewpage.action?pageId=121141784>



```

/**
 * MDB BI_SCYSCD Service
 */
@RequiredArgsConstructor
@Service
public class BiCsyscdQrySvc {
    private final BiCsyscdQryMbtsRepo csyscdDynamicQryRepo;
    private final RedisAndQrySvc redisAndQrySvc;

    public List<BiCsyscdOutQryDto> findCsyscd(BiCsyscdInQryDto inDto) {

        if (DtoUtil.getCount(inDto)<1) {
            LogUtil.info("조회 조건을 추가해주세요.");
            throw new OrderException("IOBIE0001");
        }
        List<BiCsyscdOutQryDto> list = new ArrayList<BiCsyscdOutQryDto>();

        //검색조건이 grp_id 만 있는 경우만 redis를 이용
        if(StringUtils.isNotEmpty(inDto.getGrpId()) &&
            StringUtils.isEmpty(inDto.getSaCd(),
                                inDto.getSoItemId(),
                                inDto.getCd(),
                                inDto.getName(),
                                inDto.getRef1(),
                                inDto.getRef2(),
                                inDto.getRef4(),
                                inDto.getRef5(),
                                inDto.getDefineName(),
                                inDto.getStartDate(),
                                inDto.getRemark(),
                                inDto.getWhereTxt(),
                                inDto.getEffectiveDateCheck()) &&
            inDto.getRef3() == null &&
            inDto.getOutputSeq() == null)
        {
            for(String grpId : inDto.getGrpId().split(BiqueryConstants.MDB_DELIM)) {
                String redisKey =
                    RedisKeyConstants.getRedisKey(RedisKeyConstants.REDIS_BI_MDB_CSYS_CD_GRPID.key(),
                    grpId);
                findCsyscdRedis(RedisKeyConstants.REDIS_BI_MDB_CSYS_CD_GRPID, redisKey,
                    list, BiCsyscdInQryDto.builder().grpId(grpId).build());
            }
        }
        //검색조건이 grp_id 와 cd 만 있는 경우만 redis를 이용
        else if(StringUtils.isNotEmpty(inDto.getGrpId()) &&
            StringUtils.isNotEmpty(inDto.getCd()) &&
            StringUtils.isEmpty( inDto.getSaCd(),
                                inDto.getSoItemId(),
                                inDto.getName(),
                                inDto.getRef1(),
                                inDto.getRef2(),
                                inDto.getRef4(),
                                inDto.getRef5(),
                                inDto.getDefineName(),
                                inDto.getStartDate(),

```

```

        inDto.getRemark(),
        inDto.getWhereTxt(),
        inDto.getEffectiveDateCheck()) &&
        inDto.getRef3() == null &&
        inDto.getOutputSeq() == null)
    {
        for(String grpId : inDto.getGrpId().split(BiqueryConstants.MDB_DELIM)) {
            for(String cd : inDto.getCd().split(BiqueryConstants.MDB_DELIM)) {
                String redisKey =
RedisKeyConstants.getRedisKey(RedisKeyConstants.REDIS_BI_MDB_CSYS_CD_GRPID_CD.key(),
grpId, cd);
                findCsycsdRedis(RedisKeyConstants.REDIS_BI_MDB_CSYS_CD_GRPID_CD,
redisKey, list, BiCsycsdInQryDto.builder().grpId(grpId).cd(cd).build());
            }
        }
        else {
            list = csycsdDynamicQryRepo.findCsycsd(inDto);
        }

        return list;
    }

    @SuppressWarnings("unchecked")
    private void findCsycsdRedis(RedisKeyConstants redisConstants, String key,
List<BiCsycsdOutQryDto> list, BiCsycsdInQryDto inDto) {
        Function<BiCsycsdInQryDto, List<BiCsycsdOutQryDto>> function = (fnDto) ->
csycsdDynamicQryRepo.findCsycsd(fnDto);
        list.addAll((List<BiCsycsdOutQryDto>)
redisAndQrySvc.findListValue(redisConstants, key, function, inDto));
    }
}

```