NIDS 한국의료기기안전정보원
National Institute of Medical Device Safety Information

의료기기통합정보시스템 고도화

OpenAPI 가이드

InVaKo 인포벨리코리아

문서번호 : MSIS20-MA-SG-0202

Ver. 1.9

2022.07.21

사업명	의료기기통합정보시스템 기능개선				
구 분	소속	직책	성명	일자	서명
작성자	인포벨리코리아	TA	최재만	2020.07.02	
검토자	인포벨리코리아	PM	강길용		
승인자	한국의료기기안전정보원	연구원	김지연		

〈 개 정 이 력 〉

버전	변경사유	변경내용	개정일	작성자	검토자
1.0	신규	신규작성	2019.08.22	지원근	하정우
1.1	변경	오타 및 잘못 연결된 참조 수정	2020.01.13	지원근	
1.2	추가	공급내역보고 관련 공급구분 신규목록 추가 및 공급형태 신규 항목 추가 거래처 목록 조회 신규 검색 조건 추가 요청필드 및 응답필드 타입 size 추가 공급내역 보고자료 추가 응답필드 추가 오타 및 잘못 연결된 참조 수정	2020.06.25	최재만	
1.3	추가	공급내역보고 공급구분 중 회수일 때 회수 수량 관련 항목 추가 오타 수정 및 응답 예제 현행화	2020.07.02.	최재만	
1.4	변경	25. 고유식별자(UDI-DI) 품목 정보 조회 API 응답예제 수정	2020.07.23.	최재만	
1.5	변경	공급구분이 '출고, 임대'일 때에만 공급형태 값을 입력하도록 수정 조회 API에 문자길이 정보 추가	2020.08.13.	최재만	
1.6	변경	16. 통합업체 목록조회 API 업종 상세명 응답필드 추가 25. 고유식별자(UDI-DI) 품목 정보 조회 API 응답필드 오기 수정	2020.09.29.	최재만	
1.7	변경	6. UDI-DI 통합정보 목록 조회 7. UDI-DI 코드 및 통합정보 추가 9. 통합정보 수정 응답필드 추가 25. 고유식별자(UDI-DI) 품목 정보 조회 응답필드 추가 - 사용자멸균 정형용품 여부 - 세트화 구성 여부	2021.06.30	김형성	
1.8	변경	9. 통합정보 수정 - 프탈레이트류 포함 여부 설명 변경	2022.02.18	최현정	
1.9	추가	공급내역보고 관련 중고의료기기여부 신규 항목 추가	2022.07.21	최현정	

*1) 버전: 승인을 득한 이후에는 1.0부터 시작하여 정수 단위로 변경 관리 함
변경 발생 시 소수점 아래 번호로 관리하고, 큰 변경이 발생하면 상위 정수를 변경 함
(예, V1.2 : 2번 수정됨, 목차 내용이 변경되면 V2.0 이 됨)
*2) 변경 사유 : 변경 내용이 이전 문서에 대해 신규/추가/수정/삭제 인지 선택 기입
*3) 변경 내용 : 변경 내용을 자세히 기록

목 차

1. 개요 ····· 4
2. USECASE ····· 6
3. OAUTH2 Access Token 요청 ····· 8
4. 품목 목록 조회 ····· 11
5. 모델 목록 조회 ····· 14
6. UDI-DI 통합정보 목록 조회 ····· 17
7. UDI-DI 코드 및 통합정보 추가 ····· 23
8. UDI-DI 코드 삭제 ····· 28
9. 통합정보 수정 ····· 30
10. 물류바코드 목록 조회 ····· 39
11. 물류바코드 추가 ····· 42
12. 물류바코드 수정 ····· 45
13. 물류바코드 삭제 ····· 48
14. 요양급여 정보 조회 ····· 50
15. 거래처 목록 조회 ····· 52
16. 통합업체 목록 조회 ····· 56
17. 도로명주소 조회 ····· 59
18. 거래처 등록 ····· 62
19. 거래처 수정 ····· 66
20. 공급내역 보고자료 목록 조회 ····· 70
21. 공급내역 보고자료 추가 ····· 76
22. 공급내역 보고자료 수정 ····· 81
23. 공급내역 보고자료 일괄 삭제 ····· 85
24. 공급내역 보고자료 단건 삭제 ····· 87
25. 고유식별자(UDI-DI) 품목 정보 조회 ····· 89
26. 공급내역 보고 이력 목록 조회 ····· 93
27. 공급내역 보고 현황 목록 조회 ····· 96
28. 공급내역 보고 및 재보고 처리 ····· 107

1. 개요

본 문서는 의료기기 통합정보시스템(이하 '통합정보시스템'이라 한다)에 의료기기 표준코드(UDI) 및 통합정보 등록을 위해 공개된 API 활용법을 설명하는 개발자용 문서입니다.

API 관련 업무를 개발하기 위해서는 의료기기업체의 업무 프로세스를 먼저 숙지하여야 합니다.

1.1. 구성

통합정보시스템 공개 API는 **HTTP REST**, **JSON 포맷** 통신규약을 전제하여 구성되어 있습니다. 모든 응답은 JSON Object로 진행됩니다.

1.2. 보안

통합정보시스템 공개 API는 OAuth2 스펙을 통해 인증을 제공합니다. token 요청을 제외한 모든 기능은 Header에 **Authorization: {access_token}을** 포함하여야 합니다.

1.3. 제공하는 기능

기능명칭	기능설명
Access Token 발급	
품목 목록 조회	사용자 업체의 허가(인증·신고) 품목 목록을 조회
모델 목록 조회	사용자 업체의 허가(인증·신고) 모델 목록을 조회
UDI-DI 통합정보 목록 조회	모델별 등록된 UDI-DI 코드 및 통합정보를 조회
UDI-DI 코드 및 통합정보 추가	모델별 UDI-DI 코드 등록
UDI-DI 코드 수정	등록된 UDI-DI 코드 정보를 수정
UDI-DI 코드 삭제	등록된 UDI-DI 코드를 삭제
통합정보 수정	통합정보를 수정
물류바코드 목록 조회	통합정보 항목 중 물류바코드 정보를 조회
물류바코드 추가	통합정보 항목 중 물류바코드 정보를 추가
물류바코드 수정	통합정보 항목 중 물류바코드 정보를 수정
물류바코드 삭제	통합정보 항목 중 물류바코드 정보를 삭제
요양급여 정보 조회	요양급여코드로 해당 코드의 요양급여정보를 조회
거래처 목록 조회	공급내역보고를 위해 등록된 거래처 목록 조회
통합업체 목록 조회	거래처를 등록하기 위한 통합정보시스템에 등록된 업체 목록 조회
도로명주소 조회	거래처 등록 시 필요한 도로명 주소를 조회
거래처 등록	공급내역보고를 위한 거래처 등록
거래처 수정	등록된 거래처 수정
공급내역 보고자료 목록 조회	공급내역 보고자료 목록 조회
공급내역 보고자료 추가	공급내역 보고자료 추가

공급내역 보고자료 수정	공급내역 보고자료 수정
공급내역 보고자료 일괄 삭제	기준 월의 공급내역 보고자료 일괄 삭제
공급내역 보고자료 단건 삭제	공급내역 보고자료 단건 삭제
고유식별자(UDI-DI) 품목 정보 조회	고유식별자(UDI-DI) 코드의 품목 정보 조회
공급내역 보고 이력 목록 조회	기준 월 별로 공급내역 보고된 요약정보 목록 조회
공급내역 보고 현황 목록 조회	공급내역 보고된 현황 목록 조회
공급내역 보고 및 재보고 처리	공급내역 보고자료 정보를 보고 및 재보고 처리 수행

1.4. 개발환경 구성

HTTP 요청을 보낼 수 있는 환경이면 어디에서든 이용 가능합니다.

1.4.1. API Test 제공

아래 소개하는 **OAuth2 인증을 제외한** 모든 API는 /api/v1/{resource}/** 로 구성되어있으며, 해당 기능을 테스트하기 위해서는 /api/test/v1/{resource}/** 와 같이 **api하위**로 **test경로만 추가** 하면 테스트 해 보실 수 있습니다. 조회 기능을 제외한 등록, 수정, 삭제 기능들은 실제로 등록, 수정, 삭제가 수행되지 않고 입력값 검증 결과만 확인하기 때문에 **응답값이 본 문서에 설명과는 다를 수 있음**을 알려드립니다. 응답값을 제대로 입력했는지 테스트 하는 용도로 사용하실 수 있습니다.

1.5. 제한사항

1.5.1. HTTP Method

HTTP REST 스펙을 구현함에 있어서 CRUD에 HTTP Method를 적절히 사용해야 하나, 식품의약품안전처 시스템 보안 규정상 GET, POST 외의 Method를 사용할 수 없어 부득이 변형된 형태를 사용하게 됨을 양해바랍니다.

1.5.2. 정상적인 데이터 저장 및 수정

정상적인 데이터 저장을 위해 모든 항목 전송 해주셔야합니다. 항목이 누락될 경우 오류가 발생할 수 있습니다.

ex) { "suplyFlagCode":"2",
"suplyTypeCode":"3",
"meddevItemSeq":20060048,
"seq":7149,
"udiDiSeq":1958,
"stdCode":"(01)08811344578771(10)Q1231(11)191001",
"udiDiCode":"08811344578771",
"udiPiCode":"(10)Q1231(11)191001",
"lotNo":"Q1231",

"itemSeq":"",
"manufYm":"191001",
"useTmlmt":"",
"indrtDvyfgYn":"",
"dvyfgPlaceBcncCode":"",
"bcncCode":"100013321000001",
"suplyDate":"20200502",
"suplyQty":"1000",
"remark":"API 테스트") }

※ 참고사항

요청필드 필수 항목에 "O"라고 표시되어 있는 데이터 항목을 누락한 경우 string 타입은 null, boolean 타입은 "N" 으로 업데이트됩니다.

1.5.3. 예제코드

HTTP 프로토콜에 대한 예제만 제공하며, 상세한 예제코드는 Java 기반으로만 제공합니다. 다른 언어는 Java 코드를 참고하여 개발해야 합니다. 예제코드의 restTemplate은 Spring Framework의 RestTemplate를 사용했습니다.

2. USECASE

일반적인 업무 흐름에 따른 API USECASE를 설명합니다.

2.1. AccessToken 발급

통합정보시스템의 모든 API를 사용하기 위해서는 OAuth2 기반 인증절차가 선행되어야 합니다. 관련 상세 스펙은 **[OAUTH2 Access Token 요청]**을 참고하여 API를 사용하기 위한 유효한 Access Token을 유지해야 합니다.

2.2. UDI-DI 코드 및 통합정보 등록

모델별 UDI-DI 코드 및 통합정보를 등록하는 시나리오를 설명합니다.

UDI-DI 코드 및 통합정보 등록을 위해서는 우선 UDI-DI 코드를 등록할 모델을 **[모델 목록 조회]** API로 식별하여, **[UDI-DI 코드 및 통합정보 추가]** API를 통해 UDI-DI코드를 등록하고 그에 따른 통합정보 내용을 **[통합정보 수정]** API를 통해 등록하는 절차를 따라야 합니다. 이때, 요양급여 정보는 **[요양급여 정보 조회]** API로 조회한 정보를 설정해야 합니다. **[UDI-DI 코드 및 통합정보 추가]** 과정에서 통합정보 일부 항목(허가정보 연계항목)은 자동으로 입력됩니다.

이후 상세 물류 바코드를 **[물류바코드 추가]** API로 등록하는 과정을 통해 통합정보 입력이 진행되며, 통합정보 등록이 완료되는 기준(웹서비스와 동일)을 충족하면 통합정보 등록이 완료된 것으로 처리됩니다. 물류바코드 추가 과정에서 수정 혹은 삭제 요구 발생 시에는 **[물류바코드 수정]**, **[물류바코드 삭제]** API를 통해 이를 수행할 수 있습니다.

등록된 혹은 등록중인 UDI-DI 코드 및 통합정보는 **[UDI-DI 통합정보 목록 조회]** API를

통해 조회할 수 있으며, 통합정보의 점진적 작성을 위해서는 **[UDI-DI 통합정보 목록 조회]**의 결과정보로 반복적인 통합정보 수정 작업을 진행할 수 있습니다.

2.3. 공급내역 보고 처리

업체 별 공급내역 보고를 처리하는 시나리오를 설명합니다.

공급내역 보고 처리를 위해서는 우선 특정 기준 월의 **[공급내역 보고자료 추가]** API를 통해 공급내역 보고자료를 추가합니다. 이때, 공급내역 보고자료에 추가하는 품목정보를 **[고유식별자(UDI-DI) 품목 정보 조회]** API를 통해 조회할 수 있습니다.

등록된 공급내역 보고자료 정보를 **[공급내역 보고자료 목록 조회]** API를 통해 조회합니다. 이때, 잘못 등록된 공급내역 보고자료가 있을 경우 **[공급내역 보고자료 수정]** 및 **[공급내역 보고자료 단건 삭제]** API를 통해 한 건 씩 수정, 삭제 할 수 있으며, **[공급내역 보고자료 일괄 삭제]** API를 통해 해당 기준 월의 공급내역 보고자료를 한 번에 삭제할 수 있습니다.

이후 기준 월에 등록된 공급내역 보고자료를 **[공급내역 보고 및 재보고]** API로 공급내역 보고 처리 요청을 합니다.(보고 가능 기한 내에 보고자료는 재보고 가능합니다.) **[공급내역 보고 이력 목록 조회]** API로 공급내역 보고 요약 정보를 조회할 수 있고, **[공급내역 보고 현황 목록 조회]** API를 통해 공급내역 보고 상세 정보를 확인할 수 있습니다.

3. OAUTH2 Access Token 요청

통합정보시스템의 API를 사용하기 위해서는 OAuth2기반의 인증정보가 필요합니다. 클라이언트에서 API에 제공하는 인증정보를 **Access Token**이라고 하며 Access Token을 HTTP 헤더의 **Authorization** 에 설정함으로써 시스템이 API를 호출하는 클라이언트를 식별할 수 있도록 합니다. 본 절은 Access Token을 발급받는 방법을 설명합니다.

3.1. Request

```
POST /api/oauth/token HTTP/1.1
Host: udiportal.mfds.go.kr
Authorization: Basic {auth}
Content-Type: application/x-www-form-urlencoded;charset=UTF-8
Accept: application/json;charset=UTF-8
```

통합정보시스템 웹서비스의 회원정보수정 화면에서 ClientId와 ClientSecret을 확인합니다. 이 값을 "{ClientId}:{ClientSecret}" 형태의 문자열로 만들어 Base64로 인코딩하고 HTTP Header의 Authorization 에 설정하고 아래 필드를 form submit (**form-urlencoded**) 형태로 POST 요청합니다.

- **요청필드**

키	설명	값	타입
grant_type	인증부여 타입(고정)	client_credentials	string

- **요청예제**

```
POST /api/oauth/token HTTP/1.1
Host: udiportal.mfds.go.kr
Authorization: Basic {base64 encoded clientId:clientSecret}
Content-Type: application/x-www-form-urlencoded;charset=UTF-8
Accept: application/json;charset=UTF-8

grant_type=client_credentials
```

- **예제 코드 (Java, Spring)**

```
public static String obtainAccessToken(String clientId, String clientSecret) throws JSONException {
    String authService = "https://udiportal.mfds.go.kr/api/oauth/token";

    String value = String.format("%s:%s", clientId, clientSecret);
    String authorization = "Basic " +
            Base64.getEncoder().encodeToString(value.getBytes(Charset.forName("UTF-8")));
```

```
    MultiValueMap<String, String> formParam = new LinkedMultiValueMap<>();
    formParam.add("grant_type", "client_credentials");

    HttpHeaders headers = new HttpHeaders();
    headers.add(HttpHeaders.AUTHORIZATION, authorization);
    headers.add(HttpHeaders.ACCEPT, MediaType.APPLICATION_JSON_UTF8_VALUE);

    HttpEntity<MultiValueMap<String, String>> requestEntity = new HttpEntity<>(formParam, he
aders);
    ResponseEntity<String> responseEntity =
            restTemplate.exchange(URI.create(authService),
                    HttpMethod.POST,
                    requestEntity,
                    String.class);

    return new JSONObject(responseEntity.getBody()).getString("access_token");
}
```

3.2. Response

- **응답코드**

코드	설명
200	정상 응답
401	ClientId 가 존재하지 않거나 ClientSecret이 잘못된 경우
500	처리 중 오류가 발생했을 경우

- **응답필드**

키	설명	타입
access_token	Access Token	string
token_type	Access Token의 타입. "bearer" 고정값	string
expires_in	Access Token 만료 시간(초). 기본값 86400초(1일) 예) 63849 초 후에 만료됨을 나타냄	number
scope	접근할 수 있는 권한 범위	string

- **응답예제**

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8

{"access_token":"{access_token}","token_type":"bearer","expires_in":63849,"scope":"api_call"}
```

{access_token}값을 이후 요청시 **Authorization** 헤더에 **Bearer {access_token}** 형태로 추가해 사용합니다. Access Token은 expires_in 필드의 값으로 계산하여 만료되기 전 새로운 Access Token을 발급받아 놓거나, 응답 코드가 401(인증정보 없음)일 때 재발급 받아서 다시 요청하는 등 유효기간 만료에 대비해야 합니다.

- **에러 응답 예제 – ClientId 가 존재하지 않거나 ClientSecret이 잘못된 경우**

```
HTTP/1.1 401 UNAUTHORIZED
WWW-Authenticate: Basic realm="oauth2/client"
Content-Type: application/json;charset=UTF-8

{"timestamp":"2019-08-11T09:53:58.998+0000","status":401,"error":"Unauthorized","message":"Unauthorized","path":"/oauth/token"}
```

4. 품목 목록 조회

본 API는 품목허가 목록을 조회합니다. UDI-DI 통합정보 관련 API를 사용하는데 필요한 값은 응답필드에 굵은글씨(Bold)처리 되어있습니다.

4.1. Request

```
GET /api/v1/int-info/items?offset=1&limit=50 HTTP/1.1
Host: udiportal.mfds.go.kr
Authorization: Bearer {access_token}
Accept: application/json;charset=UTF-8
```

Access_token을 Authorization Header에 담아 GET 으로 요청합니다.

◆ **요청필드**

키	설명	구분	필수	타입
offset	페이징이 시작될 숫자	Query	O	number
limit	페이징에 담길 개수 (최대 100)	Query	O	number
itemName	검색할 품목명 (LIKE 검색)	Query	X	string(500)
productName	검색할 제품명 (LIKE 검색)	Query	X	string(3000)
grade	등급	Query	X	string(1)
searchStart	검색 시작일자 format : YYYY-MM-DD	Query	X	string(10)
searchEnd	검색 종료일자 format : YYYY-MM-DD	Query	X	string(10)
udiDiCode	검색할 UDI-DI코드 (LIKE 검색) 해당 코드가 포함된 품목들을 조회	Query	X	string(60)

◆ **예제**

```
GET /api/v1/int-info/items?offset=1&limit=10 HTTP/1.1
Host: udiportal.mfds.go.kr
Authorization: Bearer {access_token}
Accept: application/json;charset=UTF-8
```

◆ **예제코드 (Java, Spring)**

```
        UriComponentsBuilder uriBuilder = UriComponentsBuilder.fromHttpUrl("https://udiportal.mfds.go.kr/api/v1/int-info/items")
                .queryParam("offset", 1)
                .queryParam("limit", 10);
        HttpHeaders headers = new HttpHeaders();
```

```
String accessToken = obtainAccessToken(clientId, clientSecret);
headers.add(HttpHeaders.AUTHORIZATION, "Bearer " + accessToken);
headers.add(HttpHeaders.ACCEPT, MediaType.APPLICATION_JSON_UTF8_VALUE);
HttpEntity httpEntity = new HttpEntity(headers);

ResponseEntity<String> responseEntity = restTemplate.exchange(uriBuilder.toUriString(),
        HttpMethod.GET,
        httpEntity,
        String.class);

JSONObject body = new JSONObject(responseEntity.getBody());
```

4.2. Response

- **응답코드**

코드	설명
200	정상응답
400	입력값이 잘못되었을 경우
403	해당 정보에 접근 권한이 없을 경우
401	인증정보가 존재하지 않을 경우
500	처리 중 오류가 발생했을 경우

◆ 응답필드

키	설명	타입
page	현재 페이지 번호	number
pageSize	페이지 크기	number
totalElements	총 갯수	number
items[].itemName	품목명	string(500)
items[].permitDate	품목허가일자	string(10)
items[].**grade**	등급	string(1)
items[].**meddevItemSeq**	품목일련번호	string(100)
items[].**isForExport**	수출용 여부	string(1)
items[].**cobFlagCode**	제조/수입 구분 • 제조 : MNFCTUR • 수입 : INCME	string(10)
items[].mebTypeName	형명(모델명)	string(10)
items[].meddevItemNo	품목 허가 번호	string(50)

◆ 응답예제

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8

{
  "pageSize": 10,
  "page": 1,
  "items": [
    {
      "itemName": "탄력밴드",
      "grade": "1",
      "permitDate": "2006-11-03",
      "meddevItemSeq": "20060105",
      "isForExport": false,
      "mebTypeName": null,
      "meddevItemNo": "부산 제신 06-121 호",
      "cobFlagCode": "MNFCTUR"
    },
    ...
  ],
  "totalElements": 120
}
```

5. 모델 목록 조회

전체 모델 목록을 조회합니다. 모델 정보는 매우 중요합니다. 향후 설명할 **UDI-DI 통합정보에 관련된 API는 모델 정보에 종속적**이기 때문입니다. 통합정보를 조회하고 등록하고 수정하는데 모델의 정보가 필수 매개변수로써 사용됩니다. UDI-DI 통합정보 관련 API를 사용하는데 필요한 값은 응답필드에 굵은글씨(Bold)처리 되어있습니다.

5.1. Request

```
GET /api/v1/int-info/models?offset=1&limit=50 HTTP/1.1
Host: udiportal.mfds.go.kr
Authorization: Bearer {access_token}
Accept: application/json;charset=UTF-8
```

Access_token을 Authorization Header에 담아 GET 으로 요청합니다.

- **요청필드**

키	설명	구분	필수	타입
offset	페이징이 시작될 숫자	Query	O	number
limit	페이징에 담길 개수 (최대 100)	Query	O	number
modelNameLike	검색할 모델명	Query	O	string(4000)

- **예제**

```
GET /api/v1/int-info/models?offset=1&limit=10 HTTP/1.1
Host: udiportal.mfds.go.kr
Authorization: Bearer {access_token}
Accept: application/json;charset=UTF-8
```

- **예제코드 (Java, Spring)**

```
        UriComponentsBuilder uriBuilder = UriComponentsBuilder.fromHttpUrl("https://udiportal.mfds.go.kr/api/v1/int-info/models")
                .queryParam("offset", 1)
                .queryParam("limit", 10);
        HttpHeaders headers = new HttpHeaders();
        String accessToken = obtainAccessToken(clientId, clientSecret);
        headers.add(HttpHeaders.AUTHORIZATION, "Bearer " + accessToken);
        headers.add(HttpHeaders.ACCEPT, MediaType.APPLICATION_JSON_UTF8_VALUE);
        HttpEntity httpEntity = new HttpEntity(headers);
        ResponseEntity<String> responseEntity = restTemplate.exchange(uriBuilder.toUriString(),
                HttpMethod.GET,
                httpEntity,
                String.class);
```

```
JSONObject body = new JSONObject(responseEntity.getBody());
```

5.2. Response

◆ 응답코드

코드	설명
200	정상응답
400	입력값이 잘못되었을 경우
403	해당 정보에 접근 권한이 없을 경우
401	인증정보가 존재하지 않을 경우
500	처리 중 오류가 발생했을 경우

◆ 응답필드

키	설명	타입
page	현재 페이지 번호	number
pageSize	페이지 크기	number
totalElements	총 갯수	number
items[].**modelName**	**모델명**	string(4000)
items[].itemName	품목명	string(500)
items[].permitItemNo	품목허가번호	string(50)
items[].permitDate	품목허가일자	string(10)
items[].grade	등급	string(1)
items[].udiDiCount	등록된 UDI-DI 통합정보 갯수	string
items[].**meddevItemSeq**	**품목일련번호**	number
items[].**isForExport**	**수출용 여부**	boolean
items[].**seq**	**모델 일련번호**	string(50)
items[].cobFlagCode	제조/수입 구분 • 제조 : MNFCTUR • 수입 : INCME	string(10)

◆ 응답예제

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8

{
  "pageSize": 10,
  "page": 1,
  "items": [
    {
      "modelName": "맨담서포트 무릎튼튼 프리미엄+",
      "itemName": "탄력밴드",
      "permitItemNo": "부산 제신 06-120 호",
      "permitDate": "2006-10-27",
      "grade": "1",
      "udidiCount": "2",
      "meddevItemSeq": 20060048,
      "isForExport": false,
      "seq": "7147",
      "cobFlagCode": null
    },
    ...
  ],
  "totalElements": 308
}
```

6. UDI-DI 통합정보 목록 조회

해당 모델의 UDI-DI 코드 및 통합정보 목록을 조회합니다. 본 API의 매개변수로 쓰이는 값들은 [**모델 목록 조회]**에서 확인할 수 있습니다.

6.1. Request

```
GET /api/v1/int-info/udidi-unity?meddevItemSeq={meddevItemSeq}&mebTypeName={mebTypeNam
e}&isForExport={isForExport} HTTP/1.1
Host: udiportal.mfds.go.kr
Authorization: Bearer {access_token}
Accept: application/json;charset=UTF-8
```

Access_token을 Authorization Header에 담아 GET 으로 요청합니다.

◆ **요청필드**

키	설명	구분	필수	타입
meddevItemSeq	품목일련번호	Query	O	number
mebTypeName	모델명 URL encoding해야 하며 아래 문자는 별도로 URL encoding 전에 치환해줘야 합니다. • "+" -> "%2B" • "&" -> "%26"	Query	O	string(4000)
isForExport	수출용 여부 true / false	Query	O	boolean

◆ **예제**

```
GET /api/v1/int-info/udidi-unity?meddevItemSeq=20060048&mebTypeName=%25ED%2583%2584%2
5EB%25A0%25A5%25EB%25B0%25B4%25EB%2593%259C(JC-7500%2520SPEC)&exportYn=false HTT
P/1.1
Host: udiportal.mfds.go.kr
Authorization: Bearer {access_token}
Accept: application/json;charset=UTF-8
```

◆ **예제코드 (Java, Spring)**

```
        UriComponentsBuilder uriBuilder = UriComponentsBuilder.fromHttpUrl("https://udiportal.mfds.go.kr/api/v1/int-info/udidi-unity")
                .queryParam("meddevItemSeq", 20060048)
                .queryParam("mebTypeName", "탄력밴드(JC-7500 SPEC)".replace("+", "%2B").replace("&", "%26"))
                .queryParam("isForExport", false);
        HttpHeaders headers = new HttpHeaders();
        String accessToken = obtainAccessToken(clientId, clientSecret);
        headers.add(HttpHeaders.AUTHORIZATION, "Bearer " + accessToken);
        headers.add(HttpHeaders.ACCEPT, MediaType.APPLICATION_JSON_UTF8_VALUE);
        HttpEntity httpEntity = new HttpEntity(headers);

        ResponseEntity<String> responseEntity = restTemplate.exchange(uriBuilder.build(false).toUriString(),
                HttpMethod.GET,
                httpEntity,
                String.class);

        JSONObject body = new JSONObject(responseEntity.getBody());
```

6.2. Response

◆ **응답코드**

코드	설명
200	정상응답
400	입력값이 잘못되었을 경우
403	해당 정보에 접근 권한이 없을 경우
401	인증정보가 존재하지 않을 경우
500	처리 중 오류가 발생했을 경우

◆ **응답필드**

키	설명	타입
items[].udiDiSeq	UDI-DI 일련번호	number
items[].productItemInfo	제품정보	object
items[].productItemInfo.meddevItemSeq	품목일련번호	string(50)
items[].productItemInfo.itemName	품목명	string(500)
items[].productItemInfo.grade	등급	string(1)
items[].productItemInfo.meddevItemNo	품목허가번호	string(50)

items[].productItemInfo.permitDate	품목허가일자	string(10)
items[].productItemInfo.mebTypeName	형명(모델명)	string(4000)
items[].productItemInfo.isForExport	수출용 여부	boolean
items[].productItemInfo.cobFlagCode	제조/수입 구분 • 제조 : MNFCTUR • 수입 : INCME	string(10)
items[].meddevItemSeq	품목일련번호	number
items[].udiDiCode	UDI-DI 코드	string(60)
items[].codeSystemFlag	구분코드 구분	string(10)
items[].packQuantity	포장 내 수량	number
items[].isUseEnd	사용종료여부	boolean
items[].useEndDate	사용종료일자	boolean
items[].useEndReason	사용종료사유	string(1000)
items[].unityInfoRegisted	통합정보 등록 여부	boolean
items[].unityInfoRegistComptDate	통합정보 등록 완료일	string(10)
items[].isSupplyHist	공급내역여부	boolean
items[].useLotNo	로트번호 사용 여부	boolean
items[].useItemSeq	일련번호 사용 여부	boolean
items[].useManufYm	제조연월 사용 여부	boolean
items[].useTimeLimit	사용기한 사용 여부	boolean
items[].isSterilizationMeddev	멸균의료기기여부	boolean
items[].needSterilization	사용전 멸균 필요 여부	boolean
items[].sterilizationMethods	멸균방법	array
items[].sterilizationMethods[].order	멸균방법 순번	number
items[].sterilizationMethods[].code	멸균방법코드 "1":고압증기멸균 "2":건열멸균 "3":에틸렌옥사이드가스멸균 "4":의료용포름알데이드가스멸균 "5":의료용저온플라즈마멸균 "6":냉액멸균 "7":마이크로파멸균 "8":이산화염소가스멸균 "99":기타	string(10)
items[].sterilizationMethods[].name	멸균방법	string(50)
items[].sterilizationMethods[].etc	멸균방법 기타 멸균방법	string(1000)
items[].isRcperSalaryTarget	요양급여대상여부	boolean

items[].rcperSalarys	요양급여 정보	array
items[].rcperSalarys[].order	요양급여정보 순번	number
items[].rcperSalarys[].code	요양급여정보 재료대 코드	string(30)
items[].rcperSalarys[].name	요양급여정보 재료대 품목명	string(4000)
items[].rcperSalarys[].mlsfcName	요양급여정보 치료재료대 중분류명	string(4000)
items[].rcperSalaryCodeNoinputReas	요양급여 코드 미입력 사유	string(1000)
items[].includeLatex	라텍스 포함 여부	boolean
items[].includePhthalate	프탈레이트 포함 여부	boolean
items[].mriSafeExpsrCode	MRI 안전 노출 여부 코드 "1" : 안전 "2" : 안전하지 않음 "3" : 조건부 안전 "4" : 평가되지 않음 "5" : 해당사항 없음	string(1)
items[].swVer	소프트웨어 버전	string(10)
items[].itemAddDescription	추가 설명	string(1000)
items[].warnTabooDetail	비고(치명적인 경고 또는 금기사항)	string(1000)
items[].storageCond	저장조건	string(1000)
items[].distbTreatCond	유통취급조건	string(1000)
items[].managerTelNo	관리책임자 연락처	string(60)
items[].managerEmail	관리책임자 이메일	string(60)
items[].consumerStateCenterName	소비자센터 명칭	string(100)
items[].consumerStateCnterTelNo	소비자센터 연락처	string(60)
items[].registDate	최초 등록일	string(6)
items[].mebTypeSeq	형명(모델) 일련번호	number
items[].**userSterilizationYn**	사용자멸균 정형용품 여부	boolean
items[].**kitYn**	세트화 구성 여부	boolean

- **응답예제**

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8

{
  "items": [
    {
      "rcperSalarys": [],
      "managerTelNo": "0435112112",
      "consumerStateCnterTelNo": "0435112111",
```

```
    "distbTreatCond": "유통취급조건",
    "consumerStateCenterName": "소비자센터",
    "useItemSeq": true,
    "itemAddDescription": "제품에대한 추가 설명",
    "meddevItemSeq": 20060048,
    "isUseEnd": false,
    "useEndReason": "",
    "includePhthalate": false,
    "warnTabooDetail": "비고",
    "mebTypeSeq": 7148,
    "sterilizationMethods": [],
    "unityInfoRegistComptDate": "2020-06-24",
    "useTimeLimit": true,
    "managerEmail": "mfg1@tnrbiofab.com",
    "swVer": "1.0",
    "codeSystemFlag": "GS1",
    "isSterilizationMeddev": false,
    "storageCond": "저장조건",
    "productItemInfo": {
      "itemName": "탄력밴드",
      "grade": "3",
      "permitDate": "2006-10-27",
      "meddevItemSeq": "20060048",
      "isForExport": false,
      "mebTypeName": "탄력밴드(JC-7500 SPEC)",
      "meddevItemNo": "부산 제신 06-120 호",
      "cobFlagCode": null
    },
    "packQuantity": 20,
    "rcperSalaryCodeNoinputReas": "해당없음",
    "isRcperSalaryTarget": false,
    "useEndDate": null,
    "mriSafeExpsrCode": "1",
    "udiDiSeq": 149389,
    "unityInfoRegisted": true,
    "udiDiCode": "08802817204044",
    "useManufYm": true,
    "isSupplyHist": true,
    "useLotNo": false,
    "needSterilization": false,
```

```
        "includeLatex": false,
        "registDate": "2020-06-24",
        "userSterilizationYn": false,
        "kitYn": false
    },  ...
]
```

7. UDI-DI 코드 및 통합정보 추가

모델별 UDI-DI 코드를 생성합니다. UDI-DI 코드를 등록하면 통합정보 일부 항목(허가정보 연계항목)이 함께 입력됩니다. 본 API의 매개변수로 쓰이는 값들은 [**모델 목록 조회]**에서 확인할 수 있습니다.

7.1. Request

```
POST /api/v1/int-info/udidi-unity HTTP/1.1
Host: udiportal.mfds.go.kr
Authorization: Bearer {access_token}
Content-Type: application/json;charset=UTF-8
```

Access_token을 Authorization Header에 담아 요청 내용을 JSON string으로 만들어 POST 로 요청합니다.

- **요청필드**

키	설명	타입
meddevItemSeq	품목허가 일련번호	number
mebTypeName	모델명	string (1330)
udiDiCode	UDI-DI 코드	string(60)
codeSystem	UDI-DI 코드체계 GS1 / HIBCC / ICCBBA	string(6)
isForExport	수출용 여부 true / false	boolean
userSterilizationYn	사용자멸균 정형용품 여부 true / false	boolean
kitYn	세트화 구성 여부 true / false	boolean

- **예제**

```
POST /api/v1/int-info/udidi-unity HTTP/1.1
Host: udiportal.mfds.go.kr
Authorization: Bearer {access_token}
Content-Type: application/json;charset=UTF-8
{
    "codeSystem": "GS1",
    "udiDiCode": "08811234787870",
    "meddevItemSeq": 20060048,
    "isForExport": false,
    "mebTypeName": "탄력밴드(JC-7500 SPEC)",
    "userSterilizationYn": false,
    "kitYn": false
}
```

◆ **예제코드 (Java, Spring)**

```
UriComponentsBuilder uriBuilder =
        UriComponentsBuilder.fromHttpUrl( "https://udiportal.mfds.go.kr/api/v1/int-info/udidi-unity");
HttpHeaders headers = new HttpHeaders();
String accessToken = obtainAccessToken(clientId, clientSecret);
headers.add(HttpHeaders.AUTHORIZATION, "Bearer " + accessToken);
headers.add(HttpHeaders.CONTENT_TYPE, MediaType.APPLICATION_JSON_UTF8_VALUE);
Map<String, Object> body = new HashMap<>();
body.put("meddevItemSeq", 20060048); // 품목일련번호. number.
body.put("mebTypeName", "탄력밴드(JC-7500 SPEC)"); // 형명(모델명). string.
body.put("udiDiCode", "08811234787870"); // UDI-DI 코드. string.
body.put("codeSystem", "GS1"); // 코드구분. string. "GS1", "HIBCC", "ICCBBA"
body.put("isForExport", false); // 수출용 여부. boolean.
body.put("userSterilizationYn", false); // 사용자멸균 정형용품 여부. boolean.
body.put("kitYn", false); // 세트화 구성 여부. boolean.
HttpEntity<Map<String, Object>> httpEntity = new HttpEntity<>(body, headers);

ResponseEntity<String> responseEntity = restTemplate.exchange(uriBuilder.toUriString(),
          HttpMethod.POST,
          httpEntity,
          String.class);

JSONObject response = new JSONObject(responseEntity.getBody());
```

참고사항

요청필드 중 userSterilizationYn(사용자멸균 정형용품 여부) 또는 kitYn(세트화 구성 여부) 항목이 생략된다면 디폴트값으로 생략된 항목은 'N'으로 저장됩니다.

7.2. Response

추가된 UDI-DI 코드 및 통합정보가 반환됩니다.

- **응답코드**

코드	설명
200	정상응답
400	입력값이 잘못되었을 경우
403	해당 정보에 접근 권한이 없을 경우
401	인증정보가 존재하지 않을 경우
500	처리 중 오류가 발생했을 경우

- **응답필드**

키	설명	타입
udiDiSeq	UDI-DI 일련번호	number
productItemInfo	제품정보	object
productItemInfo.meddevItemSeq	품목일련번호	number
productItemInfo.itemName	품목명	string
productItemInfo.grade	등급	string(10)
productItemInfo.meddevItemNo	품목허가번호	string(20)
productItemInfo.permitDate	품목허가일자	string(8)
productItemInfo.mebTypeName	형명(모델명)	string(1330)
productItemInfo.isForExport	수출용 여부	boolean
productItemInfo.cobFlagCode	제조/수입 구분 • 제조 : MNFCTUR • 수입 : INCME	string
meddevItemSeq	품목일련번호	string
mebTypeSeq	형명(모델)일련번호	number
udiDiCode	UDI-DI 코드	string(60)
codeSystemFlag	구분코드 구분	string(1)
packQuantity	포장 내 수량	number
isUseEnd	사용종료여부	boolean
useEndDate	사용종료일자	string(7)
useEndReason	사용종료사유	string(330)
unityInfoRegisted	통합정보 등록 여부	boolean
unityInfoRegistComptDate	통합정보 등록 완료일	string(10)
isSupplyHist	공급내역여부	boolean
useLotNo	로트번호 사용 여부	boolean

useItemSeq	일련번호 사용 여부	boolean
useManufYm	제조연월 사용 여부	boolean
useTimeLimit	사용기한 사용 여부	boolean
isSterilizationMeddev	멸균의료기기여부	boolean
needSterilization	사용전 멸균 필요 여부	boolean
sterilizationMethods	멸균방법	array
sterilizationMethods[].order	멸균방법 순번	number
sterilizationMethods[].code	멸균방법코드 "1":고압증기멸균 "2":건열멸균 "3":에틸렌옥사이드가스멸균 "4":의료용포름알데이드가스멸균 "5":의료용저온플라즈마멸균 "6":냉액멸균 "7":마이크로파멸균 "8":이산화염소가스멸균 "99":기타	string(5)
sterilizationMethods[].name	멸균방법	string(330)
sterilizationMethods[].etc	멸균방법 기타 멸균방법	string(330)
isRcperSalaryTarget	요양급여대상여부	boolean
rcperSalarys	요양급여 정보	array
rcperSalarys[].order	요양급여정보 순번	number
rcperSalarys[].code	요양급여정보 재료대 코드	string(30)
rcperSalarys[].name	요양급여정보 재료대 품목명	string(1330)
rcperSalarys[].mlsfcName	요양급여정보 치료재료대 중분류명	string(1330)
rcperSalaryCodeNoinputReas	요양급여 코드 미입력 사유	string(330)
includeLatex	라텍스 포함 여부	boolean
includePhthalate	프탈레이트 포함 여부	boolean
mriSafeExpsrCode	MRI 안전 노출 여부 코드 "1" : 안전 "2" : 안전하지 않음 "3" : 조건부 안전 "4" : 평가되지 않음 "5" : 해당사항 없음	string(5)
swVer	소프트웨어 버전	string(100)
itemAddDescription	추가 설명	string(330)
warnTabooDetail	비고(치명적인 경고 또는 금기사항)	string(330)
storageCond	저장조건	string(330)

distbTreatCond	유통취급조건	string(330)
managerTelNo	관리책임자 연락처	string(60)
managerEmail	관리책임자 이메일	string(60)
consumerStateCenterName	소비자센터 명칭	string(30)
consumerStateCnterTelNo	소비자센터 연락처	string(60)
registDate	최초 등록일	string(10)
userSterilizationYn	사용자멸균 정형용품 여부	boolean
kitYn	세트화 구성 여부	boolean

- **응답예제**

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8

{"udiDiSeq":1956,"productItemInfo":null,"meddevItemSeq":"20060048",“mebTypeSeq”:123123,"udiDiCode":"08801123487448","codeSystemFlag":null,"packQuantity":5,"isUseEnd":false,"useEndDate":null,"useEndReason":"","unityInfoRegisted":true,"unityInfoRegistComptDate":"2019-07-24","isSupplyHist":false,"useLotNo":true,"useItemSeq":false,"useManufYm":false,"useTimeLimit":true,"isSterilizationMeddev":false,"needSterilization":true,"sterilizationMethods":[{"order":1,"code":"1","name":"고압증기멸균","etc":""},{"order":2,"code":"99","name":"기타","etc":"기타멸균방법명"}],"isRcperSalaryTarget":false,"rcperSalarys":[{"order":1,"code":"1","name":"2","mlsfcName":"3"}],"rcperSalaryCodeNoinputReas":"요양급여대상이 아님.","includeLatex":true,"includePhthalate":false,"mriSafeExpsrCode":"1","swVer":"1.1","itemAddDescription":"","warnTabooDetail":"","storageCond":"상온보관","distbTreatCond":null,"managerTelNo":null,"managerEmail":"manager@example.com","consumerStateCenterName":"C/S센터","consumerStateCnterTelNo":"0000000000","registDate":"2019-07-23","userSterilizationYn":false,"kitYn":false}
```

- **오류 응답 예제 – 중복된 UDI-DI 코드**

```
HTTP/1.1 400 BAD_REQUEST
Content-Type: application/json;charset=UTF-8

{"message":"Invalid Input","status":400,"errors":[{"field":"udiDiCode","value":"08811234787870","reason":"UDI-DI 코드가 '탄력밴드(JC-7500 SPEC)' (품목허가 '부산 제신 06-120 호') 와 중복됩니다"}]}
```

- **오류 응답 예제 – UDI-DI 코드 정합성 오류**

```
HTTP/1.1 400 BAD_REQUEST
Content-Type: application/json;charset=UTF-8

{"message":"Invalid Input","status":400,"errors":[{"field":"udiDiCode","value":"998811234787870","reason":"코드는 13자리 또는 14자리여야 합니다"}]}
```

8. UDI-DI 코드 삭제

UDI-DI 코드를 삭제합니다. 본 API의 매개변수로 쓰이는 값들은 [**UDI-DI 통합정보 목록조회]**에서 확인할 수 있습니다.

※ 공급내역보고 실적이 있는 UDI-DI 코드는 삭제할 수 없습니다.

8.1. Request

```
POST /api/v1/int-info/udidi-code/{meddevItemSeq}/{mebTypeSeq}/{udiDiSeq} HTTP/1.1
Host: udiportal.mfds.go.kr
Authorization: Bearer {access_token}
Accept: application/json;charset=UTF-8
```

Access_token을 Authorization Header에 담아 POST 로 요청합니다.

◆ **요청필드**

키	설명	구분	필수	타입
meddevItemSeq	품목일련번호	PathVariable	O	number
mebTypeSeq	모델일련번호	PathVariable	O	number
udiDiSeq	UDI-DI일련번호	PathVariable	O	number

◆ **예제**

```
POST /api/v1/int-info/udidi-unity/20060048/1956/7147 HTTP/1.1
Host: udiportal.mfds.go.kr
Authorization: Bearer {access_token}
Content-Type: application/json;charset=UTF-8
```

◆ **예제코드 (Java, Spring)**

```
        Integer meddevItemSeq = 20060048; // 품목일련번호. number. required
        Integer mebTypeSeq = 7147; // 형명(모델) 일련번호. number. required
        Integer udiDiSeq = 1956; // UDI-DI 일련번호. number. required

        UriComponentsBuilder uriBuilder = UriComponentsBuilder.fromHttpUrl("https://udiportal.mfds.go.kr/api/v1/int-info/udidi-code")
                .pathSegment(String.valueOf(meddevItemSeq))
                .pathSegment(String.valueOf(mebTypeSeq))
                .pathSegment(String.valueOf(udiDiSeq));

        HttpHeaders headers = new HttpHeaders();
        String accessToken = TestUtil.obtainAccessToken(clientId, clientSecret);
        headers.add(HttpHeaders.AUTHORIZATION, "Bearer " + accessToken);
```

```
headers.add(HttpHeaders.ACCEPT, MediaType.APPLICATION_JSON_UTF8_VALUE);
HttpEntity httpEntity = new HttpEntity(headers);

// POST /udidi-code/{meddevItemSeq}/{mebTypeSeq}/{udiDiSeq}
ResponseEntity<String> responseEntity = restTemplate.exchange(uriBuilder.toUriString(),
        HttpMethod.POST,
        httpEntity,
        String.class);
```

8.2. Response

수정된 UDI-DI 통합정보가 반환됩니다.

- **응답코드**

코드	설명
200	정상응답
400	입력값이 잘못되었을 경우, 해당 UDI-DI 코드가 존재하지 않을 경우
403	해당 정보에 접근 권한이 없을 경우
401	인증정보가 존재하지 않을 경우
500	처리 중 오류가 발생했을 경우

- **응답필드**

코드(Response Code)만 응답합니다.

- **응답예제**

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
```

9. 통합정보 수정

통합정보 내용을 수정합니다. 본 API를 통해 수정한 통합정보가 통합정보등록요건을 만족하면 통합정보로 등록됩니다. 본 API의 매개변수로 쓰이는 값들은 [**UDI-DI 통합정보 목록 조회]**에서 확인할 수 있습니다.

9.1. Request

```
POST /api/v1/int-info/udidi-unity/{meddevItemSeq}/{mebTypeSeq}/{udiDiSeq} HTTP/1.1
Host: udiportal.mfds.go.kr
Authorization: Bearer {access_token}
Content-Type: application/json;charset=UTF-8
```

Access_token을 Authorization Header에 담아 요청 내용을 JSON string으로 만들어 POST 로 요청합니다.

◆ **요청필드**

키	설명	구분	필수	타입
meddevItemSeq	품목일련번호	PathVariable	O	number
mebTypeSeq	모델일련번호	PathVariable	O	number
udiDiSeq	UDI-DI일련번호	PathVariable	O	number
isUseEnd	사용종료 여부	Body	X	boolean
useEndDate	사용종료일자(YYYY-MM-DD)	Body	X	string(10)
useEndReason	사용종료사유	Body	X	string (330)
needSterilization	사용전 멸균 필요 여부	Body	O	boolean
sterilizationMethods	멸균방법	Body	O	array
sterilizationMethods[].order	멸균방법 순번	Body	X	number
sterilizationMethods[].code	멸균방법 코드 "1" : 고압증기멸균 "2" : 건열멸균 "3" : 에틸렌옥사이드가스멸균 "4" : 의료용포름알데이드가스멸균 "5" : 의료용저온플라즈마멸균 "6" : 냉액멸균 "7" : 마이크로파멸균 "8" : 이산화염소가스멸균 "99": 기타	Body	X	string(5)
sterilizationMethods[].name	멸균방법	Body	X	string (330)
sterilizationMethods[].etc	멸균방법 기타 멸균방법	Body	X	string (330)

isRcperSalaryTarget	요양급여대상 여부	Body	O	boolean
rcperSalaryCodeNoinputReas	요양급여정보 미입력 사유	Body	O	string (330)
rcperSalarys	요양급여정보	Body	O	array
rcperSalarys[].order	요양급여정보 순번	Body	X	number
rcperSalarys[].code	요양급여정보 재료대 코드	Body	X	string(30)
rcperSalarys[].name	요양급여정보 재료대 품목명	Body	X	string (1330)
rcperSalarys[].mlsfcName	요양급여정보 치료재료대 중분류명	Body	X	string (1330)
packQuantity	포장내 수량	Body	O	number
storageCond	저장조건	Body	O	string (330)
distbTreatCond	유통취급조건	Body	O	string (330
useLotNo	로트번호 사용 여부	Body	O	boolean
useItemSeq	일련번호 사용 여부	Body	O	boolean
useManufYm	제조연월 사용 여부	Body	O	boolean
useTimeLimit	사용기한 사용 여부	Body	O	boolean
managerTelNo	관리자 연락처	Body	O	string(60)
managerEmail	관리자 Email	Body	O	string(60)
consumerStateCenterName	소비자센터 명칭	Body	O	string(30)
consumerStateCenterTelNo	소비자센터 연락처	Body	O	string(60)
includeLatex	라텍스 포함 여부	Body	O	boolean
includePhthalate	프탈레이트류 포함 여부(수액세트 품목인 경우만 필수)	Body	O	boolean
mriSafeExpsrCode	MRI 안전 노출 코드 "1" : 안전 "2" : 안전하지 않음 "3" : 조건부 안전 "4" : 평가되지 않음 "5" : 해당사항 없음	Body	O	string(5)
swVer	S/W버전	Body	O	string (100)
warnTabooDetail	비고	Body	O	string (330)
itemAddDescription	제품에 대한 추가 설명	Body	O	string (330)
userSterilizationYn	사용자멸균 정형용품 여부	Body	O	boolean
kitYn	세트화 구성 여부	Body	O	boolean

◆ **예제**

```
POST /api/v1/int-info/udidi-unity/20060048/1956/7147 HTTP/1.1
Host: udiportal.mfds.go.kr
Authorization: Bearer {access_token}
Content-Type: application/json;charset=UTF-8

{"rcperSalaryCodeNoinputReas":"요양급여대상이 아님.","rcperSalarys":[{"name":"2","mlsfcName":"3","code":"1","order":1}],"managerTelno":"00000000000","consumerStateCnterTelNo":"0000000000","isRcperSalaryTarget":false,"distbTreatCond":"상온 유통","consumerStateCenterName":"C/S센터","useItemSeq":false,"itemAddDescription":"","includePhthalate":false,"warnTabooDetail":"","sterilizationMethods":[{"name":"고압증기멸균","code":"1","etc":"","order":1},{"name":"기타","code":"99","etc":"기타멸균방법명","order":2}],"useTimeLimit":true,"mriSafeExpsrCode":"1","managerEmail":"manager@example.com","swVer":"1.1","useManufYm":false,"packTotalQty":5,"storageCond":"상온보관","useLotNo":true,"needSterilization":true,"includeLatex":true,"userSterilizationYn":false,"kitYn":false}
```

◆ **예제코드 (Java, Spring)**

```
        Integer meddevItemSeq = 20060048; // 품목일련번호. number. required
        Integer mebTypeSeq = 7147; // 형명(모델) 일련번호. number. required
        Integer udiDiSeq = 1956; // UDI-DI 일련번호. number. required
        UriComponentsBuilder uriBuilder = UriComponentsBuilder.fromHttpUrl("https://udiportal.mfds.go.kr/api/v1/int-info/udidi-unity")
                .pathSegment(String.valueOf(meddevItemSeq))
                .pathSegment(String.valueOf(mebTypeSeq))
                .pathSegment(String.valueOf(udiDiSeq));
        HttpHeaders headers = new HttpHeaders();
        String accessToken = obtainAccessToken(clientId, clientSecret);
        headers.add(HttpHeaders.AUTHORIZATION, "Bearer " + accessToken);
        headers.add(HttpHeaders.CONTENT_TYPE, MediaType.APPLICATION_JSON_UTF8_VALUE);
        Map<String, Object> body = new HashMap<>();

        //// 멸균방법
        // 사용전 멸균 필요 여부
        body.put("needSterilization", true);
        // -- 사용전 멸균 필요 여부

        // 멸균방법 TODO 사용전멸균필요여부 Y이면 멸균방법 필수
        List<Map<String, Object>> sterilizationMethods = new ArrayList<>();
        Map<String, Object> sterilizationMethod = new HashMap<>(4); // 멸균방법
        /*
        "1" : 고압증기멸균
        "2" : 건열멸균
```

```
        "3" : 에틸렌옥사이드가스멸균
        "4" : 의료용포름알데이드가스멸균
        "5" : 의료용저온플라즈마멸균
        "6" : 냉액멸균
        "7" : 마이크로파멸균
        "8" : 이산화염소가스멸균
        "99": 기타
         */
        sterilizationMethod.put("order", 1); // 순번
        sterilizationMethod.put("code", "1"); // 멸균방법 코드
        sterilizationMethod.put("name", "고압증기멸균"); //멸균방법 TODO 코드만 넣으면 안되는
가?
        sterilizationMethod.put("etc", ""); //기타멸균방법
        sterilizationMethods.add(sterilizationMethod);
        sterilizationMethod = new HashMap<>(4);
        sterilizationMethod.put("order", 2); // 순번
        sterilizationMethod.put("code", "99"); // 멸균방법 코드
        sterilizationMethod.put("name", "기타"); //멸균방법
        sterilizationMethod.put("etc", "기타멸균방법명"); //기타멸균방법
        sterilizationMethods.add(sterilizationMethod);

//          updateParam.put("STERILIZATION_METHOD_INFO", sterilizationMethods);
        body.put("sterilizationMethods", sterilizationMethods);
        // -- 멸균방법
        //// -- 멸균방법

        //// 요양급여
        // 요양급여대상 여부
        body.put("isRcperSalaryTarget", false);
        // -- 요양급여대상 여부
        // 요양급여 미입력 사유 TODO 요양급여대상 N이면 필수입력?? 확인할것 Y인데 코드를
몰라서 못할때 써도 되는건지
        body.put("rcperSalaryCodeNoinputReas", "요양급여대상이 아님.");
        // -- 요양급여 미입력 사유

        // 요양급여 TODO 요양급여대상 Y이면 필수입력
        List<Map<String, Object>> rcperSalarys = new ArrayList<>();
        Map<String, Object> rcperSalary = new HashMap<>(1);
        rcperSalary.put("order", 1); // 요양급여 재료대 코드
        rcperSalary.put("code", "1"); // 요양급여 재료대 코드
```

```
rcperSalary.put("name", "2"); // 요양급여 재료대 품목명
rcperSalary.put("mlsfcName", "3"); // 요양급여 치료 재료대 중분류명
rcperSalarys.add(rcperSalary);

body.put("rcperSalarys", rcperSalarys);
// -- 요양급여
//// -- 요양급여

// 포장내 수량
body.put("packQuantity", 5);
// -- 포장내 수량

// 저장조건
body.put("storageCond", "상온보관");
// -- 저장조건

// 유통·취급조건
body.put("distbTreatCond", "상온 유통");
// --유통·취급조건


//// 관리형태
//// 관리형태 4개중 하나는 필수
// 로트번호 사용 여부
body.put("useLotNo", true);
// --로트번호 사용 여부

// 일련번호 사용 여부
body.put("useItemSeq", false);
// -- 일련번호 사용 여부

// 제조연월 사용 여부
body.put("useManufYm", false);
// --제조연월 사용 여부

// 사용기한 사용 여부
body.put("useTimeLimit", true);
// -- 사용기한 사용 여부
//// --관리형태
```

```
//// 업자정보
// 관리자연락처
body.put("managerTelno", "00000000000");
// -- 관리자연락처
// 관리자이메일
body.put("managerEmail", "manager@example.com");
// -- 관리자이메일
// 소비자센터 명칭
body.put("consumerStateCenterName", "C/S센터");
// -- 소비자센터 명칭
// 소비자센터 연락처
body.put("consumerStateCenterTelNo", "0000000000");
// -- 소비자센터 연락처
//// -- 업자정보

//// 제품정보
// 라텍스 포함 여부
body.put("includeLatex", true);
// --라텍스 포함 여부
// 프탈레이트류 포함 여부
body.put("includePhthalate", false);
// -- 프탈레이트류 포함 여부
// MRI 안전 노출 코드
// "1" : 안전
// "2" : 안전하지 않음
// "3" : 조건부 안전
// "4" : 평가되지 않음
// "5" : 해당사항 없음
body.put("mriSafeExpsrCode", "1"); // TODO validation
// -- MRI 안전 노출 코드
// S/W 버전
body.put("swVer", "1.1");
// -- S/W 버전
// 비고
body.put("warnTabooDetail", "");
// -- 비고
// 제품에 대한 추가 설명
body.put("itemAddDescription", "");
// -- 제품에 대한 추가 설명
//// -- 제품정보
```

```
// 사용자멸균 정형용품 여부
body.put("userSterilizationYn", false);

// 세트화 구성 여부
body.put("kitYn", false);

HttpEntity<Map<String, Object>> httpEntity = new HttpEntity<>(body, headers);

ResponseEntity<String> responseEntity = restTemplate.exchange(uriBuilder.toUriString(),
        HttpMethod.POST,
        httpEntity,
        String.class);

JSONObject response = new JSONObject(responseEntity.getBody());
```

9.2. Response

수정된 UDI-DI 통합정보가 반환됩니다.

- **응답코드**

코드	설명
200	정상응답
400	입력값이 잘못되었을 경우
403	해당 정보에 접근 권한이 없을 경우
401	인증정보가 존재하지 않을 경우
500	처리 중 오류가 발생했을 경우

- **응답필드**

키	설명	타입
udiDiSeq	UDI-DI 일련번호	number
productItemInfo	제품정보	object
productItemInfo.meddevItemSeq	품목일련번호	number
productItemInfo.itemName	품목명	string(330)
productItemInfo.grade	등급	string(10)
productItemInfo.meddevItemNo	품목허가번호	string(20)
productItemInfo.permitDate	품목허가일자	string(8)
productItemInfo.mebTypeName	형명(모델명)	string(1330)
productItemInfo.isForExport	수출용 여부	boolean

productItemInfo.cobFlagCode	제조/수입 구분 • 제조 : MNFCTUR • 수입 : INCME	string
meddevItemSeq	품목일련번호	string
udiDiCode	UDI-DI 코드	string(60)
codeSystemFlag	구분코드 구분	string(1)
packQuantity	포장 내 수량	number
isUseEnd	사용종료여부	boolean
useEndDate	사용종료일자	string(7)
useEndReason	사용종료사유	string(330)
unityInfoRegisted	통합정보 등록 여부	boolean
unityInfoRegistComptDate	통합정보 등록 완료일	string(11)
isSupplyHist	공급내역여부	boolean
useLotNo	로트번호 사용 여부	boolean
useItemSeq	일련번호 사용 여부	boolean
useManufYm	제조연월 사용 여부	boolean
useTimeLimit	사용기한 사용 여부	boolean
isSterilizationMeddev	멸균의료기기여부	boolean
needSterilization	사용전 멸균 필요 여부	boolean
sterilizationMethods	멸균방법	array
sterilizationMethods[].order	멸균방법 순번	number
sterilizationMethods[].code	멸균방법코드 "1":고압증기멸균 "2":건열멸균 "3":에틸렌옥사이드가스멸균 "4":의료용포름알데이드가스멸균 "5":의료용저온플라즈마멸균 "6":냉액멸균 "7":마이크로파멸균 "8":이산화염소가스멸균 "99":기타	string(5)
sterilizationMethods[].name	멸균방법	string(330)
sterilizationMethods[].etc	멸균방법 기타 멸균방법	string(330)
isRcperSalaryTarget	요양급여대상여부	boolean
rcperSalarys	요양급여 정보	array
rcperSalarys[].order	요양급여정보 순번	number
rcperSalarys[].code	요양급여정보 재료대 코드	string(30

rcperSalarys[].name	요양급여정보 재료대 품목명	string(1330)
rcperSalarys[].mlsfcName	요양급여정보치료재료대중분류명	string(1330)
rcperSalaryCodeNoinputReas	요양급여 코드 미입력 사유	string(330)
includeLatex	라텍스 포함 여부	boolean
includePhthalate	프탈레이트 포함 여부	boolean
mriSafeExpsrCode	MRI 안전 노출 여부 코드 "1" : 안전 "2" : 안전하지 않음 "3" : 조건부 안전 "4" : 평가되지 않음 "5" : 해당사항 없음	string(50)
swVer	소프트웨어 버전	string(100)
itemAddDescription	추가 설명	string(330)
warnTabooDetail	비고(치명적인 경고 또는 금기사항)	string(330)
storageCond	저장조건	string(330)
distbTreatCond	유통취급조건	string(330)
managerTelNo	관리책임자 연락처	string(60)
managerEmail	관리책임자 이메일	string(60)
consumerStateCenterName	소비자센터 명칭	string(30)
consumerStateCnterTelNo	소비자센터 연락처	string(60)
registDate	최초 등록일	string(10)
userSterilizationYn	사용자멸균 정형용품 여부	boolean
kitYn	세트화 구성 여부	boolean

- **응답예제**

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
{"udiDiSeq":1956,"productItemInfo":null,"meddevItemSeq":"20060048","udiDiCode":"08801123487448","codeSystemFlag":null,"packQuantity":5,"isUseEnd":false,"useEndDate":null,"useEndReason":"","unityInfoRegisted":true,"unityInfoRegistComptDate":"2019-07-24","isSupplyHist":false,"useLotNo":true,"useItemSeq":false,"useManufYm":false,"useTimeLimit":true,"isSterilizationMeddev":false,"needSterilization":true,"sterilizationMethods":[{"order":1,"code":"1","name":"고압증기멸균","etc":""},{"order":2,"code":"99","name":"기타","etc":"기타멸균방법명"}],"isRcperSalaryTarget":false,"rcperSalarys":[{"order":1,"code":"1","name":"2","mlsfcName":"3"}],"rcperSalaryCodeNoinputReas":"요양급여대상이 아님.","includeLatex":true,"includePhthalate":false,"mriSafeExpsrCode":"1","swVer":"1.1","itemAddDescription":"","warnTabooDetail":"","storageCond":"상온보관","distbTreatCond":null,"managerTelNo":null,"managerEmail":"manager@example.com","consumerStateCenterName":"C/S센터","consumerStateCnterTelNo":"0000000000","registDate":"2019-07-23","userSterilizationYn":false,"kitYn":false}
```

10. 물류바코드 목록 조회

UDI-DI코드 정보에서 물류바코드 정보를 조회합니다. 본 API의 매개변수로 쓰이는 값들은 [**UDI-DI 통합정보 목록 조회]**에서 확인할 수 있습니다.

10.1. Request

```
GET /api/v1/int-info/udidi-unity/{meddevItemSeq}/{mebTypeSeq}/{udiDiSeq}/barcodes HTTP/1.1
Host: udiportal.mfds.go.kr
Authorization: Bearer {access_token}
Accept: application/json;charset=UTF-8
```

Access_token을 Authorization Header에 담아 GET 으로 요청합니다.

◆ **요청필드**

키	설명	구분	필수	타입
meddevItemSeq	품목일련번호	PathVariable	O	number
mebTypeSeq	모델일련번호	PathVariable	O	number
udiDiSeq	UDI-DI일련번호	PathVariable	O	number

◆ **예제**

```
GET /api/v1/int-info/udidi-unity/{meddevItemSeq}/{mebTypeSeq}/{udiDiSeq}/barcodes HTTP/1.1
Host: udiportal.mfds.go.kr
Authorization: Bearer {access_token}
Accept: application/json;charset=UTF-8
```

◆ **예제코드 (Java, Spring)**

```
        Integer meddevItemSeq = 20060048; // 품목일련번호. number. required
        Integer mebTypeSeq = 7147; // 형명(모델) 일련번호. number. required
        Integer udiDiSeq = 1956; // UDI-DI 일련번호. number. required
        UriComponentsBuilder uriBuilder = UriComponentsBuilder.fromHttpUrl("https://udiportal.mfds.go.kr/api/v1/int-info/udidi-unity")
                .pathSegment(String.valueOf(meddevItemSeq))
                .pathSegment(String.valueOf(mebTypeSeq))
                .pathSegment(String.valueOf(udiDiSeq))
                .pathSegment("barcodes");

        HttpHeaders headers = new HttpHeaders();
        String accessToken = obtainAccessToken(clientId, clientSecret);
        headers.add(HttpHeaders.AUTHORIZATION, "Bearer " + accessToken);
```

```
        headers.add(HttpHeaders.ACCEPT, MediaType.APPLICATION_JSON_UTF8_VALUE);
        HttpEntity httpEntity = new HttpEntity(headers);

        // GET /udidi-unity/{meddevItemSeq}/{mebTypeSeq}/{udiDiSeq}/barcodes
        ResponseEntity<String> responseEntity = restTemplate.exchange(uriBuilder.toUriString(),
                HttpMethod.GET,
                httpEntity,
                String.class);

        JSONObject body = new JSONObject(responseEntity.getBody());
```

10.2. Response

- **응답코드**

코드	설명
200	정상응답
400	입력값이 잘못되었을 경우
403	해당 정보에 접근 권한이 없을 경우
401	인증정보가 존재하지 않을 경우
500	처리 중 오류가 발생했을 경우

- **응답필드**

키	설명	타입
items[].barcode	물류바코드	string(60)
items[].packQuantity	포장 수량	number
items[].packDegree	포장 차수	number
items[].useEndDate	사용종료 일자	string(10)
items[].useEndReason	사용종료 사유	string(1000)
items[].meddevItemSeq	품목일련번호	number
items[].mebTypeSeq	형명(모델)일련번호	number
items[].barcodeSeq	물류바코드 일련번호	number
items[].udiDiSeq	UDIDI 일련번호	number
items[].codeSystem	구분 코드	string(10)
items[].isUseEnd	사용종료 여부	boolean
items[].isSupplyHist	공급내역 여부	boolean

◆ 응답예제

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8

{
  "items": [
    {
      "packQuantity": 0,
      "codeSystem": "GS1",
      "udiDiSeq": 1956,
      "meddevItemSeq": 20060048,
      "isSupplyHist": false,
      "isUseEnd": false,
      "useEndDate": "",
      "useEndReason": "",
      "barcodeSeq": 1,
      "mebTypeSeq": 7147,
      "barcode": "08801123487448",
      "packDegree": 1
    }
  ]
}
```

11. 물류바코드 추가

UDI-DI코드 정보에 물류바코드 정보를 추가합니다. 본 API의 매개변수로 쓰이는 값들은 [**UDI-DI 통합정보 목록 조회]**에서 확인할 수 있습니다.

11.1. Request

```
POST /api/v1/int-info/udidi-unity/{meddevItemSeq}/{mebTypeSeq}/{udiDiSeq}/barcodes HTTP/1.1
Host: udiportal.mfds.go.kr
Authorization: Bearer {access_token}
Content-Type: application/json;charset=UTF-8
```

Access_token을 Authorization Header에 담아 요청 내용을 JSON string으로 만들어 POST 로 요청합니다.

◆ **요청필드**

키	설명	구분	필수	타입
meddevItemSeq	품목일련번호	PathVariable	O	number
mebTypeSeq	모델일련번호	PathVariable	O	number
udiDiSeq	UDI-DI일련번호	PathVariable	O	number
packQuantity	포장수량	Body	O	number
codeSystem	코드구분 "GS1" / "HIBCC" / "ICCBBA"	Body	O	string(6)
barcode	바코드	Body	O	string(60)
packDegree	포장차수	Body	O	number

◆ **예제**

```
POST /api/v1/int-info/udidi-unity/{meddevItemSeq}/{mebTypeSeq}/{udiDiSeq}/barcodes HTTP/1.1
Host: udiportal.mfds.go.kr
Authorization: Bearer {access_token}
Content-Type: application/json;charset=UTF-8

{"packQuantity":"100","codeSystem":"GS1","barcode":"08801123487449","packDegree":"2"}
```

◆ **예제코드 (Java, Spring)**

```
Integer meddevItemSeq = 20060048; // 품목일련번호. number. required
Integer mebTypeSeq = 7147; // 형명(모델) 일련번호. number. required
Integer udiDiSeq = 1956; // UDI-DI 일련번호. number. required
Map<String, Object> body = new HashMap<>();
body.put("barcode", "08801123487449"); // 바코드. string
```

```
        body.put("packQuantity", "100"); // 포장수량. number
        body.put("packDegree", "2"); // 포장차수. number
        body.put("codeSystem", "GS1"); // 코드구분. string. "GS1", "HIBCC", "ICCBBA"
        UriComponentsBuilder uriBuilder = UriComponentsBuilder.fromHttpUrl("https://udiportal.mfds.go.kr/api/v1/int-info/udidi-unity")
                .pathSegment(String.valueOf(meddevItemSeq))
                .pathSegment(String.valueOf(mebTypeSeq))
                .pathSegment(String.valueOf(udiDiSeq))
                .pathSegment("barcodes");

        HttpHeaders headers = new HttpHeaders();
        String accessToken = obtainAccessToken(clientId, clientSecret);
        headers.add(HttpHeaders.AUTHORIZATION, "Bearer " + accessToken);
        headers.add(HttpHeaders.CONTENT_TYPE, MediaType.APPLICATION_JSON_UTF8_VALUE);
        HttpEntity<Map<String, Object>> httpEntity = new HttpEntity<>(body, headers);

        // POST /udidi-unity/{meddevItemSeq}/{mebTypeSeq}/{udiDiSeq}/barcodes
        ResponseEntity<String> responseEntity = restTemplate.exchange(uriBuilder.toUriString(),
                HttpMethod.POST,
                httpEntity,
                String.class);

        JSONObject jsonObject = new JSONObject(responseEntity.getBody());
```

11.2. Response

저장된 바코드 정보가 반환됩니다.

- **응답코드**

코드	설명
200	정상응답
400	입력값이 잘못되었을 경우
403	해당 정보에 접근 권한이 없을 경우
401	인증정보가 존재하지 않을 경우
500	처리 중 오류가 발생했을 경우

- **응답필드**

키	설명	타입
barcode	물류바코드	string(60)
packQuantity	포장 수량	number
packDegree	포장 차수	number

useEndDate	사용종료 일자	string
useEndReason	사용종료 사유	string(330)
meddevItemSeq	품목일련번호	number
mebTypeSeq	형명(모델)일련번호	number
barcodeSeq	물류바코드 일련번호	number
udiDiSeq	UDIDI 일련번호	number
codeSystem	구분 코드	string(6)
isUseEnd	사용종료 여부	boolean
isSupplyHist	공급내역 여부	boolean

- **응답예제**

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8

{
  "packQuantity": 100,
  "codeSystem": "GS1",
  "udiDiSeq": 1956,
  "meddevItemSeq": 20060048,
  "isSupplyHist": false,
  "isUseEnd": false,
  "useEndDate": "",
  "useEndReason": "",
  "barcodeSeq": 5,
  "mebTypeSeq": 7147,
  "barcode": "08801123487449",
  "packDegree": 2
}
```

12. 물류바코드 수정

UDI-DI코드 정보에서 물류바코드의 정보를 수정합니다. 본 API의 매개변수로 쓰이는 값들은 [**물류바코드 목록 조회]**에서 확인할 수 있습니다.

12.1. Request

```
POST /api/v1/int-info/udidi-unity/{meddevItemSeq}/{mebTypeSeq}/{udiDiSeq}/barcodes/{barcodeSeq} HTTP/1.1
Host: udiportal.mfds.go.kr
Authorization: Bearer {access_token}
Content-Type: application/json;charset=UTF-8
```

Access_token을 Authorization Header에 담아 요청 내용을 JSON string으로 만들어 POST 로 요청합니다.

- **요청필드**

키	설명	구분	필수	타입
meddevItemSeq	품목일련번호	PathVariable	O	number
mebTypeSeq	모델일련번호	PathVariable	O	number
udiDiSeq	UDI-DI일련번호	PathVariable	O	number
barcodeSeq	바코드 일련번호	PathVariable	O	number
packQuantity	포장수량	Body	O	number
codeSystem	코드구분 "GS1" / "HIBCC" / "ICCBBA"	Body	O	string(6)
barcode	바코드	Body	O	string(60)
packDegree	포장차수	Body	O	number

- **예제**

```
POST /api/v1/int-info/udidi-unity/{meddevItemSeq}/{mebTypeSeq}/{udiDiSeq}/barcodes/{barcodeSeq} HTTP/1.1
Host: udiportal.mfds.go.kr
Authorization: Bearer {access_token}
Content-Type: application/json;charset=UTF-8

{"packQuantity":"100","codeSystem":"GS1","barcode":"08801123487449","packDegree":"2"}
```

◆ **예제코드 (Java, Spring)**

```
        Integer meddevItemSeq = 20060048; // 품목일련번호. number. required
        Integer mebTypeSeq = 7147; // 형명(모델) 일련번호. number. required
        Integer udiDiSeq = 1956; // UDI-DI 일련번호. number. required
        Integer barcodeSeq = 1;
        UriComponentsBuilder uriBuilder = UriComponentsBuilder.fromHttpUrl("https://udiportal.mfds.go.kr/api/v1/int-info/udidi-unity")
                .pathSegment(String.valueOf(meddevItemSeq))
                .pathSegment(String.valueOf(mebTypeSeq))
                .pathSegment(String.valueOf(udiDiSeq))
                .pathSegment("barcodes")
                .pathSegment(String.valueOf(barcodeSeq));
        Map<String, Object> body = new HashMap<>();
        // 포장차수
        body.put("packDegree", 1);
        // 포장수량
        body.put("packQuantity", 5);
        // 종료여부
        body.put("isUseEnd", false);
        // 종료일자
        body.put("useEndDate", "");
        // 종료사유
        body.put("useEndReason", "");

        HttpHeaders headers = new HttpHeaders();
        String accessToken = TestUtil.obtainAccessToken(clientId, clientSecret);
        headers.add(HttpHeaders.AUTHORIZATION, "Bearer " + accessToken);
        headers.add(HttpHeaders.CONTENT_TYPE, MediaType.APPLICATION_JSON_UTF8_VALUE);
        HttpEntity<Map<String, Object>> httpEntity = new HttpEntity<>(body, headers);

        // POST /udidi-unity/{meddevItemSeq}/{mebTypeSeq}/{udiDiSeq}/barcodes/{barcodeSeq}
        ResponseEntity responseEntity = restTemplate.exchange(uriBuilder.toUriString(),
                HttpMethod.POST,
                httpEntity,
                String.class);
```

12.2. Response

저장된 바코드 정보가 반환됩니다.

- **응답코드**

코드	설명
200	정상응답
400	입력값이 잘못되었을 경우
403	해당 정보에 접근 권한이 없을 경우
401	인증정보가 존재하지 않을 경우
500	처리 중 오류가 발생했을 경우

- **응답필드**

코드(Response Code)만 응답합니다.

13. 물류바코드 삭제

UDI-DI코드 정보에서 물류바코드 1건을 삭제합니다. 본 API의 매개변수로 쓰이는 값들은 [**물류바코드 목록 조회]**에서 확인할 수 있습니다.

13.1. Request

```
POST /api/v1/int-info/udidi-unity/{meddevItemSeq}/{mebTypeSeq}/{udiDiSeq}/barcodes/{barcodeSeq}/delete HTTP/1.1
Host: udiportal.mfds.go.kr
Authorization: Bearer {access_token}
Content-Type: application/json;charset=UTF-8
```

Access_token을 Authorization Header에 담아 POST 으로 요청합니다.

◆ **요청필드**

키	설명	구분	필수	타입
meddevItemSeq	품목일련번호	PathVariable	O	number
mebTypeSeq	모델일련번호	PathVariable	O	number
udiDiSeq	UDI-DI일련번호	PathVariable	O	number
barcodeSeq	바코드 일련번호	PathVariable	O	number

◆ **예제**

```
POST /api/v1/int-info/udidi-unity/{meddevItemSeq}/{mebTypeSeq}/{udiDiSeq}/barcodes/{barcodeSeq}/delete HTTP/1.1
Host: udiportal.mfds.go.kr
Authorization: Bearer {access_token}
Content-Type: application/json;charset=UTF-8
```

◆ **예제코드 (Java, Spring)**

```
        Integer meddevItemSeq = 20060048; // 품목일련번호. number. required
        Integer mebTypeSeq = 7147; // 형명(모델) 일련번호. number. required
        Integer udiDiSeq = 1956; // UDI-DI 일련번호. number. required
        Integer barcodeSeq = 4;
        UriComponentsBuilder uriBuilder = UriComponentsBuilder.fromHttpUrl("https://udiportal.mfds.go.kr/api/v1/int-info/udidi-unity")
                .pathSegment(String.valueOf(meddevItemSeq))
                .pathSegment(String.valueOf(mebTypeSeq))
                .pathSegment(String.valueOf(udiDiSeq))
                .pathSegment("barcodes")
```

```
                .pathSegment(String.valueOf(barcodeSeq))
                .pathSegment("delete");

        HttpHeaders headers = new HttpHeaders();
        String accessToken = obtainAccessToken(clientId, clientSecret);
        headers.add(HttpHeaders.AUTHORIZATION, "Bearer " + accessToken);
        headers.add(HttpHeaders.CONTENT_TYPE, MediaType.APPLICATION_JSON_UTF8_VALUE);
        HttpEntity httpEntity = new HttpEntity(headers);

        // POST /udidi-unity/{meddevItemSeq}/{mebTypeSeq}/{udiDiSeq}/barcodes/{barcodeSeq}/del
ete
        ResponseEntity responseEntity = restTemplate.exchange(uriBuilder.toUriString(),
                HttpMethod.POST,
                httpEntity,
                String.class);
```

13.2. Response

- **응답코드**

코드	설명
200	정상응답
400	입력값이 잘못되었을 경우
403	해당 정보에 접근 권한이 없을 경우
401	인증정보가 존재하지 않을 경우
500	처리 중 오류가 발생했을 경우

- **응답필드**

코드(Response Code)만 응답합니다.

14. 요양급여 정보 조회

요양급여코드로 해당 코드의 정보를 조회합니다. 요양급여 정보는 통합정보 수정시 사용됩니다.

14.1. Request

```
GET /api/v1/int-info/rcper-salary/{rcperSalaryCode} HTTP/1.1
Host: udiportal.mfds.go.kr
Authorization: Bearer {access_token}
Accept: application/json;charset=UTF-8
```

Access_token을 Authorization Header에 담아 GET 으로 요청합니다.

◆ **요청필드**

키	설명	구분	필수	타입
rcperSalaryCode	요양급여코드	PathVariable	O	string

◆ **예제**

```
GET /api/v1/int-info/rcper-salary/{rcperSalaryCode} HTTP/1.1
Host: udiportal.mfds.go.kr
Authorization: Bearer {access_token}
Accept: application/json;charset=UTF-8
```

◆ **예제코드 (Java, Spring)**

```
        String rcperSalaryCode = "BC1216PS"; // 요양급여코드. string. required
        UriComponentsBuilder uriBuilder = UriComponentsBuilder.fromHttpUrl("https://udiportal.mfds.go.kr/api/v1/int-info/rcper-salary")
                .pathSegment(rcperSalaryCode);

        HttpHeaders headers = new HttpHeaders();
        String accessToken = obtainAccessToken(clientId, clientSecret);
        headers.add(HttpHeaders.AUTHORIZATION, "Bearer " + accessToken);
        headers.add(HttpHeaders.ACCEPT, MediaType.APPLICATION_JSON_UTF8_VALUE);
        HttpEntity httpEntity = new HttpEntity(headers);

        // GET /rcper-salary/{rcperSalaryCode}
        ResponseEntity<String> responseEntity = restTemplate.exchange(uriBuilder.toUriString(),
                HttpMethod.GET,
                httpEntity,
```

```
                String.class);
        JSONObject body = new JSONObject(responseEntity.getBody());
```

14.2. Response

◆ 응답코드

코드	설명
200	정상응답
204	해당 코드의 정보가 존재하지 않을 경우
400	입력값이 잘못되었을 경우
403	해당 정보에 접근 권한이 없을 경우
401	인증정보가 존재하지 않을 경우
500	처리 중 오류가 발생했을 경우

◆ 응답필드

키	설명	타입
code	요양급여 코드	string(30)
name	요양급여 재료대 품목명	string(4000)
mlsfcName	요양급여 치료 재료대 중분류명	string(4000)
order	순서	number

◆ 응답예제

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8

{
  "code": "BC1216PS",
  "name": "TLSO (흉,요추부)",
  "mlsfcName": "압박고정용 SPLINT",
  "order": 0
}
```

15. 거래처 목록 조회

본 API는 거래처 목록을 조회합니다. 거래처관리 관련 API를 사용하는데 필요한 값은 응답필드에 **굵은글씨(Bold)**처리 되어있습니다.

15.1. Request

```
GET /api/v1/company-info/bcnc?offset=1&limit=10 HTTP/1.1
Host: udiportal.mfds.go.kr
Authorization: Bearer {access_token}
Accept: application/json;charset=UTF-8
```

Access_token을 Authorization Header에 담아 GET 으로 요청합니다.

◆ **요청필드**

키	설명	구분	필수	타입
offset	페이지 번호	Query	O	number
limit	페이징에 담길 개수 (최대 100)	Query	O	number
companyName	검색할 거래처 명(LIKE검색)	Query	X	string(100)
taxNo	검색할 사업자등록번호	Query	X	string(10)
cobFlagCode	검색할 업종코드 1:제조, 2:수입, 9:판매(임대), 10:요양기관, 99:기타	Query	X	number
bossName	검색할 대표자 명(LIKE검색)	Query	X	string(100)
area	검색할 지역코드 1:서울, 2:경기, 3:강원, 4:충북or충청북도, 5:충남or충청남도, 6:인천, 7:세종, 8:대전, 9:광주, 10:전남or전라남도, 11:전북or전라북도, 12:경북or경상북도, 13:경남or경상남도, 14:대구 15:울산, 16:부산, 17:제주	Query	X	number

◆ **예제**

```
GET /api/v1/company-info/bcnc?offset=1&limit=10 HTTP/1.1
Host: udiportal.mfds.go.kr
Authorization: Bearer {access_token}
Accept: application/json;charset=UTF-8
```

◆ **예제코드 (Java, Spring)**

```
UriComponentsBuilder uriBuilder = UriComponentsBuilder.fromHttpUrl("https://udiportal.mfds.go.kr/api/v1/company-info/bcnc")
                    .queryParam("offset", 1)
                    .queryParam("limit", 10)
                    .queryParam("companyName", "(주)서울")
                    ;
HttpHeaders headers = new HttpHeaders();
String accessToken = TestUtil.obtainAccessToken(clientId, clientSecret);
headers.add(HttpHeaders.AUTHORIZATION, "Bearer " + accessToken);
headers.add(HttpHeaders.ACCEPT, MediaType.APPLICATION_JSON_UTF8_VALUE);
HttpEntity httpEntity = new HttpEntity(headers);

ResponseEntity<String> responseEntity = restTemplate.exchange(uriBuilder.toUriString(),
                    HttpMethod.GET,
                    httpEntity,
                    String.class);

assertTrue("정상응답 검증", responseEntity.getStatusCode().is2xxSuccessful());
JSONObject body = new JSONObject(responseEntity.getBody());
```

15.2. Response

◆ **응답코드**

코드	설명
200	정상응답
400	입력값이 잘못되었을 경우
403	해당 정보에 접근 권한이 없을 경우
401	인증정보가 존재하지 않을 경우
500	처리 중 오류가 발생했을 경우

◆ 응답필드

키	설명	타입
page	현재 페이지 번호	number
pageSize	페이지 크기	number
items[].area	지역	string(10)
items[].no	순번	string(1000)
items[].bcncUnityCompanySeqs	거래처 통합업체 일련번호	string(50)
items[].bossName	대표자 명	string(100)
items[].bcncCode	거래처 코드	string(30)
items[].companyName	거래처 명	string(100)
items[].hptlSymbl	요양기관코드	string(50)
items[].entpAddr	거래처 전체 주소(거래처 주소 1+2)	string(1100)
items[].entpAddr1	거래처 주소1	string(600)
items[].entpAddr2	거래처 주소2	string(500)
items[].bcncCobDetailName	업종 상세명	string(100)
items[].bcncCobFlagCodeNm	거래처 업종 코드	string(1000)
items[].taxNo	거래처 사업자 등록번호	string(10)
totalElements	총 갯수	number

◆ 응답예제

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8

{
  "pageSize": 10,
  "page": 1,
  "items": [
    {
      "area": "인천",
      "no": "1",
      "bcncUnityCompanySeqs": "11624184",
      "bossName": "이기영",
```

```
        "bcncCode": "100013320000060",
        "companyName": "(주)서울광학",
        "hptlSymbl": null,
        "entpAddr": "인천광역시 남동구 남동대로 301  남동공단 22부럭 4롯트 (논현동)",
        "entpAddr2": " 남동공단 22부럭 4롯트 (논현동)",
        "entpAddr1": "인천광역시 남동구 남동대로 301",
        "bcncCobDetailName": "의료기기",
        "bcncCobFlagCodeNm": "제조업",
        "taxNo": "139-81-07256"
    },
    {
        "area": "서울",
        "no": "2",
        "bcncUnityCompanySeqs": "11627270",
        "bossName": "김기준",
        "bcncCode": "100013320000044",
        "companyName": "케이엠에이치(주)서울지점",
        "hptlSymbl": null,
        "entpAddr": "서울특별시 금천구 디지털로9길 65  ",
        "entpAddr2": null,
        "entpAddr1": "서울특별시 금천구 디지털로9길 65 ",
        "bcncCobDetailName": "의료기기",
        "bcncCobFlagCodeNm": "제조업",
        "taxNo": "119-85-19038"
    }
  ],
  "totalElements": 2
}
```

16. 통합업체 목록 조회

거래처 등록 시 통합업체목록을 조회하여 조회한 업체를 거래처로 등록할 수 있습니다.

16.1. Request

```
GET /api/v1/company-info/entrps?offset=1&limit=10 HTTP/1.1
Host: udiportal.mfds.go.kr
Authorization: Bearer {access_token}
Accept: application/json;charset=UTF-8
```

Access_token을 Authorization Header에 담아 GET 으로 요청합니다.

◆ **요청필드**

키	설명	구분	필수	타입
offset	페이지 번호	Query	O	number
limit	페이징에 담길 개수 (최대 100)	Query	O	number
entpName	업체명	Query	O	string(100)
cobFlagCode	업종코드 1:제조, 2:수입, 3:수리, 9:판매(임대), 10:요양기관	Query	O	number

◆ **예제**

```
GET /api/v1/company-info/entrps?offset=1&limit=10&entpName=제주&cobFlagCode=1 HTTP/1.1
Host: udiportal.mfds.go.kr
Authorization: Bearer {access_token}
Accept: application/json;charset=UTF-8
```

◆ **예제코드 (Java, Spring)**

```
UriComponentsBuilder uriBuilder = UriComponentsBuilder.fromHttpUrl("https://udiportal.mfds.go.kr/api/v1/company-info/entrps")
                    .queryParam("offset", 1)
                    .queryParam("limit", 10)
                    .queryParam("entpName", "제주마그마")
                    .queryParam("cobFlagCode", "1")
```

```
                    ;
HttpHeaders headers = new HttpHeaders();
String accessToken = TestUtil.obtainAccessToken(clientId, clientSecret);
headers.add(HttpHeaders.AUTHORIZATION, "Bearer " + accessToken);
headers.add(HttpHeaders.ACCEPT, MediaType.APPLICATION_JSON_UTF8_VALUE);
HttpEntity httpEntity = new HttpEntity(headers);

ResponseEntity<String> responseEntity = restTemplate.exchange(uriBuilder.toUriString(),
        HttpMethod.GET,
        httpEntity,
        String.class);

assertTrue("정상응답 검증", responseEntity.getStatusCode().is2xxSuccessful());
JSONObject body = new JSONObject(responseEntity.getBody());
```

16.2. Response

◆ 응답코드

코드	설명
200	정상응답
400	입력값이 잘못되었을 경우
403	해당 정보에 접근 권한이 없을 경우
401	인증정보가 존재하지 않을 경우
500	처리 중 오류가 발생했을 경우

◆ 응답필드

키	설명	타입
page	현재 페이지 번호	number
pageSize	페이지 크기	number
totalElements	총 개수	number
items[].no	순번	string(1000)
items[].closeDate	폐업일	string(10)
items[].rprsntName	대표자명	string(500)
items[].entpName	업체명	string(100)
items[].cobFlagCodeName	업종명	string(1000)
items[].cobDetailName	업종 상세명	string(100)

키	설명	타입
items[].entpAddrAll	업체 전체 주소(업체주소1+업체주소2)	string(1100)
items[].entpAddr1	업체 주소1	string(600)
items[].entpAddr2	업체 주소2	string(500)
items[].hptlSymbl	요양기관 기호	string(50)
items[].unityCompanySeq	업체일련번호	string(100)
items[].taxNo	사업자 번호	string(10)

◆ 응답예제

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8

{
  "pageSize": 10,
  "page": 1,
  "items": [
    {
      "closeDate": null,
      "no": "1",
      "unityCompanySeq": "11622338",
      "rprsntName": "지인수",
      "entpName": "(주)제주마그마에너지",
      "entpAddrAll": "제주특별자치도 제주시 애월읍 무수천길 13",
      "hptlSymbl": null,
      "cobFlagCodeName": "제조업",
      "entpAddr2": null,
      "cobDetailName": "의료기기",
      "taxNo": "6168147075",
      "entpAddr1": "제주특별자치도 제주시 애월읍 무수천길 13"
    }
  ],
  "totalElements": 1
}
```

17. 도로명주소 조회

도로명주소를 조회할 수 있는 API입니다. 도로명주소 조회 시 지번정보도 함께 보여줍니다. 통합업체정보에 등록되어 있지 않은 업체를 거래처로 등록할 경우 주소정보를 조회할 수 있습니다.

17.1. Request

```
POST /api/v1/company-info/street?offset=1&limit=10&streetName=오송생명로 HTTP/1.1
Host: udiportal.mfds.go.kr
Authorization: Bearer {access_token}
Accept: application/json;charset=UTF-8
```

Access_token을 Authorization Header에 담아 POST로 요청합니다.

◆ **요청필드**

키	설명	구분	필수	타입
offset	페이지 번호	Query	O	number
limit	페이징에 담길 개수 (최대 100)	Query	O	number
streetName	검색할 주소명	Query	O	string(80)

◆ **예제**

```
GET /api/v1/company-info/street?offset=1&limit=10&streetName=오송생명로 HTTP/1.1
Host: udiportal.mfds.go.kr
Authorization: Bearer {access_token}
Accept: application/json;charset=UTF-8
```

◆ **예제코드 (Java, Spring)**

```
UriComponentsBuilder uriBuilder = UriComponentsBuilder.fromHttpUrl("https://udiportal.mfds.go.kr/api/v1/company-info/street")
                        .queryParam("offset", 1)
                        .queryParam("limit", 10)
                        .queryParam("streetName", "오송생명로 123")
                        ;
HttpHeaders headers = new HttpHeaders();
```

```
String accessToken = TestUtil.obtainAccessToken(clientId, clientSecret);
headers.add(HttpHeaders.AUTHORIZATION, "Bearer " + accessToken);
headers.add(HttpHeaders.ACCEPT, MediaType.APPLICATION_JSON_UTF8_VALUE);
HttpEntity httpEntity = new HttpEntity(headers);

ResponseEntity<String> responseEntity = restTemplate.exchange(uriBuilder.toUriString(),
                                                              HttpMethod.GET,
                                                              httpEntity,
                                                              String.class);

assertTrue("정상응답 검증", responseEntity.getStatusCode().is2xxSuccessful());
JSONObject body = new JSONObject(responseEntity.getBody());
```

17.2. Response

◆ 응답코드

코드	설명
200	정상응답
400	입력값이 잘못되었을 경우
403	해당 정보에 접근 권한이 없을 경우
401	인증정보가 존재하지 않을 경우
500	처리 중 오류가 발생했을 경우

◆ 응답필드

키	설명	타입
page	현재 페이지 번호	number
pageSize	페이지 크기	number
totalElements	총 갯수	number
items[].fullStreetName	도로명주소	string(300)
items[].fullLandName	지번주소	string(300)

◆ 응답예제

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8

{
  "pageSize": 10,
  "page": 1,
  "items": [
    {
      "fullStreetName": "충청북도 청주시 흥덕구 오송읍 오송생명로 123 ",
      "fullLandName": "충청북도 청주시 흥덕구 오송읍 연제리 679 실험동물센터"
    },
    {
      "fullStreetName": "충청북도 청주시 흥덕구 오송읍 오송생명로 123 ",
      "fullLandName": "충청북도 청주시 흥덕구 오송읍 연제리 679 첨단의료기기개발연구센터"
    },
    {
      "fullStreetName": "충청북도 청주시 흥덕구 오송읍 오송생명로 123 ",
      "fullLandName": "충청북도 청주시 흥덕구 오송읍 연제리 679 임상시업신약생산센터"
    },
    {
      "fullStreetName": "충청북도 청주시 흥덕구 오송읍 오송생명로 123 ",
      "fullLandName": "충청북도 청주시 흥덕구 오송읍 연제리 679 신약개발지원센터"
    }
  ],
  "totalElements": 4
}
```

18. 거래처 등록

거래처를 등록하는 API입니다.

- **[통합업체 목록조회]** 에서 거래처로 등록할 업체를 조회한 후 조회된 해당 업체를 거래처로 등록

18.1. Request

```
POST /api/v1/company-info/bcnc HTTP/1.1
Host: udiportal.mfds.go.kr
Authorization: Bearer {access_token}
Accept: application/json;charset=UTF-8
```

Access_token을 Authorization Header에 담아 요청 내용을 JSON string으로 만들어 POST 로 요청합니다.

◆ **요청필드**

키	설명	구분	필수	타입
cobFlagCode	거래처 업종 코드 1: 제조, 2:수입, 9:판매(임대), 10:요양기관, 99:기타	Body	O	number
taxNo	거래처 사업자등록번호	Body	X	string(10)
companyName	거래처명	Body	O	string(30)
bcncUnityCompanySeq	거래처 통합업체 일련번호	Body	O	string
entpAddr1	거래처 주소1	Body	O	string(200)
entpAddr2	거래처 주소2	Body	X	string(160)

◆ **예제**

```
POST /api/v1/company-info/bcnc HTTP/1.1
Host: udiportal.mfds.go.kr
Authorization: Bearer {access_token}
Content-Type: application/json;charset=UTF-8

{
    "cobFlagCode" : "99",
    "taxNo" : "1231231231",
    "companyName" : "(테)오송실업",
```

```
    "bcncUnityCompanySeq" : "",
    "entpAddr1" : "충청북도 청주시 흥덕구 오송읍 오송생명로1",
    "entpAddr2" : ""
}
```

- **예제코드 (Java, Spring)**

```
UriComponentsBuilder uriBuilder = UriComponentsBuilder.fromHttpUrl("https://udiportal.mfds.go.kr/api/v1/company-info/bcnc");
  HttpHeaders headers = new HttpHeaders();
  String accessToken = TestUtil.obtainAccessToken(clientId, clientSecret);
  headers.add(HttpHeaders.AUTHORIZATION, "Bearer " + accessToken);
  headers.add(HttpHeaders.ACCEPT, MediaType.APPLICATION_JSON_UTF8_VALUE);
  Map<String, Object> body = new HashMap<>();
  body.put("unityCompanySeq", "10001332"); //업체일련번호
  body.put("cobFlagCode", "99"); //거래처 업종 코드
  body.put("taxNo", "1231231231"); //거래처 사업자 등록 번호
  body.put("companyName", "(테)오송실업"); //거래처명
  body.put("bcncUnityCompanySeq", ""); //거래처 통합업체 일련번호
  body.put("entpAddr1", "충청북도 청주시 흥덕구 오송읍 오송생명로1");//거래처 주소1
  body.put("entpAddr2", "");//거래처 주소2
  HttpEntity httpEntity = new HttpEntity(body, headers);

  ResponseEntity<String> responseEntity = restTemplate.exchange(uriBuilder.toUriString(),
                  HttpMethod.POST,
                  httpEntity,
                  String.class);

  assertThat("정상응답 검증", responseEntity.getStatusCode().is2xxSuccessful(), is(true));
  JSONObject response = new JSONObject(responseEntity.getBody());
```

18.2. Response

등록된 결과 정보가 반환됩니다.

- **응답코드**

코드	설명
200	정상응답
400	입력값이 잘못되었을 경우
403	해당 정보에 접근 권한이 없을 경우

401	인증정보가 존재하지 않을 경우
500	처리 중 오류가 발생했을 경우

◆ 응답필드

키	설명	타입
no	순번	string
bcncUnityCompanySeqs	거래처 통합업체 일련번호 (통합업체정보를 조회하지 않고 '기타'업종으로 등록한 경우 null)	string
bcncCode	거래처 코드	string(30)
companyName	거래처명	string(30)
hptlSymbl	거래처 요양기관기호 (요양기관을 거래처로 등록한 경우)	string(50)
bcncCobFlagCodeNm	거래처 업종명	string
entpAddr	거래처 주소	string
taxNo	거래처 사업자등록번호	string(10)

◆ 응답예제

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8

{
   "no": "1",
   "bcncUnityCompanySeqs": null,
   "bcncCode": "134177000000061",
   "companyName": "(테)오송실업",
   "hptlSymbl": null,
   "bcncCobFlagCodeNm": "기타",
   "entpAddr": "충청북도 청주시 흥덕구 오송읍 오송생명로1 ",
   "taxNo": "1231231231"
}
```

◆ **오류 응답 예제 – 필수 입력 값 미 입력**

```
HTTP/1.1 400 BAD_REQUEST
Content-Type: application/json;charset=UTF-8
{"message":"거래처 업종코드가 존재해야 합니다.","status":400,"errors":[]}
```

◆ **오류 응답 예제 – 업종 코드 오류**

```
HTTP/1.1 400 BAD_REQUEST
Content-Type: application/json;charset=UTF-8
{"message":"올바르지 않은 업종코드 입니다.","status":400,"errors":[]}
```

◆ **오류 응답 예제 – 사업자 등록번호 자릿수 및 정규식 오류**

```
HTTP/1.1 400 BAD_REQUEST
Content-Type: application/json;charset=UTF-8
{"message":"올바르지 않은 사업자등록번호 입니다.","status":400,"errors":[]}
```

◆ **오류 응답 예제 – 사업자 등록번호 입력 오류(하이픈(-)포함 입력 한 경우)**

```
HTTP/1.1 400 BAD_REQUEST
Content-Type: application/json;charset=UTF-8
{"message":"사업자 등록번호는 하이픈(-)을 제외한 10자리여야 합니다.","status":400,"errors":[]}
```

◆ **오류 응답 예제 – 거래처 통합업체 일련번호 오류**
(업종코드는 있으나 거래처 통합업체일련번호가 없을 경우(기타업체 등록 시 예외))

```
HTTP/1.1 400 BAD_REQUEST
Content-Type: application/json;charset=UTF-8
{"message":"거래처 통합업체 일련번호가 존재하지 않습니다. 통합업체정보에서 조회한 후 입력하시기 바랍니다.","status":400,"errors":[]}
```

19. 거래처 수정

거래처를 수정하는 API입니다.

19.1. Request

```
POST /api/v1/company-info/bcnc/{bcncCode} HTTP/1.1
Host: udiportal.mfds.go.kr
Authorization: Bearer {access_token}
Accept: application/json;charset=UTF-8
```

Access_token을 Authorization Header에 담아 POST 로 요청합니다.

- **요청필드**

키	설명	구분	필수	타입(길이)
bcncCode	거래처 코드	PathVariable	O	number
cobFlagCode	거래처 업종 코드 1: 제조, 2:수입, 9:판매(임대), 10:요양기관, 99:기타	Body	O	number
taxNo	거래처 사업자등록번호	Body	X	string(10)
companyName	거래처명	Body	O	string(30)
bcncUnityCompanySeq	거래처 통합업체 일련번호	Body	X	string
entpAddr1	거래처 주소1	Body	O	string(200)
entpAddr2	거래처 주소2	Body	X	string(160)

- **예제**

```
POST /api/v1/company-info/bcnc/{bcncCode} HTTP/1.1
Host: udiportal.mfds.go.kr
Authorization: Bearer {access_token}
Content-Type: application/json;charset=UTF-8
```

◆ **예제코드 (Java, Spring)**

```
String unityCompanySeq = "10001332";
String bcncCode = "134177000000065";

UriComponentsBuilder uriBuilder = UriComponentsBuilder.fromHttpUrl(baseUrl + "/bcnc")
                        .pathSegment(bcncCode)
                        ;
HttpHeaders headers = new HttpHeaders();
String accessToken = TestUtil.obtainAccessToken(clientId, clientSecret);
headers.add(HttpHeaders.AUTHORIZATION, "Bearer " + accessToken);
headers.add(HttpHeaders.ACCEPT, MediaType.APPLICATION_JSON_UTF8_VALUE);
Map<String, Object> body = new HashMap<>();
body.put("unityCompanySeq", "10001332"); //업체일련번호
body.put("cobFlagCode", "99"); //거래처 업종 코드
body.put("taxNo", "1231231231"); //거래처 사업자 등록 번호
body.put("companyName", "(테)오송실업");//거래처명
body.put("bcncUnityCompanySeq", "");//거래처 통합업체 일련번호
body.put("entpAddr1", "충청북도 청주시 흥덕구 오송읍 오송생명로1123123");//거래처 주소1
body.put("entpAddr2", "");//거래처 주소2
body.put("registId", "10001332");//등록자ID
HttpEntity httpEntity = new HttpEntity(body, headers);

ResponseEntity<String> responseEntity = restTemplate.exchange(uriBuilder.toUriString(),
                        HttpMethod.POST,
                        httpEntity,
                        String.class);

assertThat("정상응답 검증", responseEntity.getStatusCode().is2xxSuccessful(), is(true));
JSONObject response = new JSONObject(responseEntity.getBody());
```

19.2. Response

수정된 결과 정보가 반환됩니다.

◆ **응답코드**

코드	설명
200	정상응답
400	입력값이 잘못되었을 경우, 해당 UDI-DI 코드가 존재하지 않을 경우

403	해당 정보에 접근 권한이 없을 경우
401	인증정보가 존재하지 않을 경우
500	처리 중 오류가 발생했을 경우

◆ 응답필드

키	설명	타입
no	순번	string
bcncUnityCompanySeqs	거래처 통합업체 일련번호 (통합업체정보를 조회하지 않고 '기타'업종으로 등록한 경우 null)	string
bcncCode	거래처 코드	string(30)
companyName	거래처명	string(30)
hptlSymbl	거래처 요양기관기호 (요양기관을 거래처로 등록한 경우)	string(50)
bcncCobFlagCodeNm	거래처 업종명	string
entpAddr	거래처 주소	string
taxNo	거래처 사업자등록번호	string(10)

◆ 응답예제

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8

{
  "no": "1",
  "bcncUnityCompanySeqs": null,
  "bcncCode": "134177000000065",
  "companyName": "(테)오송실업",
  "hptlSymbl": null,
  "bcncCobFlagCodeNm": null,
  "entpAddr": "충청북도 청주시 흥덕구 오송읍 오송생명로1123123 ",
  "taxNo": "1231231231"
}
```

◆ 오류 응답 예제 – 필수 입력 값 미 입력

```
HTTP/1.1 400 BAD_REQUEST
Content-Type: application/json;charset=UTF-8
{"message":"거래처 업종코드가 존재해야 합니다.","status":400,"errors":[]}
```

◆ 오류 응답 예제 – 업종 코드 수정

```
HTTP/1.1 400 BAD_REQUEST
Content-Type: application/json;charset=UTF-8
{"message":"기타 이외 업종은 수정 할 수 없습니다.","status":400,"errors":[]}
```

◆ 오류 응답 예제 – 사업자 등록번호 자릿수 및 정규식 오류

```
HTTP/1.1 400 BAD_REQUEST
Content-Type: application/json;charset=UTF-8
{"message":"올바르지 않은 사업자등록번호 입니다.","status":400,"errors":[]}
```

◆ 오류 응답 예제 – 사업자 등록번호 입력 오류(하이픈(-)포함 입력 한 경우)

```
HTTP/1.1 400 BAD_REQUEST
Content-Type: application/json;charset=UTF-8
{"message":"사업자 등록번호는 하이픈(-)을 제외한 10자리여야 합니다.","status":400,"errors":[]}
```

20. 공급내역 보고자료 목록 조회

본 API는 **[공급내역 보고자료 추가]** API를 통해 생성된 공급내역 보고자료 목록을 조회합니다. 공급내역 보고 관련 API를 사용하는데 필요한 값은 응답필드에 굵은글씨(Bold)처리 되어있습니다.

20.1. Request

```
GET /api/v1/supply-info/manages?offset=1&limit=50&suplyContStdmt=201909 HTTP/1.1
Host: udiportal.mfds.go.kr
Authorization: Bearer {access_token}
Accept: application/json;charset=UTF-8
```

Access_token을 Authorization Header에 담아 GET 으로 요청합니다.

◆ **요청필드**

키	설명	구분	필수	타입
offset	페이징이 시작될 숫자	Query	O	number
limit	페이징에 담길 개수 (최대 100)	Query	O	number
suplyContStdmt	공급내역 보고 기준 월	Query	O	string(6)
suplyFlagCode	공급 구분 코드 (1:출고, 2:반품, 3:폐기, 4:임대, 5:회수)	Query	X	string(1)
suplyTypeCode	공급 형태 코드 "1" : 제조・수입・판매(임대)에 공급 "2" : 의료기관에 공급 "3" : 약국개설자 또는 의약품도매상에 공급 "4" : 견본품, 기부용 또는 군납용 등으로 공급	Query	X	string(1)
bcncName	공급받은 자(반품한 자) 업체명 (LIKE 검색)	Query	X	string(100)
typeName	공급한 제품의 모델명 (LIKE 검색) URL encoding해야 하며 아래 문자는 별도로 URL encoding 전에 치환해줘야 합니다. "+" -> "%2B" "&" -> "%26"	Query	X	string(4000)

sortable	정렬 조건 기본 정렬 조건 공급일자 내림차순 suplyDateAsc : 공급일자 오름차순 suplyFlagCode : 공급내역 구분 순(출고, 반품, 폐기, 임대, 회수 순) bcncName : 공급받은 자(반품한 자) 업체명 오름차순	Query	X	string(30)

- **예제**

```
GET /api/v1/supply-info/manages?offset=1&limit=50&suplyContStdmt=201909 HTTP/1.1
Host: udiportal.mfds.go.kr
Authorization: Bearer {access_token}
Accept: application/json;charset=UTF-8
```

- **예제코드 (Java, Spring)**

```
UriComponentsBuilder uriBuilder = UriComponentsBuilder.fromHttpUrl("https://udiportal.mfds.go.kr/api/v1/supply-info/manages")
                    .queryParam("offset", 1)
                    .queryParam("limit", 10)
                    .queryParam("suplyContStdmt", "201909");
HttpHeaders headers = new HttpHeaders();
String accessToken = TestUtil.obtainAccessToken(clientId, clientSecret);
headers.add(HttpHeaders.AUTHORIZATION, "Bearer " + accessToken);
headers.add(HttpHeaders.ACCEPT, MediaType.APPLICATION_JSON_UTF8_VALUE);
HttpEntity httpEntity = new HttpEntity(headers);

ResponseEntity<String> responseEntity = restTemplate.exchange(uriBuilder.toUriString(),
                    HttpMethod.GET,
                    httpEntity,
                    String.class);

JSONObject body = new JSONObject(responseEntity.getBody());
```

20.2. Response

- **응답코드**

코드	설명
200	정상응답

400	입력값이 잘못되었을 경우
403	해당 정보에 접근 권한이 없을 경우
401	인증정보가 존재하지 않을 경우
500	처리 중 오류가 발생했을 경우

◆ 응답필드

키	설명	타입
page	현재 페이지 번호	number
pageSize	페이지 크기	number
totalElements	총 갯수	number
items[].no	순번	number
items[].suplyContStdmt	공급내역 보고 기준 월 형식 : YYYYMM	string(6)
items[].suplyContSeq	공급내역 보고자료 일련번호	number
items[].suplyEntpName	공급한 자 업체명	string(100)
items[].cobTypeName	공급한 자 업체 영업형태	string(1000)
items[].suplyFlagCode	공급 구분 코드 (1:출고, 2:반품, 3:폐기, 4:회수, 5:임대)	string(1)
items[].suplyTypeCode	공급 형태 코드 "1" : 제조・수입・판매(임대)에 공급 "2" : 의료기관에 공급 "3" : 약국개설자 또는 의약품도매상에 공급 "4" : 견본품, 기부용 또는 군납용 등으로 공급	string(1)
items[].meddevItemSeq	의료기기 품목 일련번호	number(50)
items[].seq	모델명 일련번호	number(50)
items[].udiDiSeq	UDI-DI 코드 일련번호	number(10)
items[].stdCode	표준코드	string(200)
items[].udiDiCode	고유식별자(UDI-DI) 코드	string(60)
items[].udiPiCode	생산식별자(UDI-PI) 코드	string(100)
items[].entpName	품목 제조(수입) 업체명	string(100)
items[].itemName	품목명	string(500)
items[].meaClassNo	품목 분류번호	string(10)

items[].permitNo	품목 허가번호	string(50)
items[].typeName	모델명	string(500)
items[].packQuantity	포장내 수량	number
items[].lotNo	로트번호	string(100)
items[].itemSeq	일련번호	string(50)
items[].manufYm	제조연월 형식 : YYMMDD(표준코드 및 생산식별자 코드에 일 정보가 없는 경우 '00'으로 대체)	string(6)
items[].useTmlmt	사용기한 형식 : YYMMDD(표준코드 및 생산식별자 코드에 일 정보가 없는 경우 '00'으로 대체)	string(6)
items[].bcncCode	거래처 코드	string(30)
items[].bcncEntpName	거래처 명	string(100)
items[].bcncCobTypeName	거래처 업종	string(1000)
items[].bcncEntpAddr	거래처 소재지	string(1100)
items[].bcncTaxNo	거래처 사업자등록번호	string(10)
items[].bcncHptlCode	거래처 요양기관 기호	string(50)
items[].isDiffDvyfg	납품장소 다름 여부	boolean
items[].dvyfgPlaceBcncCode	납품장소 업체 거래처 코드	string(30)
items[].dvyfgEntpName	납품장소 업체명	string(100)
items[].dvyfgCobTypeName	납품장소 업체 영업형태	string(1000)
items[].dvyfgEntpAddr	납품장소 업체 소재지	string(1100)
items[].dvyfgTaxNo	납품장소 업체 사업자등록번호	string(10)
items[].dvyfgHptlCode	납품장소 업체 요양기관 기호	string(50)
items[].suplyDate	출고・반품・폐기・임대・회수 일자	string(8)
items[].suplyQty	출고・반품・폐기・임대・회수 수량	string
items[].indvdlzSuplyQty	포장내수량중 낱개 회수수량	string
items[].suplyUntpc	공급단가(VAT포함, 원)	string
items[].suplyAmt	공급금액(VAT포함, 원)	string
items[].remark	비고	string(2000)
items[].usedMeddevYn	중고 의료기기 여부	string(1)

◆ **응답예제**

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8

{
  "pageSize": 10,
  "page": 1,
  "items": [
    {
      "no": 1,
      "suplyContSeq": 11183,
      "suplyEntpName": "(주)네오메드",
      "suplyAmt": "100000",
      "suplyFlagCode": "1",
      "suplyTypeCode": "2",
      "bcncCode": "100013320000012",
      "bcncTaxNo": "0000000000",
      "meddevItemSeq": 20060048,
      "typeName": "탄력밴드(JC-7500 UNI)",
      "isDiffDvyfg": true,
      "remark": "",
      "dvyfgEntpName": "SEJONG hospital",
      "manufYm": "191001",
      "dvyfgPlaceBcncCode": "100013320000010",
      "itemName": "탄력밴드",
      "dvyfgHptlCode": "",
      "lotNo": "Q1231",
      "entpName": "(주)네오메드",
      "bcncCobTypeName": "판매(임대)업",
      "suplyUntpc": "100000",
      "seq": 7149,
      "bcncEntpAddr": "서울특별시 은평구 은평로 221-2 4층",
      "suplyDate": "20190912",
      "meaClassNo": "B07090",
      "packQuantity": 1,
      "bcncHptlCode": "",
      "permitNo": "부산 제신 06-120 호",
```

```
        "itemSeq": "",
        "stdCode": "(01)08811344578771(10)Q1231(11)191001",
        "useTmlmt": "",
        "bcncEntpName": "개발팀",
        "dvyfgEntpAddr": "세종특별자치시  보람로 95 도램마을13단지 22",
        "dvyfgTaxNo": "",
        "suplyQty": "1000",
        "indvdlzSuplyQty": "12",
        "udiDiSeq": 1958,
        "udiPiCode": "(10)Q1231(11)191001",
        "udiDiCode": "08811344578771",
        "cobTypeName": "제조업",
        "suplyContStdmt": "201909",
        "dvyfgCobTypeName": "기타"
        "usedMeddevYn":"N"
    }
  ],
  "totalElements": 1
}
```

21. 공급내역 보고자료 추가

공급내역 보고자료를 생성합니다. 공급내역 보고자료 생성은 공급내역 보고 및 재보고 시 자료로 활용되며, 공급내역 보고 완료된 상태는 아닙니다. 해당 보고자료는 공급내역 보고 및 재보고 수행 처리가 완료되어야 정상적인 공급내역 보고입니다.

21.1. Request

```
POST /api/v1/supply-info/manages/{suplyContStdmt} HTTP/1.1
Host: udiportal.mfds.go.kr
Authorization: Bearer {access_token}
Content-Type: application/json;charset=UTF-8
```

Access_token을 Authorization Header에 담아 요청 내용을 JSON string으로 만들어 POST 로 요청합니다.

◆ **요청필드**

키	설명	구분	필수	타입
suplyContStdmt	공급내역 보고 기준 월 형식 : YYYYMM	PathVariable	O	string(6)
suplyFlagCode	공급 구분 코드 (1:출고, 2:반품, 3:폐기, 4:임대, 5: 회수)	Body	O	string(5)
suplyTypeCode	공급 형태 코드 "1" : 제조・수입・판매(임대)에 공급 "2" : 의료기관에 공급 "3" : 약국개설자 또는 의약품도매상에 공급 "4" : 견본품, 기부용 또는 군납용 등으로 공급 (공급구분이 출고,임대인 경우 필수)	Body	O	string(5)
meddevItemSeq	의료기기 품목 일련번호	Body	O	number
seq	모델명 일련번호	Body	O	number
udiDiSeq	UDI-DI 코드 일련번호	Body	O	number
stdCode	표준코드 (식별 가능한 AI가 포함된 문자)	Body	O	string(200)
udiDiCode	고유식별자(UDI-DI) 코드	Body	O	string(60)
udiPiCode	생산식별자(UDI-PI) 코드 (식별 가능한 AI가 포함된 문자)	Body	O	string(100)

키	설명	구분	필수	타입
lotNo	로트번호(표준코드 및 생산식별자 코드에 포함된 경우 필수)	Body	X	string(100)
itemSeq	일련번호(표준코드 및 생산식별자 코드에 포함된 경우 필수)	Body	X	string(50)
manufYm	제조연월(표준코드 및 생산식별자 코드에 포함된 경우 필수) 형식 : YYMMDD(표준코드 및 생산식별자 코드에 일 정보가 없는 경우 '00'으로 대체)	Body	X	string(6)
useTmlmt	사용기한(표준코드 및 생산식별자 코드에 포함된 경우 필수) 형식 : YYMMDD(표준코드 및 생산식별자 코드에 일 정보가 없는 경우 '00'으로 대체)	Body	X	string(6)
bcncCode	거래처 코드(공급 구분 코드가 출고/반품의 경우 필수)	Body	X	string(30)
isDiffDvyfg	납품장소 다름 여부(공급 구분 코드가 출고, 임대, 회수의 경우 필수)	Body	X	boolean
dvyfgPlaceBcncCode	납품장소 업체 거래처 코드(공급 구분 코드가 출고, 납품장소 다름 여부가 'true'의 경우 필수)	Body	X	string(30)
suplyDate	출고・반품・폐기・임대・회수 일자 형식 : YYYYMMDD	Body	O	string(8)
suplyQty	출고・반품・폐기・임대・회수 수량(공급구분이 '회수'일 때 suplyQty와 indvdlzSuplyQty 둘중 한 개 필드는 필수)	Body	O	string(10)
indvdlzSuplyQty	포장내수량중 낱개 회수수량 (공급구분이 '회수'일 때 suplyQty와 indvdlzSuplyQty 둘중 한 개 필드는 필수)	Body	X	string(10)
suplyUntpc	공급단가(VAT포함, 원)(공급 구분 코드가 출고, 거래처가 요양기관의 경우 필수) 형식 : 숫자로만 이루어진 문자열(콤마 등의 문자 제외)	Body	X	string(10)
suplyAmt	공급금액(VAT포함, 원)(공급 구분 코드가 출고, 거래처가 요양기관의 경우 필수) 공급금액 = 수량*공급단가 형식 : 숫자로만 이루어진 문자열	Body	X	string(10)

키	설명	구분	필수	타입
	(콤마 등의 문자 제외)			
remark	비고	Body	X	string(650)
usedMeddevYn	중고 의료기기 여부(Y:해당, N:미해당)	Body	X	string(1)

◆ **예제**

```
POST /api/v1/supply-info/manages/201909 HTTP/1.1
Host: udiportal.mfds.go.kr
Authorization: Bearer {access_token}
Content-Type: application/json;charset=UTF-8
{
    "suplyFlagCode": "1",
    "suplyTypeCode": "3",
    "meddevItemSeq": 20060048,
    "seq": 7149,
    "udiDiSeq": 1958,
    "stdCode": "(01)08811344578771(10)Q1231(11)191001",
    "udiDiCode": "08811344578771",
    "udiPiCode": "(10)Q1231(11)191001",
    "lotNo": "Q1231",
    "manufYm": "191001",
    "bcncCode": "100013320000012",
    "suplyDate": "20190902",
    "suplyQty": "1000",
    "indvdlzSuplyQty": "12",
    "remark": "API 테스트"
}
```

◆ **예제코드 (Java, Spring)**

```
String suplyContStdmt = "201909";
UriComponentsBuilder uriBuilder = UriComponentsBuilder.fromHttpUrl("https://udiportal.mfds.go.kr/api/v1/supply-info/manages)
                                .pathSegment(suplyContStdmt);
HttpHeaders headers = new HttpHeaders();
String accessToken = TestUtil.obtainAccessToken(clientId, clientSecret);
headers.add(HttpHeaders.AUTHORIZATION, "Bearer " + accessToken);
headers.add(HttpHeaders.CONTENT_TYPE, MediaType.APPLICATION_JSON_UTF8_VALUE);
Map<String, Object> body = new HashMap<>();
body.put("suplyFlagCode", "1");
body.put("suplyTypeCode", "3");
```

```
body.put("meddevItemSeq", 20060048);
body.put("seq", 7149);
body.put("udiDiSeq", 1958);
body.put("stdCode", "(01)08811344578771(10)Q1231(11)191001");
body.put("udiDiCode", "08811344578771");
body.put("udiPiCode", "(10)Q1231(11)191001");
body.put("lotNo", "Q1231");
body.put("manufYm", "191001");
body.put("bcncCode", "100013320000012");
body.put("suplyDate", "20190902");
body.put("suplyQty", "1000");
body.put("indvdlzSuplyQty", "12");
body.put("remark", "API 테스트");
HttpEntity<Map<String, Object>> httpEntity = new HttpEntity<>(body, headers);

ResponseEntity<String> responseEntity = restTemplate.exchange(uriBuilder.toUriString(),
                    HttpMethod.POST,
                    httpEntity,
                    String.class);

JSONObject response = new JSONObject(responseEntity.getBody());
```

21.2. Response

- **응답코드**

코드	설명
200	정상응답
400	입력값이 잘못되었을 경우
403	해당 정보에 접근 권한이 없을 경우
401	인증정보가 존재하지 않을 경우
500	처리 중 오류가 발생했을 경우

- **응답필드**

키	설명	타입
suplyContSeq	공급내역 보고자료 일련번호	string(10)
suplyContStdmt	공급내역 기준 월	string(6)

◆ 성공 응답 예제

```
HTTP/1.1 400 BAD_REQUEST
Content-Type: application/json;charset=UTF-8

{"suplyContSeq":"1","suplyContStdmt":"202005"}
```

◆ 오류 응답 예제 - 공급 구분 코드 미입력

```
HTTP/1.1 400 BAD_REQUEST
Content-Type: application/json;charset=UTF-8

{"message":"공급 구분 값이 없습니다.","status":400,"errors":[]}
```

◆ 오류 응답 예제 – 표준코드와 제조연월 입력 정보가 다른 경우

```
HTTP/1.1 400 BAD_REQUEST
Content-Type: application/json;charset=UTF-8

{"message":"제조연월의 날짜 형식이 잘못되었습니다.(형식 : YYMMDD, 일이 없을경우 DD는 00으로 입력)","status":400,"errors":[]}
```

참고사항

보고하고자 하는 품목이 사용자멸균 정형용품이고 표준코드 체계가 GS1, ICCBBA인 경우 요청 필드의 표준코드에 식별자(ex. GS1 : (01), ICCBBA : =/)가 포함되어 있어야 하지만 HIBCC의 경우 구분자(/)는 필수조건이 아닙니다. 또한 사용자멸균 정형용품인 경우 UDI-PI 및 관리형태(로트번호, 일련번호, 제조연월, 사용기한)가 입력되면 오류처리 됩니다.

body 입력예시.

표준코드(UDI)	UDI-DI	UDI-PI	처리	비고
(01)08812345678	08812345678		정상	
08812345678	08812345678		오류	식별자를 누락한 경우
+H123SD12D475/	+H123SD12D475		정상	
+H123SD12D475	+H123SD12D475		정상	
(01)08812345678(10)LP123(17)20210101			오류	표준코드에 PI가 포함된 경우
(01)08812345678	(01)08812345678		오류	UDI-DI에 식별자가 입력된 경우
=/W123456789454	W123456789454		정상	

22. 공급내역 보고자료 수정

수정해야할 필드만 입력한 뒤 **[공급내역 보고자료 수정]** API를 호출하여 공급내역 보고자료를 수정합니다. 공급내역 보고 및 재보고 처리가 완료된 상태에서 공급내역 보고자료의 정보를 수정하게 된 경우, 해당 보고자료는 공급내역 재보고 수행 처리가 완료되어야 정상적인 공급내역 보고입니다.

22.1. Request

```
POST /api/v1/supply-info/manages/update/{suplyContStdmt}/{suplyContSeq} HTTP/1.1
Host: udiportal.mfds.go.kr
Authorization: Bearer {access_token}
Content-Type: application/json;charset=UTF-8
```

Access_token을 Authorization Header에 담아 요청 내용을 JSON string으로 만들어 POST 로 요청합니다.

◆ **요청필드**

키	설명	구분	필수	타입
suplyContStdmt	공급내역 보고 기준 월 형식 : YYYYMM	PathVariable	O	string(6)
suplyContSeq	공급내역 보고자료 일련번호	PathVariable	O	number
suplyFlagCode	공급 구분 코드 (1:출고, 2:반품, 3:폐기, 4:임대, 5: 회수)	Body	X	string(5)
suplyTypeCode	공급 형태 코드 "1" : 제조・수입・판매(임대)에 공급 "2" : 의료기관에 공급 "3" : 약국개설자 또는 의약품도매상에 공급 "4" : 견본품, 기부용 또는 군납용 등으로 공급 (공급구분이 출고,임대인 경우 필수)	Body	X	string(5)
meddevItemSeq	의료기기 품목 일련번호	Body	X	number
seq	모델명 일련번호	Body	X	number
udiDiSeq	UDI-DI 코드 일련번호	Body	X	number
stdCode	표준코드 (식별 가능한 AI가 포함된 문자)	Body	X	string(200)
udiDiCode	고유식별자(UDI-DI) 코드	Body	X	string(60)
udiPiCode	생산식별자(UDI-PI) 코드	Body	X	string(100)

키	설명	구분	필수	타입
	(식별 가능한 AI가 포함된 문자)			
lotNo	로트번호(표준코드 및 생산식별자 코드에 포함된 경우 필수)	Body	X	string(100)
itemSeq	일련번호(표준코드 및 생산식별자 코드에 포함된 경우 필수)	Body	X	string(50)
manufYm	제조연월(표준코드 및 생산식별자 코드에 포함된 경우 필수) 형식 : YYMMDD(표준코드 및 생산식별자 코드에 일 정보가 없는 경우 '00'으로 대체)	Body	X	string(6)
useTmlmt	사용기한(표준코드 및 생산식별자 코드에 포함된 경우 필수) 형식 : YYMMDD(표준코드 및 생산식별자 코드에 일 정보가 없는 경우 '00'으로 대체)	Body	X	string(6)
bcncCode	거래처 코드(공급 구분 코드가 출고/반품의 경우 필수)	Body	X	string(30)
isDiffDvyfg	납품장소 다름 여부(공급 구분 코드가 출고의 경우 필수)	Body	X	boolean
dvyfgPlaceBcncCode	납품장소 업체 거래처 코드(공급 구분 코드가 출고, 납품장소 다름 여부가 'true'의 경우 필수)	Body	X	string(30)
suplyDate	출고・반품・폐기・임대・회수 일자 형식 : YYYYMMDD	Body	X	string(8)
suplyQty	출고・반품・폐기・임대・회수 수량 (공급구분이 '회수'일 때 suplyQty와 indvdlzSuplyQty 둘중 한 개 필드는 필수)	Body	X	string(10)
indvdlzSuplyQty	포장내수량중 낱개 회수수량 (공급구분이 '회수'일 때 suplyQty와 indvdlzSuplyQty 둘중 한 개 필드는 필수)	Body	X	string(10)
suplyUntpc	공급단가(VAT포함, 원)(공급 구분 코드가 출고, 거래처가 요양기관의 경우 필수) 형식 : 숫자로만 이루어진 문자열(콤마 등의 문자 제외)	Body	X	string(10)
suplyAmt	공급금액(VAT포함, 원)(공급 구분 코드가 출고, 거래처가 요양기관의 경우 필수) 공급금액 = 수량*공급단가 형식 : 숫자로만 이루어진 문자열(콤	Body	X	string(10)

키	설명	구분	필수	타입
	마 등의 문자 제외)			
remark	비고	Body	X	string(650)
usedMeddevYn	중고 의료기기 여부(Y:해당, N:미해당)	Body	X	string(1)

◆ **예제**

```
POST /api/v1/supply-info/manages/201909/11203 HTTP/1.1
Host: udiportal.mfds.go.kr
Authorization: Bearer {access_token}
Content-Type: application/json;charset=UTF-8
{
    "suplyQty": "150",
    "remark": "API 테스트"
}
```

◆ **예제코드 (Java, Spring)**

```
String suplyContStdmt = "201909";
int suplyContSeq = 11203;
UriComponentsBuilder uriBuilder = UriComponentsBuilder.fromHttpUrl("https://udiportal.mfds.go.kr/
api/v1/supply-info/manages/update")
                        .pathSegment(suplyContStdmt)
                        .pathSegment(String.valueOf(suplyContSeq));
HttpHeaders headers = new HttpHeaders();
String accessToken = TestUtil.obtainAccessToken(clientId, clientSecret);
headers.add(HttpHeaders.AUTHORIZATION, "Bearer " + accessToken);
headers.add(HttpHeaders.CONTENT_TYPE, MediaType.APPLICATION_JSON_UTF8_VALUE);
Map<String, Object> body = new HashMap<>();
body.put("suplyQty", "150");
body.put("remark", "API 수정 테스트");
HttpEntity<Map<String, Object>> httpEntity = new HttpEntity<>(body, headers);

ResponseEntity<String> responseEntity = restTemplate.exchange(uriBuilder.toUriString(),
                HttpMethod.POST,
                httpEntity,
                String.class);
```

22.2. Response

- **응답코드**

코드	설명
200	정상응답
400	입력값이 잘못되었을 경우
403	해당 정보에 접근 권한이 없을 경우
401	인증정보가 존재하지 않을 경우
500	처리 중 오류가 발생했을 경우

- **응답필드**

  코드(Response Code)만 응답합니다.

- **오류 응답 예제 – 수정된 출고・반품・폐기・임대・회수 일자 정보가 기준 월을 벗어나는 경우**

```
HTTP/1.1 400 BAD_REQUEST
Content-Type: application/json;charset=UTF-8

{"message":"반품일자 값이 기준 월을 벗어납니다.","status":400,"errors":[]}
```

- **오류 응답 예제 – 표준코드 변경없이 다른 일련번호 정보 입력 시**

```
HTTP/1.1 400 BAD_REQUEST
Content-Type: application/json;charset=UTF-8

{"message":"표준코드의 일련번호 값과 입력한 일련번호의 값이 다릅니다.","status":400,"errors":[]}
```

23. 공급내역 보고자료 일괄 삭제

공급내역 보고자료를 일괄 삭제합니다. 공급내역 보고 및 재보고 처리가 완료된 상태에서 공급내역 보고자료의 정보를 삭제하게 된 경우, 해당 보고자료는 공급내역 재보고 수행 처리가 완료되어야 정상적인 공급내역 보고입니다.

23.1. Request

```
POST /api/v1/supply-info/manages/update/{suplyContStdmt} HTTP/1.1
Host: udiportal.mfds.go.kr
Authorization: Bearer {access_token}
Content-Type: application/json;charset=UTF-8
```

Access_token을 Authorization Header에 담아 요청 내용을 JSON string으로 만들어 POST 로 요청합니다.

◆ **요청필드**

키	설명	구분	필수	타입
suplyContStdmt	공급내역 보고 기준 월 형식 : YYYYMM	PathVariable	O	string(6)

◆ **예제**

```
POST /api/v1/supply-info/delete/201909 HTTP/1.1
Host: udiportal.mfds.go.kr
Authorization: Bearer {access_token}
Content-Type: application/json;charset=UTF-8
```

◆ **예제코드 (Java, Spring)**

```
String suplyContStdmt = "201909";
UriComponentsBuilder uriBuilder = UriComponentsBuilder.fromHttpUrl("https://udiportal.mfds.go.kr/
api/v1/supply-info/manages/delete")
                    .pathSegment(suplyContStdmt);
HttpHeaders headers = new HttpHeaders();
String accessToken = TestUtil.obtainAccessToken(clientId, clientSecret);
headers.add(HttpHeaders.AUTHORIZATION, "Bearer " + accessToken);
headers.add(HttpHeaders.CONTENT_TYPE, MediaType.APPLICATION_JSON_UTF8_VALUE);
HttpEntity httpEntity = new HttpEntity(headers);

ResponseEntity responseEntity = restTemplate.exchange(uriBuilder.toUriString(),
```

```
                HttpMethod.POST,
                httpEntity,
                String.class);
```

23.2. Response

◆ **응답코드**

코드	설명
200	정상응답
400	입력값이 잘못되었을 경우
403	해당 정보에 접근 권한이 없을 경우
401	인증정보가 존재하지 않을 경우
500	처리 중 오류가 발생했을 경우

◆ **응답필드**

코드(Response Code)만 응답합니다.

24. 공급내역 보고자료 단건 삭제

공급내역 보고자료를 삭제합니다. 공급내역 보고 및 재보고 처리가 완료된 상태에서 공급내역 보고자료의 정보를 삭제하게 된 경우, 해당 보고자료는 공급내역 재보고 수행 처리가 완료되어야 정상적인 공급내역 보고입니다.

24.1. Request

```
POST /api/v1/supply-info/manages/delete/{suplyContStdmt}/{suplyContSeq} HTTP/1.1
Host: udiportal.mfds.go.kr
Authorization: Bearer {access_token}
Content-Type: application/json;charset=UTF-8
```

Access_token을 Authorization Header에 담아 요청 내용을 JSON string으로 만들어 POST 로 요청합니다.

◆ **요청필드**

키	설명	구분	필수	타입
suplyContStdmt	공급내역 보고 기준 월 형식 : YYYYMM	PathVariable	O	string(6)
suplyContSeq	공급내역 보고자료 일련번호	PathVariable	O	number

◆ **예제**

```
POST /api/v1/supply-info/manages/201909/11203 HTTP/1.1
Host: udiportal.mfds.go.kr
Authorization: Bearer {access_token}
Content-Type: application/json;charset=UTF-8
```

◆ **예제코드 (Java, Spring)**

```
String suplyContStdmt = "201909";
int suplyContSeq = 11203;
UriComponentsBuilder uriBuilder = UriComponentsBuilder.fromHttpUrl("https://udiportal.mfds.go.kr/api/v1/supply-info/manages/delete")
                    .pathSegment(suplyContStdmt)
                    .pathSegment(String.valueOf(suplyContSeq));
HttpHeaders headers = new HttpHeaders();
String accessToken = TestUtil.obtainAccessToken(clientId, clientSecret);
```

```
headers.add(HttpHeaders.AUTHORIZATION, "Bearer " + accessToken);
headers.add(HttpHeaders.CONTENT_TYPE, MediaType.APPLICATION_JSON_UTF8_VALUE);
HttpEntity httpEntity = new HttpEntity(headers);

ResponseEntity responseEntity = restTemplate.exchange(uriBuilder.toUriString(),
                    HttpMethod.POST,
                    httpEntity,
                    String.class);
```

24.2. Response

- **응답코드**

코드	설명
200	정상응답
400	입력값이 잘못되었을 경우
403	해당 정보에 접근 권한이 없을 경우
401	인증정보가 존재하지 않을 경우
500	처리 중 오류가 발생했을 경우

- **응답필드**

코드(Response Code)만 응답합니다.

- **오류 응답 예제 – 삭제할 대상이 없는 경우**

```
HTTP/1.1 400 BAD_REQUEST
Content-Type: application/json;charset=UTF-8

{"message":"삭제할 대상이 존재하지 않습니다.","status":400,"errors":[]}
```

25. 고유식별자(UDI-DI) 품목 정보 조회

본 API는 품목 일련번호, 모델 일련번호, UDI-DI 코드 일련번호 등 공급내역 보고자료 추가 시 입력 필수 사항 정보 및 기타 품목정보를 조회합니다. [공급내역 보고자료 추가] API를 사용하는데 필요한 값은 응답필드에 굵은글씨(Bold)처리 되어있습니다.

25.1. Request

```
GET /api/v1/supply-info/udidi-product HTTP/1.1
Host: udiportal.mfds.go.kr
Authorization: Bearer {access_token}
Accept: application/json;charset=UTF-8
```

Access_token을 Authorization Header에 담아 GET 으로 요청합니다.

- **요청필드**

키	설명	구분	필수	타입
udiDiCode	고유식별자(UDI-DI) 코드 URL encoding해야 하며 아래 문자는 별도로 URL encoding 전에 치환해줘야 합니다. "+" -> "%2B" "&" -> "%26"	Query	O	string(60)

- **예제**

```
GET /api/v1/supply-info/udidi-product HTTP/1.1
Host: udiportal.mfds.go.kr
Authorization: Bearer {access_token}
Accept: application/json;charset=UTF-8
```

- **예제코드 (Java, Spring)**

```
UriComponentsBuilder uriBuilder = UriComponentsBuilder.fromHttpUrl("https://udiportal.mfds.go.kr/api/v1/supply-info/udidi-product")

HttpHeaders headers = new HttpHeaders();
String accessToken = TestUtil.obtainAccessToken(clientId, clientSecret);
headers.add(HttpHeaders.AUTHORIZATION, "Bearer " + accessToken);
headers.add(HttpHeaders.CONTENT_TYPE, MediaType.APPLICATION_FORM_URLENCODED_VALUE);
headers.add(HttpHeaders.ACCEPT, MediaType.APPLICATION_JSON_UTF8_VALUE);
```

```
MultiValueMap<String, String> params = new LinkedMultiValueMap<>();
params.add("udiDiCode", "+B0953021121".replace("+", "%2B").replace("&", "%26"));

HttpEntity httpEntity = new HttpEntity(headers);
uriBuilder.queryParams(params);
        String uriString = uriBuilder.build(false).toUriString();
        ResponseEntity<String> responseEntity = restTemplate.exchange(uriString,
                HttpMethod.GET,
                httpEntity,
                String.class);

JSONObject body = new JSONObject(responseEntity.getBody());
```

25.2. Response

◆ 응답코드

코드	설명
200	정상응답
400	입력값이 잘못되었을 경우
403	해당 정보에 접근 권한이 없을 경우
401	인증정보가 존재하지 않을 경우
500	처리 중 오류가 발생했을 경우

◆ 응답필드

키	설명	타입
items[].meddevItemSeq	품목 일련번호	number
items[].seq	모델 일련번호	number
items[].udiDiSeq	UDI-DI 일련번호	number
items[].udiDiCode	고유식별자(UDI-DI) 코드	string(60)
items[].entpName	업체명	string(100)

items[].cobTypeName	업체 영업 형태	string(1000)
items[].itemName	품목명	string(500)
items[].typeName	모델명	string(4000)
items[].brandName	제품명	string(3000)
items[].permitNo	품목 허가 번호	string(50)
items[].meaClassNo	품목 분류 번호	string(10)
items[].grade	품목 등급	string(1)
items[].useLotNo	로트번호 사용 여부	boolean
items[].useItemSeq	일련번호 사용 여부	boolean
items[].useManufYm	제조연월 사용 여부	boolean
items[].useTimeLimit	사용기한 사용 여부	boolean
items[].packQuantity	포장내 수량	number
items[].userSterilizationYn	사용자멸균 정형용품 여부	boolean
items[].kitYn	세트화 구성 여부	boolean

◆ 응답예제

```
HTTP/1.1 200 O
Content-Type: application/json;charset=UTF-8

{
  "items": [
    {
      "meaClassNo": "B03010.01",
      "packQuantity": 1,
      "brandName": "Hintermann Series",
      "useItemSeq": false,
      "meddevItemSeq": 200204237,
      "typeName": "Talar Implant 302112",
      "permitNo": "수허 02-1233 호",
      "useTimeLimit": true,
      "itemName": "인공발목관절",
      "udiDiSeq": 1111385,
      "entpName": "(주)이건메디칼",
      "udiDiCode": "+B0953021121",
      "useManufYm": false,
```

```
        "grade": "3",
        "cobTypeName": "수입업",
        "useLotNo": true,
        "seq": 430310,
        "userSterilizationYn":false,
        "kitYn": false
      }
   ]
}
```

26. 공급내역 보고 이력 목록 조회

본 API는 **[공급내역 보고자료 추가]** API와 **[공급내역 보고 및 재보고 처리]** API 통해 생성된 공급내역 보고 이력 목록을 조회합니다. **[공급내역 보고 및 재보고 처리]** API를 사용하는데 필요한 값은 응답필드에 굵은글씨(Bold)처리 되어있습니다.

26.1. Request

```
GET /api/v1/supply-info/report-summary HTTP/1.1
Host: udiportal.mfds.go.kr
Authorization: Bearer {access_token}
Accept: application/json;charset=UTF-8
```

Access_token을 Authorization Header에 담아 GET 으로 요청합니다.

- **요청필드**

키	설명	구분	필수	타입
startSuplyContStdmt	공급내역 보고 이력 검색 시작 기준 월 형식 : YYYYMM	Query	X	string(6)
endSuplyContStdmt	공급내역 보고 이력 검색 종료 기준 월 형식 : YYYYMM	Query	X	string(6)

- **예제**

```
GET /api/v1/supply-info/report-summary HTTP/1.1
Host: udiportal.mfds.go.kr
Authorization: Bearer {access_token}
Accept: application/json;charset=UTF-8
```

- **예제코드 (Java, Spring)**

```
UriComponentsBuilder uriBuilder = UriComponentsBuilder.fromHttpUrl("https://udiportal.mfds.go.kr/api/v1/supply-info/report-summary")
                    .queryParam("startSuplyContStdmt", "201901")
                    .queryParam("endSuplyContStdmt", "201912")
HttpHeaders headers = new HttpHeaders();
String accessToken = TestUtil.obtainAccessToken(clientId, clientSecret);
headers.add(HttpHeaders.AUTHORIZATION, "Bearer " + accessToken);
headers.add(HttpHeaders.ACCEPT, MediaType.APPLICATION_JSON_UTF8_VALUE);
HttpEntity httpEntity = new HttpEntity(headers);
```

```
ResponseEntity<String> responseEntity = restTemplate.exchange(uriBuilder.toUriString(),
                    HttpMethod.GET,
                    httpEntity,
                    String.class);

JSONObject body = new JSONObject(responseEntity.getBody());
```

26.2. Response

- **응답코드**

코드	설명
200	정상응답
400	입력값이 잘못되었을 경우
403	해당 정보에 접근 권한이 없을 경우
401	인증정보가 존재하지 않을 경우
500	처리 중 오류가 발생했을 경우

- **응답필드**

키	설명	타입
items[].suplyContStdmt	공급내역 보고 기준 월 형식 : YYYYMM	string(6)
items[].reportStatus	보고상태(미보고, 보고, 처리중)	string(10)
items[].lastReportTime	최종 보고 일시	string(20)
items[].suplyCnt	보고(공급) 건수	number
items[].reportCnt	보고 횟수	number
items[].lastReportDateLimit	최종 보고 가능 일	string(10)

- **응답예제**

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8

{
```

```
  "items": [
    {
      "lastReportTime": "2019-9-21 18:56:44",
      "reportCnt": 8,
      "suplyCnt": 1,
      "reportStatus": "보고",
      "suplyContStdmt": "201908",
      "lastReportDateLimit": "2019-09-30"
    },
    {
      "lastReportTime": "2019-10-22 12:26:46",
      "reportCnt": 1,
      "suplyCnt": 23,
      "reportStatus": "보고",
      "suplyContStdmt": "201909",
      "lastReportDateLimit": "2019-10-31"
    }
  ]
}
```

27. 공급내역 보고 현황 목록 조회

본 API는 **[공급내역 보고 및 재보고 처리]** API 통해 보고된 공급내역 보고 현황 목록을 조회합니다.

27.1. Request

```
GET /api/v1/supply-info/reports HTTP/1.1
Host: udiportal.mfds.go.kr
Authorization: Bearer {access_token}
Accept: application/json;charset=UTF-8
```

Access_token을 Authorization Header에 담아 GET 으로 요청합니다.

◆ **요청필드**

키	설명	구분	필수	타입
offset	페이징이 시작될 숫자	Query	O	number
limit	페이징에 담길 개수 (최대 100)	Query	O	number
suplyFlagCode	공급 구분 코드 (1:출고, 2:반품, 3:폐기, 4:임대, 5: 회수)	Query	X	string(1)
suplyTypeCode	공급 형태 코드 "1" : 제조・수입・판매(임대)에 공급 "2" : 의료기관에 공급 "3" : 약국개설자 또는 의약품도매상에 공급 "4" : 견본품, 기부용 또는 군납용 등으로 공급	Query	X	string(1)
startSuplyDate	공급일자(출고・반품・폐기・임대・회수일자) 검색 시작 일 형식 : YYYYMMDD	Query	O	string(6)
endSuplyDate	공급일자(출고・반품・폐기・임대・회수일자) 검색 종료 일 형식 : YYYYMMDD	Query	X	string(8)
meddevItemSeq	품목 일련번호	Query	X	number
itemName	품목명	Query	X	string(500)
typeName	형명(모델명)	Query	X	string(4000)
bcncEntpName	거래처 명칭	Query	X	string(100)
bcncCobFlagCode	거래처 영업 형태	Query	X	string(1000)

udidiList	고유식별자(UDI-DI) 코드 목록 URL encoding해야 하며 아래 문자는 별도로 URL encoding 전에 치환해줘야 합니다. "+" -> "%2B" "&" -> "%26"	Query	X	array
lotNo	로트번호	Query	X	string(100)
itemSeq	일련번호	Query	X	string(50)
manufYm	제조연월	Query	X	string(6)
useTmlmt	사용기한	Query	X	string(6)
isTraceManageTarget	추적 관리 대상 여부	Query	X	boolean

- **예제**

```
GET /api/v1/supply-info/reports HTTP/1.1
Host: udiportal.mfds.go.kr
Authorization: Bearer {access_token}
Accept: application/json;charset=UTF-8
```

- **예제코드 (Java, Spring)**

```
List<String> udiDiList = new ArrayList<String>();
udiDiList.add("08811344578771".replace("+", "%2B").replace("&", "%26"));
udiDiList.add("+7745412131213".replace("+", "%2B").replace("&", "%26"));
udiDiList.add("08912395678905".replace("+", "%2B").replace("&", "%26"));
UriComponentsBuilder uriBuilder = UriComponentsBuilder.fromHttpUrl("https://udiportal.mfds.go.kr/
api/v1/supply-info/reports");

MultiValueMap<String, String> params = new LinkedMultiValueMap<>();
params.add("offset", "1");
params.add("limit", "10");
params.add("startSuplyDate", "20190101");
params.add("endSuplyDate", "20191231");
params.add("meddevItemSeq", "20060048");
params.add("udidiList", udiDiList.toString());
params.add("isTraceManageTarget", "false");

HttpHeaders headers = new HttpHeaders();
String accessToken = TestUtil.obtainAccessToken(clientId, clientSecret);
headers.add(HttpHeaders.AUTHORIZATION, "Bearer " + accessToken);
headers.add(HttpHeaders.CONTENT_TYPE, MediaType.APPLICATION_FORM_URLENCODED_VALUE);
```

```
headers.add(HttpHeaders.ACCEPT, MediaType.APPLICATION_JSON_UTF8_VALUE);
HttpEntity httpEntity = new HttpEntity(headers);

uriBuilder.queryParams(params);
String uriString = uriBuilder.build(false).toUriString();
ResponseEntity<String> responseEntity = restTemplate.exchange(uriString,
                    HttpMethod.GET,
                    httpEntity,
                    String.class);

JSONObject body = new JSONObject(responseEntity.getBody());
```

27.2. Response

- **응답코드**

코드	설명
200	정상응답
400	입력값이 잘못되었을 경우
403	해당 정보에 접근 권한이 없을 경우
401	인증정보가 존재하지 않을 경우
500	처리 중 오류가 발생했을 경우

- **응답필드**

키	설명	타입
page	현재 페이지 번호	number
pageSize	페이지 크기	number
totalElements	총 갯수	number
items[].no	순번	number
items[].suplyEntpName	공급한 자 업체명	string(100)
items[].cobTypeName	공급한 자 업체 영업형태	string(1000)
items[].suplyFlagCode	공급 구분 코드 (1:출고, 2:반품, 3:폐기, 4:임대, 5:회수)	string(1)

items[].suplyTypeCode	공급 형태 코드 "1" : 제조・수입・판매(임대)에 공급 "2" : 의료기관에 공급 "3" : 약국개설자 또는 의약품도매상에 공급 "4" : 견본품, 기부용 또는 군납용 등으로 공급	string(1)
items[].udiDiCode	고유식별자(UDI-DI) 코드	string(60)
items[].itemName	품목명	string(500)
items[].meaClassNo	품목 분류번호	string(10)
items[].permitNo	품목 허가번호	string(50)
items[].typeName	모델명	string(500)
items[].lotNo	로트번호	string(100)
items[].itemSeq	일련번호	string(50)
items[].manufYm	제조연월 형식 : YYMMDD(표준코드 및 생산식별자 코드에 일 정보가 없는 경우 '00'으로 대체)	string(6)
items[].useTmlmt	사용기한 형식 : YYMMDD(표준코드 및 생산식별자 코드에 일 정보가 없는 경우 '00'으로 대체)	string(6)
items[].bcncEntpName	거래처 명	string(100)
items[].bcncCobTypeName	거래처 업종	string(1000)
items[].suplyDate	출고・반품・폐기・임대・회수 일자	string(10)
items[].suplyQty	출고・반품・폐기・임대・회수 수량 (공급구분이 '회수'일 때 suplyQty와 indvdlzSuplyQty 둘중 한 개 필드는 필수)	string
item[].indvdlzSuplyQty	포장내수량중 낱개 회수수량 (공급구분이 '회수'일 때 suplyQty와 indvdlzSuplyQty 둘중 한 개 필드는 필수)	string
items[].suplyUntpc	공급단가(VAT포함, 원)	string
items[].suplyAmt	공급금액(VAT포함, 원)	string
items[].isTraceManageTarget	추적 관리 대상 여부	boolean

◆ 응답예제

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
```

```
{
  "pageSize": 10,
  "page": 1,
  "items": [
    {
      "meaClassNo": "B03100.02",
      "no": 1,
      "suplyEntpName": "(주)네오메드",
      "suplyAmt": "68265",
      "suplyFlagCode": "1",
      "suplyTypeCode": "2",
      "typeName": "AR-1924B",
      "isTraceManageTarget": false,
      "manufYm": "130202",
      "permitNo": "수허 10-1038 호",
      "itemSeq": "XYZ456789012345678",
      "useTmlmt": "200202",
      "bcncEntpName": "개발팀",
      "itemName": "특수재질골절합용나사",
      "suplyQty": "555",
      "indvdlzSuplyQty": "12";
      "lotNo": "LOT123456789012345",
      "bcncCobTypeName": "판매(임대)업",
      "suplyUntpc": "123",
      "udiDiCode": "+4312312",
      "cobTypeName": "제조업",
      "suplyDate": "20190813"
    },
    {
      "meaClassNo": "B07090",
      "no": 2,
      "suplyEntpName": "(주)네오메드",
      "suplyAmt": "1500000",
      "suplyFlagCode": "1",
      "suplyTypeCode": "2",
      "typeName": "탄력밴드(JC-7500 UNI)",
      "isTraceManageTarget": false,
      "manufYm": null,
      "permitNo": "부산 제신 06-120 호",
      "itemSeq": null,
```

```
        "useTmlmt": "220501",
        "bcncEntpName": "개발팀",
        "itemName": "탄력밴드",
        "suplyQty": "100",
        "indvdlzSuplyQty": "31";
        "lotNo": "Q1231",
        "bcncCobTypeName": "판매(임대)업",
        "suplyUntpc": "15000",
        "udiDiCode": "08811344578771",
        "cobTypeName": "제조업",
        "suplyDate": "20190813"
    },
    {
        "meaClassNo": "B07090",
        "no": 3,
        "suplyEntpName": "(주)네오메드",
        "suplyAmt": "7500000",
        "suplyFlagCode": "1",
        "suplyTypeCode": "2",
        "typeName": "탄력밴드(JC-7500 UNI)",
        "isTraceManageTarget": false,
        "manufYm": null,
        "permitNo": "부산 제신 06-120 호",
        "itemSeq": null,
        "useTmlmt": "220501",
        "bcncEntpName": "바른아산정형외과의원",
        "itemName": "탄력밴드",
        "suplyQty": "150",
        "indvdlzSuplyQty": "23";
        "lotNo": "Q1231",
        "bcncCobTypeName": "요양기관",
        "suplyUntpc": "50000",
        "udiDiCode": "08811344578771",
        "cobTypeName": "제조업",
        "suplyDate": "20190813"
    },
    {
        "meaClassNo": "B07090",
        "no": 4,
        "suplyEntpName": "(주)네오메드",
```

```
        "suplyAmt": "35000",
        "suplyFlagCode": "1",
        "suplyTypeCode": "2",
        "typeName": "탄력밴드(JC-7500 UNI)",
        "isTraceManageTarget": false,
        "manufYm": null,
        "permitNo": "부산 제신 06-120 호",
        "itemSeq": null,
        "useTmlmt": "220501",
        "bcncEntpName": "더바른성모정형외과의원",
        "itemName": "탄력밴드",
        "suplyQty": "10",
        "indvdlzSuplyQty": "2";
        "lotNo": "Q1231",
        "bcncCobTypeName": "요양기관",
        "suplyUntpc": "3500",
        "udiDiCode": "08811344578771",
        "cobTypeName": "제조업",
        "suplyDate": "20190813"
    },
    {
        "meaClassNo": "B07090",
        "no": 5,
        "suplyEntpName": "(주)네오메드",
        "suplyAmt": "1000",
        "suplyFlagCode": "1",
        "suplyTypeCode": "2",
        "typeName": "탄력밴드(JC-7500 UNI)",
        "isTraceManageTarget": false,
        "manufYm": null,
        "permitNo": "부산 제신 06-120 호",
        "itemSeq": null,
        "useTmlmt": "220501",
        "bcncEntpName": "한국베드",
        "itemName": "탄력밴드",
        "suplyQty": "100",
        "indvdlzSuplyQty": "1";
        "lotNo": "Q1231",
        "bcncCobTypeName": "제조업",
        "suplyUntpc": "10",
```

```
      "udiDiCode": "08811344578771",
      "cobTypeName": "제조업",
      "suplyDate": "20190813"
    },
    {
      "meaClassNo": "B07090",
      "no": 6,
      "suplyEntpName": "(주)네오메드",
      "suplyAmt": "300000",
      "suplyFlagCode": "1",
      "suplyTypeCode": "2",
      "typeName": "탄력밴드(JC-7500 UNI)",
      "isTraceManageTarget": false,
      "manufYm": null,
      "permitNo": "부산 제신 06-120 호",
      "itemSeq": null,
      "useTmlmt": "220625",
      "bcncEntpName": "바른한방병원",
      "itemName": "탄력밴드",
      "suplyQty": "20",
      "indvdlzSuplyQty": "12";
      "lotNo": "Q1231",
      "bcncCobTypeName": "요양기관",
      "suplyUntpc": "15000",
      "udiDiCode": "08811344578771",
      "cobTypeName": "제조업",
      "suplyDate": "20190813"
    },
    {
      "meaClassNo": "B05010",
      "no": 7,
      "suplyEntpName": "(주)네오메드",
      "suplyAmt": "120",
      "suplyFlagCode": "1",
      "suplyTypeCode": "2",
      "typeName": "네오메드 컴팩트 써포트 JC-104",
      "isTraceManageTarget": false,
      "manufYm": null,
      "permitNo": "부산 제신 99-696 호",
      "itemSeq": "0000241",
```

```
        "useTmlmt": "160109",
        "bcncEntpName": "개발팀",
        "itemName": "부목",
        "suplyQty": "10",
        “indvdlzSuplyQty”: “8”;
        "lotNo": null,
        "bcncCobTypeName": "판매(임대)업",
        "suplyUntpc": "12",
        "udiDiCode": "A9999XYZ100T0479",
        "cobTypeName": "제조업",
        "suplyDate": "20190813"
    },
    {
        "meaClassNo": "B07090",
        "no": 8,
        "suplyEntpName": "(주)네오메드",
        "suplyAmt": null,
        "suplyFlagCode": "2",
        "suplyTypeCode": "3",
        "typeName": "탄력밴드(JC-7500 UNI)",
        "isTraceManageTarget": false,
        "manufYm": "191001",
        "permitNo": "부산 제신 06-120 호",
        "itemSeq": null,
        "useTmlmt": null,
        "bcncEntpName": "유한킴벌리(주)대전",
        "itemName": "탄력밴드",
        "suplyQty": "9",
        “indvdlzSuplyQty”: “4”;
        "lotNo": "Q1231",
        "bcncCobTypeName": "수입업",
        "suplyUntpc": null,
        "udiDiCode": "08811344578771",
        "cobTypeName": "제조업",
        "suplyDate": "20190813"
    },
    {
        "meaClassNo": "B07090",
        "no": 9,
        "suplyEntpName": "(주)네오메드",
```

```
        "suplyAmt": null,
        "suplyFlagCode": "2",
        "suplyTypeCode": "3",
        "typeName": "탄력밴드(JC-7500 UNI)",
        "isTraceManageTarget": false,
        "manufYm": null,
        "permitNo": "부산 제신 06-120 호",
        "itemSeq": null,
        "useTmlmt": "220501",
        "bcncEntpName": "바른아산정형외과의원",
        "itemName": "탄력밴드",
        "suplyQty": "50",
        "indvdlzSuplyQty": "16";
        "lotNo": "Q1231",
        "bcncCobTypeName": "요양기관",
        "suplyUntpc": null,
        "udiDiCode": "08811344578771",
        "cobTypeName": "제조업",
        "suplyDate": "20190813"
    },
    {
        "meaClassNo": "B07090",
        "no": 10,
        "suplyEntpName": "(주)네오메드",
        "suplyAmt": null,
        "suplyFlagCode": "2",
        "suplyTypeCode": "3",
        "typeName": "탄력밴드(JC-7500 UNI)",
        "isTraceManageTarget": false,
        "manufYm": null,
        "permitNo": "부산 제신 06-120 호",
        "itemSeq": null,
        "useTmlmt": "220501",
        "bcncEntpName": "한국베드",
        "itemName": "탄력밴드",
        "suplyQty": "2",
        "indvdlzSuplyQty": "2";
        "lotNo": "Q1231",
        "bcncCobTypeName": "제조업",
        "suplyUntpc": null,
```

```
        "udiDiCode": "08811344578771",
        "cobTypeName": "제조업",
        "suplyDate": "20190813"
      }
   ],
   "totalElements": 36
}
```

28. 공급내역 보고 및 재보고 처리

등록된 공급내역 보고자료를 기준으로 공급내역 보고 및 재보고 처리를 수행합니다. 공급내역 보고는 최종 보고 가능일 이내에 처리가 완료되어야 정상적인 공급내역 보고입니다.

28.1. Request

```
POST /api/v1/supply-info/manages/report/{suplyContStdmt} HTTP/1.1
Host: udiportal.mfds.go.kr
Authorization: Bearer {access_token}
Content-Type: application/json;charset=UTF-8
```

Access_token을 Authorization Header에 담아 요청 내용을 JSON string으로 만들어 POST 로 요청합니다.

◆ **요청필드**

키	설명	구분	필수	타입
suplyContStdmt	공급내역 보고 기준 월 형식 : YYYYMM	PathVariable	O	string(6)

◆ **예제**

```
POST /api/v1/supply-info/manages/report/201909 HTTP/1.1
Host: udiportal.mfds.go.kr
Authorization: Bearer {access_token}
Content-Type: application/json;charset=UTF-8
```

◆ **예제코드 (Java, Spring)**

```
String suplyContStdmt = "201909";
UriComponentsBuilder uriBuilder = UriComponentsBuilder.fromHttpUrl("https://udiportal.mfds.go.kr/
api/v1/supply-info/manages/report")
                                .pathSegment(suplyContStdmt);
HttpHeaders headers = new HttpHeaders();
String accessToken = TestUtil.obtainAccessToken(clientId, clientSecret);
headers.add(HttpHeaders.AUTHORIZATION, "Bearer " + accessToken);
headers.add(HttpHeaders.CONTENT_TYPE, MediaType.APPLICATION_JSON_UTF8_VALUE);
HttpEntity httpEntity = new HttpEntity(headers);

// when
ResponseEntity<String> responseEntity = restTemplate.exchange(uriBuilder.toUriString(),
```

```
                HttpMethod.POST,
                httpEntity,
                String.class);
```

28.2. Response

◆ **응답코드**

코드	설명
200	정상응답
400	입력값이 잘못되었을 경우
403	해당 정보에 접근 권한이 없을 경우
401	인증정보가 존재하지 않을 경우
500	처리 중 오류가 발생했을 경우

◆ **응답필드**

코드(Response Code)만 응답합니다.

◆ **오류 응답 예제 – 공급내역 최종 보고 가능일이 지난 경우**

```
HTTP/1.1 400 BAD_REQUEST
Content-Type: application/json;charset=UTF-8

{"message":"공급내역 보고 가능일이 지났습니다.","status":400,"errors":[]}
```