SellerPort - Google Ads API Integration Design Document

Version: 1.0 **Date:** December 2024 **Company:** SellerPort **Website:** https://sellerport.app

---

1. Executive Summary

SellerPort is a SaaS platform designed for Korean e-commerce sellers to analyze and optimize their advertising performance across multiple channels. This document outlines our integration with the Google Ads API to provide users with comprehensive advertising analytics.

---

2. Company Overview

2.1 Business Description

SellerPort helps online sellers track, analyze, and optimize their advertising spend across multiple platforms including:

- Google Ads
- Meta Ads (Facebook/Instagram)
- Naver Search Ads

2.2 Target Users

- Korean e-commerce sellers
- Online store owners
- Small to medium-sized businesses running digital ads

2.3 Value Proposition

- Unified dashboard for multi-channel ad performance
- ROAS (Return on Ad Spend) calculation and tracking
- Cross-platform attribution and conversion tracking

---

3. Google Ads API Usage

3.1 Purpose

We use the Google Ads API to **READ** advertising performance data from our users' accounts. We do NOT create, modify, or delete any campaigns or ads.

3.2 Data Retrieved

Data Type	API Resource	Purpose
Campaign Performance	`campaign`	Display campaign-level metrics
Ad Group Performance	`ad_group`	Detailed ad group analytics
Cost Metrics	`metrics.cost_micros`	Calculate ad spend
Click Metrics	`metrics.clicks`	Track engagement

Impression Metrics	`metrics.impressions`	Measure reach
Conversion Metrics	`metrics.conversions`	Track conversions

3.3 API Methods Used

- `GoogleAdsService.Search` - Query performance data
- `CustomerService.ListAccessibleCustomers` - List linked accounts

3.4 Access Level Required

- **Read-only access** to campaign performance data
- No write operations performed

---

4. OAuth 2.0 Authentication Flow

4.1 Flow Diagram

```
┌─────────────┐     ┌─────────────┐     ┌─────────────┐
│   User      │     │  SellerPort │     │  Google     │
│  (Seller)   │     │   Server    │     │  OAuth      │
└──────┬──────┘     └──────┬──────┘     └──────┬──────┘
       │                   │                   │
       │ 1. Click          │                   │
       │ "Connect Google"  │                   │
       │──────────────────>│                   │
       │                   │                   │
       │                   │ 2. Redirect to    │
       │                   │ Google OAuth      │
       │<─────────────────────────────────────>│
       │                   │                   │
       │ 3. User grants    │                   │
       │ permission        │                   │
       │──────────────────────────────────────>│
       │                   │                   │
       │                   │ 4. Authorization  │
       │                   │ code callback     │
       │                   │<──────────────────│
       │                   │                   │
       │                   │ 5. Exchange for   │
       │                   │ access token      │
       │                   │──────────────────>│
       │                   │                   │
       │                   │ 6. Access token   │
       │                   │ + refresh token   │
       │                   │<──────────────────│
       │                   │                   │
       │ 7. Connection     │                   │
       │ successful        │                   │
       │<──────────────────│                   │
       │                   │                   │
```

4.2 OAuth Scopes Requested

```
https://www.googleapis.com/auth/adwords
```

4.3 Token Management

- Access tokens are stored securely in encrypted database
- Refresh tokens are used to maintain access
- Users can revoke access at any time from their dashboard

---

5. Data Security & Privacy

5.1 Data Storage

- All data stored in Supabase (PostgreSQL) with encryption at rest
- Access tokens encrypted using AES-256
- Database hosted in secure cloud infrastructure

5.2 Data Access Controls

- Users can only access their own advertising data
- Role-based access control (RBAC) implemented
- Admin access logged and audited

5.3 Data Retention

- Performance data retained while user account is active
- Users can request data deletion at any time
- Upon account deletion, all associated data is removed

5.4 Compliance

- GDPR compliant data handling
- Korean PIPA (Personal Information Protection Act) compliant

---

6. User Interface Screenshots

6.1 Dashboard Overview

The main dashboard displays:

- Total ad spend across all platforms
- ROAS metrics and trends
- Click and conversion statistics
- Campaign performance comparison

6.2 Google Ads Connection

Users connect their Google Ads account via:

1. Navigate to "Ad Channels" page
2. Click "Connect Google Ads"
3. Complete OAuth consent flow
4. View synced campaigns

---

7. Technical Architecture

7.1 Technology Stack

Component	Technology
Frontend	Next.js 14 (React)
Backend	Next.js API Routes
Database	Supabase (PostgreSQL)
Authentication	Supabase Auth
Hosting	Vercel

7.2 API Integration Architecture

```
┌─────────────────────────────────────────────────────────┐
│                      SellerPort                          │
├─────────────────────────────────────────────────────────┤
│                                                         │
│  ┌─────────────┐    ┌─────────────┐    ┌─────────────┐   │
│  │  Frontend   │    │ API Routes  │    │  Database   │   │
│  │  (Next.js)  │────│  (Next.js)  │────│ (Supabase)  │   │
│  └─────────────┘    └──────┬──────┘    └─────────────┘   │
│                            │                            │
└────────────────────────────┼────────────────────────────┘
                             │
              ┌──────────────┼──────────────┐
              │              │              │
              ▼              ▼              ▼
        ┌──────────┐    ┌──────────┐    ┌──────────┐
        │  Google  │    │   Meta   │    │  Naver   │
        │ Ads API  │    │ Ads API  │    │ Ads API  │
        └──────────┘    └──────────┘    └──────────┘
```

8. Rate Limiting & Best Practices

8.1 API Usage Patterns

- Batch requests where possible
- Cache frequently accessed data
- Implement exponential backoff for retries

8.2 Request Frequency

- Daily sync of campaign performance data
- Real-time sync on user request
- Maximum 10,000 requests per day expected

9. Error Handling

9.1 Error Scenarios

Error Type	Handling
Token expired	Auto-refresh using refresh token
Rate limited	Exponential backoff retry
Permission denied	Notify user to re-authenticate
Network error	Retry with timeout

9.2 User Notifications

- Clear error messages displayed to users
- Automatic retry for transient errors
- Manual reconnection option for auth errors

---

10. Contact Information

Company: SellerPort **Website:** https://sellerport.app **API Contact Email:** usisst8888@gmail.com **Technical Support:** support@sellerport.app

---

11. Appendix

11.1 Sample API Request

```
// Fetch campaign performance data
const query = `
  SELECT
    campaign.id,
    campaign.name,
    metrics.impressions,
    metrics.clicks,
    metrics.cost_micros,
    metrics.conversions
  FROM campaign
  WHERE segments.date DURING LAST_30_DAYS
`;
```

11.2 Data Flow Summary

1. User connects Google Ads account via OAuth
2. SellerPort receives authorization code
3. Exchange code for access/refresh tokens
4. Store tokens securely in database
5. Fetch performance data using Google Ads API
6. Display metrics in user dashboard
7. Refresh data periodically or on-demand

Document prepared for Google Ads API Basic Access Application