Amazon Advertising API Setup Guide

📋 Prerequisites

- Amazon Seller Central or Vendor Central account
- Active Amazon Advertising account
- Products with existing PPC campaigns (optional but recommended)

🔑 Getting API Credentials

Step 1: Register as a Developer

1. Go to Amazon Advertising API (https://advertising.amazon.com/API/docs/en-us/get-started)
2. Click “Register as a developer”
3. Accept the API License Agreement
4. Fill out the developer registration form

Step 2: Create an Application

1. Log in to Amazon Advertising Console (https://advertising.amazon.com/)
2. Navigate to API section
3. Click “Create App”
4. Fill in app details:
   - **App Name**: “PPC Optimizer” (or your choice)
   - **Description**: “Automated PPC optimization tool”
   - **Redirect URI**: `https://localhost` (for testing)

Step 3: Get Client ID and Client Secret

After creating the app, you’ll receive:
- **Client ID**: `amzn1.application-oa2-client.xxxxx`
- **Client Secret**: `xxxxxxxxxxxxxxxxxxxx`

Save these securely!

Step 4: Generate Refresh Token

Method 1: Using OAuth 2.0 Flow (Recommended)

1. Build authorization URL:

```
https://www.amazon.com/ap/oa?client_id=YOUR_CLIENT_ID&scope=advertising::campaign_management&response_type=code&redirect_uri=https://localhost
```

1. Open this URL in browser
2. Log in with your Amazon Advertising account
3. Authorize the application
4. You’ll be redirected to: `https://localhost?code=AUTH_CODE&scope=...`
5. Copy the `AUTH_CODE` from the URL

6. Exchange auth code for refresh token:

Windows PowerShell:

```
$body = @{
    grant_type = 'authorization_code'
    code = 'YOUR_AUTH_CODE'
    redirect_uri = 'https://localhost'
    client_id = 'YOUR_CLIENT_ID'
    client_secret = 'YOUR_CLIENT_SECRET'
}

Invoke-RestMethod -Uri 'https://api.amazon.com/auth/o2/token' -Method Post -Body $body
```

Python:

```
import requests

data = {
    'grant_type': 'authorization_code',
    'code': 'YOUR_AUTH_CODE',
    'redirect_uri': 'https://localhost',
    'client_id': 'YOUR_CLIENT_ID',
    'client_secret': 'YOUR_CLIENT_SECRET'
}

response = requests.post('https://api.amazon.com/auth/o2/token', data=data)
tokens = response.json()
print('Refresh Token:', tokens['refresh_token'])
```

1. Save the **refresh_token** - it doesn't expire!

Step 5: Get Profile ID

Use this script to list your profiles:

```python
import requests

# Use your credentials
access_token = 'YOUR_ACCESS_TOKEN'
client_id = 'YOUR_CLIENT_ID'

headers = {
    'Authorization': f'Bearer {access_token}',
    'Amazon-Advertising-API-ClientId': client_id,
    'Content-Type': 'application/json'
}

response = requests.get(
    'https://advertising-api.amazon.com/v2/profiles',
    headers=headers
)

profiles = response.json()
for profile in profiles:
    print(f"Profile ID: {profile['profileId']}")
    print(f"Name: {profile['accountInfo']['name']}")
    print(f"Marketplace: {profile['countryCode']}")
    print('---')
```

Step 6: Update config.json

Put all credentials in `config.json`:

```json
{
  "amazon_api": {
    "region": "NA",
    "profile_id": "1234567890",
    "client_id": "amzn1.application-oa2-client.xxxxx",
    "client_secret": "your_client_secret_here",
    "refresh_token": "Atzr|IwEBxxxx"
  }
}
```

🌍 Region Codes

- **NA**: North America (US, CA, MX)
- **EU**: Europe (UK, DE, FR, IT, ES)
- **FE**: Far East (JP, AU, IN)

🔒 Security Best Practices

1. **Never commit config.json** to version control
2. **Use environment variables** for production:
   `cmd`
   `set AMAZON_CLIENT_ID=your_client_id`
   `set AMAZON_CLIENT_SECRET=your_client_secret`
   `set AMAZON_REFRESH_TOKEN=your_refresh_token`
3. **Rotate credentials** periodically

4. **Limit API permissions** to only what's needed
5. **Monitor API usage** in Amazon Advertising Console

🧪 Test Your Setup

Run this test script:

```
import requests
import json

# Load config
with open('config.json', 'r') as f:
    config = json.load(f)

# Get access token
token_response = requests.post(
    'https://api.amazon.com/auth/o2/token',
    data={
        'grant_type': 'refresh_token',
        'refresh_token': config['amazon_api']['refresh_token'],
        'client_id': config['amazon_api']['client_id'],
        'client_secret': config['amazon_api']['client_secret']
    }
)

if token_response.status_code == 200:
    print("✅ Authentication successful!")
    access_token = token_response.json()['access_token']

    # Test API call
    headers = {
        'Authorization': f'Bearer {access_token}',
        'Amazon-Advertising-API-ClientId': config['amazon_api']['client_id'],
        'Amazon-Advertising-API-Scope': config['amazon_api']['profile_id']
    }

    campaigns_response = requests.get(
        f"https://advertising-api.amazon.com/v2/sp/campaigns",
        headers=headers
    )

    if campaigns_response.status_code == 200:
        campaigns = campaigns_response.json()
        print(f"✅ Found {len(campaigns)} campaigns")
        print("Setup complete! You're ready to use the optimizer.")
    else:
        print(f"❌ API call failed: {campaigns_response.status_code}")
        print(campaigns_response.text)
else:
    print(f"❌ Authentication failed: {token_response.status_code}")
    print(token_response.text)
```

Save as `test_api.py` and run:

```
python test_api.py
```

🆘 Troubleshooting

“Invalid client_id”

- Double-check your client ID matches the one from Amazon Advertising Console
- Ensure no extra spaces or line breaks

“Invalid refresh_token”

- Refresh tokens can expire if not used for 6 months
- Generate a new one using the OAuth flow

“Access denied”

- Ensure you’re using the correct profile ID
- Check that your app has necessary permissions
- Re-authorize your app in Amazon Advertising Console

“Rate limit exceeded”

- The optimizer respects rate limits automatically
- If you’re testing, wait a few minutes between calls

📚 Additional Resources

- Amazon Advertising API Documentation (https://advertising.amazon.com/API/docs/)
- OAuth 2.0 Guide (https://advertising.amazon.com/API/docs/en-us/guides/get-started/oauth)
- API Reference (https://advertising.amazon.com/API/docs/en-us/reference/)

Next: Return to README.md (README.md) to configure and run the optimizer