

오너클랜 API

Version

v0.10.13

Document version

v0.10.13.0

Changes

v0.11.2 -> v0.11.4

- 상품 등록(createItem) 및 수정(updateItem)시 상품에 대한 검색 키워드(Item.searchKeywords) 등록 제한 개수 변경.
 - 기존: 40개 / 변경: 20개

v0.10.10 -> v.10.13

- 송장번호 입력일 기준으로 주문 검색하는 기능 추가.
 - allOrders 파라미터에 shippedAfter, shippedBefore 추가.
 - shippedAfter: 해당 시점 이후에 송장번호가 입력된 주문건만 검색.
 - shippedBefore: 해당 시점 이전에 송장번호가 입력된 주문건만 검색.

v0.9.8 -> v0.10.10

- C/S WRITE API 추가.
 - 추가 내역:
 - createVendorQnaArticle - 1:1 문의 작성.
 - createDirectTransferSellerQnaArticleReply - 이관된 판매사 1:1 문의에 답변.
 - 자세한 사용 방법은 각 API method 항목 참조.

v0.9.4 -> v0.9.8

- 상품 등록(createItem) 및 수정(updateItem) 시 상품 상세정보(Item.content)에 iframe 태그 입력 불가능.

v0.9.0 -> v0.9.4

- 상품명 길이 제한 적용
 - 제한 내용은 오너클랜 웹사이트와 동일: 한글 2bytes 기준(encoding: euc-kr)으로 최대 50bytes.
 - createItem, updateItem에 모두 적용.

v0.9.0 -> v0.9.1

- 다중 목록이미지 지원.

- `createItem`, `updateItem`: `Item.images` 필드에 입력하는 URL이 1개인 경우 해당 URL의 이미지가 대표 목록이미지로 지정. 2개 이상인 경우 첫 번째 URL의 이미지가 대표 목록이미지로, 나머지 URL의 이미지가 추가 목록이미지로 저장.
- `item`, `allItems`: `Item.images`에 제공되는 링크 중 첫 번째 링크가 대표 목록이미지, 나머지가 추가 목록이미지.

v0.8.0 -> v0.9.0

- C/S READ API 추가.
 - 추가 내역:
 - `vendorQnaArticle` / `allVendorQnaArticles` - 공급사 1:1 문의 게시판 글 조회.(단일 / 복수)
 - 자세한 사용 방법은 각 API method 항목 참조.

v0.7.59 -> v0.8.0

- Order Update API 추가.
 - 추가 내역:
 - `checkOrder` - 주문을 발송 준비중 상태로 변경.
 - `setTrackingInfo` - 주문에 대해 운송장 번호를 입력하고 발송 완료 상태로 변경.
 - `setOrderProductsTrackingInfo` - 주문 제품에 대해 각각 운송장 번호를 변경.
 - `acceptOrderCancellationRequest` - 주문 취소 요청을 수락.
 - `rejectOrderCancellationRequest` - 주문 취소 요청을 거절.
 - 자세한 사용 방법은 각 API method 항목 참조.
- GUI endpoint(GraphiQL -> GraphQL Playground) 변경
 - 기존에는 별도의 URL이 있었으나, v0.8부터는 GraphQL API Endpoint와 동일한 URL을 사용하고 브라우저로 접속시 GUI 사용 가능.
- `allItems` 쿼리의 `sortBy` 기준 추가.
 - 기존의 `keyAsc`, `keyDesc` 동작 버그 개선 및 상품코드 기준 정렬에서 내부 자체 기준 정렬로 변경.
 - 변경된 `keyAsc`, `keyDesc`는 반환하는 Item의 순서를 항상 보장함.
 - 상품코드(Item.key)로 정렬하는 `itemKeyAsc`, `itemKeyDesc` 추가. 변경 이전의 `keyAsc`, `keyDesc`와 동일함.
- 기타 버그 개선.

v0.7.56 -> v0.7.59

- 오픈마켓 카테고리 등록 기능 추가.
 - 오픈마켓 SCM에서 상품을 등록할 때 "마켓 카테고리 설정"에서 설정하는 기능과 동일.
 - `Item.metadata.externalCategories` 필드로 설정하며, 자세한 내용은 `createItem` method의 설명을 참조.

v0.7.52 -> v0.7.56

- 상품 상단문구(판매자 한정 설명)(`Item.sellerOnlyContent`) 조회 불가로 변경
 - 기존에는 판매자 한정 설명을 상품 등록 및 수정시에만 사용할 수 없게 했으나, 오픈마켓 웹사이트 업데이트에 맞추어 조회까지 불가능하도록 변경.
 - 앞으로 사용하지 않음.
- `allItems` 쿼리 정렬 조건(`sortBy`) 관련 변경

- `default`는 기존에 오너클랜 웹사이트 내 검색 결과와 동일한 순서를 보장하기 위한 정렬 기준이었으나, 해당 정렬이 DB 부하를 많이 발생시켜서 일반적인 조건에서는 `registerDateDesc`와 동일하게 변경. 다만 `search` 파라미터를 같이 사용하는 경우에는 기존과 동일하게 동작.
- `allItems`의 일부 쿼리 형태에 대해 Quota 제약 강화
 - `sortBy`에서 `priceAsc`, `priceDesc`를 사용하는 경우
 - `search` 파라미터를 사용하는 경우
 - `vprdcde` 없이 `vcode`만 명시하는 경우
- 상품 등록 및 수정시에 옵션명, 옵션값, 공급사 관리코드값에 대한 trim 처리.
 - `Item.options.optionAttributes.name`, `Item.options.optionAttributes.value`, `Item.metadata.vcode`, `Item.metadata.vprdcde`, `Item.options.metadata.vcode`, `Item.options.metadata.vprdcde`의 필드들에 대해서 입력된 값이 있다면 모두 trim처리하고 입력된 값 중간에 `\r`, `\n`, `\t` 문자는 모두 삭제 처리.
- 기타 버그 수정.

v0.7.52 -> v0.7.54

- 상품 등록 및 수정시 판매자 한정 설명(`Item.sellerOnlyContent`) 설정 불가능.
 - 기존에 등록한 상품의 `Item.sellerOnlyContent`는 조회할 수 있으나, 오너클랜 웹사이트에서 노출하지 않게 변경되는 경우 API에서도 조회 불가능으로 변경될 예정.

v0.7.45 -> v0.7.52

- 카테고리에 전체 이름 추가.
 - 기존에는 카테고리의 현재 레벨에서의 이름(ex: "청소기")만을 `Category.name` 필드로 반환했으나, `Category.fullName` 필드에서 전체 이름(ex: "디지털/가전>생활가전>청소기")를 반환하도록 추가.
- 오픈마켓 연동 카테고리 정보에 각 카테고리에 대한 전체 이름 추가. 정보가 없는 경우 `null` 또는 빈 문자열 반환.
 - `Item.metadata.coupangCategoryFullName`
 - `Item.metadata.auctionCategoryFullName`
 - `Item.metadata.gmarketCategoryFullName`
 - `Item.metadata.st11CategoryFullName`
 - `Item.metadata.interparkCategoryFullName`
 - `Item.metadata.smartstoreCategoryFullName`
 - `Item.metadata.playautoCategoryFullName`
 - `Item.metadata.esellersCategoryFullName`

v0.7.42 -> v0.7.45

- 외부 카테고리 정보 제공시 플레이오토, 이셀러스 카테고리 코드 추가 제공.
- `Item.options.status`에서 `null`로 표시되는 경우 개선.

v0.7.39 -> v0.7.42

- 오너클랜 카테고리 체계 개편에 따른 카테고리 코드 변경
 - 카테고리 코드가 쓰이는 모든 필드 및 검색 조건은 신규 카테고리 정보로 변경.

v0.7.35 -> v0.7.39

- v0.7.28 -> v0.7.35

- v0.7.27 -> v0.7.28

- v0.7.18 -> v0.7.27

- v0.7.12 -> v0.7.18

- v0.7.11 -> v0.7.12

- v0.7.10 -> v0.7.11

- 4 / 76

- `true`이면 입력한 상품명, 키워드, 옵션명, 옵션값에 치환 규칙(매뉴얼 하단의 *특수문자 치환 정책* 섹션을 참고)을 적용한 후에 입력값 검사 및 등록, 수정 진행.
- `false`면 치환 규칙을 적용하지 않고 입력값 검사 및 등록, 수정 진행.
- 명시하지 않으면 `false`로 간주.

v0.7.7 -> v0.7.10

- 상품명, 키워드, 옵션명, 옵션값에 특수문자 제한 정책 반영.
 - 오너클랜 웹사이트에 적용되는 정책과 동일.
 - 상품명(`Item.name`), 키워드(`Item.searchKeywords`)는 동일한 정책을 반영하며, 해당 사항은 매뉴얼 하단의 *특수문자 제한 정책* 섹션을 참고.
 - 옵션명(`Item.options.optionAttributes.name`), 옵션값(`Item.options.optionAttributes.value`)는 동일한 정책을 반영하며, 해당 사항은 매뉴얼 하단의 *특수문자 제한 정책* 섹션을 참고.
- 상품 인증정보(`Item.metadata.certificateInformation`) 관련 정책 변경
 - 오너클랜 웹사이트에 적용되는 정책과 동일.
 - 기존:
 - 입력할 인증 타입(`certificateType` 필드)을 지정한 후 해당 필드에 필수로 입력해야 하는 값을 모두 지정하면 통과.
 - 별도의 validation 없음.
 - 가능한 인증 타입: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 81, 82, 83, 84
 - 변경:
 - 입력할 인증 타입(`certificateType` 필드)을 지정한 후 해당 필드에 필수로 입력해야 하는 값을 지정.
 - 인증 타입이 11, 14, 18, 81, 82, 84인 경우 필수 필드가 모두 입력되면 통과.
 - 인증 타입이 9, 10, 12, 13, 16, 17인 경우 모두 `certificateCode` 필드를 입력받고, 해당 필드의 값을 통해 실제로 등록된 인증정보와 일치하는지 검증. 검증에 통과하면 해당 검증 정보를 기반으로 데이터 저장.
 - 인증 타입 중 1, 2, 3, 4, 5, 6, 7, 8, 15, 83은 더이상 등록하거나 수정할 때 사용할 수 없으며 기존 데이터를 읽기만 가능. 만약 등록하거나 수정할 때 사용하면 deprecation error 반환.
 - 인증 정보를 직접 `certificateInformation` 필드에 입력하는 경우 `certificateInformation.inContent` 필드의 값을 `false`로, 상품 상세정보에 있어 상세 정보 별도표기로 진행해야 하는 경우 인증 타입만 입력하고 `certificateInformation.inContent` 필드의 값을 `true`로 설정.

v0.7.3 -> v0.7.7

- `allItems` 쿼리의 `category` 파라미터는 기존의 일치 검색에서 `LIKE` 검색으로 변경.
 - 입력한 카테고리 및 하위 카테고리(모든 레벨)에 해당하는 상품을 검색.
- `createItem`, `updateItem`에서 상품 신규 등록 및 기존 상품 수정시 `Item.searchKeywords`의 각 항목에 길이 제한 추가.
 - `euc-kr` 인코딩 기준 20 bytes 이하의 값만 등록 가능.
 - 개수 제한은 기존과 동일하게 10개 이하.
- `createItem`, `updateItem`에서 상품 신규 등록 및 기존 상품 수정시 옵션 가격은 최저값의 150%를 넘을 수 없음.
 - 예를 들어, 각 옵션에 공급가를 입력할 때, 옵션 가격 중 최저값을 10000이라고 하면 모든 옵션의 가격은 15000을 넘을 수 없음.

- `createItem`에서 API request quota 제한과 별도로 기존 오너클랜 시스템과 동일하게 하루 상품 등록 개수 제한 설정.
 - 각 공급사별 제한 수량은 오너클랜 시스템과 동일.

v0.7.0 -> v0.7.3

- 필드 삭제
 - `Item.brand` -> 삭제
 - 현재 오너클랜 시스템에서 사용하지 않는 필드로, 삭제.

v0.6.41 -> v0.7.0

- 필드명 및 위치 변경
 - `Item.isTaxFree` -> `Item.taxFree`
 - `Item.isMinorSellable` -> `Item.adultOnly`
 - `adultOnly = not isMinorSellable`의 관계에 있습니다.
 - (New) `Item.price`
 - 상품의 기준 가격입니다. 가격이 서로 다른 옵션이 있는 경우 이 가격을 기준으로 추가 혹은 할인 가격이 계산됩니다.
 - `Item.metadata.madein` -> `Item.origin`
 - `Item.metadata.production` -> `Item.production`
 - `Item.metadata.isStrictSellprice` -> `Item.pricePolicy`
 - Type: Boolean -> Enum { free, fixed }
 - `isStrictSellprice`가 true인 경우는 fixed와 같고, false인 경우는 free와 같습니다.
 - `Item.metadata.prContent` -> `Item.sellerOnlyContent`
 - `Item.metadata.deliveryFee` -> `Item.shippingFee`
 - `Item.metadata.deliveryType` -> `Item.shippingType`
 - `Item.metadata.consumerPrice` -> `Item.fixedPrice`
 - Type: Int -> Price
 - `Item.metadata.isRefundable` -> `Item.returnable`
 - `Item.metadata.rtRefuseMsg` -> `Item.noReturnReason`
 - `Item.metadata.safeDeliveryDay` -> `Item.guaranteedShippingPeriod`
 - `Item.metadata.isOpenmarketSellable` -> `Item.openmarketSellable`
 - `Item.metadata.minq` -> 삭제
 - `Item.metadata.maxq` -> `Item.boxQuantity`
 - `Item.metadata.vendorAlertMessage` -> `Item.metadata.vendorNotice`
 - `Item.metadata.closetime` -> `Item.closingTime`
 - `Item.metadata.nfCategoryInformation` -> `Item.metadata.productNotificationInformation`
- 필드 액세스 방식 변경
 - Price
 - 기존: 화폐 단위 이름의 하위 필드를 명시함으로써 액세스

```
# 입력
{
  priceField {
    KRW
  }
}
```

```
{
  "priceField": {
    "KRW": 12345
  }
}
```

- 변경: 필드의 인자로 **currency**를 지정함. **currency**는 기존 방식에서 사용하던 화폐 단위 이름과 같은 enum.

```
# 입력
{
  priceField (currency: KRW)
}
```

```
{
  "priceField": 12345
}
```

- 참고사항: 여러 화폐 단위의 값이 필요한 경우나 여러 번 요청해야 하는 경우 GraphQL Alias 를 다음과 같이 사용.

```
# 입력
{
  price_KRW: priceField (currency: KRW)
  price_USD: priceField (currency: USD)
}
```

```
{
  "price_KRW": 12345,
  "price_USD": 12.345
}
```

- 기존: 사이즈 이름의 하위 필드를 명시함으로써 액세스

```
# 입력
{
  images {
    large
    medium
    small
  }
}
```

```
{
  "images": {
    "large": [
      "url1",
      "url2"
    ],
    "medium": [
      "url3",
      "url4"
    ],
    "small": [
      "url5",
      "url6"
    ]
  }
}
```

- 변경: 필드의 인자로 **size**를 지정함. **size**는 기존 방식에서 사용하던 사이즈 이름과 같은 enum.

```
# 입력
{
  imagesField (size: large)
}
```

```
{
  "imagesField": [
    "url1",
    "url2"
  ]
}
```


- 참고: 여러 사이즈의 이미지가 필요한 경우나 여러 번 요청해야 하는 경우 GraphQL Alias를 다음과 같이 사용.

```
# 입력
{
  images_large: imagesField (size: large)
  images_medium: imagesField (size: medium)
  images_small: imagesField (size: small)
}
```

```
{
  "images_large": [
    "url1",
    "url2"
  ],
  "images_medium": [
    "url3",
    "url4"
  ],
  "images_small": [
    "url5",
    "url6"
  ]
}
```

- 기타 버그 수정.

v0.6.31 -> v0.6.41

- 오픈마켓 카테고리 코드 제공(쿠팡, 스토어팜, 옥션, 지마켓, 11번가, 인터파크).
- 기타 버그 수정.

Endpoints

API Endpoint

- URL:
 - Production: <https://api.ownerclan.com/v1/graphql>
 - Sandbox: <https://api-sandbox.ownerclan.com/v1/graphql>
- 사용법: **GraphQL 쿼리 보내기** 섹션 참고.

GUI Test Endpoint (GraphQL Playground)

- URL:
 - Production: <https://api.ownerclan.com/v1/graphql>
 - Sandbox: <https://api-sandbox.ownerclan.com/v1/graphql>
- 사용법: **GraphQL Playground** 섹션 참고.

Authentication Endpoint

- URL:
 - Production: <https://auth.ownerclan.com/auth>
 - Sandbox: <https://auth-sandbox.ownerclan.com/auth>
- 인증 방식: JWT
- 사용법: **JWT 인증** 섹션 참고.

사용법

GraphQL

오너클랜 API는 [GraphQL](#)로 구현되어 있습니다. GraphQL은 URL과 Method에 따라 query가 구분되었던 REST API와 다르게 GraphQL 언어를 통해 쿼리를 작성하고 단 하나의 엔드포인트로 모든 요청을 보내도록 되어 있습니다. 쿼리의 내용에 어떤 데이터를 받을 지, 어떤 데이터를 수정할 지 명시하면 됩니다. 예를 들면 다음과 같은 방식으로 상품 1개에 대한 정보를 얻어올 수 있습니다.

```
# 예시 쿼리.  
# 그대로 붙여넣기 해도 돌아가지는 않습니다.  
query {  
  item(key: "W000000") {  
    name  
    model  
    options {  
      price  
      quantity  
      optionAttributes {  
        name  
        value  
      }  
    }  
  }  
}
```

```
# 예시 응답.  
{  
  "data": {  
    "name": "예시 상품",  
    "model": "예시 모델",  
    "options": [  
      {  
        "price": 35000,  
        "quantity": 23,  
        "optionAttributes": [  
          {  
            "name": "색상",  
            "value": "RED"  
          },  
          {  

```

```

        "name": "사이즈",
        "value": "95"
      }
    ]
  },
  {
    "price": 35000,
    "quantity": 23,
    "optionAttributes": [
      {
        "name": "색상",
        "value": "RED"
      },
      {
        "name": "사이즈",
        "value": "100"
      }
    ]
  }
]
}
}
}

```

예시 쿼리와 응답을 보면 알 수 있듯이, GraphQL 쿼리는 쿼리에 적힌 그대로 응답을 주게 됩니다.

GraphQL 쿼리 보내기

GraphQL 쿼리는 REST API에서 요청의 종류에 따라(READ, CREATE, ...) HTTP method가 달라졌던 것에 비해 READ 요청은 모두 GET method를, CREATE, UPDATE, DELETE 요청은 모두 POST method를 사용합니다. 아래 **JWT 인증** 섹션에서 설명하는 JWT 토큰을 받아 Header에 **Authorization** header를 추가하면 되며, Bearer token 타입이므로 **Bearer YOUR_TOKEN**과 같은 형식으로 넣어주시면 됩니다.

- READ (query): GET method를 사용합니다. 쿼리 내용은 URL의 query parameter로 넘겨주시면 됩니다.
 - 예시: https://api-sandbox.ownerclan.com/v1/graphql?query=EXAMPLE_QUERY
 - 코드:

```

// 아래 코드는 작동하는 코드가 아닙니다.
// 작동하는 코드는 별도로 첨부된 [create|read|update]_query_example.js 파일을
참고해주세요.

// ajax를 위한 XMLHttpRequest 객체를 초기화합니다.
var client = new XMLHttpRequest();

// GraphQL_Playground에서 사용한 쿼리를 그대로 사용할 수 있습니다.
var readQuery = "test query";

// ajax 요청을 셋팅합니다.
client.open("GET", "https://api-sandbox.ownerclan.com/v1/graphql?query=" +
encodeURIComponent(readQuery), true);
client.setRequestHeader("Authorization", "Bearer YOUR_TOKEN");

```

```

client.onreadystatechange = function (aEvt) {
    if (client.readyState === 4) {
        if (client.status === 200) {
            var response = JSON.parse(client.responseText);
            if (response.errors) {
                // 에러가 있다면 에러를 콘솔에 씁니다.
                console.log(JSON.stringify(response.errors));
            } else {
                // 에러가 없다면 반환된 데이터를 콘솔에 씁니다.
                console.log(JSON.stringify(response.data));
            }
        } else {
            // API 서버 응답이 정상인 경우 에러와 HTTP status code를 콘솔에
            // 씁니다.
            console.error(client.status, client.responseText);
        }
    }
}

// ajax 요청을 보냅니다.
client.send(null);

```

- CREATE, UPDATE, DELETE (mutation): POST method를 사용합니다.

- 예시: <https://api-sandbox.ownerclan.com/v1/graphql>
- 코드:

```

// 아래 코드는 작동하는 코드가 아닙니다.
// 작동하는 코드는 별도로 첨부된 [create|read|update]_query_example.js 파일을
// 참고해주세요.

// ajax를 위한 XMLHttpRequest 객체를 초기화합니다.
var client = new XMLHttpRequest();

// GraphQL_Playground에서 사용한 쿼리를 그대로 사용할 수 있습니다.
var updateQuery = `test query`;

// JSON 포맷으로 넣어주면 됩니다. 역시 GraphQL_Playground에서 사용한 데이터를 그
// 대로 사용합니다.
var inputVariables = {
    "input": {
        "name": {
            "ko_KR": "테스트 입력"
        },
        "searchKeywords": [
            "테스트 키워드 1",
            "테스트 키워드 2"
        ]
    }
};

```

```
// ajax 요청을 셋팅합니다.
client.open("POST", "https://api-sandbox.ownerclan.com/v1/graphql", true);
client.setRequestHeader("Authorization", "Bearer YOUT_TOKEN");
client.setRequestHeader("Content-Type", "application/json");
client.onreadystatechange = function (aEvt) {
    if (client.readyState === 4) {
        if (client.status === 200) {
            var response = JSON.parse(client.responseText);
            if (response.errors) {
                // 에러가 있다면 에러를 콘솔에 씁니다.
                console.log(JSON.stringify(response.errors));
            } else {
                // 에러가 없다면 반환된 데이터를 콘솔에 씁니다.
                console.log(JSON.stringify(response.data));
            }
        } else {
            // API 서버 응답이 정상인 경우 에러와 HTTP status code를 콘솔에
            // 씁니다.
            console.error(client.status, client.responseText);
        }
    }
}

// 요청을 보냅니다.
client.send(JSON.stringify({
    operationName: "updateItem",
    query: updateQuery,
    variables: inputVariables
})));
```

GraphQL Playground (Formerly GraphiQL)

오너클랜 API에서는 GraphQL Playground에서 UI를 갖추고 쿼리를 테스트할 수 있는 환경을 제공합니다. 어떤 쿼리가 있는지, 쿼리는 어떤 인자를 받고 어떤 데이터를 반환하는지까지 모두 볼 수 있습니다. GraphQL Playground는 Production 환경과 Sandbox 환경에서 모두 제공되며, API Endpoint와 동일한 Endpoint를 브라우저로 접속하면 됩니다. 특히 아래 **Method** 섹션에서 각 method별로 주어진 예시 쿼리를 시도해 볼 수 있습니다. 인증 token은 아래 **JWT 인증** 섹션에서 나온 방법대로 받아서 GraphQL Playground의 *Http headers* 항목에 JSON object 형식으로 지정하면 됩니다. (아래 예시 참조)

```
{
  "Authorization": "Bearer YOUR_TOKEN"
}
```

JWT 인증

오너클랜 API는 JWT를 기반으로 한 토큰 인증 방식을 사용합니다. 토큰은 [API Authentication endpoint\(Production\)](#) 혹은 [API Authentication endpoint\(Sandbox\)](#)로 오너클랜 아이디와 비밀번호를 POST 요청으로 보내서 받을 수 있습니다. 자바스크립트 예시는 다음과 같습니다.

```

var authData = {
  service: "ownerclan",
  userType: "vendor",
  username: "공급사ID", // 2010으로 시작하는 오너클랜 공급사 ID.
  password: "공급사PW" // 공급사 비밀번호.
};

$.ajax({
  url: "https://auth-sandbox.ownerclan.com/auth", // Production 환경에서는
  "https://auth.ownerclan.com/auth"를 사용합니다.
  type: "POST",
  contentType: "application/json",
  processData: false,
  data: JSON.stringify(authData),
  success: function(data) {
    // 성공한 경우, 토큰을 콘솔에 출력합니다.
    console.log(data);
  },
  error: function(data) {
    // 실패한 경우 에러와 HTTP status code를 기록합니다.
    console.error(data.responseText, data.status);
  }
});

```

Method

item

단일 상품 정보를 조회합니다. 공급사 API에서는 토큰을 발급받은 공급사로 등록한 상품만 조회할 수 있습니다.

- 파라미터:
 - **key**: ID - 정보를 조회할 상품의 key입니다. 실제 타입은 **String**이며, 오너클랜 상품 코드와 같습니다.
 - **lang**: Language - Text 타입의 필드들의 **lang** 파라미터 값을 한 번에 설정합니다. 하위 필드에서 별도로 설정하는 경우 이 설정은 덮어씌워집니다. 사용 가능한 값은 *데이터 타입 - Language* 파트를 참고해주시면 됩니다. 기본값은 **ko_KR**입니다.
 - **currency**: Currency - 가격 정보 필드들의 **currency** 파라미터 값을 한 번에 설정합니다. 하위 필드에서 별도로 설정하는 경우 이 설정은 덮어씌워집니다. 사용 가능한 값은 *데이터 타입 - Currency* 항목을 참고해주시면 됩니다. 기본값은 **KRW**입니다.
- 반환 데이터: 상품 정보를 반환하며, 다음 데이터를 액세스 할 수 있습니다.
 - **createdAt**: Int - 상품이 DB에 등록된 시각입니다. Unix timestamp로 주어집니다.
 - **updatedAt**: Int - 상품이 최종적으로 업데이트 된 시각입니다. Unix timestamp로 주어집니다.
 - **key**: String - 상품의 오너클랜 코드입니다.
 - **name**: Text - 상품의 이름입니다. **lang** 파라미터를 가질 수 있으며, 언어를 지정할 수 있습니다. 현재 지원되는 언어는 **ko_KR**입니다.
 - **model**: String - 상품의 모델명입니다.
 - **production**: String - 제조사입니다.
 - **origin**: String - 제조 국가 (국내 제조의 경우 지역 포함)입니다.

- **id: String** - 상품의 API ID입니다.
- **price: Float** - 상품의 가격입니다. 옵션마다 가격이 다른 경우 기준 가격이 됩니다. **currency** 파라미터를 가질 수 있으며, 현재 지원되는 **Currency**는 **KRW**입니다.
- **pricePolicy: PricePolicy** - 상품 가격 정책입니다. 가능한 값들은 *데이터 타입 - PricePolicy* 항목을 참고해주시면 됩니다.
- **fixedPrice: Float** - **pricePolicy**가 **fixed**인 경우 소비자 준수가격입니다. **currency** 파라미터를 가질 수 있으며, 현재 지원되는 **Currency**는 **KRW**입니다.
- **searchKeywords: [String]** - 상품에 대한 검색 키워드 목록입니다.
- **category: Category** - 상품에 대한 카테고리 정보입니다.
- **content: String** - 상품의 상세정보입니다.
- **shippingFee: Int** - 상품 배송비입니다.
- **shippingType: ShippingType** - 상품 배송비 부과 타입입니다. 가능한 값들은 *데이터 타입 - ShippingType* 항목을 참고해주시면 됩니다.
- **images: [URL]** - 상품의 이미지를 제공합니다. **size** 파라미터를 가지며, 현재 지원되는 **ImageSize**는 **auto, xlarge, large, medium, small** 입니다. 제공되는 URL 중 첫 번째 URL이 대표 목록이미지, 2개 이상의 URL이 제공된 경우 나머지 URL들이 추가 목록이미지입니다.
- **status: String** - 상품의 현재 상태입니다. "**soldout**"은 품절, "**available**"은 판매중, "**unavailable**"은 수량이 남아있으나 판매하지 않는(진열 X) 상태, "**discontinued**"은 단종을 의미합니다.
- **options: [ItemOption]** - 상품의 옵션 정보입니다. 옵션별 옵션 속성, 가격을 조회할 수 있습니다. **ItemOption**에 대한 내용은 *데이터 타입 - ItemOption* 항목을 참고해주시면 됩니다.
- **taxFree: Boolean** - 면세 상품인지의 여부입니다.
- **adultOnly: Boolean** - 미성년자 판매 가능 상품인지의 여부입니다. **false**면 가능, **true**면 불가능입니다.
- **returnable: Boolean** - 반품 가능 여부입니다.
- **noReturnReason: String** - 반품 불가 상품의 경우, 반품 불가 사유입니다.
- **guaranteedShippingPeriod: Int** - 안전배송일입니다.
- **openmarketSellable: Boolean** - 오픈마켓 판매가 가능한 상품인지의 여부입니다.
- **boxQuantity: Int** - 1박스 당 최대 묶음배송 가능 수량입니다.
- **attributes: [String]** - 상품 속성의 리스트입니다.
- **closingTime: String** - 배송 마감 시각입니다. **null**이나 **undefined**인 경우 따로 지정되지 않은 것이며, 지정된 경우 양식은 **HH:MM:SS**입니다.
- **returnCriteria: ReturnCriteria** - 반품접수 유형입니다. 자세한 내용은 *데이터 타입 - ReturnCriteria* 항목을 참고해주시면 됩니다.
- **metadata: JSON** - 상품에 대한 추가 정보입니다.
 - **returnAddressCode: Int** - 반품지 코드입니다. **0**은 공급사별 기본 배송지입니다.
 - **vendorKey: String** - 해당 상품을 공급하는 공급사의 고유 key입니다.
 - **productNotificationInformation: Object** - 상품 정보고시 정보입니다. 하단의 **정보고시** 섹션을 참고해주세요.
 - **certificateInformation: [Object]** - 상품 인증정보 내역입니다. 인증정보는 하단의 **인증정보** 섹션을 참고해주세요.
 - **vendorNotice: String** - 공급사 알림 메시지입니다. 오너클랜 사이트의 상품 상세페이지에서 상품 상세설명 바로 위에 노출되는, 상단문구와는 별개의 문구입니다.
 - **auctionCategoryCode: String** - 이 item의 옥션 카테고리 코드입니다.
 - **auctionCategoryFullName: String** - 이 item의 옥션 카테고리 이름입니다.
 - **gmarketCategoryCode: String** - 이 item의 지마켓 카테고리 코드입니다.

- `gmarketCategoryFullName: String` - 이 item의 지마켓 카테고리 이름입니다.
- `st11CategoryCode: String` - 이 item의 11번가 카테고리 코드입니다.
- `st11CategoryFullName: String` - 이 item의 11번가 카테고리 이름입니다.
- `interparkCategoryCode: String` - 이 item의 인터파크 카테고리 코드입니다.
- `interparkCategoryFullName: String` - 이 item의 인터파크 카테고리 이름입니다.
- `smartstoreCategoryCode: String` - 이 item의 스마트스토어(스토어팜) 카테고리 코드입니다.
- `smartstoreCategoryFullName: String` - 이 item의 스마트스토어(스토어팜) 카테고리 이름입니다.
- `coupangCategoryCode: String` - 이 item의 쿠팡 카테고리 코드입니다.
- `coupangCategoryFullName: String` - 이 item의 쿠팡 카테고리 이름입니다.
- `playautoCategoryCode: String` - 이 item의 플레이오토 카테고리 코드입니다.
- `playautoCategoryFullName: String` - 이 item의 플레이오토 카테고리 이름입니다.
- `esellersCategoryCode: String` - 이 item의 이셀러스 카테고리 코드입니다.
- `esellersCategoryFullName: String` - 이 item의 이셀러스 카테고리 이름입니다.
- `shippingName: String` - 이 item의 택배 송장명입니다.
- `groupCode: String` - 그룹 상품인 경우 이 item의 그룹 코드입니다.
- `vcode: String` - 4자리의 상품 발주업체 코드(공급사 자체 상품코드 1)입니다. 판매사에게는 노출되지 않는 항목으로, 공급사 측에서 자체 관리 용도로 사용하면 됩니다.
- `vprdcod: String` - 상품 발주업체의 상품 관리코드(공급사 자체 상품코드 2)입니다. 판매사에게는 노출되지 않는 항목으로, 공급사 측에서 자체 관리 용도로 사용하면 됩니다.
- `returnShippingFee: Int` - 편도 반품배송비입니다.
- `auctionCategoryIsItemMatching: Boolean` - 이 item의 옥션 카테고리 정보가 직접 지정한 카테고리 매칭 정보에 의한 것인지의 여부입니다.
- `gmarketCategoryIsItemMatching: Boolean` - 이 item의 지마켓 카테고리 정보가 직접 지정한 카테고리 매칭 정보에 의한 것인지의 여부입니다.
- `st11CategoryIsItemMatching: Boolean` - 이 item의 11번가 카테고리 정보가 직접 지정한 카테고리 매칭 정보에 의한 것인지의 여부입니다.
- `interparkCategoryIsItemMatching: Boolean` - 이 item의 인터파크 카테고리 정보가 직접 지정한 카테고리 매칭 정보에 의한 것인지의 여부입니다.
- `coupangCategoryIsItemMatching: Boolean` - 이 item의 쿠팡 카테고리 정보가 직접 지정한 카테고리 매칭 정보에 의한 것인지의 여부입니다.

• 예시 쿼리:

```
query testQuery {
  item(key: "W0000000") {
    createdAt
    updatedAt
    key
    name
    model
    production
    origin
    id
    price
    pricePolicy
    fixedPrice
  }
}
```



```

    searchKeywords
    category {
      key
      name
      fullName
    }
    content
    shippingFee
    shippingType
    images
    status
    options {
      optionAttributes {
        name
        value
      }
      price
      quantity
      key
    }
    taxFree
    adultOnly
    returnable
    noReturnReason
    guaranteedShippingPeriod
    openmarketSellable
    boxQuantity
    attributes
    closingTime
    returnCriteria
    metadata
  }
}

```

- 예시 코드:
 - [example1 - 단일 상품 정보 조회하기](#)
 - [example2 - 이미지 여러 사이즈 조회하기](#)

allItems

복수의 상품을 조회합니다. 한 번에 최대 1000개의 상품 정보를 조회할 수 있으며, pagination을 위한 cursor를 제공합니다. 자세한 사용법은 하단 예시를 참고하면 됩니다.

- 파라미터:
 - Pagination 관련 파라미터:
 - **after: String** - API 고유 cursor 값으로, 지정된 경우 이 cursor 값 이후의 상품만을 불러옵니다.
 - **before: String** - API 고유 cursor 값으로, 지정된 경우 이 cursor 값 이전의 상품만을 불러옵니다.
 - **first: Int** - 조건을 만족하는 모든 상품들 중 처음 몇 개의 상품을 불러올 지를 나타냅니다.

- **last: Int** - 조건을 만족하는 모든 상품들 중 처음 몇 개의 상품을 불러올 지를 나타냅니다.
 - Search 관련 파라미터:
 - **dateFrom: Timestamp** - 상품 수정 일자가 이 값 이후인 것들만을 불러옵니다. 생략하면 1 달 전으로 설정됩니다.
 - **dateTo: Timestamp** - 상품 수정 일자가 이 값 이후인 것들만을 불러옵니다. 생략하면 현재 로 설정됩니다.
 - **minPrice: Int** - 가격 검색시 최저가입니다. 이 필드가 값을 가지면, 명시된 가격 이상의 가격을 가지는 상품만을 불러옵니다.
 - **maxPrice: Int** - 가격 검색시 최고가입니다. 이 필드가 값을 가지면, 명시된 가격 이하의 가격을 가지는 상품만을 불러옵니다.
 - **search: String** - 검색어입니다. 오너클랜 상품코드를 넣을 경우(W로 시작하는 7자리 코드) 해당 상품코드로 검색하고, 그렇지 않을 경우 일반 검색을 진행합니다.
 - **vendor: ID** - 주어진 공급사 key를 가진 공급사의 상품만을 불러옵니다.
 - **vcode: String** - 주어진 발주처 코드를 가진 상품만을 불러옵니다.
 - **vprdcod: String** - 주어진 발주처 상품코드를 가진 상품만을 불러옵니다.
 - **status: ItemStatus** - 주어진 상태에 해당하는 상품만을 불러옵니다. 가능한 값들은 *ItemStatus* 항목을 참고해주세요.
 - **category: ID** - 주어진 카테고리 코드와 해당 카테고리의 하위 카테고리에 해당하는 상품들만을 불러옵니다.
 - **attributes: [String]** - 주어진 속성을 모두 갖는 상품만을 불러옵니다. 가능한 속성 값들은 아래를 참고해주세요.
 - **sortBy: ItemSortCriteria** - 정렬 조건을 설정합니다. 설정하지 않으면 key로 정렬된 결과를 반환합니다. 가능한 정렬 기준은 *ItemSortCriteria* 항목을 참고해주세요.
 - **shippingLocationType: ShippingLocationType** - 국내 배송 상품 혹은 해외 배송 상품을 구분하여 요청할 수 있습니다. 설정하지 않으면 국내/해외 배송 상품 모두 반환합니다. 가능한 기준은 *ShippingLocationType* 항목을 참고해주세요.
 - 데이터 관련 파라미터
 - **lang: Language** - 현재 쿼리의 **Text** 타입의 필드들에 대한 기본 언어를 선택합니다. 별도로 설정하지 않으면 기본값은 **ko_KR**입니다. 만약 **Text** 타입의 하위 필드에 **lang** 파라미터 값이 설정되면 해당 필드는 설정된 언어로 값을 반환합니다.
 - **currency: Currency** - 현재 쿼리의 **Price** 타입의 필드들에 대한 기본 화폐단위를 선택합니다. 별도로 설정하지 않으면 기본값은 **KRW**입니다. 만약 **Price** 타입의 하위 필드에 **currency** 파라미터 값이 설정되면 해당 필드는 설정된 화폐단위로 값을 반환합니다.
- 반환 값: **allItems**는 pagination을 위한 정보와 상품 정보를 동시에 담고 있습니다.
 - **pageInfo**: pagination을 위한 정보입니다.
 - **hasNextPage**: 데이터가 뒤에 더 있는지를 나타냅니다.
 - **hasPreviousPage**: 데이터가 앞에 더 있는지를 나타냅니다.
 - **startCursor**: 현재 페이지의 첫 데이터에 대한 cursor 값입니다. 이 값을 **before** 파라미터에 넘겨주면 정확하게 이전 페이지의 데이터를 가져올 수 있습니다.
 - **endCursor**: 현재 페이지의 마지막 데이터에 대한 cursor 값입니다. 이 값을 **after** 파라미터에 넘겨주면 정확하게 다음 페이지의 데이터를 가져올 수 있습니다.
 - **edges**: 상품 데이터입니다.
 - **cursor**: 해당 상품의 pagination cursor입니다.
 - **node**: 해당 상품의 데이터입니다. 반환되는 데이터는 **item** 쿼리의 반환 값과 같습니다.
- 예시 쿼리:

검색 없이 기본 조건만으로 가져오는 쿼리입니다.

```
query {  
  allItems {  
    pageInfo {  
      hasNextPage  
      hasPreviousPage  
      startCursor  
      endCursor  
    }  
    edges {  
      cursor  
      node {  
        createdAt  
        updatedAt  
        key  
        name  
        model  
        production  
        origin  
        id  
        price  
        pricePolicy  
        fixedPrice  
        searchKeywords  
        category {  
          key  
          name  
          fullName  
        }  
        content  
        shippingFee  
        shippingType  
        images  
        status  
        options {  
          optionAttributes {  
            name  
            value  
          }  
          price  
          quantity  
          key  
        }  
        taxFree  
        adultOnly  
        returnable  
        noReturnReason  
        guaranteedShippingPeriod  
        openmarketSellable  
        boxQuantity  
        attributes  
        closingTime  
        returnCriteria
```

```

    metadata
  }
}
}
}

```

카테고리 key가 "50000108" 혹은 그 하위 카테고리의 key이고 가격이 10000원 이상, 50000원 이하인 상품 중 첫 50개만 가져옵니다.

```

query {
  allItems(minPrice: 10000, maxPrice: 50000, category: "50000108", first: 50) {
    pageInfo {
      hasNextPage
      hasPreviousPage
      startCursor
      endCursor
    }
    edges {
      cursor
      node {
        createdAt
        updatedAt
        key
        name
        model
        production
        origin
        id
        price
        pricePolicy
        fixedPrice
        searchKeywords
        category {
          key
          name
          fullName
        }
        content
        shippingFee
        shippingType
        images
        status
        options {
          optionAttributes {
            name
            value
          }
          price
          quantity
          key
        }
        taxFree
      }
    }
  }
}

```

```

        adultOnly
        returnable
        noReturnReason
        guaranteedShippingPeriod
        openmarketSellable
        boxQuantity
        attributes
        closingTime
        returnCriteria
        metadata
    }
}
}
}

```

카테고리 key가 "50000108" 혹은 그 하위 카테고리의 key인 상품을 수정일 기준 내림차순 정렬 후 첫 30개를 가져옵니다.

```

query {
  allItems(category: "50000108", sortBy: dateDesc, first: 30) {
    pageInfo {
      hasNextPage
      hasPreviousPage
      startCursor
      endCursor
    }
    edges {
      cursor
      node {
        createdAt
        updatedAt
        key
        name
        model
        production
        origin
        id
        price
        pricePolicy
        fixedPrice
        searchKeywords
        category {
          key
          name
          fullName
        }
        content
        shippingFee
        shippingType
        images
        status
        options {

```

```

    optionAttributes {
      name
      value
    }
    price
    quantity
    key
  }
  taxFree
  adultOnly
  returnable
  noReturnReason
  guaranteedShippingPeriod
  openmarketSellable
  boxQuantity
  attributes
  closingTime
  returnCriteria
  metadata
}
}
}
}

```

• 예시 코드:

- example1 - 상품 수정일 기준 검색
- example2 - 가격 검색
- example3 - 공급사 코드 검색
- example4 - 카테고리 코드 검색
- example5 - 복합 검색(가격 & 수정일 & 카테고리 코드)
- example6 - 페이지네이션1(특정 상품 이후의 상품 중 첫 500개)
- example7 - 페이지네이션2(특정 상품 이전의 상품 중 마지막 100개)
- example8 - 페이지네이션 & 복합검색

createItem

새 상품을 등록합니다.

- 유의사항: 하루에 등록할 수 있는 상품의 수(createItem request에 대한 quota와는 다름)에도 제한이 있습니다. 오너클랜 SCM에서 확인할 수 있는 제한과 동일합니다.
- 파라미터:
 - **autoReplace: boolean** - 자동 치환 규칙을 적용할 지의 여부입니다. 치환 규칙은 하단의 *특수문자 치환 정책* 섹션을 참고해주세요.
 - **autoTranslate: boolean** - ko_KR 필드에 값이 주어지지 않을 경우 자동으로 번역할 지의 여부입니다. 만약 **autoTranslate**가 **false**인데 **TextInput** 필드에 ko_KR 값이 주어지지 않으면 에러를 반환합니다.
 - **input: object(ItemInput)** - 입력할 데이터입니다.
- 입력 데이터 (굵은 글씨는 필수 필드입니다):
 - **name: TextInput** - 상품의 이름입니다. **TextInput** 필드에서 지원하는 입력 중 적어도 하나는 넣어야 합니다. ko_KR 필드를 입력한 경우 해당 필드의 값이, 아니면 한글로 번역 후의 값이 **euc-**

kr 인코딩 기준으로 50 bytes 이내여야 합니다.

- **content: String** - 상품 상세페이지 내용입니다.
- **production: String** - 제조사입니다.
- **origin: String** - 원산지입니다. 오너클랜 SCM에서 등록할 때 사용하는 값과 동일한 양식이어야 합니다. 예를 들어 "국내|서울|금천구", "해외|아시아|중국"과 같은 형식의 값만을 허용합니다.
- **images: [URL]** - 상품 이미지입니다. URL 문자열의 배열입니다. URL이 1개가 주어진 경우 해당 URL의 이미지를 대표 목록이미지로 설정하며, 2개 이상이 주어진 경우 첫 번째 URL의 이미지를 대표 목록이미지로, 나머지 URL의 이미지를 추가 목록이미지로 설정합니다.
- **category: String** - 상품에 해당하는 오너클랜 카테고리 코드입니다. 반드시 8자리여야합니다.
- **options: ItemOptionInput** - 상품 옵션 리스트입니다.
- **taxFree: Boolean** - 면세 상품인지의 여부입니다.
- **adultOnly: Boolean** - 미성년자 판매 가능 여부입니다.
- **openmarketSellable: Boolean** - 오픈마켓 판매 가능 여부입니다.
- **returnable: Boolean** - 반품 가능 여부입니다.
- **pricePolicy: PricePolicy** - 가격 정책입니다. 가능한 값들에 대해서는 *PricePolicy* 섹션을 참고해주세요.
- **shippingType: ShippingType** - 배송비 부과 타입입니다. 가능한 값들에 대해서는 *ShippingType* 섹션을 참고해주세요.
- **shippingFee: Int** - 배송비입니다.
- **attributes: [String]** - 상품 속성 리스트입니다. 상품 속성의 목록은 **상품속성** 섹션을 참고해주세요.
- **metadata: JSON** - 상품에 대한 기타 정보입니다. 여기에 들어가는 정보는 아래를 참고하세요.
- **model: String** - 상품 모델명입니다. 기본 값은 null입니다.
- **searchKeywords: [String]** - 상품 키워드 리스트입니다. 기본 값은 []입니다. 각 항목은 euc-kr 인코딩 기준으로 20 bytes 이내여야 합니다. 최대 40개까지 입력할 수 있습니다.
- **noReturnReason: String** - 반품 불가 상품인 경우 반품 불가 사유입니다. 기본 값은 ""입니다.
- **boxQuantity: Int** - 묶음 배송 최대 수량입니다. 기본 값은 null입니다.
- **fixedPrice: Price** - *pricePolicy*가 *fixed*인 경우 소비자 준수 가격입니다. *fixed*인 경우 필수 항목입니다.
- **closingTime: String** - 배송 마감 시각입니다. HH:MM:SS 형식이어야 합니다. 기본 값은 00:00:00입니다.
- **guaranteedShippingPeriod: Int** - 안전 배송일입니다. 기본 값은 3입니다.
- **returnCriteria: String** - 반품접수 유형입니다. 가능한 값들에 대해서는 *ReturnCriteria* 섹션을 참고해주세요. 기본값은 "seller"입니다.
- **metadata:**
 - **productNameDeli: String** - 택배송장 상품명입니다.
 - **productNotificationInformation: Object** - 정보고시 항목입니다.
 - **vcode: String** - 4자리의 발주업체 코드입니다.
 - **vprcode: String** - 발주업체의 상품코드입니다.
 - **returnAddressCode: Int** - 반품지 주소 코드입니다. 입력하지 않으면 공급사 기본값이 적용됩니다. 반품지 주소 코드로 주소를 조회하는 기능은 추후 오픈 예정입니다.
 - **returnShippingFee: Int** - 편도 반품배송비입니다. *shippingType*이 *all*, *inAdvance*인 경우 입력한 값이 있으면 입력한 값으로, 없으면 배송비와 동일하게 설정되며, *shippingType*이 *free*면 필수로 입력해야 합니다. *shippingType*이 *uponArrival*이면 입력 여부에 관계없이 0으로 설정됩니다.

- **certificateInformation: [Object]** - 인증 관련 정보입니다. 자세한 항목은 **인증정보** 섹션을 참고해주세요. 없으면 입력하지 않아도 됩니다.
- **externalCategories: Object** - 마켓 카테고리 매칭 정보입니다. 현재 쿠팡, 옥션, 지마켓, 인터파크, 11번가의 5개 마켓 카테고리 매칭 정보를 직접 지정할 수 있습니다. 자세한 내용과 입력 방식은 **마켓 카테고리 매칭 정보** 섹션을 참고해주세요. 지정하지 않으려면 이 필드 자체를 생략하면 됩니다.
- 반환 데이터:
 - 새로 등록된 상품 정보를 반환합니다.
- 예시 쿼리: 아래 2개의 코드 중 첫 번째는 실제 쿼리이고, 두 번째는 쿼리에서 쓰이는 **\$input** variable을 정의한 것입니다. GraphQL Playground에서 테스트할 때는 쿼리를 입력하는 공간 아래에 **QUERY VARIABLES**에 입력하면 됩니다.

```
mutation createItem($input: ItemInput) {
  createItem(autoTranslate: false, input: $input){
    createdAt
    updatedAt
    key
    name
    model
    production
    origin
    id
    price
    pricePolicy
    fixedPrice
    searchKeywords
    category {
      key
      name
    }
    content
    shippingFee
    shippingType
    images
    status
    options {
      optionAttributes {
        name
        value
      }
      price
      quantity
      key
    }
    taxFree
    adultOnly
    returnable
    noReturnReason
    guaranteedShippingPeriod
    openmarketSellable
    boxQuantity
  }
}
```



```

    attributes
    closingTime
    returnCriteria
    metadata
  }
}

```

```

{
  "input": {
    "name": {
      "ko_KR": "테스트 상품명"
    },
    "model": "테스트 모델명",
    "content": "테스트 콘텐츠",
    "production": "테스트 제조사",
    "origin": "국내|서울|금천구",
    "images": [
      "http://1.bp.blogspot.com/-
i5mpLKgaqQc/VASXbrMgSNI/AAAAAAAAADM/FoUS8mkYDpQ/s1600/20140814_122930.jpg"
    ],
    "searchKeywords": [
      "테스트 키워드 1",
      "테스트 키워드 2"
    ],
    "category": "50000108",
    "options": [
      {
        "optionAttributes": [
          {
            "name": {
              "ko_KR": "색상"
            },
            "value": {
              "ko_KR": "스카이블루"
            }
          }
        ],
        "price": {
          "currency": "KRW",
          "value": 10000
        },
        "productId": "",
        "quantity": 1,
        "metadata": {}
      },
      {
        "optionAttributes": [
          {
            "name": {
              "ko_KR": "색상"
            }
          }
        ]
      }
    ]
  }
}

```

```

        "value": {
            "ko_KR": "레드"
        }
    },
    ],
    "price": {
        "currency": "KRW",
        "value": 10000
    },
    "productId": "",
    "quantity": 1,
    "metadata": {}
}
],
"taxFree": false,
"adultOnly": false,
"openmarketSellable": true,
"returnable": true,
"shippingType": "inAdvance",
"shippingFee": 9600,
"pricePolicy": "free",
"returnCriteria": "vendor",
"metadata": {
    "certificateInformation": [
        {
            inContent: true,
            certificateType: 11
        }
    ],
    "productNotificationInformation": {
        "code": 35,
        "categorySpecific": [
            "상품 상세설명 참조",
            "상품 상세설명 참조",
            "해외 북아메리카 미국",
            "상품 상세설명 참조",
            "상품 상세설명 참조",
            "7 - 15일"
        ],
        "common": [
            "상품 상세정보에 별도 표기",
            "상품 상세정보에 별도 표기",
            "상품 상세정보에 별도 표기",
            "상품 상세정보에 별도 표기"
        ]
    },
    "productnameDeli": "테스트 송장명",
    "returnAddressCode": 0,
    "returnShippingFee": 9600
}
}
}

```

- 예시 코드:
 - [example1 - 단일 상품 등록하기](#)

UpdateItem

기존에 있는 상품을 업데이트합니다. 모든 필드에 대해, 값이 주어지지 않으면 기존에 있는 값을 유지하고 값이 주어진 경우에만 해당 필드의 값을 업데이트합니다.

- 파라미터:
 - **key:** ID - 업데이트할 상품의 key입니다. 실제 타입은 **String**이며, 오너클랜 상품 코드와 같습니다.
 - **autoReplace:** boolean - 자동 치환 규칙을 적용할 지의 여부입니다. 치환 규칙은 하단의 **특수문자 치환 정책** 섹션을 참고해주세요.
 - **autoTranslate:** boolean - ko_KR 필드에 값이 주어지지 않을 경우 자동으로 번역할 지의 여부입니다. 만약 **autoTranslate**가 **false**인데 **TextInput** 필드에 ko_KR 값이 주어지지 않으면 에러를 반환합니다.
 - **input:** object(**ItemInput**) - 입력할 데이터입니다.
- 입력 데이터 (굵은 글씨는 필수 필드입니다):
 - **name:** **TextInput** - 상품의 이름입니다. **TextInput** 필드에서 지원하는 입력 중 적어도 하나는 넣어야 합니다. ko_KR 필드를 입력한 경우 해당 필드의 값이, 아니면 한글로 번역 후의 값이 **euc-kr** 인코딩 기준으로 **50 bytes** 이내여야 합니다.
 - **content:** **String** - 상품 상세페이지 내용입니다.
 - **production:** **String** - 제조사입니다.
 - **origin:** **String** - 원산지입니다. 오너클랜 SCM에서 등록할 때 사용하는 값과 동일한 양식이어야 합니다. 예를 들어 "국내|서울|금천구", "해외|아시아|중국"과 같은 형식의 값만을 허용합니다.
 - **images:** [URL] - 상품 이미지입니다. URL 문자열의 배열입니다. URL이 1개가 주어진 경우 해당 URL의 이미지를 대표 목록이미지로 설정하며, 2개 이상이 주어진 경우 첫 번째 URL의 이미지를 대표 목록이미지로, 나머지 URL의 이미지를 추가 목록이미지로 설정합니다.
 - **category:** **String** - 상품에 해당하는 오너클랜 카테고리 코드입니다. 반드시 8자리여야 합니다.
 - **options:** **ItemOptionInput** - 상품 옵션 리스트입니다.
 - **taxFree:** **Boolean** - 면세 상품인지의 여부입니다.
 - **adultOnly:** **Boolean** - 미성년자 판매 가능 여부입니다.
 - **openmarketSellable:** **Boolean** - 오픈마켓 판매 가능 여부입니다.
 - **returnable:** **Boolean** - 반품 가능 여부입니다.
 - **pricePolicy:** **PricePolicy** - 가격 정책입니다. 가능한 값들에 대해서는 **PricePolicy** 섹션을 참고해주세요.
 - **shippingType:** **ShippingType** - 배송비 부과 타입입니다. 가능한 값들에 대해서는 **ShippingType** 섹션을 참고해주세요.
 - **shippingFee:** **Int** - 배송비입니다.
 - **attributes:** [String] - 상품 속성 리스트입니다. 상품 속성의 목록은 **상품속성** 섹션을 참고해주세요.
 - **metadata:** **JSON** - 상품에 대한 기타 정보입니다. 여기에 들어가는 정보는 아래를 참고하세요.
 - **model:** **String** - 상품 모델명입니다.
 - **searchKeywords:** [String] - 상품 키워드 리스트입니다. 각 항목은 **euc-kr** 인코딩 기준으로 **20 bytes** 이내여야 합니다. 최대 40개까지 입력할 수 있습니다.

- `noReturnReason: String` - 반품 불가 상품인 경우 반품 불가 사유입니다.
 - `boxQuantity: Int` - 묶음 배송 최대 수량입니다.
 - `fixedPrice: Price` - `pricePolicy`가 `fixed`인 경우 소비자 준수 가격입니다. `fixed`인 경우 필수 항목입니다.
 - `closingTime: String` - 배송 마감 시각입니다. `HH:MM:SS` 형식이어야 합니다.
 - `guaranteedShippingPeriod: Int` - 안전 배송일입니다.
 - `returnCriteria: String` - 반품접수 유형입니다. 가능한 값들에 대해서는 *ReturnCriteria* 섹션을 참고해주세요.
- **metadata:**
 - `productNameDeli: String` - 택배송장 상품명입니다.
 - `productNotificationInformation: Object` - 정보고시 항목입니다.
 - `vcode: String` - 4자리의 발주업체 코드입니다.
 - `vprcode: String` - 발주업체의 상품코드입니다.
 - `returnAddressCode: Int` - 반품지 주소 코드입니다. 입력하지 않으면 공급사 기본값이 적용됩니다. 반품지 주소 코드로 주소를 조회하는 기능은 추후 오픈 예정입니다.
 - `returnShippingFee: Int` - 편도 반품배송비입니다. 값을 입력한 경우 입력한 값으로 변경되며, 입력하지 않으면 변경되지 않습니다. 단, `shippingFee`를 변경하거나, `shippingType`을 변경한 경우 새롭게 변경되는 배송비와 배송비 부과 타입에 따른 반품배송비가 계산되어 반영됩니다.
 - `certificateInformation: [Object]` - 인증 관련 정보입니다. 자세한 항목은 **인증정보** 섹션을 참고해주세요.
 - `externalCategories: Object` - 마켓 카테고리 매칭 정보입니다. 현재 쿠팡, 옥션, 지마켓, 인터파크, 11번가의 5개 마켓 카테고리 매칭 정보를 직접 지정할 수 있습니다. `updateItem` 쿼리에서는 입력한 내용에 대해 기존에 있던 마켓 카테고리 매칭 정보인 경우 새 값으로 덮어쓰우며, 기존에 없던 값은 추가하고, 기존에 있었으나 명시되지 않은 마켓에 대한 매칭 정보는 삭제합니다. 자세한 내용과 입력 방식은 **마켓 카테고리 매칭 정보** 섹션을 참고해주세요. 지정하지 않으려면 이 필드 자체를 생략하면 됩니다.
 - **반환 데이터:**
 - 업데이트된 상품 정보를 반환합니다.
 - 예시 쿼리: 아래 2개의 코드 중 첫 번째는 실제 쿼리이고, 두 번째는 쿼리에서 쓰이는 `$input` variable을 정의한 것입니다. GraphQL Playground에서 테스트할 때는 쿼리를 입력하는 공간 아래에 `QUERY VARIABLES`에 입력하면 됩니다.

```
mutation updateItem($input: ItemUpdateInput) {
  updateItem(autoTranslate: false, key: "W123A1A", input: $input){
    createdAt
    updatedAt
    key
    name
    model
    production
    origin
    id
    price
    pricePolicy
  }
}
```

```

    fixedPrice
    searchKeywords
    category {
      key
      name
      fullName
    }
    content
    shippingFee
    shippingType
    images
    status
    options {
      optionAttributes {
        name
        value
      }
      price
      quantity
      key
    }
    taxFree
    adultOnly
    returnable
    noReturnReason
    guaranteedShippingPeriod
    openmarketSellable
    boxQuantity
    attributes
    closingTime
    returnCriteria
    metadata
  }
}

```

```

{
  "input": {
    "name": {
      "ko_KR": "테스트 상품명"
    },
    "model": "테스트 모델명",
    "content": "테스트 콘텐츠",
    "production": "테스트 제조사",
    "origin": "국내|서울|구로구",
    "images": [
      "http://1.bp.blogspot.com/-
i5mpLKgaqQc/VASXbrMgSNI/AAAAAAAAADM/FoUS8mkYDpQ/s1600/20140814_122930.jpg"
    ],
    "searchKeywords": [
      "테스트 키워드 1",
      "테스트 키워드 2"
    ]
  }
}

```

```
],
"category": "50000108",
"options": [
  {
    "optionAttributes": [
      {
        "name": {
          "ko_KR": "색상"
        },
        "value": {
          "ko_KR": "스카이블루"
        }
      }
    ],
    "price": {
      "currency": "KRW",
      "value": 10000
    },
    "productId": "",
    "quantity": 1,
    "metadata": {}
  },
  {
    "optionAttributes": [
      {
        "name": {
          "ko_KR": "색상"
        },
        "value": {
          "ko_KR": "레드"
        }
      }
    ],
    "price": {
      "currency": "KRW",
      "value": 10000
    },
    "productId": "",
    "quantity": 1,
    "metadata": {}
  }
],
"taxFree": false,
"adultOnly": false,
"openmarketSellable": true,
"returnable": true,
"shippingType": "inAdvance",
"shippingFee": 9600,
"pricePolicy": "free",
"returnCriteria": "vendor",
"metadata": {
  "certificateInformation": [
    {
      "certificateType": 12,
```

```

        "certificateName": "한국건설생활시험연구원(KCL)",
        "certificateCode": "CERTIFICATE_CODE",
        "companyName": "COMPANY_NAME",
        "certifiedDate": "2018-02-21",
        "inContent": false
    }
],
"productNotificationInformation": {
    "code": 35,
    "categorySpecific": [
        "상품 상세설명 참조",
        "상품 상세설명 참조",
        "해외 북아메리카 미국",
        "상품 상세설명 참조",
        "상품 상세설명 참조",
        "7 - 15일"
    ],
    "common": [
        "상품 상세정보에 별도 표기",
        "상품 상세정보에 별도 표기",
        "상품 상세정보에 별도 표기",
        "상품 상세정보에 별도 표기"
    ]
},
"productnameDeli": "테스트 송장명",
"returnAddressCode": 0
}
}
}

```

- 예시 코드:
 - [example1 - 단일 상품 업데이트 하기](#)

discontinueItem

상품을 단종시킵니다. 옵션이 있는 상품의 경우, 상품을 단종시키면 모든 옵션들 역시 단종상태로 전환됩니다.

- 파라미터:
 - **key**: ID - 단종시킬 상품의 key입니다. 실제 타입은 **String**이며, 오너클랜 상품 코드와 같습니다.
- 반환 데이터:
 - 이 쿼리는 성공적으로 단종되었는지의 여부를 **Boolean**으로 반환합니다.
- 예시 쿼리

```

mutation {
  discontinueItem(key: "W123A1A")
}

```

```
{
  "data": true
}
```

- 예시 코드:
 - [example1 - 단일 상품 단종시키](#)

soldoutItem

상품을 품절시킵니다. 옵션이 있는 상품의 경우, 상품을 품절시키면 모든 옵션들 역시 품절상태로 전환됩니다.

- 파라미터:
 - key: ID** - 품절시킬 상품의 key입니다. 실제 타입은 **String**이며, 오너클랜 상품 코드와 같습니다.
- 반환 데이터:
 - 이 쿼리는 성공적으로 품절되었는지의 여부를 **Boolean**으로 반환합니다.
- 예시 쿼리

```
mutation {
  soldoutItem(key: "W123A1A")
}
```

```
{
  "data": true
}
```

- 예시 코드:
 - [example1 - 단일 상품 품절시키기](#)

category

오너클랜의 단일 카테고리 정보를 조회합니다.

- 참고사항:
 - 모든 최상위 카테고리의 상위 카테고리는 **"ROOT"**라는 이름을 가지며, 코드는 **"000000000"**입니다. 이를 이용해 최상위 카테고리만 전부 가져오거나, 모든 카테고리를 가져올 수 있습니다. 자세한 내용은 아래 예시 코드를 참고해주세요.
- 파라미터:
 - key: ID** - 정보를 조회할 카테고리의 key입니다. 실제 타입은 **String**이며, 오너클랜 카테고리 코드와 같습니다.

- 반환 데이터: 카테고리 정보를 반환하며, 다음 데이터에 액세스 할 수 있습니다.
 - **key: String** - 카테고리의 오너클랜 코드입니다.
 - **id: String** - 카테고리의 API ID입니다.
 - **name: String** - 카테고리의 이름입니다. **lang** 파라미터를 가질 수 있으며, 언어를 지정할 수 있습니다. 현재 지원되는 언어는 "ko_KR"입니다. **lang** 파라미터는 생략하면 "ko_KR"로 적용됩니다.
 - **fullName: String** - 현재 카테고리의 전체 이름입니다. 최상단 카테고리의 이름부터 현재 카테고리까지의 이름을 모두 포함합니다.
 - **attributes: [String]** - 카테고리의 속성 정보입니다. 자세한 항목은 아래를 참고해주세요.
 - **parent: Category** - 상위 카테고리(1단계 위)입니다. 상위 카테고리에 대해서는 **key, id, name**만 조회할 수 있습니다.
 - **children: [Category]** - 하위 카테고리(1단계 아래)들입니다. 하위 카테고리 각각에 대해서는 **key, id, name**만 조회할 수 있습니다.
 - **ancestors: [Category]** - 상위 카테고리 전체입니다. 상위 카테고리 각각에 대해서는 **key, id, name**만 조회할 수 있습니다.
 - **descendants: [Category]** - 하위 카테고리 전체이고, pagination을 위한 정보와 카테고리 정보를 동시에 담고 있습니다.
 - **pageInfo:** pagination을 위한 정보입니다.
 - **hasNextPage:** 데이터가 뒤에 더 있는지를 나타냅니다.
 - **hasPreviousPage:** 데이터가 앞에 더 있는지를 나타냅니다.
 - **startCursor:** 현재 페이지의 첫 데이터에 대한 cursor 값입니다. 이 값을 **before** 파라미터에 넘겨주면 정확하게 이전 페이지의 데이터를 가져올 수 있습니다.
 - **endCursor:** 현재 페이지의 마지막 데이터에 대한 cursor 값입니다. 이 값을 **after** 파라미터에 넘겨주면 정확하게 다음 페이지의 데이터를 가져올 수 있습니다.
 - **edges:** 카테고리 데이터입니다.
 - **cursor:** 해당 카테고리의 pagination cursor입니다.
 - **node:** 해당 카테고리의 데이터입니다. 반환되는 데이터는 **category** 쿼리가 반환하는 값들 중 **key, id, name**만 액세스 할 수 있습니다.
- 예시 코드:

```
# 단일 카테고리 조회
query {
  category(key: "50000108") {
    key
    name
    parent {
      key
      name
      parent {
        key
        name
      }
    }
  }
  children {
    key
    name
  }
  ancestors {
```

```

        key
        name
      }
      descendants {
        edges {
          node {
            key
            name
          }
        }
      }
    }
  }
}

```

가상의 최상위 카테고리 정보 가져오기.

```

query {
  category(key: "00000000") {
    key
    name
  }
}

```

실제로 쓰이는 최상위 카테고리의 리스트 가져오기. (가상의 카테고리 이용)
 # 반환된 데이터의 children 필드의 값들이 실제로 쓰이는 최상위 카테고리들입니다.

```

query {
  category(key: "00000000") {
    key
    name
    children {
      key
      name
    }
  }
}

```

전체 카테고리 정보 가져오기(1~100)

```

query {
  category(key: "00000000") {
    descendants(first: 100) {
      pageInfo {
        hasNextPage
        hasPreviousPage
        startCursor
        endCursor
      }
      edges {
        cursor
      }
    }
  }
}

```

```

        node {
            key
            name
        }
    }
}
}
}

```

```

# 전체 카테고리 정보 가져오기(101~200)
query {
  category(key: "00000000") {
    descendants(after: "100", first: 100) {
      pageInfo {
        hasNextPage
        hasPreviousPage
        startCursor
        endCursor
      }
      edges {
        cursor
        node {
          key
          name
        }
      }
    }
  }
}

```

- 예시 코드:
 - [example1](#) - 단일 카테고리 정보 가져오기
 - [example2](#) - 전체 카테고리 가져오기
 - [example3](#) - 전체 카테고리 가져오기 & 페이지네이션

order

단일 주문 정보를 조회합니다.

- 파라미터:
 - **key: String** - 정보를 조회할 주문 내역의 key입니다. 오너클랜 주문 코드와 같습니다.
- 반환 데이터: 주문 정보를 반환하며, 다음 내용을 액세스 할 수 있습니다.
 - **key: String** - 주문 내역의 key입니다. 오너클랜 주문 코드와 같습니다.
 - **id: String** - 주문 내역의 API ID입니다.
 - **products: [OrderProduct]** - 주문 제품 정보의 배열입니다. 하위 필드로 조회할 수 있는 내역은 아래와 같습니다.
 - **quantity: Int** - 주문 제품의 수량입니다.

- **price: Float** - 수량이 반영되지 않은, 주문 제품의 가격입니다. **currency** 파라미터를 가질 수 있으며, 현재 지원되는 **Currency**는 **KRW**입니다.
- **sourceAddress: Object** - 상품에 설정된 반품지 주소입니다. 하위 필드로 조회할 수 있는 내역은 아래와 같습니다.
 - **addr1: String** - 기본 주소입니다.
 - **addr2: String** - 상세 주소입니다.
 - **postalCode: String** - 우편번호입니다.
- **shippingType: ShippingType** - 배송비 결제 방식입니다. 가능한 값들은 *데이터 타입* - *ShippingType* 항목을 참고해주시면 됩니다.
- **itemKey: String** - 주문 제품의 상품 key입니다.
- **productName: String** - 주문 제품의 주문 당시 이름입니다.
- **itemOptionInfo: Object** - 주문 제품의 옵션 정보입니다. 하위 필드로 조회할 수 있는 내역은 아래와 같습니다.
 - **optionAttributes: [Object]** - 상품의 옵션 정보입니다.
 - **name: String** - 옵션명입니다.
 - **value: String** - 옵션값입니다.
 - **price: Float** - 주문 제품의 옵션 가격입니다. 옵션 추가금액이 아닌, 추가금액이 반영된 전체 금액이므로 참고해주시면 됩니다.
 - **shippingName: String** - 택배 송장 상품명입니다.
- **trackingNumber: String** - 택배 운송장 번호입니다.
- **trackingCompanyCode: String** - 택배사 코드입니다.
- **shippingCompanyName: String** - 택배사 이름입니다.
- **shippedDate: Int** - 택배 운송장 입력 일시입니다. Unix timestamp로 주어집니다.
- **additionalAttributes: [Object]** - 입력형 옵션을 포함한, 추가 속성 정보입니다.
 - **name: String** - 속성 이름입니다.
 - **value: String** - 속성 값입니다.
- **taxFree: Boolean** - 면세상품인지의 여부입니다.
- **status: OrderStatus** - 주문 상태입니다. 가능한 값들은 *데이터 타입* - *OrderStatus* 항목을 참고해주시면 됩니다.
- **shippingInfo: ShippingInfo** - 배송 정보입니다. 하위 필드로 조회할 수 있는 내역은 아래와 같습니다.
 - **sender: Sender** - 보내는 사람의 정보입니다. 판매사 계정의 정보와 동일합니다.
 - **name: String** - 보내는 사람의 이름입니다.
 - **phoneNumber: String** - 보내는 사람의 연락처입니다.
 - **recipient: Recipient** - 받는 사람의 정보입니다.
 - **name: String** - 받는 사람의 이름입니다.
 - **phoneNumber: String** - 받는 사람의 연락처입니다.
 - **destinationAddress: Address** - 받는 주소입니다. **Address**에 대한 내용은 *데이터 타입* - *Address* 항목을 참고해주시면 됩니다.
 - **customsClearanceCode: String** - 개인 고유 통관
 - **shippingFee: Float** - 배송비입니다.
- **createdAt: Int** - 주문이 DB에 등록된 시각입니다. Unix timestamp로 주어집니다.
- **updatedAt: Int** - 주문이 최종적으로 업데이트 된 시각입니다. Unix timestamp로 주어집니다.
- **ordererNote: String** - 최종 소비자가 요청한 주문 메모입니다. 오너클랜 웹사이트 주문내역 조회 페이지의 *고객메모* 항목에 해당합니다.
- **isBeingMediated: Boolean** - 중재중인 주문인지의 여부입니다.

- **adjustments:** [Adjustment] - 조정 내역이 있는 경우 조정 내역입니다. Adjustment에 대한 내용은 *데이터 타입 - Adjustment* 항목을 참고해주시면 됩니다.
- **transactions:** [Transaction] - 기본 상품대 및 배송비 정산 이외에 생성된 정산 관련 내역입니다. Transaction에 대한 내용은 *데이터 타입 - Transaction* 항목을 참고해주시면 됩니다.
- 예시 쿼리:

```

query {
  order(key: "2019000000000000000A") {
    key
    id
    products {
      quantity
      price
      sourceAddress {
        addr1
        addr2
        postalCode
      }
      shippingType
      itemKey
      productName
      itemOptionInfo {
        optionAttributes {
          name
          value
        }
        price
      }
      trackingNumber
      shippingCompanyCode
      shippingCompanyName
      shippedDate
      additionalAttributes {
        key
        value
      }
      taxFree
    }
    status
    shippingInfo {
      sender {
        name
        phoneNumber
      }
      recipient {
        name
        phoneNumber
        destinationAddress {
          addr1
          addr2
          postalCode
        }
      }
    }
  }
}

```

```

        customsClearanceCode
    }
    shippingFee
}
createdAt
updatedAt
ordererNote
isBeingMediated
adjustments {
    reason
    costPrice
    taxFree
}
transactions {
    key
    id
    kind
    status
    amount {
        currency
        value
    }
    createdAt
    updatedAt
    closedAt
    note
}
}
}

```

- 예시 코드:
 - [example1 - 단일 주문내역 조회하기](#)

allOrders

복수의 주문내역을 조회합니다. 한 번에 최대 1000개의 주문 내역을 조회할 수 있으며, pagination을 위한 cursor를 제공합니다. 자세한 사용법은 하단 예시를 참고하면 됩니다.

- 파라미터:
 - Pagination 관련 파라미터:
 - **after: String** - API 고유 cursor 값으로, 지정된 경우 이 cursor 값 이후의 주문내역만을 불러옵니다.
 - **before: String** - API 고유 cursor 값으로, 지정된 경우 이 cursor 값 이전의 주문내역만을 불러옵니다.
 - **first: Int** - 조건을 만족하는 모든 주문내역들 중 처음 몇 개의 주문내역을 불러올 지를 나타냅니다.
 - **last: Int** - 조건을 만족하는 모든 주문내역들 중 마지막 몇 개의 주문내역을 불러올 지를 나타냅니다.
 - Search 관련 파라미터:

- **dateFrom: Timestamp** - 주문 내역이 생성된 시각이 이 값 이후인 것들만을 불러옵니다. Unix timestamp이며, 생략하면 90일 전으로 설정됩니다.
 - **dateTo: Timestamp** - 주문 내역이 생성된 시각이 이 값 이전인 것들만을 불러옵니다. Unix timestamp이며, 생략하면 현재로 설정됩니다.
 - **status: OrderStatus** - 주어진 상태에 해당하는 주문내역만을 불러옵니다. **OrderStatus**의 가능한 값들은 아래를 참고해주세요.
 - **shippedAfter: Timestamp** - 송장번호가 입력된 시각이 이 값 이후인 것들만을 불러옵니다. Unix timestamp이며, 생략하면 90일 전으로 설정됩니다.
 - **shippedBefore: Timestamp** - 송장번호가 입력된 시각이 이 값 이전인 것들만을 불러옵니다. Unix timestamp이며, 생략하면 현재로 설정됩니다.
- 반환 값: **allOrders**는 pagination을 위한 정보와 주문내역 정보를 동시에 담고 있습니다.
 - **pageInfo**: pagination을 위한 정보입니다.
 - **hasNextPage**: 데이터가 뒤에 더 있는지를 나타냅니다.
 - **hasPreviousPage**: 데이터가 앞에 더 있는지를 나타냅니다.
 - **startCursor**: 현재 페이지의 첫 데이터에 대한 cursor 값입니다. 이 값을 **before** 파라미터에 넘겨주면 정확하게 이전 페이지의 데이터를 가져올 수 있습니다.
 - **endCursor**: 현재 페이지의 마지막 데이터에 대한 cursor 값입니다. 이 값을 **after** 파라미터에 넘겨주면 정확하게 다음 페이지의 데이터를 가져올 수 있습니다.
 - **edges**: 주문내역 데이터입니다.
 - **cursor**: 해당 주문내역의 pagination cursor입니다.
 - **node**: 주문내역에 대한 정보입니다. **order** 쿼리에서 노출하는 정보와 동일합니다.
- 예시 쿼리:

```
# 검색 없이 기본 조건만으로 가져오는 쿼리입니다.
query {
  allOrders {
    edges {
      node {
        key
        id
        products {
          quantity
          price
          sourceAddress {
            addr1
            addr2
            postalCode
          }
        }
        shippingType
        itemKey
        productName
        itemOptionInfo {
          optionAttributes {
            name
            value
          }
        }
        price
      }
    }
  }
}
```

```
    }
    trackingNumber
    shippingCompanyCode
    shippingCompanyName
    shippedDate
    additionalAttributes {
      key
      value
    }
    taxFree
  }
  status
  shippingInfo {
    sender {
      name
      phoneNumber
    }
    recipient {
      name
      phoneNumber
      destinationAddress {
        addr1
        addr2
        postalCode
      }
      customsClearanceCode
    }
    shippingFee
  }
  createdAt
  updatedAt
  ordererNote
  isBeingMediated
  adjustments {
    reason
    costPrice
    taxFree
  }
  transactions {
    key
    id
    kind
    status
    amount {
      currency
      value
    }
    createdAt
    updatedAt
    closedAt
    note
  }
}
```



```
}
}
```

- 예시 코드:
 - example1 - 기본 검색
 - example2 - 날짜 범위 검색
 - example3 - 날짜 범위 및 주문 상태 검색
 - example4 - 페이지네이션
 - example5 - 송장번호 입력 날짜 범위 검색

checkOrder

결제 완료상태인 주문건을 발송 준비중 상태로 변경합니다. 오너클랜 웹사이트에서 주문서를 확인하는 동작과 동일한 기능입니다.

- 파라미터:
 - key: String** - 발송 준비중으로 변경할 주문의 Key입니다. 오너클랜 주문코드와 동일합니다.
- 반환 값: 발송 준비중 상태로 변경된 주문 정보를 반환합니다. **order** 쿼리의 반환 결과와 같습니다.
- 예시 쿼리:

```
mutation {
  checkOrder(key: "20200220000000000000A") {
    key
    id
    products {
      quantity
      price
      sourceAddress {
        addr1
        addr2
        postalCode
      }
      shippingType
      itemKey
      productName
      itemOptionInfo {
        optionAttributes {
          name
          value
        }
        price
      }
      trackingNumber
      shippingCompanyCode
      shippingCompanyName
      shippedDate
      additionalAttributes {
        key
        value
      }
    }
  }
}
```

```
    taxFree
  }
  status
  shippingInfo {
    sender {
      name
      phoneNumber
    }
    recipient {
      name
      phoneNumber
      destinationAddress {
        addr1
        addr2
        postalCode
      }
      customsClearanceCode
    }
    shippingFee
  }
  createdAt
  updatedAt
  ordererNote
  isBeingMediated
  adjustments {
    reason
    costPrice
    taxFree
  }
  transactions {
    key
    id
    kind
    status
    amount {
      currency
      value
    }
    createdAt
    updatedAt
    closedAt
    note
  }
}
```

- 예시 코드:
 - [example1](#) - 발송 준비중 상태로 변경

setTrackingInfo

주문건에 대해 택배 운송장 번호를 입력합니다. 단일 주문에 대해 입력할 수 있으며, 이 method를 통해 입력한 결과는 해당 주문의 모든 주문 제품에 적용됩니다.

- 파라미터:
 - **key: String** - 운송장 번호를 입력할 주문의 Key입니다. 오너클랜 주문코드와 동일합니다.
 - **input: Object** - 운송장 정보입니다. 입력해야 하는 필드는 아래와 같습니다.
 - **shippingCompanyCode: String** - 택배사 코드입니다. 코드 목록은 *택배사 코드 정보* 항목을 참고하면 됩니다.
 - **trackingNumber: String** - 운송장 번호입니다. 숫자 이외의 문자는 모두 삭제된 상태로 입력됩니다.
- 반환 값: 운송장 번호가 등록된 주문 정보를 반환합니다. **order** 쿼리의 반환 결과와 같습니다.
- 예시 쿼리: 아래 2개의 코드 중 첫 번째는 실제 쿼리이고, 두 번째는 쿼리에서 쓰이는 **\$input** variable을 정의한 것입니다. GraphQL Playground에서 테스트할 때는 쿼리를 입력하는 공간 아래에 **QUERY VARIABLES**에 입력하면 됩니다.

```
mutation SetTrackingInfo($input: TrackingInfoInput!) {
  setTrackingInfo(key: "2020022000000000000A", input: $input) {
    key
    id
    products {
      quantity
      price
      sourceAddress {
        addr1
        addr2
        postalCode
      }
      shippingType
      itemKey
      productName
      itemOptionInfo {
        optionAttributes {
          name
          value
        }
        price
      }
      trackingNumber
      shippingCompanyCode
      shippingCompanyName
      shippedDate
      additionalAttributes {
        key
        value
      }
      taxFree
    }
    status
    shippingInfo {
      sender {
        name
      }
    }
  }
}
```

```

        phoneNumber
      }
      recipient {
        name
        phoneNumber
        destinationAddress {
          addr1
          addr2
          postalCode
        }
        customsClearanceCode
      }
      shippingFee
    }
    createdAt
    updatedAt
    ordererNote
    isBeingMediated
    adjustments {
      reason
      costPrice
      taxFree
    }
    transactions {
      key
      id
      kind
      status
      amount {
        currency
        value
      }
      createdAt
      updatedAt
      closedAt
      note
    }
  }
}

```

```

{
  "input": {
    "shippingCompanyCode": "3",
    "trackingNumber": "1234567891011"
  }
}

```

- 예시 코드:
 - [example1 - 택배 송장 입력](#)

setOrderProductsTrackingInfo

주문건의 주문 제품 각각에 대해 택배 운송장 번호를 입력합니다. 단일 주문에 대해 입력할 수 있으며, 이 method를 통해 입력한 결과는 해당하는 주문 제품에만 적용됩니다.

- 파라미터:
 - **key:** *String* - 운송장 번호를 입력할 주문의 Key입니다. 오너클랜 주문코드와 동일합니다.
 - **input:** *[Object]* - 운송장 정보입니다. 입력해야 하는 필드는 아래와 같습니다.
 - **shippingCompanyCode:** *String* - 택배사 코드입니다. 코드 목록은 *택배사 코드 정보* 항목을 참고하면 됩니다.
 - **trackingNumber:** *String* - 운송장 번호입니다. 숫자 이외의 문자는 모두 삭제된 상태로 입력됩니다.
 - **itemKey:** *String* - 운송장 번호를 입력할 대상 주문 제품의 상품 KEY입니다.
 - **optionAttributes:** *[String]* - 운송장 번호를 입력할 대상 주문 제품의 옵션 정보입니다. 단일 옵션 상품인 경우 빈 배열을, 옵션이 있는 상품의 경우 옵션1부터 차례로 선택 값을 넣으면 됩니다.
- 반환 값: 운송장 번호가 등록된 주문 정보를 반환합니다. *order* 쿼리의 반환 결과와 같습니다.
- 예시 쿼리: 아래 2개의 코드 중 첫 번째는 실제 쿼리이고, 두 번째는 쿼리에서 쓰이는 *\$input* variable을 정의한 것입니다. GraphQL Playground에서 테스트할 때는 쿼리를 입력하는 공간 아래에 *QUERY VARIABLES*에 입력하면 됩니다.

```
mutation SetOrderProductsTrackingInfo($input: [OrderproductTrackingInfoInput!]) {
  setOrderProductsTrackingInfo(key: "20200220000000000000A", input: $input) {
    key
    id
    products {
      quantity
      price
      sourceAddress {
        addr1
        addr2
        postalCode
      }
      shippingType
      itemKey
      productName
      itemOptionInfo {
        optionAttributes {
          name
          value
        }
        price
      }
      trackingNumber
      shippingCompanyCode
      shippingCompanyName
      shippedDate
      additionalAttributes {
        key
        value
      }
    }
  }
}
```

```

    }
    taxFree
  }
  status
  shippingInfo {
    sender {
      name
      phoneNumber
    }
    recipient {
      name
      phoneNumber
      destinationAddress {
        addr1
        addr2
        postalCode
      }
      customsClearanceCode
    }
    shippingFee
  }
  createdAt
  updatedAt
  ordererNote
  isBeingMediated
  adjustments {
    reason
    costPrice
    taxFree
  }
  transactions {
    key
    id
    kind
    status
    amount {
      currency
      value
    }
    createdAt
    updatedAt
    closedAt
    note
  }
}

```

```

{
  "input": [
    {
      "trackingNumber": "1234567891011",

```

```

    "shippingCompanyCode": "3",
    "itemKey": "W000000",
    "optionAttributes": ["XL", "red"]
  }
]
}

```

- 예시 코드:
 - example1 - 개별 주문 제품 택배 송장 입력

acceptOrderCancellationRequest

주문 취소 요청을 수락합니다.

- 파라미터:
 - key: String - 주문 취소 요청을 수락할 주문의 주문코드입니다.
- 반환 데이터: 변경된 주문 정보를 반환합니다. order 쿼리의 반환 결과와 같습니다.
- 예시 쿼리: 아래 2개의 코드 중 첫 번째는 실제 쿼리이고, 두 번째는 쿼리에서 쓰이는 \$key variable을 정의한 것입니다. GraphQL Playground에서 테스트할 때는 쿼리를 입력하는 공간 아래에 QUERY VARIABLES에 입력하면 됩니다.(예시 데이터이므로 그대로 붙여넣어도 작동하지는 않습니다.)

```

mutation AcceptOrderCancellationRequest($key: ID!) {
  acceptOrderCancellationRequest(key: $key) {
    key
    id
    products {
      quantity
      price
      sourceAddress {
        addr1
        addr2
        postalCode
      }
      shippingType
      itemKey
      productName
      itemOptionInfo {
        optionAttributes {
          name
          value
        }
        price
      }
      trackingNumber
      shippingCompanyCode
      shippingCompanyName
      shippedDate
      additionalAttributes {
        key
        value
      }
    }
  }
}

```

```
    taxFree
  }
  status
  shippingInfo {
    sender {
      name
      phoneNumber
    }
    recipient {
      name
      phoneNumber
      destinationAddress {
        addr1
        addr2
        postalCode
      }
      customsClearanceCode
    }
    shippingFee
  }
  createdAt
  updatedAt
  ordererNote
  isBeingMediated
  adjustments {
    reason
    costPrice
    taxFree
  }
  transactions {
    key
    id
    kind
    status
    amount {
      currency
      value
    }
    createdAt
    updatedAt
    closedAt
    note
  }
}
```

```
{
  "key": "2020000000000000000A"
}
```


- 예시 코드:
 - [example1 - 주문 취소 요청 수락](#)

rejectOrderCancellationRequest

주문 취소 요청을 거절합니다.

- 파라미터:
 - **key: String** - 주문 취소 요청을 거절할 주문의 주문코드입니다.
 - **input: object(RejectOrderCancellationInput)** - 주문 취소 요청 거절시 필요한 데이터입니다.
 - **rejectReason: String** - 주문 취소 요청 거절 사유입니다.
 - **trackingInfo: object(RejectOrderCancellationTrackingInfo)** - 주문 취소 요청 거절 건에 대한 배송 정보입니다.
 - **shippingCompanyCode: String** - 택배사 코드입니다. 코드 목록은 [택배사 코드 정보](#) 항목을 참고하면 됩니다.
 - **trackingNumber: String** - 운송장 번호입니다.
- 반환 데이터: 변경된 주문 정보를 반환합니다. **order** 쿼리의 반환 결과와 같습니다.
- 예시 쿼리: 아래 2개의 코드 중 첫 번째는 실제 쿼리이고, 두 번째는 쿼리에서 쓰이는 **\$key**와 **\$input variable**을 정의한 것입니다. GraphQL Playground에서 테스트할 때는 쿼리를 입력하는 공간 아래에 **QUERY VARIABLES**에 입력하면 됩니다.(예시 데이터이므로 그대로 붙여넣어도 작동하지는 않습니다.)

```
mutation RejectOrderCancellationRequest($key: ID!, $input:
RejectOrderCancellationInput!) {
  rejectOrderCancellationRequest(key: $key, input: $input) {
    key
    id
    products {
      quantity
      price
      sourceAddress {
        addr1
        addr2
        postalCode
      }
      shippingType
      itemKey
      productName
      itemOptionInfo {
        optionAttributes {
          name
          value
        }
        price
      }
      trackingNumber
      shippingCompanyCode
      shippingCompanyName
      shippedDate
      additionalAttributes {
```

```

        key
        value
    }
    taxFree
}
status
shippingInfo {
    sender {
        name
        phoneNumber
    }
    recipient {
        name
        phoneNumber
        destinationAddress {
            addr1
            addr2
            postalCode
        }
        customsClearanceCode
    }
    shippingFee
}
createdAt
updatedAt
ordererNote
isBeingMediated
adjustments {
    reason
    costPrice
    taxFree
}
transactions {
    key
    id
    kind
    status
    amount {
        currency
        value
    }
    createdAt
    updatedAt
    closedAt
    note
}
}
}

```

```

{
  "key": "2020000000000000000A",

```

```

"input": {
  "rejectReason": "주문 취소 요청 거절 사유.",
  "trackingInfo": {
    "shippingCompanyCode": "3",
    "trackingNumber": "1234567891011"
  }
}
}

```

- 예시 코드:
 - [example1](#) - 주문 취소 요청 거절

vendorQnaArticle

단일 공급사 문의 게시판 글을 조회합니다.

- 파라미터:
 - **key: String** - 정보를 조회할 공급사 문의 게시판 글의 key입니다.
- 반환 데이터: 공급사 문의 게시판 글의 정보를 반환하며, 다음 내용을 액세스 할 수 있습니다.
 - **key: String** - 공급사 문의 게시판 글의 key입니다.
 - **id: String** - 공급사 문의 게시판 글의 API ID입니다.
 - **type: VendorQnaType** - 공급사 문의 게시판 글의 타입입니다. 가능한 값들은 *데이터 타입* - *VendorQnaType*을 참고해주시면 됩니다.
 - **title: String** - 글의 제목입니다.
 - **content: String** - 글의 내용입니다.
 - **files: [String]** - 첨부파일의 URL 리스트입니다.
 - **relatedItemKey: String** - 연관 상품이 있는 경우 상품코드입니다.
 - **relatedOrderKey: String** - 연관 주문이 있는 경우 주문코드입니다.
 - **authorType: VendorQnaAuthorType** - 게시자 타입입니다. 가능한 값은 *데이터 타입* - *VendorQnaAuthorType* 항목을 참고해주시면 됩니다.
 - **createdAt: Timestamp** - 등록 시각입니다.
 - **repliedAt: Timestamp** - 답변이 있는 경우, 답변 등록 시각입니다.
 - **reply: String** - 답변이 있는 경우, 답변 내용입니다.
 - **replierType: VendorQnaAuthorType** - 답변이 있는 경우, 답변을 작성한 사용자 타입입니다. 가능한 값은 *데이터 타입* - *VendorQnaAuthorType* 항목을 참고해주시면 됩니다.
- 예시 쿼리: 아래 2개의 코드 중 첫 번째는 실제 쿼리이고, 두 번째는 쿼리에서 쓰이는 **\$key** variable을 정의한 것입니다. GraphQL Playground에서 테스트할 때는 쿼리를 입력하는 공간 아래에 **QUERY VARIABLES**에 입력하면 됩니다.

```

query VendorQnaArticle($key: ID!) {
  vendorQnaArticle(key: $key) {
    key
    id
    type
    title
    content
    files
    relatedItemKey
  }
}

```

```

    relatedOrderKey
    authorType
    createdAt
    repliedAt
    reply
    replierType
  }
}
```

```

{
  "key": "1000000"
}
```

- 예시 코드:
 - [example1 - 단일 공급사 문의 게시판 글 조회](#)

allVendorQnaArticles

복수의 공급사 문의 게시판 글 목록을 조회합니다. 검색 조건을 설정한 경우 검색 조건에 부합하는 글만 조회합니다.

- 파라미터:
 - Pagination 관련 파라미터:
 - after: String** - API 고유 cursor 값으로, 지정된 경우 이 cursor 값 이후의 글만을 불러옵니다.
 - before: String** - API 고유 cursor 값으로, 지정된 경우 이 cursor 값 이전의 글만을 불러옵니다.
 - first: Int** - 조건을 만족하는 모든 내역들 중 처음 몇 개의 글을 불러올 지를 나타냅니다.
 - last: Int** - 조건을 만족하는 모든 내역들 중 마지막 몇 개의 글을 불러올 지를 나타냅니다.
 - Search 관련 파라미터:
 - search: [VendorQnaSearch]**: 텍스트 검색을 지원합니다.
 - value: String** - 검색할 값입니다.
 - by: VendorQnaSearchBy**: 검색할 필드입니다. 가능한 필드는 **title**, **content**, **orderKey**, **itemKey**입니다.
 - type: VendorQnaType**: 주어진 타입인 문의 글만을 불러옵니다. 가능한 값은 *데이터 타입 - VendorQnaType* 항목을 참고해주시면 됩니다.
 - authorType: VendorQnaAuthorType** - 글 게시자 타입으로 검색합니다. 가능한 값은 *데이터 타입 - VendorQnaAuthorType* 항목을 참고해주시면 됩니다.
 - dateFrom: Timestamp** - 글이 등록된 일자가 이 값 이후인 것들만을 불러옵니다. Unix timestamp이며, 생략하면 7일 전으로 설정됩니다. 최대 30일 간격으로 조회가 가능합니다.
 - dateTo: Timestamp** - 글이 등록된 일자가 이 값 이전인 것들만을 불러옵니다. Unix timestamp이며, 생략하면 현재로 설정됩니다. 최대 30일 간격으로 조회가 가능합니다.
- 반환 값: **allSellerQnaArticles**는 pagination을 위한 정보와 내용에 대한 정보를 동시에 담고 있습니다.
 - pageInfo**: pagination을 위한 정보입니다.
 - hasNextPage**: 데이터가 뒤에 더 있는지를 나타냅니다.

- **hasPreviousPage**: 데이터가 앞에 더 있는지를 나타냅니다.
 - **startCursor**: 현재 페이지의 첫 데이터에 대한 cursor 값입니다. 이 값을 **before** 파라미터에 넘겨주면 정확하게 이전 페이지의 데이터를 가져올 수 있습니다.
 - **endCursor**: 현재 페이지의 마지막 데이터에 대한 cursor 값입니다. 이 값을 **after** 파라미터에 넘겨주면 정확하게 다음 페이지의 데이터를 가져올 수 있습니다.
- **edges**: 1:1 문의 글 데이터입니다.
 - **cursor**: 해당 글의 pagination cursor입니다.
 - **node**: 해당 글의 내용입니다. 타입은 **vendorQnaArticle** 쿼리의 반환 결과와 동일합니다.
- 예시 쿼리:

```
# 기본 검색. 7일전 내역까지 최대 100건 조회.
query AllVendorQnaArticles {
  allVendorQnaArticles() {
    pageInfo {
      hasNextPage
      hasPreviousPage
      startCursor
      endCursor
    }
    edges {
      cursor
      node {
        key
        id
        type
        title
        content
        files
        relatedItemKey
        relatedOrderKey
        authorType
        createdAt
        repliedAt
        reply
        replierType
      }
    }
  }
}
```

- 예시 코드:
 - [example1](#) - 기본 조건 조회
 - [example2](#) - 14일간 내역 조회
 - [example3](#) - 제목 검색
 - [example4](#) - 게시자 타입 검색
 - [example5](#) - 문의글 타입 검색

createDirectTransferSellerQnaArticleReply

이관된 판매사 1:1 문의에 답변을 작성합니다.

- 파라미터:
 - **input**: `object(DirectTransferSellerQnaArticleReplyInput)` - 입력할 데이터입니다.
 - **answerType**: `SellerQnaDirectAnswerType` - 답변 타입입니다. 가능한 값은 *데일리 타입* - `SellerQnaDirectAnswerType` 항목을 참고해주시면 됩니다.
 - **key**: `String` - 이관된 판매사 1:1 문의글의 key입니다.
- 입력 데이터 (굵은 글씨는 필수 필드입니다):
 - **deliveryInfo**: `String` - 출고 관련 기타 정보입니다.
 - **estimatedShippingDay**: `EstimatedShippingDayType` - 출고 예정일 타입입니다. 가능한 값은 *데일리 타입* - `EstimatedShippingDayType` 항목을 참고해주시면 됩니다.
 - **deliveryUnavailableReason**: `DeliveryUnavailableReasonType` - 출고 불가 사유 타입입니다. 가능한 값은 *데일리 타입* - `DeliveryUnavailableReasonType` 항목을 참고해주시면 됩니다.
 - **content**: `String` - 기타 답변 내용입니다.
- 반환 데이터:
 - 답변을 작성한 1:1 문의글을 반환합니다.
- 예시 쿼리:

```
mutation CreateDirectTransferSellerQnaArticleReply($key: ID!, $answerType:
SellerQnaDirectAnswerType!, $input: DirectTransferSellerQnaArticleReplyInput!) {
  createDirectTransferSellerQnaArticleReply(key: $key, answerType: $answerType,
input: $input) {
    id
    key
    type
    isSecret
    title
    content
    files
    relatedItemKey
    relatedOrderKey
    createdAt
    recipientName
    subArticles {
      id
      key
      type
      isSecret
      title
      content
      files
      relatedItemKey
      relatedOrderKey
      createdAt
      recipientName
      comments
    }
    parentArticles {
      id
      key
```

```

    type
    isSecret
    title
    content
    files
    relatedItemKey
    relatedOrderKey
    createdAt
    recipientName
    comments
  }
  comments
}
}

```

```

{
  "key": "5000000",
  "answerType": "Etc",
  "input": {
    "content": "답변 내용"
  }
}

```

- 예시 코드:
 - [example](#) - 이관된 판매사 1:1 문의 답변 작성

createVendorQnaArticle

공급사 1:1 문의 게시판에 글을 작성합니다.

- 파라미터:
 - **input**: `object(VendorQnaArticleInput)` - 입력할 데이터입니다.
- 입력 데이터 (굵은 글씨는 필수 필드입니다):
 - **type**: `VendorQnaType` - 문의글 타입입니다. 가능한 값은 *데이터 타입* - `VendorQnaType` 항목을 참고해주시면 됩니다.
 - **title**: `String` - 글 제목입니다.
 - **content**: `String` - 문의 내용입니다.
 - **files**: `[Upload]` - 첨부파일의 리스트입니다.
 - **relatedItemKey**: `String` - 연관 상품이 있는 경우 상품의 key입니다.
 - **relatedOrderKey**: `String` - 연관 주문이 있는 경우 주문의 key입니다.
- 반환 데이터:
 - 새로 생성된 문의글을 반환합니다.
- 예시 쿼리:

```

mutation CreateVendorQnaArticle($input: VendorQnaArticleInput!) {
  createVendorQnaArticle(input: $input) {
    id
  }
}

```

```

    key
    type
    title
    content
    files
    relatedItemKey
    relatedOrderKey
    createdAt
    updatedAt
    authorType
    authorKey
    repliedAt
    reply
    repliedByKey
    replierType
  }
}
```

```

{
  "input": {
    "type": "item",
    "title": "제목",
    "content": "문의 내용",
    "files": [Upload!],
    "relatedItemKey": "w999999",
    "relatedOrderKey": "20200101000000000000A"
  }
}
```

- 예시 코드:
 - [example - 공급사 1:1 문의 작성](#)

데이터 형식

ShippingType

배송비 유형은 3가지로 구분됩니다. API에서는 아래 3가지 값들 중 하나를 사용하고, 반환하게 됩니다.

- **inAdvance**: 선불 전용 상품
- **uponArrival**: 착불 전용 상품
- **free**: 무료 배송 상품

ItemStatus

상품의 현재 상태를 나타내는 방식입니다.

- **soldout**: 일시 품절 상태입니다.
- **available**: 판매 가능 상태입니다.
- **unavailable**: 판매 불가인 상품입니다. 단, 재입고되거나 재판매 될 수 있습니다.

- **discontinued**: 단종된 상품이고, 재입고 되지 않는 상품입니다.

ItemSortCriteria

상품 정렬 기준입니다.

- **dateDesc**: 상품 수정일 기준 내림차순 정렬입니다.
- **dateAsc**: 상품 수정일 기준 오름차순 정렬입니다.
- **default**: key 기준 기본 정렬입니다.
- **nameAsc**: 상품명 기준 오름차순 정렬입니다.
- **nameDesc**: 상품명 기준 내림차순 정렬입니다.
- **priceAsc**: 가격 기준 오름차순 정렬입니다.
- **priceDesc**: 가격 기준 내림차순 정렬입니다.
- **registerDateAsc**: 상품 등록일 기준 오름차순 정렬입니다.
- **registerDateDesc**: 상품 등록일 기준 내림차순 정렬입니다.
- **keyAsc**: key 기준 오름차순 정렬입니다.
- **keyDesc**: key 기준 내림차순 정렬입니다.

ShippingLocationType

상품 배송지 타입입니다.

- **domestic**: 국내 배송 상품입니다.
- **overseas**: 해외 배송 상품입니다.

ImageSize

상품 이미지 사이즈입니다. **auto**, **xlarge**, **large**, **medium**, **small**의 3가지 사이즈를 제공합니다.

- **auto** - 원본 이미지 사이즈를 기반으로 최적화된 사이즈의 이미지를 제공합니다.
- **xlarge** - 가로 1000px, 세로 1000px의 이미지.
- **large** - 가로 640px, 세로 640px의 이미지.
- **medium** - 가로 300px, 세로 300px의 이미지.
- **small** - 가로 100px, 세로 100px의 이미지.

Currency

화폐 단위 타입입니다. ISO 4217에 따른 값을 지원하며, 현재 지원하는 값의 목록은 다음과 같습니다.

- **KRW**: 한국 원화 단위입니다.
- **USD**: 미국 달러화 단위입니다.
- **CNY**: 중국 위안화 단위입니다.

다만 현재 API에서는 기본값이 **KRW**이며, **KRW** 이외의 단위로 명시된 경우에는 null을 반환합니다.

Language

언어 타입입니다. ISO 639-1과 ISO 3166-1 alpha-2를 조합한 형태의 값을 지원하며, 현재 지원하는 값의 목록은 다음과 같습니다.

- **ko_KR**: 대한민국 - 한국어

- `en_US`: 미국 - 영어
- `zh_CN`: 중국 - 중국어

다만 현재 API에서는 기본값이 `ko_KR`이며, `ko_KR` 이외의 코드로 명시된 경우에는 `null`을 반환합니다.

PricePolicy

가격 정책 타입입니다.

- `free`: 가격 자율 제품입니다.
- `fixed`: 소비자 준수 가격이 있는 제품입니다. 반드시 `Item.fixedPrice` 필드를 확인해주시기 바랍니다.

Text

텍스트 정보를 보여주는 형식입니다. 다국어 지원이 가능한 항목에 대해서 최대 3가지 언어(`ko_KR`, `en_US`, `zh_CN`)를 지원합니다. `ko_KR` 이외의 언어의 경우 현재 오너클랜 API에서 지원하지 않습니다. 요청하더라도 `null`이 반환됩니다. 언어는 `lang` 필드의 값으로 지정할 수 있으며, 지정하지 않으면 기본으로 `ko_KR`일 때의 값을 반환합니다.

- 파라미터
 - `lang: Language` - 텍스트 필드의 출력 언어를 지정할 수 있습니다. 기본값은 `ko_KR`이며, 현재 `ko_KR` 이외의 언어는 지원하지 않으나 추후에 다른 언어를 지원할 예정입니다.
- 반환값
 - `String`
- 예시 쿼리

```
# 실제로 테스트 할 수 있는 쿼리는 아닙니다.
query {
  test {
    textField(lang: ko_KR)
  }
}
```

```
{
  "data": {
    "test": {
      "textField": "테스트 필드입니다."
    }
  }
}
```

TextInput

텍스트 정보를 입력하는 형식입니다. 다국어 지원이 가능한 항목에 대해서 최대 3가지 언어(**ko_KR**, **en_US**, **zh_CN**)의 입력을 지원합니다. **ko_KR** 이외의 언어만 입력된 경우, 쿼리가 **autoTranslate** 옵션이 있는 쿼리이고 해당 파라미터가 **true**인 경우에 한해 **ko_KR** 필드의 값은 API에서 자동으로 번역해서 사용합니다.

- 입력
 - **ko_KR**: 텍스트의 한국어 데이터입니다. (쿼리의 **autoTranslate**가 **false**인 경우에는 반드시 입력해야 합니다.)
 - **en_US**: 텍스트의 영어 데이터입니다.
 - **zh_CN**: 텍스트의 중국어 데이터입니다.
- 예시 쿼리 1

```
# 실제로 테스트 할 수 있는 쿼리는 아닙니다.
query {
  test(autoTranslate: true, input: {
    textField {
      en_US: "This is for test."
    }
  }) {
    textField(lang: "en_US")
  }
}
```

```
{
  "data": {
    "test": {
      "textField": "This is for test."
    }
  }
}
```

- 예시 쿼리 2

```
# 실제로 테스트 할 수 있는 쿼리는 아닙니다.
query {
  test(input: {
    textField {
      en_US: "This is for test."
    }
  }) {
    textField
  }
}
```

```
{
  "data": {
    "test": null
  },
  "errors": [
    {
      "message": "Currently, ko_KR attribute of TextInput is required.",
      "path": "..."
    }
  ]
}
```

Price

가격 정보를 위한 형식입니다. **KRW** 외에 **USD**, **CNY**를 지원합니다. **KRW** 이외의 화폐에 대해서는 현재 오너클랜 API에서 지원하지 않습니다.

- 파라미터
 - **currency**: **Currency** - 가격 정보의 화폐 단위를 지정할 수 있습니다. 기본값은 **KRW**와 같습니다.
- 반환값
 - **Int** | **Float**
- 예시 쿼리 1

```
# 실제로 테스트 할 수 있는 쿼리는 아닙니다.
query {
  test {
    price
  }
}
```

```
{
  "data": {
    "test": {
      "price": 10000
    }
  }
}
```

- 예시 쿼리 2

```
# 실제로 테스트 할 수 있는 쿼리는 아닙니다.
query {
```

```

test {
  price_KRW: price(currency: KRW)
  price_USD: price(currency: USD)
  price_CNY: price(currency: CNY)
}

```

```

{
  "data": {
    "test": {
      "price_KRW": 10000,
      "price_USD": 10.00,
      "price_CNY": 55.56
    }
  }
}

```

PriceInput

가격 정보 입력을 위한 형식입니다. 화폐단위와 해당 화폐단위로의 가격을 입력하면 됩니다. **KRW**가 아닌 경우 미리 지정된 환율을 이용해 **KRW** 값을 API에서 자동으로 계산합니다.

- 입력
 - **currency: String** - 화폐 단위입니다. "**KRW**", "**USD**", "**CNY**"(각각 **Currency** 타입의 문자열 표현입니다.)를 지원합니다.
 - **value: Int|Float** - 가격입니다.
- 예시 쿼리

실제로 테스트 할 수 있는 쿼리는 아닙니다.

```

query {
  test(input: {
    price: {
      currency: "KRW"
      value: 15000
    }
  }) {
    price {
      KRW
      USD
    }
  }
}

```

```

{
  "data": {

```

```

    "test": {
      "price": {
        "KRW": 15000,
        "USD": null
      }
    }
  }
}

```

ItemOption

상품 옵션 정보입니다. 오너클랜 API에서는 하나의 옵션 정보를 SKU(Stock Keeping Unit, 재고 관리 단위) 혹은 SKU의 모음으로 가정합니다. 따라서 오너클랜 웹사이트에서 옵션이 없는 것으로 조회되는 상품이라 하더라도, SKU가 있어야하므로 `Item.options` 필드에 값이 1개 제공되며, 옵션이 1개 있는 것으로 조회되는 상품도 `Item.options` 필드에 값이 1개 제공됩니다. 단, 옵션이 없는 것으로 조회되는 경우에는, `Item.options.optionAttributes`에 값이 없는 빈 배열이 반환되고, 옵션이 1개 있는 것으로 조회되는 상품은 `Item.options.optionAttributes`에 값이 있는, 원소 1개인 배열이 반환됩니다. 이를 통해 옵션이 없는 상품과 있는 상품을 구분할 수 있습니다.

- `optionAttributes: [ItemOptionAttribute]` - 옵션 속성에 대한 이름 - 값 배열입니다.
- `price: Price` - 옵션 가격입니다.
- `quantity: Int` - 옵션 재고 수량입니다.

ItemOptionAttribute

상품 옵션에 대한 속성 정보입니다.

- `name: String` - 옵션 속성 이름입니다.
- `value: String` - 옵션 속성 값입니다. 만약 이 필드의 값이 `null`이라면 해당 옵션 속성은 입력형 옵션 속성임을 나타냅니다. 입력형 옵션은 현재 등록 및 수정시에는 사용이 불가능하고(선택형 옵션만 가능) 상품 정보 조회시에만 조회가 가능합니다.

ItemOptionInput

옵션 정보 입력을 위한 형식입니다.

- **참고사항**
 - 옵션이 없더라도, 즉 단일 상품이어도 옵션 1개로 등록해야 합니다. 즉, 단일 상품과 단일 옵션 상품은 구분하지 않습니다.
 - 단일 상품을 등록하고자 하는 경우, `optionAttributes` 필드에 빈 배열을 넣고, `options` 필드의 길이가 1이 되도록 입력하면 됩니다.
 - 현재 API에서는 옵션의 수량은 재고가 있는 경우와 없는 경우만 구분합니다. 즉, `quantity`에 양수 값을 넣어주면 해당 옵션은 판매중인 옵션으로 인식하며, 0을 넣어주면 해당 옵션은 품절된 것으로 인식하고 품절로 처리하게 됩니다.
 - 상품의 품절을 위해서는 모든 옵션의 `quantity`를 0으로 설정해주면 됩니다. 품절된 상품의 재입고를 위해서는 `quantity`가 0보다 큰 숫자인 옵션이 1개 이상만 있으면 됩니다.
 - 또한 옵션의 부분 업데이트(예를 들어 전체 옵션이 5개이나, 이 중 1개를 품절처리하고 2개는 옵션 값을 변경하는 등의 업데이트)는 지원하지 않으므로, 옵션을 변경하고자 한다면, 변경하고자 하는

옵션의 내용과 유지하고자 하는 옵션의 내용을 모두 명시해야 합니다. 기존에 있던 옵션이지만 업데이트 쿼리의 내용에 없는 옵션은 단종으로 처리됩니다.

- **ItemOptionInput**에서 **productId** 필드는 이후 버전에서 쓰일 예정입니다. 문자열로 아무 값이나 넣으면 됩니다.
- **ItemOptionInput**에서 **metadata** 필드는 필요한 경우에만 쓰입니다. 일반적으로는 쓰이지 않는 필드이므로 별도의 요청사항이 없다면 빈 오브젝트 **{}**를 넣어주시면 됩니다.
- 각 옵션의 가격은 자유롭게 설정할 수 있으나, 옵션 가격들의 최저값을 기준으로 100% ~ 150%의 범위 이내(양 끝 포함)로 설정되어야 합니다. 옵션이 없는 상품이나 단일 옵션 상품의 경우에는 신경쓰지 않아도 되나, 옵션이 2개 이상인 경우에 유의해주시기 바랍니다. 예를 들어 옵션이 3개인 상품을 등록할 때 옵션 A가 10000원, 옵션 B가 12000원, 옵션 C가 15000원인 경우는 가능하나(옵션 가격 최저가 = 10000원, 허용 범위 = 10000원 ~ 15000원), 옵션 A가 10000원, 옵션 B가 15000원, 옵션 C가 16000원인 경우는 불가능합니다.(옵션 가격 최저가 = 10000원, 허용 범위 = 10000원 ~ 15000원, 유효하지 않은 옵션 = 옵션 C)

• 입력

- **optionAttributes: [ItemOptionAttributes]** - 옵션 속성의 배열입니다. 필수 필드이며, 옵션이 없는 단일 상품을 등록하는 경우, 빈 배열로 설정하면 됩니다.
- **price: PriceInput** - 옵션의 가격입니다. 반드시 상품의 기준가격에 대한 옵션 증감 금액이 아닌 *전체 가격*을 입력해야 합니다. 예를 들어 10,000원인 옵션과 11,000원인 옵션이 있고 기준 가격을 10,000원이라고 생각하여 옵션 가격을 각각 0원, 1,000원이라고 생각할 수 있지만 반드시 *10,000원, 11,000원*을 입력해야 합니다.
- **quantity: Int** - 옵션 재고 수량입니다. 반드시 입력해야 하며, 음수는 허용하지 않습니다. 0인 옵션은 품절 상태로 처리합니다.
- **productId: String** - 옵션 고유 ID이나 현재는 사용하지 않고 추후 사용할 예정인 값입니다. 빈 문자열을 넣어주시면 됩니다.
- **metadata: Object** - 별도의 요청사항이 있는 경우 사용하는 필드이나 대부분의 경우 사용되지 않습니다. 특별한 경우가 아니라면 빈 오브젝트(**{}**)로 설정하면 됩니다.

• 예시 쿼리

실제로 테스트 할 수 있는 쿼리는 아닙니다.

```
mutation {
  test(input: {
    options: [
      {
        optionAttributes: [
          {
            name: {
              ko_KR: "색상"
            }
            value: {
              ko_KR: "파랑"
            }
          }
        ]
        name: {
          ko_KR: "사이즈"
        }
        value: {
          ko_KR: "95"
        }
      }
    ]
  })
}
```

```
    }
  }
]
price: {
  currency: "KRW"
  value: 20000
}
productId: "test"
quantity: 100
metadata: {}
}
{
  optionAttributes: [
    {
      name: {
        ko_KR: "색상"
      }
      value: {
        ko_KR: "파랑"
      }
    }
    {
      name: {
        ko_KR: "사이즈"
      }
      value: {
        ko_KR: "100"
      }
    }
  ]
  price: {
    currency: "KRW"
    value: 20000
  }
  productId: "test"
  quantity: 100
  metadata: {}
}
{
  optionAttributes: [
    {
      name: {
        ko_KR: "색상"
      }
      value: {
        ko_KR: "빨강"
      }
    }
    {
      name: {
        ko_KR: "사이즈"
      }
      value: {
        ko_KR: "100"
      }
    }
  ]
}
```



```

        }
    }
    ]
    price: {
        currency: "KRW"
        value: 20000
    }
    productId: "test"
    quantity: 100
    metadata: {}
}
{
    optionAttributes: [
        {
            name: {
                ko_KR: "색상"
            }
            value: {
                ko_KR: "빨강"
            }
        }
        {
            name: {
                ko_KR: "사이즈"
            }
            value: {
                ko_KR: "95"
            }
        }
    ]
    price: {
        currency: "KRW"
        value: 20000
    }
    productId: "test"
    quantity: 100
    metadata: {}
}
    ]
}) {
    options
}
}

```

```

{
    "data": {
        "test": {
            "options": [
                {
                    "optionAttributes": [
                        {

```

```
        "name": "옵션1",
        "value": "파랑 95"
    },
],
"price": {
    "KRW": 20000,
    "USD": null
},
"quantity": 1,
"product": {
    "id": "Product,..."
}
},
{
    "optionAttributes": [
        {
            "name": "옵션1",
            "value": "파랑 100"
        },
    ],
    "price": {
        "KRW": 20000,
        "USD": null
    },
    "quantity": 1,
    "product": {
        "id": "Product,..."
    }
},
{
    "optionAttributes": [
        {
            "name": "옵션1",
            "value": "빨강 95"
        },
    ],
    "price": {
        "KRW": 20000,
        "USD": null
    },
    "quantity": 1,
    "product": {
        "id": "Product,..."
    }
},
{
    "optionAttributes": [
        {
            "name": "옵션1",
            "value": "빨강 100"
        },
    ],
    "price": {
        "KRW": 20000,
```

```

        "USD": null
      },
      "quantity": 1,
      "product": {
        "id": "Product,..."
      }
    }
  ]
}
}
}

```

ReturnCriteria

상품의 반품시 반품접수 유형입니다.

- **vendor**: **공급사 반품접수** 유형으로, 공급사가 직접 택배사로 반품 택배를 접수합니다.
- **seller**: **원운송장 반품접수** 유형으로, 판매사가 원운송장에 기재되어 있는 주소로 반품 택배를 접수합니다.

OrderStatus

주문의 상태를 나타냅니다.

- **placed**: **미처리** 상태입니다.
- **paid**: **입금완료** 상태입니다.
- **preparing**: **발송 준비** 상태입니다.
- **cancelRequested**: **취소 요청** 상태입니다.
- **cancelled**: **주문 취소** 상태입니다.
- **shipped**: **발송 완료** 상태입니다.
- **exchangeRequested**: **교환 요청** 상태입니다.
- **exchanged**: **교환 완료** 상태입니다.
- **refundRequested**: **반품 요청** 상태입니다.
- **refundAccepted**: **반품 접수** 상태입니다.
- **refundShipped**: **반송 물품 발송 완료** 상태입니다.
- **refunded**: **반품 완료** 상태입니다.
- **refundClosed**: **반품 완료 후 PG 정산 완료** 상태입니다.

Address

주소 정보입니다. 기본주소와 상세주소, 우편번호 필드로 구성되어 있습니다.

- **addr1**: **String** - 기본주소입니다.
- **addr2**: **String** - 상세주소입니다.
- **postalCode**: **String** - 우편번호입니다.

Adjustment

주문 내역에 대해 조정이 발생한 경우, 조정 내역에 대한 정보입니다.

- **reason: String** - 조정이 발생한 사유입니다.
- **costPrice: String** - 조정 대상 금액입니다.
- **taxFree: Boolean** - 조정 대상 금액에 대한 면세 여부입니다.

Transaction

거래 내역 정보입니다.

- **key: String** - 거래 내역 정보의 key값입니다.
- **id: String** - 거래 내역 정보의 API ID 값입니다.
- **kind: TransactionKind** - 거래 내역 타입입니다. 정산 관련 내역은 **settlement**로 값이 고정되어 있습니다.
- **status: TransactionStatus** - 거래 내역의 상태입니다. 자세한 내용은 **TransactionStatus** 항목을 참고해주세요.
- **amount: Price** - 거래 내역에 해당하는 금액입니다. **currency**와 **value**로 구성되어 있습니다.
- **createdAt: Int** - 거래 내역이 생성된 시각입니다. Unix timestamp로 주어집니다.
- **updatedAt: Int** - 거래 내역의 상태가 최종적으로 변경된 시각입니다. Unix timestamp로 주어집니다.
- **closedAt: Int** - 거래 내역이 마감된 시각입니다. Unix timestamp로 주어집니다. 마감되지 않은 내역의 경우 **null**이 반환됩니다.
- **note: String** - 거래 내역에 대한 상세 정보입니다.

TransactionStatus

거래 내역의 상태입니다.

- **placed: 처리 대기중** 상태입니다.
- **ongoing: 처리중** 상태입니다.
- **closed: 처리 완료** 상태입니다.

VendorQnaType

공급사 1:1 문의 글의 타입입니다.

- **item**: 상품 관련 문의글 타입입니다.
- **order**: 주문 관련 문의글 타입입니다.
- **returnOrRefund**: 반품/교환 관련 문의글 타입입니다.
- **system**: 오퍼클랜 시스템 관련 문의글 타입입니다.
- **scm**: 매출/SCM 이용 방법 관련 문의글 타입입니다.
- **reportOrCertificate**: 신고, 제재 또는 인증 관련 문의글 타입입니다.
- **tax**: 회계, 세금계산서 또는 사업자정보 관련 문의글 타입입니다.
- **etc**: 기타 문의글 타입입니다.

VendorQnaAuthorType

공급사 1:1 문의 글의 작성자 타입입니다.

- **admin**: 관리자.
- **seller**: 판매사.
- **vendor**: 공급사.

SellerQnaDirectAnswerType

상품출고/재고문의 답변 타입입니다.

- **deliveryCompleted**: 출고 완료 답변입니다.
- **deliveryEstimated**: 출고 예정 답변입니다.
- **deliveryUnavailable**: 출고 불가 답변입니다.
- **etc**: 기타 답변입니다.

EstimatedShippingDayType

출고 예정일 타입입니다.

- **today**: 당일 출고 예정입니다.
- **tomorrow**: 익일 출고 예정입니다.

DeliveryUnavailableReasonType

출고 불가 사유 타입입니다.

- **outOfStock**: 재고가 없는 상태입니다.
- **soldout**: 품절인 상태입니다.

오픈마켓 참고 정보(카테고리)

Item 타입의 정보(**item** 쿼리의 반환 데이터, **allItems** 쿼리의 반환 데이터)에서 **Item.metadata**에서는 옥션, 지마켓, 인터파크, 11번가, 스마트스토어(구 스토어팜), 쿠팡, 플레이오토, 이셀러스에 대해 해당 상품의 오픈마켓 또는 외부 솔루션 등록 참고 정보(카테고리 코드)를 제공합니다. 하지만 해당 카테고리 정보는 오픈마켓, 외부 솔루션 측의 정책에 의해 언제든지 변경될 수 있으며, 오너클랜 측에서 해당 변경 사항을 인지한 후 대다수의 상품에 반영이 되기까지 시간이 걸릴 수 있습니다. 따라서 해당 정보는 참고용으로만 사용하시고 정확한 정보가 아닐 수 있음을 염두에 두어야 합니다.

지원하는 오픈마켓 별 필드 정보는 다음과 같습니다.

- 옥션: **Item.metadata.auctionCategoryCode** - String / **Item.metadata.auctionCategoryFullName** - String
- 지마켓: **Item.metadata.gmarketCategoryCode** - String / **Item.metadata.gmarketCategoryFullName** - String
- 인터파크: **Item.metadata.interparkCategoryCode** - String / **Item.metadata.interparkCategoryFullName** - String
- 11번가: **Item.metadata.st11CategoryCode** - String / **Item.metadata.11stCategoryFullName** - String
- 스마트스토어(구 스토어팜): **Item.metadata.smartstoreCategoryCode** - String / **Item.metadata.smartstoreCategoryFullName** - String
- 쿠팡: **Item.metadata.coupangCategoryCode** - String / **Item.metadata.coupangCategoryFullName** - String
- 플레이오토: **Item.metadata.playautoCategoryCode** - String / **Item.metadata.playautoCategoryFullName** - String

- 이셀러스: `Item.metadata.esellersCategoryCode` - String / `Item.metadata.esellersCategoryFullName` - String

해당 마켓에 대한 카테고리 데이터가 없는 경우 `null`을 반환합니다.

마켓 카테고리 매칭 정보

오너클랜에서는 기본적으로 카테고리-카테고리 매칭을 통해 외부 마켓과의 카테고리 매칭 정보를 제공하고 있습니다. 다만 상품을 등록하거나 수정할 때 상품 단위로 상품에 대한 마켓 카테고리 매칭 정보를 직접 지정할 수 있습니다. 현재 쿠팡, 옥션, 지마켓, 인터파크, 11번가의 5개 마켓에 대해 카테고리 매칭 정보를 직접 지정할 수 있습니다. 지정은 `createItem`, `updateItem` 쿼리의 `input`에서 `metadata.externalCategories` 필드를 통해 할 수 있으며, 다음과 같이 사용하실 수 있습니다.

(사용할 수 있는 예제는 아닙니다.)

```
{
  "input": {
    "metadata": {
      "externalCategories": {
        "coupang": "80601",
        "gmarket": "300024380"
      }
    }
  }
}
```

`externalCategories` object 내부에 각 마켓에 대한 카테고리 매칭 정보를 명시할 때는 다음과 같은 규칙을 지켜주셔야 합니다.

- 쿠팡: 필드명 `coupang`, 타입 `string`
- 옥션: 필드명 `auktion`, 타입 `string`
- 지마켓: 필드명 `gmarket`, 타입 `string`
- 인터파크: 필드명 `interpark`, 타입 `string`
- 11번가: 필드명 `st11`, 타입 `string`

상품 정보를 수정할 때 마켓 카테고리 매칭 정보를 명시하는 경우 다음과 같은 방식으로 진행됩니다.

- 기존에 지정하지 않았던 마켓의 카테고리 매칭 정보가 입력된 경우 해당 내용을 추가합니다.
- 기존에 지정했던 마켓의 카테고리 매칭 정보가 재입력된 경우 재입력된 내용으로 변경합니다.
- 기존에 지정했던 마켓의 카테고리 매칭 정보가 입력되지 않은 경우 해당 마켓의 카테고리 매칭 정보를 삭제합니다.

상품을 조회할 때는 직접 매칭 정보를 명시한 것인지 아닌지의 여부를 다음 필드들에서 확인할 수 있습니다.

- `Item.metadata.coupangCategoryIsItemMatching`: 쿠팡 카테고리 매칭 정보가 직접 입력된 정보인지의 여부.
- `Item.metadata.auktionCategoryIsItemMatching`: 옥션 카테고리 매칭 정보가 직접 입력된 정보인지의 여부.

- `Item.metadata.gmarketCategoryIsItemMatching`: 지마켓 카테고리 매칭 정보가 직접 입력된 정보인지의 여부.
- `Item.metadata.st11CategoryIsItemMatching`: 11번가 카테고리 매칭 정보가 직접 입력된 정보인지의 여부.
- `Item.metadata.interparkCategoryIsItemMatching`: 인터파크 카테고리 매칭 정보가 직접 입력된 정보인지의 여부.

예제는 아래 링크에서 확인할 수 있습니다.

- 예시 코드:
 - [example1](#) - 상품 등록시 마켓 카테고리 매칭 정보 입력하기
 - [example2](#) - 상품 수정시 마켓 카테고리 매칭 정보 입력하기

인증정보(certificateInformation)

HACCP(해썹), 자율안전인증, 특허인증 등 상품에 해당하는 인증 정보들입니다. 입력과 관련된 내용은 다음과 같습니다.

- 입력할 인증 타입(`certificateType` 필드)을 지정한 후 해당 필드에 필수로 입력해야 하는 값을 지정.
- 인증 타입이 11, 14, 18, 81, 82, 84인 경우 필수 필드가 모두 입력되면 통과.
- 인증 타입이 9, 10, 12, 13, 16, 17인 경우 모두 `certificateCode` 필드를 입력받고, 해당 필드의 값을 통해 실제로 등록된 인증정보와 일치하는지 검증. 검증에 통과하면 해당 검증 정보를 기반으로 데이터 저장.
- 인증 타입 중 1, 2, 3, 4, 5, 6, 7, 8, 15, 83은 더이상 등록하거나 수정할 때 사용할 수 없으며 기존 데이터를 읽기만 가능. 만약 등록하거나 수정할 때 사용하면 deprecation error 반환.
- 인증 정보를 직접 `certificateInformation` 필드에 입력하는 경우 `certificateInformation.inContent` 필드의 값을 `false`로, 상품 상세정보에 있어 상세정보 별도표기로 진행해야 하는 경우 인증 타입만 입력하고 `certificateInformation.inContent` 필드의 값을 `true`로 설정.

인증정보는 현재 12종의 인증정보를 지원합니다. 각 인증 정보는 인증 정보 타입을 나타내는 `certificateType`와 각 인증 정보에 필요한 필드들로 구성됩니다. 자세한 내용은 다음과 같습니다.

- HACCP(해썹) 지정업소
 - **DEPRECATED (신규 등록 / 수정시 사용 불가)**
 - `certificateType: 1`
 - `companyName: String` - 지정업소의 이름입니다.
 - `certifiedDate: String` - 인증 일자입니다. 포맷: YYYY-MM-DD
- 친환경 농축산물
 - **DEPRECATED (신규 등록 / 수정시 사용 불가)**
 - `certificateType: 2`
 - `certificateName: String` - 인증명입니다.
 - `certificateCode: String` - 인증 번호입니다.
 - `expiryDate: String` - 인증 만료일자입니다. 포맷: YYYY-MM-DD
- 유기가공식품
 - **DEPRECATED (신규 등록 / 수정시 사용 불가)**
 - `certificateType: 3`
 - `certificateName: String` - 인증명입니다.
 - `certificateCode: String` - 인증번호입니다.

- `expiryDate: String` - 인증 만료일자입니다. 포맷: `YYYY-MM-DD`
- 기타
 - **DEPRECATED (신규 등록 / 수정시 사용 불가)**
 - `certificateType: 4`
 - `certificateName: String` - 인증명입니다.
 - `certificateCode: String` - 인증번호입니다.
 - `certifiedDate: String` - 인증 일자입니다. 포맷: `YYYY-MM-DD`
 - `expiryDate: String` - 인증 만료일자입니다. 포맷: `YYYY-MM-DD`
- 자율안전 / 안전인증
 - **DEPRECATED (신규 등록 / 수정시 사용 불가)**
 - `certificateType: 5`
 - `certificateName: String` - 인증명입니다.
 - `certificateCode: String` - 인증번호입니다.
 - `certifiedDate: String` - 인증 일자입니다. 포맷: `YYYY-MM-DD`
- 적합성평가
 - **DEPRECATED (신규 등록 / 수정시 사용 불가)**
 - `certificateType: 6`
 - `certificateName: String` - 인증명입니다.
 - `certificateCode: String` - 인증번호입니다.
 - `certifiedDate: String` - 인증 일자입니다. 포맷: `YYYY-MM-DD`
- 특허
 - **DEPRECATED (신규 등록 / 수정시 사용 불가)**
 - `certificateType: 7`
 - `certificateCode: String` - 인증번호입니다.
- HACCP(해썹) 축산물
 - **DEPRECATED (신규 등록 / 수정시 사용 불가)**
 - `certificateType: 8`
 - `certificateName: String` - 인증명입니다.
 - `certifiedDate: String` - 인증 일자입니다. 포맷: `YYYY-MM-DD`
- [어린이제품] 안전인증
 - `certificateType: 9`
 - `certificateIssuer: String` - 인증기관명입니다.
 - `certificateCode: String` - 인증번호입니다.
 - `companyName: String` - 인증받은 상호명입니다.
 - `certifiedDate: String` - 인증 일자입니다. 포맷: `YYYY-MM-DD`
- [어린이제품] 안전확인
 - `certificateType: 10`
 - `certificateIssuer: String` - 인증기관명입니다.
 - `certificateCode: String` - 인증번호입니다.
 - `companyName: String` - 인증받은 상호명입니다.
 - `certifiedDate: String` - 인증 일자입니다. 포맷: `YYYY-MM-DD`
- [어린이제품] 공급자적합성확인
 - `certificateType: 11`
 - `certificateIssuer: String` - 인증기관명입니다.
 - `companyName: String` - 인증받은 상호명입니다.
 - `model: String` - 모델명입니다.

- [생활용품] 안전인증
 - `certificateType`: 12
 - `certificateIssuer`: `String` - 인증기관명입니다.
 - `certificateCode`: `String` - 인증번호입니다.
 - `companyName`: `String` - 인증받은 상호명입니다.
 - `certifiedDate`: `String` - 인증 일자입니다. 포맷: `YYYY-MM-DD`
- [생활용품] 안전확인
 - `certificateType`: 13
 - `certificateIssuer`: `String` - 인증기관명입니다.
 - `certificateCode`: `String` - 인증번호입니다.
 - `companyName`: `String` - 인증받은 상호명입니다.
 - `certifiedDate`: `String` - 인증 일자입니다. 포맷: `YYYY-MM-DD`
- [생활용품] 공급자적합성확인
 - `certificateType`: 14
 - `certificateIssuer`: `String` - 인증기관명입니다.
 - `companyName`: `String` - 인증받은 상호명입니다.
 - `model`: `String` - 모델명입니다.
- [생활용품] 어린이보호포장
 - **DEPRECATED (신규 등록 / 수정시 사용 불가)**
 - `certificateType`: 15
 - `certificateIssuer`: `String` - 인증기관명입니다.
 - `certificateCode`: `String` - 인증번호입니다.
- [전기용품] 안전인증
 - `certificateType`: 16
 - `certificateIssuer`: `String` - 인증기관명입니다.
 - `certificateCode`: `String` - 인증번호입니다.
 - `companyName`: `String` - 인증받은 상호명입니다.
 - `certifiedDate`: `String` - 인증 일자입니다. 포맷: `YYYY-MM-DD`
- [전기용품] 자율안전확인
 - `certificateType`: 17
 - `certificateIssuer`: `String` - 인증기관명입니다.
 - `certificateCode`: `String` - 인증번호입니다.
 - `companyName`: `String` - 인증받은 상호명입니다.
 - `certifiedDate`: `String` - 인증 일자입니다. 포맷: `YYYY-MM-DD`
- [전기용품] 공급자적합성확인
 - `certificateType`: 18
 - `certificateIssuer`: `String` - 인증기관명입니다.
 - `companyName`: `String` - 인증받은 상호명입니다.
 - `model`: `String` - 모델명입니다.
- [방송통신기자재] 적합인증
 - `certificateType`: 19,
 - `certificateCode`: `String` - 인증번호입니다.
 - `companyName`: `String` - 인증받은 상호명입니다.
 - `modelName`: `String` - 인증받은 모델명입니다.
 - `certifiedDate`: `String` - 인증 일자입니다. 포맷: `YYYY-MM-DD`
- [방송통신기자재] 적합등록

- `certificateType: 20`,
- `certificateCode: String` - 인증번호입니다.
- `companyName: String` - 인증받은 상호명입니다.
- `modelName: String` - 인증받은 모델명입니다.
- `certifiedDate: String` - 인증 일자입니다. 포맷: YYYY-MM-DD
- [방송통신기자재] 잠정인증
 - `certificateType: 21`,
 - `certificateCode: String` - 인증번호입니다.
 - `companyName: String` - 인증받은 상호명입니다.
 - `modelName: String` - 인증받은 모델명입니다.
 - `certifiedDate: String` - 인증 일자입니다. 포맷: YYYY-MM-DD
- 의료기기 판매업 신고
 - `certificateType: 81`
 - `certificateIssuer: String` - 인증기관명입니다.
 - `certificateCode: String` - 인증번호입니다.
- 의료기기 품목 허가
 - `certificateType: 82`
 - `certificateCode: String` - 인증번호입니다.
- 사전광고 심의
 - **DEPRECATED (신규 등록 / 수정시 사용 불가)**
 - `certificateType: 83`
 - `certificateCode: String` - 인증번호입니다.
- 건강기능식품 판매업신고
 - `certificateType: 84`
 - `certificateIssuer: String` - 인증기관명입니다.
 - `certificateCode: String` - 인증번호입니다.

상품속성

상품에 대한 속성값들입니다.

- `"ATTR_H_FOOD"` - 건강식품 여부입니다.
- `"ATTR_MEDICAL"` - 의료용품 여부입니다.
- `"ATTR_OVERSEA"` - 해외직배송 상품 여부입니다. 설정 가능한 필드가 아닙니다.
- `"ATTR_OVERSEA_DELI"` - 해외직구 상품 여부입니다. 설정 가능한 필드가 아닙니다.
- `"PROHIBIT"` - 유통금지 상품 여부입니다. 이 값이 있는 경우 유통금지상품이므로 상품 수정이 불가능합니다. 수정해야 하는 경우, 오너클랜 모니터링팀으로 문의해주시기 바랍니다.

카테고리 속성

카테고리에 대한 속성값들입니다.

- `"HFOOD"` - 건강기능식품 여부입니다.
- `"MEDI"` - 의료용품 여부입니다.

상품 등록 및 운영시 참조사항

택배사 코드 정보

택배 운송장 정보를 등록할 때 사용할 수 있는 택배사 코드 정보는 다음과 같습니다.

- "1": 롯데택배
- "2": 한진택배
- "3": CJ대한통운
- "4": 우체국택배
- "8": KGB택배
- "9": 로젠택배
- "14": 건영택배
- "19": 일양택배
- "22": 호남택배
- "24": 경동택배
- "25": 대신택배
- "26": 천일택배
- "27": CVSnet(GS25편의점)
- "28": 합동택배
- "30": CUpost(CU편의점)
- "31": HI택배
- "32": FLF퍼레버택배

특수문자 제한 정책

오너클랜 API의 경우 상품을 등록할 때, **상품명 (Item.productname)**, **택배 송장명 (Item.metadata.shippingName)**, **키워드(Item.searchKeywords)** **옵션명 (Item.options.optionAttributes.name)**, ****옵션값(Item.options.optionAttributes.value)****에 대하여 정해진 특수문자만을 허용하고 있습니다.

각 필드별로 허용되는 문자 집합은 다음과 같습니다.

- 상품명(Item.name), 택배 송장명(Item.metadata.shippingName), 키워드(Item.searchKeywords):
 - 한글
 - 영문
 - 숫자
 - 한자: 유니코드 범위 \u4E00-\u9FFF에 해당하는 한자만 가능.
 - 공백(): 허용하되 2개 이상의 공백이 연달아 나타나는 경우 1개로 치환.
 - 탭(\t): 공백으로 치환.
 - +, ~, _ /, ., %, -
 - (,): 허용하되 2개 이상의 동일 괄호가 연달아 나타나는 경우 해당 괄호 1개로 치환.
 - {, [: (으로 치환.
 - },]:)으로 치환.
- 옵션명(Item.options.optionAttributes.name), 옵션값(Item.options.optionAttributes.value)
 - 한글
 - 영문
 - 숫자
 - 한자: 유니코드 범위 \u4E00-\u9FFF에 해당하는 한자만 가능.
 - 공백(): 허용하되 2개 이상의 공백이 연달아 나타나는 경우 1개로 치환.
 - 탭(\t): 공백으로 치환.
 - ', !, ", #, \$, &, *, :, ;, ^, \, @, ?, +, ~, _ /, ., <, >, %, -

- (,): 허용하되 2개 이상의 동일 괄호가 연달아 나타나는 경우 해당 괄호 1개로 치환.
- {, [: (으로 치환.
- },],)으로 치환.
- 다음 특수문자 집합에 속하는 문자들은 입력이 가능하나 마켓에는 정상적으로 노출되지 않을 수 있으니 참고하시기 바랍니다.
 - $\pm, \neq, \leq, \geq, \infty, \approx, \int, \phi, \Sigma, \text{£}, \text{¥}, \text{¤}, ^\circ\text{F}, \text{‰}, \text{€}, \mu\text{ℓ}, \text{dℓ}, \text{kℓ}, \text{mm}^3, \text{cm}^3, \text{km}^3, \text{fm}, \text{nm}, \mu\text{m}, \text{km}, \text{km}^2, \text{ha}, \mu\text{g}, \text{mg}, \text{kg}, \text{kt}, \text{cal}, \text{kcal}, \text{dB}, \text{mJ/s}, \text{mJ/s}^2, \text{ps}, \text{ns}, \mu\text{s}, \text{ms}, \text{pV}, \text{nV}, \mu\text{V}, \text{mV}, \text{kV}, \text{MV}, \text{pA}, \text{nA}, \mu\text{A}, \text{mA}, \text{kA}, \text{pW}, \text{nW}, \mu\text{W}, \text{mW}, \text{kW}, \text{MW}, \text{Hz}, \text{kHz}, \text{MHz}, \text{GHz}, \text{THz}, \Omega, \text{k}\Omega, \text{M}\Omega, \text{pF}, \text{nF}, \mu\text{F}, \text{mol}, \text{rad}, \text{rad/s}, \text{rad/s}^2, \text{kPa}, \text{MPa}, \text{GPa}, \text{Wb}, \text{lm}, \text{lx}, \text{C/kg}, \text{S}, \text{Co}, \text{a.m.}, \text{p.m.}, ^\circ, \Theta, \Lambda, \Pi, \Sigma, \Phi, \Omega, \alpha, \beta, \delta, \eta, \theta, \lambda, \mu, \nu, \pi, \rho, \sigma, \tau, \upsilon, \phi, \emptyset, \emptyset, \beta, \Phi, \exists, \forall, \text{Я}, \text{б}, \text{д}, \phi$

특수문자 치환 정책

`createItem`, `updateItem` 쿼리에서 `autoReplace` 옵션을 `true`로 지정한 경우, 다음 규칙에 의해 치환된 결과를 기준으로 입력값 검사 및 등록, 수정이 진행됩니다.

- 공백(): 2개 이상의 공백이 연달아 나타나는 경우 1개로 치환하며 문자열 맨 앞과 맨 뒤의 공백들은 trim.
- 탭(\t): 공백으로 치환한 뒤 공백 치환 규칙을 따름.
- {, }: 각각 (,)으로 치환.
- [,]: 각각 (,)으로 치환.

반품 배송비

오너클랜 API에서 사용하는 반품배송비는 모두 **편도** 기준입니다. 즉, 예를 들어 반품 및 교환에 배송비가 왕복 기준 10000원이 소요될 것으로 예상되면 5000원으로 등록해주시면 됩니다.

배송비 타입에 따른 반품 배송비 정책은 다음과 같습니다.

- 선불 전용(`Item.shippingType = inAdvance`): 반품 배송비가 입력되면 해당 값을, 입력되지 않으면 오너클랜에 노출되는 배송비와 동일하게 책정됩니다.
- 착불 전용(`Item.shippingType = uponArrival`): 0원으로 설정됩니다.
- 무료 배송 상품(`Item.shippingType = free`): 반품 배송비는 필수로 입력해야 하며, 입력된 반품 배송비로 설정됩니다.