

# 디플샵 Open API v2

공급사 상품, 주문 관리 API 가이드 (수정일: 2026-03-20)

## 목차

1. 인증 체계
2. 공통 에러 응답
3. API 목록
  - Access Token 발급
  - Access Token 갱신
  - 공급사 상품 목록 조회
  - 공급사 SKU 목록 조회
  - 공급사 주문 조회
  - 공급사 상품준비중 처리
  - 공급사 파일오류 처리
  - 공급사 배송중 처리
  - 공급사 배송정보 수정
  - 공급사 생산불가 처리
4. 부록

## 인증 체계

디폴샵 API는 3종류의 키/토큰을 사용합니다.

구분	형식	유효기간	설명
API Key	UUID 기반 문자열	90일	설정 화면에서 발급 - Access Token 발급 시 사용
Access Token	base64url 랜덤 문자열	2시간	API 호출 시 인증 헤더에 사용
Refresh Token	base64url 랜덤 문자열	2주	Access Token 만료 시 갱신용

### API Key

- 설정 화면에서 발급하며, 발급 시 **1회만 확인** 가능합니다. 이후에는 재확인할 수 없으므로 반드시 안전한 곳에 보관하세요.
- 유효기간은 **90일**이며, 만료 30일 전부터 설정 화면에 갱신 알림이 표시됩니다.
- 만료된 API Key로는 토큰 발급이 불가능합니다. 설정 화면에서 갱신하거나 재발급해야 합니다.
- 갱신 시 키 값은 변경되지 않으며 만료일만 연장됩니다. 재발급 시 기존 키는 폐기됩니다.

## 인증 흐름

1. **Access Token 발급** - API Key를 사용하여 `POST /keys/token` 으로 Access Token과 Refresh Token을 발급받습니다.
2. **API 호출** - 발급받은 Access Token을 `Authorization: Bearer {access_token}` 헤더에 담아 API를 호출합니다.
3. **토큰 갱신** - Access Token 만료 시 Refresh Token을 사용하여 `POST /keys/token/refresh` 로 갱신합니다. Access Token과 Refresh Token이 모두 새로 발급됩니다.

## 만료 시 대응

상황	대응
Access Token 만료	Refresh Token으로 갱신 ( <code>POST /keys/token/refresh</code> )
Refresh Token 만료	API Key로 토큰 재발급 ( <code>POST /keys/token</code> )
API Key 만료	설정 화면에서 갱신 (만료 30일 전부터 알림)

## 보안 주의사항

- API Key는 절대 외부에 노출하지 마세요.
- API Key 유출이 의심되는 즉시 재발급 받으세요.
- Access Token은 서버 측에서만 사용하고, 클라이언트(브라우저)에 노출하지 마세요.

## 공통 에러 응답

모든 API는 오류 발생 시 `{ "error": "에러 메시지" }` 형식으로 응답합니다.

**HTTP 상태 코드:** `400` 잘못된 요청 / `401` 인증 실패 / `404` 리소스 없음 / `500` 서버 내부 오류

## 1. Access Token 발급 API

API Key를 사용하여 Access Token과 Refresh Token을 발급합니다.

```
POST https://dp1.shop/api/v2/keys/token
```

### Headers

Name	Type	Required	Description
Authorization	string	<input checked="" type="checkbox"/>	Bearer {API_KEY} (dp1_xxxx 형식)

응답 - 성공 (200 OK)

```
{
  "access_token": "Jge1SEoRWKVUXKRgGZjUPDqw8nB3xYhT2vLmA5fK0wQ",
  "access_token_expires_at": "2026-03-18T12:00:00.000Z",
  "refresh_token": "PmXqZ8kLnRfWs3tYcVbN7uJgFaH9xN3wQ5kGTFp1yA",
  "refresh_token_expires_at": "2026-04-01T10:00:00.000Z",
  "issued_at": "2026-03-18T10:00:00.000Z"
}
```

필드	설명
access_token	Access Token (2시간 유효)
access_token_expires_at	Access Token 만료 시각
refresh_token	Refresh Token (2주 유효)
refresh_token_expires_at	Refresh Token 만료 시각
issued_at	발급 시각

예러: { "error": "API key has expired" } / { "error": "API key not found or inactive" }

참고: 기존 토큰이 있으면 새 토큰으로 교체됩니다.

## 2. Access Token 갱신 API

Refresh Token을 사용하여 새 Access Token과 Refresh Token을 발급합니다.

```
POST https://dpl.shop/api/v2/keys/token/refresh
```

### Headers

Name	Type	Required	Description
Authorization	string	✔	Bearer {REFRESH_TOKEN}

응답 - 성공 (200 OK)

```
{
  "access_token": "Ht7nKpWqX2vRsLmYcBdE4uJgFaZ9xN3wM8kQ5TfP1yA",
  "access_token_expires_at": "2026-03-18T14:00:00.000Z",
  "refresh_token": "Zw4tRpMnK8vXsLqY2cBdE7uJgFaH9xN3wQ5kGTfP1yA",
  "refresh_token_expires_at": "2026-04-01T12:00:00.000Z",
  "issued_at": "2026-03-18T12:00:00.000Z"
}
```

에러: { "error": "Refresh token has expired" } / { "error": "Access token has expired" }

참고: Refresh Token이 만료된 경우 API Key로 /keys/token 을 다시 호출해야 합니다.

### 3. 공급사 상품 목록 조회 API

공급사에 등록된 상품 목록을 조회합니다.

```
GET https://dpl.shop/api/v2/partner/product
```

#### Headers

Name	Type	Required	Description
Authorization	string	✔	Bearer {ACCESS_TOKEN}

#### Query Parameters

Name	Type	Required	Description
name	string	✘	상품명 검색 (부분 일치)
page	integer	✘	페이지 번호 (기본값: 1)
limit	integer	✘	한 페이지당 조회 수 (기본값: 20, 최대: 100)

#### 요청 예제

```
GET https://dpl.shop/api/v2/partner/product?name=텀블러&page=1&limit=20
```

#### 응답 - 성공 (200 OK)

```
{
  "data": [
    {
      "seqno": 1,
      "name": "텀블러 A",
      "selStatus": "00",
      "supType": "00",
      "designType": "N",
      "optionCategory": null,
      "createdAt": "2024-01-01T00:00:00.000Z",
      "updatedAt": "2024-01-15T12:00:00.000Z"
    }
  ],
  "pagination": { "page": 1, "limit": 20, "totalCount": 45, "totalPages": 3 },
  "timeStamp": "2024-01-15T12:00:00.000Z"
}
```

## 4. 공급사 SKU 목록 조회 API

공급사에 등록된 상품의 SKU 목록을 조회합니다.

```
GET https://dpl.shop/api/v2/partner/product/sku
```

### Headers

Name	Type	Required	Description
Authorization	string	✔	Bearer {ACCESS_TOKEN}

### Query Parameters

Name	Type	Required	Description
productSeqno	integer	✔	조회할 상품 번호
sku	string	✘	SKU 코드 검색 (부분 일치)
page	integer	✘	페이지 번호 (기본값: 1)
limit	integer	✘	한 페이지당 조회 수 (기본값: 100, 최대: 1000)

### 요청 예제

```
GET https://dpl.shop/api/v2/partner/product/sku?productSeqno=1&page=1&limit=100
```

### 응답 - 성공 (200 OK)

```
{
  "data": [
    {
      "sku": "A100100001",
      "price": 10000,
      "priceDelv": 3000,
      "stock": 100,
      "selStatus": "00",
      "options": [
        { "name": "색상", "value": "블랙" },
        { "name": "사이즈", "value": "L" }
      ],
      "createdAt": "2024-01-01T00:00:00.000Z",
      "updatedAt": "2024-01-15T12:00:00.000Z",
      "product": { "seqno": 1, "name": "텀블러 A" }
    }
  ],
  "pagination": { "page": 1, "limit": 100, "totalCount": 250, "totalPages": 3 },
  "timeStamp": "2024-01-15T12:00:00.000Z"
}
```

## 5. 공급사 주문 조회 API

공급사 주문 내역을 조회합니다.

```
GET https://dpl.shop/api/v2/partner/order
```

### Headers

Name	Type	Required	Description
Authorization	string	✓	Bearer {ACCESS_TOKEN}

### Query Parameters

Name	Type	Required	Description
startDate	string<date-time>	✓	조회 시작일시 (ISO 8601 형식)
endDate	string<date-time>	✓	조회 종료일시 (ISO 8601 형식)
rangeType	string	✗	조회 기준 필드 (아래 참고, 기본값: <code>createdAt</code> )
orderStatus	string	✗	조회할 주문 상태 (콤마로 구분)
payStatus	string	✗	조회할 결제 상태 (콤마로 구분)

**rangeType** 값: `createdAt` 주문일 / `deliveredAt` 배송완료일 / `departedAt` 출고일

제약사항: 조회 기간 최대 90일 / startDate < endDate / 승인 완료된 주문만 조회 가능

조회 가능한 주문 상태: 15 발주대기 / 20 접수대기 / 30 상품준비중 / 35 파일오류 / 60 배송중 / 70 배송완료 / 80 구매확정 / 85 반품신청 / 90 취소 / 95 생산불가

조회 가능한 결제 상태: 00 결제대기중 / 10 입금대기 / 80 결제완료 / 90 결제실패 / 99 결제취소

요청 예제

```
GET https://dpl.shop/api/v2/partner/order?startDate=2024-01-01T00:00:00.000Z&endDate=2024-03-31T23:59:59.999Z&orderStatus=30,60&rangeType=createdAt
```

## 응답 타입

```
{ data: Order[]; timeStamp: Date; }

type Order = {
  orderNum: string; name: string; payType: string; orderStatus: string;
  orderId: string; orderName: string; orderTel: string; orderEmail: string;
  orderDate: Date; deliveryName: string; deliveryTel: string;
  deliveryAddr1: string; deliveryAddr2: string; deliveryZip: string;
  deliveryComment: string; payStatus: string; paymentDate: Date;
  shop: { name: string }; data: OrderDetail[];
}

type OrderDetail = {
  quantity: number; orderDetailNum: string; price: number;
  options: { name: string; value: string }[];
  shopDetailComment: string | null; deliveryPrice: number;
  orderSku: string | null; status: string; payStatus: string;
  deliveryCmp: string | null; deliveryNum: string | null;
  departedAt: Date | null; deliveredAt: Date | null;
  confirmedAt: Date | null; addOptions: object;
}
```

## 응답 예제

```
{
  "data": [
    {
      "orderNum": "20241119-0000029-01",
      "name": "장패드 외 1건",
      "payType": "10",
      "orderStatus": "30",
      "orderId": "user123",
      "orderName": "홍길동",
      "orderTel": "010-1234-5678",
      "orderEmail": "user@example.com",
      "orderDate": "2024-01-01T10:30:00.000Z",
      "deliveryName": "김철수",
      "deliveryTel": "010-9876-5432",
      "deliveryAddr1": "서울시 강남구 테헤란로 123",
      "deliveryAddr2": "3층",
      "deliveryZip": "06234",
      "deliveryComment": "부재 시 문앞에 놔주세요",
      "payStatus": "80",
      "paymentDate": "2024-01-01T10:35:00.000Z",
      "shop": { "name": "테스트 쇼핑몰" },
      "data": [
        {
          "quantity": 2,
          "orderDetailNum": "20241119-0000029-01",
          "price": 25000,
          "options": [{ "name": "종류", "value": "전체커스텀 장패드" }],
          "shopDetailComment": "빠른 배송 부탁드립니다",
          "deliveryPrice": 3000,
          "orderSku": "A10010012345",
          "status": "30",
          "payStatus": "80",
          "deliveryCmp": null,
          "deliveryNum": null,
          "departedAt": null,
          "deliveredAt": null,
          "confirmedAt": null,
          "addOptions": {}
        }
      ]
    }
  ],
  "timeStamp": "2024-01-15T10:30:00.000Z"
}
```

## 에러 응답

```
{ "error": "startDate와 endDate는 필수입니다" }  
{ "error": "사용 불가능한 orderStatus가 포함되어 있습니다..." }  
{ "error": "조회 기간은 최대 90일까지 가능합니다" }
```

## 6. 공급사 상품준비중 처리 API

주문 상태를 \*\*상품준비중(30)\*\*으로 변경합니다.

PATCH <https://dpl.shop/api/v2/partner/order/preparation>

Name	Type	Required	Description
Authorization	string	✓	Bearer {ACCESS_TOKEN} (Header)
Content-Type	string	✓	application/json (Header)
orderDetailNum	string	✓	주문상세번호 (Body)

상태 변경 가능 조건: 주문 상태 15 (발주대기), 20 (접수대기), 25 (발주실패) / 결제 상태 70 (내부결제완료) 또는 80 (결제완료)

요청 예제: { "orderDetailNum": "20260225-000001-01" }

응답 - 성공 (200 OK)

```
{
  "success": true,
  "data": {
    "quantity": 2,
    "orderDetailNum": "20260225-000001-01",
    "price": 25000,
    "options": [{ "name": "종류", "value": "전체커스텀 장패드" }],
    "shopDetailComment": "빠른 배송 부탁드립니다",
    "deliveryPrice": 3000,
    "orderSku": "A100100001",
    "status": "30",
    "payStatus": "80",
    "deliveryCmp": null,
    "deliveryNum": null,
    "departedAt": null,
    "deliveredAt": null,
    "confirmedAt": null,
    "addOptions": {}
  },
  "timeStamp": "2024-01-15T12:00:00.000Z"
}
```

에러: { "error": "orderDetailNum은 필수 값 입니다" } / { "error": "주문이 존재하지 않습니다." } / { "error": "처리 가능한 주문 상태가 아닙니다..." }

참고: 판매종료 상품인 경우 에러를 반환합니다.

## 7. 공급사 파일오류 처리 API

주문 상태를 \*\*파일오류(35)\*\*로 변경합니다. 파일오류 사유를 반드시 입력해야 합니다.

PATCH <https://dpl.shop/api/v2/partner/order/error>

Name	Type	Required	Description
Authorization	string	✔	Bearer {ACCESS_TOKEN} (Header)
Content-Type	string	✔	application/json (Header)
orderDetailNum	string	✔	주문상세번호 (Body)
fileErrorReason	string	✔	파일오류 사유 (Body) - 고객에게 안내될 오류 내용

상태 변경 가능 조건: 주문 상태 15 (발주대기), 20 (접수대기), 25 (발주실패), 30 (상품준비중) / 결제 상태 70 (내부결제한료) 또는 80 (결제완료)

요청 예제:

```
{
  "orderDetailNum": "20260225-000001-01",
  "fileErrorReason": "해상도가 낮아 인쇄 불가"
}
```

응답 - 성공 (200 OK)

```
{
  "success": true,
  "data": {
    "quantity": 2,
    "orderDetailNum": "20260225-000001-01",
    "price": 25000,
    "options": [{ "name": "종류", "value": "전체커스텀 장패드" }],
    "shopDetailComment": null,
    "deliveryPrice": 3000,
    "orderSku": "A10010012345",
    "status": "35",
    "payStatus": "80",
    "deliveryCmp": null,
    "deliveryNum": null,
    "departedAt": null,
    "deliveredAt": null,
    "confirmedAt": null,
    "addOptions": {}
  },
  "timeStamp": "2024-01-15T12:00:00.000Z"
}
```

## 8. 공급사 배송중 처리 API

주문 상태를 \*\*배송중(60)\*\*으로 변경하고 배송정보를 등록합니다. 배송 추적은 디플샵에서 자동 처리됩니다.

PATCH <https://dpl.shop/api/v2/partner/order/transit>

Name	Type	Required	Description
Authorization	string	✓	Bearer {ACCESS_TOKEN} (Header)
Content-Type	string	✓	application/json (Header)
orderDetailNum	string	✓	주문상세번호 (Body)
deliveryCmp	string	✓	배송회사 코드 (Body)
deliveryNum	string	✓	송장번호 - 숫자 10자리 이상 (Body)

상태 변경 가능 조건: 주문 상태 30 (상품준비중) / 결제 상태 70 (내부결제완료) 또는 80 (결제완료)

배송회사 코드: 01 CJ대한통운 / 02 우체국택배 / 03 한진택배 / 04 로젠택배

송장번호 규칙: 숫자만 10자리 이상 / 하이픈 자동 제거 / 영문·특수문자 포함 시 에러

## 요청 예제

```
{
  "orderDetailNum": "20260225-000001-01",
  "deliveryCmp": "01",
  "deliveryNum": "1234567890"
}
```

## 응답 - 성공 (200 OK)

```
{
  "success": true,
  "data": {
    "quantity": 2,
    "orderDetailNum": "20260225-000001-01",
    "price": 25000,
    "options": [{ "name": "종류", "value": "전체커스텀 장패드" }],
    "shopDetailComment": "빠른 배송 부탁드립니다",
    "deliveryPrice": 3000,
    "orderSku": "A10010012345",
    "status": "60",
    "payStatus": "80",
    "deliveryCmp": "01",
    "deliveryNum": "1234567890",
    "departedAt": "2024-01-15T12:00:00.000Z",
    "deliveredAt": null,
    "confirmedAt": null,
    "addOptions": {}
  },
  "timeStamp": "2024-01-15T12:00:00.000Z"
}
```

## 에러 응답

```
{ "error": "배송중 상태로 변경시 deliveryCmp와 deliveryNum은 필수입니다" }
{ "error": "유효하지 않은 배송회사 코드입니다..." }
{ "error": "유효하지 않은 송장번호입니다. 숫자만 10자리 이상이어야 합니다" }
```

## 9. 공급사 배송정보 수정 API

배송중 상태에서 배송 정보(배송회사, 송장번호)를 수정합니다.

PATCH <https://dpl.shop/api/v2/partner/order/delivery>

Name	Type	Required	Description
Authorization	string	✓	Bearer {ACCESS_TOKEN} (Header)
Content-Type	string	✓	application/json (Header)
orderDetailNum	string	✓	주문상세번호 (Body)
deliveryCmp	string	✓	배송회사 코드 (Body)
deliveryNum	string	✓	송장번호 - 숫자 10자리 이상 (Body)

상태 변경 가능 조건: 주문 상태 60 (배송중) / 결제 상태 70 (내부결제완료) 또는 80 (결제완료)

요청 예제: { "orderDetailNum": "20241119-0000011-01", "deliveryCmp": "03", "deliveryNum": "9876543210" }

응답 - 성공 (200 OK)

```
{
  "success": true,
  "data": {
    "quantity": 2,
    "orderDetailNum": "20241119-0000011-01",
    "price": 25000,
    "options": [{ "name": "종류", "value": "전체커스텀 장패드" }],
    "shopDetailComment": "빠른 배송 부탁드립니다",
    "deliveryPrice": 3000,
    "orderSku": "A10010012345",
    "status": "60",
    "payStatus": "80",
    "deliveryCmp": "03",
    "deliveryNum": "9876543210",
    "departedAt": "2024-01-15T09:00:00Z",
    "deliveredAt": null,
    "confirmedAt": null,
    "addOptions": {}
  },
  "timeStamp": "2024-01-15T10:30:00Z"
}
```

에러: { "error": "orderDetailNum은 필수 값 입니다" } / { "error": "유효하지 않은 배송회사 코드입니다..." } / { "error": "주문이 존재하지 않습니다." }

## 10. 공급사 생산불가 처리 API

주문 상태를 \*\*생산불가(95)\*\*로 변경합니다. 주의: 결제완료 상태인 경우 자동으로 결제를 취소합니다.

PATCH <https://dpl.shop/api/v2/partner/order/unable>

Name	Type	Required	Description
Authorization	string	✓	Bearer {ACCESS_TOKEN} (Header)
Content-Type	string	✓	application/json (Header)
orderDetailNum	string	✓	주문상세번호 (Body)

상태 변경 가능 조건: 주문 상태 15 (발주대기), 20 (접수대기), 25 (발주실패), 30 (상품준비중) / 결제 상태 70 (내부결제완료) 또는 80 (결제완료)

요청 예제: { "orderDetailNum": "20241119-0000011-01" }

응답 - 성공 (200 OK)

```
{
  "success": true,
  "data": {
    "quantity": 2,
    "orderDetailNum": "20241119-0000011-01",
    "price": 25000,
    "options": [{ "name": "종류", "value": "전체커스텀 장패드" }],
    "shopDetailComment": "빠른 배송 부탁드립니다",
    "deliveryPrice": 3000,
    "orderSku": "A10010012345",
    "status": "95",
    "payStatus": "99",
    "deliveryCmp": null,
    "deliveryNum": null,
    "departedAt": null,
    "deliveredAt": null,
    "confirmedAt": null,
    "addOptions": {}
  },
  "timeStamp": "2024-01-15T12:00:00Z"
}
```

## 부록

### 판매 상태 코드 (selStatus)

코드	설명	코드	설명
00	임시저장	60	판매안함
05	SKU변경사항있음	67	품질예정
10	승인요청	68	품질
20	승인완료	69	판매중지예정
30	승인거절	70	판매중지
40	수정요청	71	판매중지요청
50	판매함	95	삭제
		98	판매종료요청
		99	판매종료

### 공급 유형 코드 (supType)

00 전체 / 10 그룹

### 디자인 유형 코드 (designType)

N 없음 / S 쇼핑몰 시안제출 / G 파일만 / E 에디터만 / F 판매사 파일 / T 판매사 템플릿 / A 후편집

## 주문 상태 코드 (orderStatus / status)

코드	설명	코드	설명
00	승인대기	60	배송중
10	결제대기	70	배송완료
15	발주대기	80	구매확정
20	접수대기	85	반품신청
25	발주실패	90	취소
30	상품준비중	95	생산불가
35	파일오류	99	삭제

## 결제 상태 코드 (payStatus)

코드	설명	코드	설명
00	결제대기중	60	출금완료
10	입금대기	70	내부결제완료
20	추가입금대기	80	결제완료
30	입금완료(수동)	90	결제실패
40	입금완료(자동)	99	결제취소
50	출금전		

## 기타 코드

결제 수단 (payType): 10 카드결제 / 20 무통장입금 / 30 계좌이체 / 40 후불결제

배송회사 (deliveryCmp): 01 CJ대한통운 / 02 우체국택배 / 03 한진택배 / 04 로젠택배

날짜 형식: ISO 8601 — YYYY-MM-DDTHH:mm:ss.sssZ (예: 2024-01-01T00:00:00.000Z)