플레이오토 주문 조회 API 활용 가이드

분류	웹 서비스

인증 정보

항목	값
API Key	GqOHPcT90B3RmzPtF6xWQ5kDhaYcpImx8UcDtj3t
Email	jinman@biocom.kr
Password	bico1016!
Base URL	https://openapi.playauto.io/api

* AI혁신팀 계정으로도 조회 가능
* API 키는 회사별 발급으로 보임(공정관리팀과 AI혁신팀의 API키가 동일)

---

1. 인증 (토큰 발급)

먼저 토큰을 발급받아야 합니다. 토큰은 24시간 유효합니다.

```
const response = await fetch('https://openapi.playauto.io/api/auth', {
  method: 'POST',
  headers: {
    'x-api-key': 'GqOHPcT90B3RmzPtF6xWQ5kDhaYcpImx8UcDtj3t',
    'Content-Type': 'application/json'
  },
  body: JSON.stringify({
    email: 'jinman@biocom.kr',
    password: 'bico1016!'
  })
});

const data = await response.json();
```

```
const token = data[0].token;  // 이 토큰을 저장해서 사용
console.log('토큰:', token);
```

응답:

```
[
  {
    "token": "eyJhbGciOiJIUzUxMiIsInR5cCI6IkpXVCJ9...(생략)...",
    "sol_no": 113565
  }
]
```

---

2. 쇼핑몰 코드 조회 (먼저 실행 권장)

주문 조회 전에 어떤 쇼핑몰이 연동되어 있는지 확인하세요.

```
const response = await fetch('https://openapi.playauto.io/api/shops?used=true', {
  method: 'GET',
  headers: {
    'x-api-key': 'GqOHPcT90B3RmzPtF6xWQ5kDhaYcpImx8UcDtj3t',
    'Authorization': `Token ${token}`
  }
});

const shops = await response.json();
console.log('연동된 쇼핑몰 목록:', shops);
```

응답 예시:

```
[
  {
    "shop_name": "스마트스토어-B",
    "shop_cd": "1077",
    "shop_id": "biocom015@gmail.com",
    "seller_nick": "미리바이오 스마트스토어"
  },
  {
```

```
    "shop_name": "아임웹-C",
    "shop_cd": "2005",
    "shop_id": "acc@biocom.kr",
    "seller_nick": "더클린커피 자사몰"
  }
]
```

현재 바이오컴 연동 쇼핑몰 (2025-01 기준)

shop_cd	shop_name	seller_nick
1005	아임웹-B	바이오컴펫 자사몰
1077	스마트스토어-B	미리바이오 스마트스토어
2005	아임웹-C	더클린커피 자사몰
3005	아임웹-D	에이아이바이오 자사몰
4005	아임웹-E	미리바이오 자사몰
A077	스마트스토어	더클린스토어 스마트스토어
A100	롯데아이몰	-
B005	아임웹	바이오컴 자사몰
B378	쿠팡	쿠팡 (2개 계정: accbiocom, biocomkr)
ULPL	바이오컴-앱(수동)	바이오컴 앱 쇼핑몰 운영환경
UVP2	바이오컴-앱-테스트(수동)	바이오컴 앱 쇼핑몰 개발환경

3. 주문 조회 API

URL: `POST https://openapi.playauto.io/api/orders`

4. 요청 파라미터 (필수/선택 정리)

필수 파라미터 (3개)

파라미터	타입	설명	예시
date_type	String	날짜 검색 기준	"ord_time" (주문일 기준)
sdate	String	검색 시작일	"2024-01-01"
edate	String	검색 종료일	"2024-12-31"

자주 쓰는 선택 파라미터

파라미터	타입	기본값	설명
`start`	Number	0	검색 시작 위치 (페이징용)
`length`	Number	500	조회 건수 (최대 3000)
`status`	String[]	전체	주문 상태 필터
`shop_cd`	String	전체	특정 쇼핑몰만 조회
`masking_yn`	Boolean	true	개인정보 마스킹 여부

---

5. 실제 사용 예시

예시 1: 1년치 전체 주문 조회

```
const response = await fetch('https://openapi.playauto.io/api/orders', {
  method: 'POST',
  headers: {
    'x-api-key': 'GqOHPcT90B3RmzPtF6xWQ5kDhaYcpImx8UcDtj3t',
    'Authorization': `Token ${token}`,
    'Content-Type': 'application/json'
  },
  body: JSON.stringify({
    date_type: 'ord_time',
    sdate: '2024-01-01',
    edate: '2024-12-31',
    start: 0,
    length: 3000,
    status: ['ALL']
  })
});

const data = await response.json();
console.log('주문 목록:', data.results);
console.log('전체 건수:', data.recordsTotal);
```

예시 2: 스마트스토어 주문만 조회

```
const response = await fetch('https://openapi.playauto.io/api/orders', {
  method: 'POST',
  headers: {
    'x-api-key': 'GqOHPcT90B3RmzPtF6xWQ5kDhaYcpImx8UcDtj3t',
    'Authorization': `Token ${token}`,
    'Content-Type': 'application/json'
  },
  body: JSON.stringify({
    date_type: 'ord_time',
    sdate: '2024-01-01',
    edate: '2024-12-31',
    shop_cd: '1077',  // 스마트스토어-B
    start: 0,
    length: 3000
  })
});

const data = await response.json();
console.log('스마트스토어 주문:', data.results);
```

예시 3: 배송완료된 주문만 조회

```
const response = await fetch('https://openapi.playauto.io/api/orders', {
  method: 'POST',
  headers: {
    'x-api-key': 'GqOHPcT90B3RmzPtF6xWQ5kDhaYcpImx8UcDtj3t',
    'Authorization': `Token ${token}`,
    'Content-Type': 'application/json'
  },
  body: JSON.stringify({
    date_type: 'ord_time',
    sdate: '2024-01-01',
    edate: '2024-12-31',
    status: ['배송완료', '구매결정'],
    start: 0,
    length: 3000
  })
```

```
});

const data = await response.json();
console.log('배송완료 주문:', data.results);
```

6. 전체 데이터 가져오기 (3000건 초과 시)

한 번에 최대 3000건까지만 조회 가능합니다. 전체 데이터를 가져오려면 반복 호출하세요.

```
async function getAllOrders(token, sdate, edate) {
  let allOrders = [];
  let start = 0;
  const length = 3000;

  while (true) {
    const response = await fetch('https://openapi.playauto.io/api/orders', {
      method: 'POST',
      headers: {
        'x-api-key': 'GqOHPcT90B3RmzPtF6xWQ5kDhaYcpImx8UcDtj3t',
        'Authorization': `Token ${token}`,
        'Content-Type': 'application/json'
      },
      body: JSON.stringify({
        date_type: 'ord_time',
        sdate: sdate,
        edate: edate,
        start: start,
        length: length
      })
    });

    const data = await response.json();
    allOrders = allOrders.concat(data.results);

    console.log(`${start}~${start + data.results.length}건 조회 완료`);

    // 더 이상 데이터가 없으면 종료
    if (data.results.length < length) break;
```

```
    start += length;
  }

  console.log(`총 ${allOrders.length}건 조회 완료`);
  return allOrders;
}

// 사용 예시
const orders = await getAllOrders(token, '2024-01-01', '2024-12-31');
```

7. 채널별 매출 집계 예시

```
// 주문 데이터를 채널별로 집계
function aggregateByChannel(orders) {
  const summary = {};

  orders.forEach(order => {
    const channel = order.shop_name;

    if (!summary[channel]) {
      summary[channel] = {
        shop_cd: order.shop_cd,
        shop_name: order.shop_name,
        seller_nick: order.seller_nick,
        orderCount: 0,
        totalSales: 0,
        totalPayAmt: 0
      };
    }

    summary[channel].orderCount += 1;
    summary[channel].totalSales += order.sales || 0;
    summary[channel].totalPayAmt += order.pay_amt || 0;
  });

  return Object.values(summary);
```

```
}

// 사용 예시
const orders = await getAllOrders(token, '2024-01-01', '2024-12-31');
const channelSummary = aggregateByChannel(orders);
console.table(channelSummary);
```

8. 복사해서 바로 쓰는 전체 코드

```
// ===== 플레이오토 API 호출 전체 코드 =====

const API_KEY = 'GqOHPcT90B3RmzPtF6xWQ5kDhaYcpImx8UcDtj3t';
const BASE_URL = 'https://openapi.playauto.io/api';

// 1. 토큰 발급
async function getToken() {
  const response = await fetch(`${BASE_URL}/auth`, {
    method: 'POST',
    headers: {
      'x-api-key': API_KEY,
      'Content-Type': 'application/json'
    },
    body: JSON.stringify({
      email: 'jinman@biocom.kr',
      password: 'bico1016!'
    })
  });

  const data = await response.json();
  return data[0].token;
}

// 2. 연동된 쇼핑몰 목록 조회
async function getShops(token) {
  const response = await fetch(`${BASE_URL}/shops?used=true`, {
    method: 'GET',
    headers: {
```

```
      'x-api-key': API_KEY,
      'Authorization': `Token ${token}`
    }
  });

  return await response.json();
}

// 3. 주문 조회 (페이징 자동 처리)
async function getOrders(token, sdate, edate, options = {}) {
  let allOrders = [];
  let start = 0;
  const length = 3000;

  while (true) {
    const response = await fetch(`${BASE_URL}/orders`, {
      method: 'POST',
      headers: {
        'x-api-key': API_KEY,
        'Authorization': `Token ${token}`,
        'Content-Type': 'application/json'
      },
      body: JSON.stringify({
        date_type: options.date_type || 'ord_time',
        sdate: sdate,
        edate: edate,
        start: start,
        length: length,
        status: options.status || ['ALL'],
        shop_cd: options.shop_cd,
        masking_yn: options.masking_yn ?? true
      })
    });

    const data = await response.json();
    allOrders = allOrders.concat(data.results);

    if (data.results.length < length) break;
```

```
    start += length;
  }

  return allOrders;
}

// 4. 채널별 집계
function aggregateByChannel(orders) {
  const summary = {};

  orders.forEach(order => {
    const key = order.shop_cd;

    if (!summary[key]) {
      summary[key] = {
        shop_cd: order.shop_cd,
        shop_name: order.shop_name,
        seller_nick: order.seller_nick,
        orderCount: 0,
        totalSales: 0,
        totalPayAmt: 0
      };
    }

    summary[key].orderCount += 1;
    summary[key].totalSales += order.sales || 0;
    summary[key].totalPayAmt += order.pay_amt || 0;
  });

  return Object.values(summary);
}

// ===== 실행 =====
async function main() {
  // 토큰 발급
  const token = await getToken();
  console.log('토큰 발급 완료');
```

```
  // 쇼핑몰 목록 확인
  const shops = await getShops(token);
  console.log('연동된 쇼핑몰:', shops);

  // 1년치 주문 조회
  const orders = await getOrders(token, '2024-01-01', '2024-12-31');
  console.log(`총 ${orders.length}건 조회`);

  // 채널별 집계
  const summary = aggregateByChannel(orders);
  console.table(summary);
}

main();
```

9. date_type 옵션 설명

값	의미	언제 사용?
ord_time	주문일	고객이 주문한 날짜 기준 (가장 많이 사용)
pay_time	결제완료일	실제 결제가 완료된 날짜 기준
wdate	수집일	플레이오토에 수집된 날짜 기준
mdate	변경일	주문 정보가 수정된 날짜 기준
out_time	출고완료일	출고 처리된 날짜 기준

10. 주문 상태 (status)

상태	의미
결제완료	결제 완료, 아직 처리 전
신규주문	신규 접수된 주문
출고대기	출고 대기 중
배송중	배송 진행 중
배송완료	배송 완료
구매결정	구매 확정 (정산 가능)
취소요청	취소 요청됨

상태	의미
취소완료	취소 처리 완료
반품요청	반품 요청됨
반품완료	반품 처리 완료

11. 응답에서 중요한 필드

채널 분석용

```
{
  shop_cd: "1077",          // 채널 코드
  shop_name: "스마트스토어-B", // 채널명
  seller_nick: "미리바이오"   // 스토어명
}
```

매출 분석용

```
{
  pay_amt: 115000,    // 실결제금액 (고객이 낸 돈)
  sales: 322000,      // 판매금액 (할인 전 원가)
  sale_cnt: 1,        // 주문 수량
  discount_amt: 207000 // 총 할인금액
}
```

시간 분석용

```
{
  ord_time: "2024-12-31 20:50:51",  // 주문일시
  pay_time: "2024-12-31 20:50:55",  // 결제일시
  ship_com_time: "2025-01-03 14:11:22" // 배송완료일시
}
```

12. 응답값 전체 필드 (주석 포함)

```
{
  results: [
    {
      // ===== 플레이오토 시스템 정보 =====
      sol_no: 113565,                  // 솔루션번호 (플레이오토 계정 고유번호)
      uniq: "3961692873863462201",       // 주문 고유번호 (플레이오토 내부 ID)
      ori_uniq: null,                  // 원본 주문 고유번호 (사본주문인 경우)
      pa_bundle_no: null,              // 상위 묶음번호
      bundle_no: "3961765173941969",     // 주문 묶음번호 (합포장 그룹)
      ori_bundle_no: null,             // 변경 전 묶음번호
      multi_bundle_yn: 0,              // 묶음주문 여부 (0: 단일, 1: 묶음)

      // ===== 주문 상태 =====
      ord_status: "구매결정",            // 주문상태 (결제완료/신규주문/출고대기/배송
중/배송완료/구매결정/취소요청 등)

      // ===== 택배사 정보 =====
      carr_no: 5,                      // 택배사 코드 (5: 한진택배, 7: 로젠택배 등)
      carr_ser_no: null,               // 택배사 서비스 코드
      carr_name: "한진택배",             // 택배사 이름
      carr_ser_name: null,             // 택배사 서비스 이름
      invoice_no: "534163466911",        // 운송장 번호 ⭐
      ship_plan_date: "2025-01-06",      // 배송예정일

      // ===== 판매 채널 정보 ⭐ (채널별 분석에 핵심) =====
      shop_cd: "1077",                 // 쇼핑몰 코드 ⭐ (1077: 스마트스토어, 2005:
아임웹 등)
      pa_shop_cd: "A077",              // 상위 쇼핑몰 코드
      shop_name: "스마트스토어-B",         // 쇼핑몰 이름 ⭐
      logo_img: "logo-smartstore",       // 쇼핑몰 로고
      detail_url: "http://smartstore.naver.com/",  // 쇼핑몰 URL
      shop_id: "biocom015@gmail.com",    // 쇼핑몰 계정 (판매자 ID)
      seller_nick: "미리바이오 스마트스토어", // 판매자 별칭 ⭐

      // ===== 주문 정보 =====
      shop_ord_no: "2024123199044921 2024123132110711",  // 쇼핑몰 주문번
호
      shop_ord_no_real: "2024123199044921",            // 원본 주문번호 (가공
```

```
전)
    shop_sale_no: "9250736478",                    // 쇼핑몰 판매번호 (상품번호)
    shop_sale_name: "바이오컴 뉴로마스터 은행잎추출물...",  // 상품명 ⭐
    shop_sku_cd: null,              // 쇼핑몰 SKU 코드
    shop_opt_name: "구성: 3+1 세트(4개월분/ 개당 29250원)",  // 옵션명 ⭐
    shop_add_opt_name: "",          // 추가 구매 옵션명
    ship_method: "무료배송",          // 배송방법 (무료배송/선결제/착불 등)

    // ===== 일시 정보 =====
    out_time: "2025-01-02 11:05:58",        // 출고일
    invoice_time: "2025-01-02 11:04:41",    // 운송장 등록일
    invoice_print_time: "2025-01-02 11:04:41", // 운송장 출력일
    wdate: "2025-01-01 00:00:37",           // 주문 수집일 (플레이오토에 수집된
시간)
    ord_time: "2024-12-31 20:50:51",        // 주문일 ⭐ (고객이 주문한 시간)
    pay_time: "2024-12-31 20:50:55",        // 결제완료일 ⭐
    mdate: "2025-04-13 01:23:14",           // 최종 수정일
    ord_confirm_time: null,                 // 주문확인일
    ord_status_mdate: "2025-01-12 00:00:36", // 상태 변경일
    claim_com_time: null,                   // 클레임 완료일
    claim_time: null,                       // 클레임 상태 변경일
    ship_hope_time: null,                   // 배송 희망일
    ship_com_time: "2025-01-03 14:11:22",    // 배송 완료일
    api_read_time: "",                      // API 호출로 조회한 시간
    api_read_status: "NONE",                // API 호출 여부 (NONE/CALLED/EDIT
ED)

    // ===== 금액 정보 ⭐ (매출 분석에 핵심) =====
    pay_amt: 115000,                // 실결제금액 ⭐ (고객이 실제 결제한 금액)
    discount_amt: 207000,           // 총 할인금액
    shop_discount: 0,               // 쇼핑몰 부담 할인
    seller_discount: 2000,          // 판매자 부담 할인
    coupon_discount: 2000,          // 쿠폰 할인
    point_discount: 0,              // 포인트 할인
    sales: 322000,                  // 판매금액 ⭐ (할인 전 원가)
    sale_cnt: 1,                    // 주문 수량 ⭐
    sales_tax: 0,                   // 부가세
    sales_unit: null,               // 판매 단위
```

```
shop_cost_price: 0,              // 쇼핑몰 원가
shop_supply_price: 109222,       // 쇼핑몰 공급가 (수수료 제외 정산액)
ship_delay_yn: 0,                // 배송 지연 여부

// ===== 주문자 정보 =====
order_name: "이응철",             // 주문자 명 ⭐
order_id: "dmdc****",            // 주문자 ID (마스킹됨)
order_tel: "010-4***-2958",      // 주문자 전화번호 (마스킹됨)
order_htel: "010-4***-2958",     // 주문자 휴대폰 (마스킹됨)
order_email: "",                 // 주문자 이메일
ship_msg: "",                    // 배송 메시지 ⭐
ship_delay_msg: "",              // 배송 지연 메시지
out_order_time: "2025-01-02 11:03:15",  // 출고 지시일

// ===== 수령자 정보 =====
to_ctry_cd: "KR",                // 수령자 국가 코드
to_name: "이응*",                 // 수령자 명 (마스킹됨)
to_name_eng: "",                 // 수령자 영문명
to_ctry_name_eng: "Republic of Korea",  // 국가명 (영문)
to_tel: "",                      // 수령자 전화번호
to_htel: "010-4***-2958",        // 수령자 휴대폰
to_email: "",                    // 수령자 이메일
to_state: "",                    // 수령자 주/도
to_city: "",                     // 수령자 시/군
to_addr1: "경기도 부천시 부일로 515 **********",  // 주소1
to_addr2: "",                    // 주소2
to_zipcd: "14642",               // 우편번호
ship_cost: 0,                    // 배송비

// ===== 상품 관리 정보 =====
c_sale_cd: "m2023100467b8fb58d",   // 판매자 관리코드 (내부 SKU)
ord_curr_cd: "KRW",              // 통화 코드
curr_sign: "₩",                  // 통화 기호
gprivate_no: "",                 // 개인통관번호 (해외배송용)
barcode: "",                     // 바코드
model_no: "",                    // 모델번호
depot_no: 129627,                // 배송처(창고) 코드
depot_name: "창고",               // 배송처 이름
```

```
    invoice_send_time: "2025-01-02 16:50:29",  // 운송장 전송일
    gift_name: "",                  // 사은품 정보
    tag_pack: null,                 // 태그팩

    // ===== 상태 플래그 =====
    map_yn: 1,                      // SKU 상품 매칭 여부 (1: 매칭됨)
    ship_avail_yn: 0,               // 출고 가능 여부 (0: 불가, 1: 가능)
    ship_unable_reason: "출고가능상태(결제완료/신규주문/출고대기/출고보류/운송장출력) 아님",  // 출고 불가 사유
    ord_status_msg: null,           // 클레임 사유
    exchange_yn: 0,                 // 교환 여부
    exchange_ord_yn: 0,             // 교환 주문 여부
    now_time: "2026-01-08 17:43:18",   // 현재 시간 (서버)
    opt_custom_cd: null,            // 옵션 관리코드
    opt_sale_cd_pack: null,         // 옵션 판매코드 팩
    memo_yn: 0,                     // 메모 유무
    memo_complete_yn: 0,            // 메모 완료 여부
    bundle_avail_yn: 1,             // 합포장 가능 여부

    // ===== 세트 상품 정보 =====
    set_no: 247301,                 // 세트 번호
    set_cd: "T49717825762",         // 세트 코드
    pack_unit: 4,                   // 출고 수량 ⭐
    out_cnt: 4,                     // 총 출고 수량
    set_name: "뉴로마스터 3+1 세트",    // 세트 상품명 ⭐
    main_prod_no: 36820529,         // 메인 상품 번호

    // ===== 기타 =====
    pay_method: null,               // 결제 방법 (0:현금, 1:카드, 2:이머니)
    misc17: null,                   // 기타 필드
    misc19: null,
    misc20: null,
    misc_etc: null,
    notice_msg: "",                 // 특이사항 메시지 (예: "오늘출발")
    shop_ship_no: "2024123184304161",  // 쇼핑몰 배송번호
    order_msg: null,                // 추가 메시지
    api_id: "2024123199044921 2024123132110711",  // API ID
    claim_refund_sales: null,       // 환불 금액
```

```
      claim_fault_type: null,          // 클레임 귀책 유형
      dupl_doubt_yn: 0,                // 중복 의심 주문 여부
      cart_cd: null,                   // 장바구니 코드

      // ===== 반품/반송 정보 =====
      return_wdate: null,              // 반송 접수 요청일
      return_carr_no: null,            // 반송 택배사 코드
      return_carr_id: null,            // 반송 택배사 ID
      return_invoice_no: null,         // 반송 송장 번호

      // ===== 해외배송 정보 =====
      global_carr_no: null,            // 해외 택배사 코드
      global_pa_carr_no: null,         // 해외 상위 택배사 코드
      global_carr_name: null,          // 해외 택배사 이름
      global_invoice_no: null,         // 해외 송장 번호
      global_invoice_time: null,       // 해외 송장 등록일
      global_invoice_print_time: null,  // 해외 송장 출력일
      carr_label_url: null,            // 택배 라벨 URL

      // ===== 기타 시스템 필드 =====
      sol_no_t: null,
      uniq_t: null,
      shop_prod_url: "http://smartstore.naver.com/miribio/products/9250736
478",  // 상품 URL
      delivery_attribute_type: "normal"  // 배송 속성 (normal/guarantee/hope
등)
    }
  ],

  // ===== 페이지네이션 정보 =====
  recordsTotal: 500,          // 전체 주문건수 (묶음 기준)
  recordsFiltered: 500,       // 검색된 주문건수 (묶음 기준)
  recordsTotalCount: 520      // 전체 주문건수 (단건 기준, 합포장 포함)
}
```

13. 플레이오토 공식 문서

- 주문 리스트 조회

- 쇼핑몰 코드 조회
- AI혁신팀 계정으로 로그인 하시면 됩니다.