Social Media Providers Guide

Overview

The Social Media Service supports multiple social media management providers through a flexible provider architecture. This guide covers setup and usage for all supported providers.

Supported Providers

1. Ayrshare (Default - Recommended)

2. Buffer (Alternative)

Ayrshare Integration (Default Provider)

Overview

Ayrshare is the default social media management provider for this service. It provides white-label reselling capabilities, supports 13+ platforms, and offers comprehensive analytics.

Why Ayrshare?

- \u2705 **13+ Platforms**: Facebook, Twitter, Instagram, LinkedIn, TikTok, YouTube, Pinterest, Reddit, Telegram, Threads, Bluesky, Google Business, Snapchat
- \u2705 **White-Label Reselling**: Build and resell your own social media management solution
- \u2705 **Revenue Model**: Business Plan at $499/month with reselling support
- \u2705 **Advanced Features**: Webhooks, multi-user accounts, comprehensive analytics
- \u2705 **Better API**: More reliable, higher rate limits, better documentation

Prerequisites

1. Ayrshare account (https://www.ayrshare.com)
2. Business Plan subscription ($499/month for white-label features)
3. Ayrshare API key

Setup Steps

Step 1: Create Ayrshare Account

1. Visit https://www.ayrshare.com
2. Sign up for Business Plan
3. Complete account setup
4. Go to Dashboard → API Keys

5. Generate a new API key
6. Copy the API key securely

Step 2: Connect Social Media Accounts

1. Log in to Ayrshare dashboard
2. Click **Add Social Account**
3. Authenticate with each platform:
   - Facebook, Twitter, Instagram, LinkedIn
   - TikTok, YouTube, Pinterest, Reddit
   - Telegram, Threads, Bluesky, etc.
4. Note the profile IDs for each account

Step 3: Configure Service

Add to `.env` file:

```
# Provider Selection
SOCIAL_MEDIA_PROVIDER=ayrshare

# Ayrshare Configuration
AYRSHARE_API_URL=https://app.ayrshare.com/api
AYRSHARE_API_KEY=your_ayrshare_api_key_here
```

Step 4: Test Connection

```
# Start the service
./start.sh

# Test Ayrshare connection
curl -X GET "http://localhost:8007/api/v1/social-accounts" \
  -H "Authorization: Bearer YOUR_JWT_TOKEN"
```

API Usage Examples

Get Connected Profiles

```
curl -X GET "http://localhost:8007/api/v1/social-accounts" \
  -H "Authorization: Bearer YOUR_JWT_TOKEN"
```

Create a Post

```
curl -X POST "http://localhost:8007/api/v1/posts" \
  -H "Authorization: Bearer YOUR_JWT_TOKEN" \
  -H "Content-Type: application/json" \
  -d '{
    "content": "Check out our new feature! #innovation",
    "media_urls": ["https://blog-static.userpilot.com/blog/wp-content/uploads/
2023/03/10-effective-new-feature-announcement-examples-how-to-write-
one_6c5f1f10cba1bbc799eb6345bbef5ff6_2000.png
    "platforms": ["facebook", "twitter", "linkedin"],
    "scheduled_time": "2025-01-15T10:00:00Z"
  }'
```

Get Post Analytics

```
curl -X GET "http://localhost:8007/api/v1/analytics/{post_id}" \
  -H "Authorization: Bearer YOUR_JWT_TOKEN"
```

Ayrshare-Specific Features

Webhooks

Configure webhooks to receive real-time notifications:

```
curl -X POST "https://app.ayrshare.com/api/webhooks" \
  -H "Authorization: Bearer YOUR_AYRSHARE_API_KEY" \
  -H "Content-Type: application/json" \
  -d '{
    "url": "https://yourdomain.com/webhooks/ayrshare",
    "events": ["post.published", "post.failed", "analytics.updated"]
  }'
```

Multi-User Support

Ayrshare supports multi-user accounts with separate API keys:

1. Go to Dashboard → Team

2. Invite team members

3. Assign roles and permissions

4. Each user gets their own API key

Platform-Specific Content

```
curl -X POST "http://localhost:8007/api/v1/posts" \
  -H "Authorization: Bearer YOUR_JWT_TOKEN" \
  -H "Content-Type: application/json" \
  -d '{
    "platforms": {
      "facebook": {
        "post": "Check out our new feature!",
        "link": "https://example.com"
      },
      "twitter": {
        "post": "New feature alert! \ud83d\ude80 #innovation",
        "mediaUrls": ["https://octopuscrm.io/wp-content/uploads/2023/07/Thumbnail-How-
to-Announce-New-Product-Features-Using-Email-Marketing.webp
      },
      "instagram": {
        "post": "New feature \u2728",
        "mediaUrls": ["https://cdn.prod.website-files.com/685d3f27e667cd-
f05fe197f8/685d3f27e667cdf05fe1b16c_6733550da114276f65c927b6_670013740c890997004597b1_
618025b83ab97db87bb355e1_Venmo%252525202.jpeg
      }
    }
  }'
```

Buffer Integration (Alternative Provider)

Overview

Buffer is an alternative social media management provider. While it supports fewer platforms than Ayrshare, it's a solid choice for basic scheduling needs.

Prerequisites

1. Buffer account (https://buffer.com)
2. Buffer API access token
3. Connected social media profiles in Buffer

Step 1: Create Buffer Account

1. Visit https://buffer.com
2. Sign up or log in
3. Connect your social media accounts:
   - Twitter/X
   - Facebook
   - LinkedIn
   - Instagram
   - TikTok

Step 2: Get Buffer API Access Token

Option A: Using Buffer's Developer Portal

1. Visit https://publish.buffer.com/profile
2. Go to **Settings → API**
3. Click **Create New Access Token**
4. Copy the access token

Option B: OAuth Flow (Production)

For production applications, implement OAuth 2.0:

1. Register your app at https://buffer.com/developers
2. Get your Client ID and Client Secret
3. Implement OAuth flow:
   ```
   https://bufferapp.com/oauth2/authorize?
       client_id=YOUR_CLIENT_ID&
       redirect_uri=YOUR_REDIRECT_URI&
       response_type=code
   ```
4. Exchange authorization code for access token

Step 3: Configure Service

Environment Variables

Add to `.env` file:

```
# Buffer Configuration
BUFFER_API_URL=https://api.bufferapp.com/1
BUFFER_ACCESS_TOKEN=your_buffer_access_token_here
```

Test Configuration

```
# Start the service
./start.sh

# Test Buffer connection
curl -X GET "http://localhost:8007/api/v1/buffer/profiles" \
  -H "Authorization: Bearer YOUR_JWT_TOKEN"
```

Step 4: Connect Social Accounts

Create Social Account in Service

```
curl -X POST "http://localhost:8007/api/v1/social-accounts" \
  -H "Authorization: Bearer YOUR_JWT_TOKEN" \
  -H "Content-Type: application/json" \
  -d '{
    "platform": "twitter",
    "account_name": "@yourusername",
    "buffer_profile_id": "BUFFER_PROFILE_ID",
    "access_token": "BUFFER_ACCESS_TOKEN"
  }'
```

Get Buffer Profile IDs

```
curl -X GET "http://localhost:8007/api/v1/buffer/profiles" \
  -H "Authorization: Bearer YOUR_JWT_TOKEN"
```

Response:

```
[
  {
    "id": "5f7c8b9a1e4f3a0001234567",
    "service": "twitter",
    "formatted_service": "Twitter",
    "formatted_username": "@yourusername"
  }
]
```

Step 5: Schedule Posts

Create Draft Post

```
curl -X POST "http://localhost:8007/api/v1/posts" \
  -H "Authorization: Bearer YOUR_JWT_TOKEN" \
  -H "Content-Type: application/json" \
  -d '{
    "social_account_id": 1,
    "content": "Hello from Social Media Service! 🚀",
    "scheduled_time": "2025-12-25T10:00:00",
    "platform": "twitter"
  }'
```

Schedule via Buffer

```
curl -X POST "http://localhost:8007/api/v1/posts/1/schedule" \
  -H "Authorization: Bearer YOUR_JWT_TOKEN"
```

Buffer API Features Used

1. Get User Profile

```
GET /user.json?access_token=TOKEN
```

2. Get Profiles (Social Accounts)

```
GET /profiles.json?access_token=TOKEN
```

3. Create Update (Post)

```
POST /updates/create.json
{
  "profile_ids": ["PROFILE_ID"],
  "text": "Post content",
  "scheduled_at": 1640433600,
  "media": {"photo": "https://example.com/image.jpg"}
}
```

4. Get Update Statistics

```
GET /updates/{id}/statistics.json?access_token=TOKEN
```

Platform-Specific Notes

Twitter/X

- Character limit: 280
- Supports images, GIFs, videos
- Hashtags recommended

Facebook

- Character limit: 63,206 (but shorter is better)
- Supports links, images, videos
- Best times: 1-4 PM weekdays

LinkedIn

- Character limit: 3,000
- Professional tone
- Best times: Tuesday-Thursday, 7-8 AM

Instagram

- Requires image/video
- Character limit: 2,200
- First comment can include additional hashtags

Error Handling

Common Errors

401 Unauthorized

```json
{"error": "Unauthorized", "message": "Invalid access token"}
```

Solution: Refresh your Buffer access token

403 Forbidden

```json
{"error": "Forbidden", "message": "Profile not accessible"}
```

Solution: Reconnect the social account in Buffer

503 Service Unavailable

```json
{"error": "Buffer API unavailable"}
```

Solution: Wait and retry, Buffer may be experiencing issues

Rate Limits

- **Buffer API**: 60 requests per minute per access token
- **Service handles**: Automatic retry with exponential backoff
- **Best practice**: Cache Buffer profile data

Security Best Practices

1. **Never commit tokens** to version control
2. **Use environment variables** for sensitive data
3. **Rotate tokens regularly** (every 90 days)
4. **Use OAuth in production** instead of personal access tokens
5. **Implement token encryption** in database

Testing

Test Buffer Connection

```
pytest tests/test_buffer_integration.py -v
```

Manual Testing

```
# Get profiles
curl https://api.bufferapp.com/1/profiles.json?access_token=YOUR_TOKEN

# Test authentication
curl https://api.bufferapp.com/1/user.json?access_token=YOUR_TOKEN
```

Troubleshooting

Issue: "Buffer profile not found"

Solution: Sync profiles using `/social-accounts/{id}/sync-buffer`

Issue: "Post scheduling failed"

Checks:
1. Verify Buffer access token is valid
2. Check social account is connected in Buffer
3. Ensure scheduled time is in the future
4. Verify content meets platform requirements

Issue: "Analytics not syncing"

Solution: Use `/analytics/sync` endpoint to manually trigger sync

Resources

- **Buffer API Docs**: https://buffer.com/developers/api
- **Buffer Status**: https://status.buffer.com
- **Support**: https://support.buffer.com

Next Steps

1. ✅ Create Buffer account and get access token
2. ✅ Configure service with Buffer credentials
3. ✅ Connect social accounts
4. ✅ Test post scheduling
5. ✅ Monitor analytics
6. 🚀 Start scheduling posts!