

# Provider Setup Guide

---

## Overview

This guide explains how to configure and use different social media providers with the Social Media Service.

## Supported Providers

---

### 1. Ayrshare (Default - Recommended)

#### Why Ayrshare?

- 13+ social media platforms
- White-label reselling support
- Revenue generation opportunities
- Comprehensive analytics
- Multi-user accounts
- Webhooks for real-time updates

#### Setup Steps:

##### 1. Sign Up for Ayrshare

- Visit: <https://www.ayrshare.com>
- Choose Business Plan (\$499/month) for white-label features
- Get your API key from the dashboard

##### 2. Configure Environment

```
bash
# .env
SOCIAL_MEDIA_PROVIDER="ayrshare"
AYRSHARE_API_KEY="your-ayrshare-api-key-here"
AYRSHARE_API_URL="https://app.ayrshare.com/api"
```

##### 3. Connect Social Accounts

- Log in to Ayrshare dashboard
- Connect your social media accounts
- Copy profile IDs for use in API calls

##### 4. Test Connection

```
bash
curl -X GET "http://localhost:8007/api/v1/social-accounts" \
-H "Authorization: Bearer your-jwt-token"
```

#### Supported Platforms:

- Facebook (Pages & Groups)
- Twitter/X
- Instagram (Feed & Stories)
- LinkedIn (Personal & Company)
- TikTok
- YouTube (Community Posts)

- Pinterest
- Reddit
- Telegram
- Threads
- Bluesky
- Google Business Profile
- Snapchat

## 2. Buffer (Alternative)

### Why Buffer?

- Easy to use
- Good for small teams
- Basic scheduling features
- Limited to 5 platforms
- No white-label support

### Setup Steps:

#### 1. Sign Up for Buffer

- Visit: <https://buffer.com>
- Sign up for a plan
- Generate OAuth access token

#### 2. Configure Environment

```
bash
# .env
SOCIAL_MEDIA_PROVIDER="buffer"
BUFFER_ACCESS_TOKEN="your-buffer-access-token-here"
BUFFER_API_URL="https://api.bufferapp.com/1"
```

#### 3. Get Access Token

- Go to <https://buffer.com/developers/apps>
- Create a new app
- Complete OAuth flow
- Copy access token

#### 4. Test Connection

```
bash
curl -X GET "http://localhost:8007/api/v1/social-accounts" \
-H "Authorization: Bearer your-jwt-token"
```

### Supported Platforms:

- Facebook
- Twitter/X
- Instagram
- LinkedIn
- Pinterest

# Configuration Options

---

## Environment Variables

### Required Variables:

```
# Provider Selection
SOCIAL_MEDIA_PROVIDER="ayrshare" # or "buffer"

# Ayrshare (if using Ayrshare)
AYRSHARE_API_KEY="your-key"

# Buffer (if using Buffer)
BUFFER_ACCESS_TOKEN="your-token"
```

### Optional Variables:

```
# Custom API URLs (advanced)
AYRSHARE_API_URL="https://app/ayrshare.com/api"
BUFFER_API_URL="https://api.bufferapp.com/1"
```

## Switching Providers

To switch from Buffer to Ayrshare:

### 1. Update .env:

```
bash
SOCIAL_MEDIA_PROVIDER="ayrshare"
AYRSHARE_API_KEY="your-ayrshare-key"
```

### 2. Restart Service:

```
bash
./restart.sh
```

### 3. Verify:

```
bash
curl http://localhost:8007/api/v1/health
```

No code changes needed!

## Usage Examples

---

### 1. Get Connected Profiles

```
from app.services.providers.provider_factory import get_provider

provider = get_provider() # Uses default from settings
profiles = await provider.get_profiles()

for profile in profiles:
    print(f"{profile['platform']}: {profile['username']}")
```

## 2. Create a Post

```
from datetime import datetime, timedelta

provider = get_provider()

# Schedule post for tomorrow
tomorrow = datetime.now() + timedelta(days=1)

post = await provider.create_post(
    profile_ids=['profile1', 'profile2'],
    text="Check out our new product! #innovation",
    media={
        'photos': ['https://i.ytimg.com/vi/g5t3IoQWFzs/hqdefault.jpg'],
        'link': 'https://example.com/product'
    },
    scheduled_at=tomorrow
)

print(f"Post created: {post['id']}")
```

## 3. Get Post Analytics

```
provider = get_provider()

analytics = await provider.get_post_analytics('post-id-123')

print(f"Likes: {analytics['likes']}")  

print(f"Comments: {analytics['comments']}")  

print(f"Engagement Rate: {analytics['engagement_rate']}%)")
```

## 4. Use Specific Provider

```
# Force Ayrshare
ayrshare = get_provider('ayrshare', api_key='custom-key')

# Force Buffer
buffer = get_provider('buffer', access_token='custom-token')
```

## API Endpoints

All endpoints work with any configured provider:

## Social Accounts

```
# List accounts
GET /api/v1/social-accounts

# Get account
GET /api/v1/social-accounts/{id}

# Create account
POST /api/v1/social-accounts

# Sync with provider
POST /api/v1/social-accounts/{id}/sync-provider

# Test connection
POST /api/v1/social-accounts/{id}/test
```

## Posts

```
# Create post
POST /api/v1/posts

# Schedule post
POST /api/v1/posts/{id}/schedule

# Get calendar
GET /api/v1/posts/calendar

# Bulk schedule
POST /api/v1/posts/bulk-schedule
```

## Analytics

```
# Get analytics
GET /api/v1/analytics/{id}

# Sync analytics
POST /api/v1/analytics-sync

# Get summary
GET /api/v1/analytics/summary
```

## Troubleshooting

### Provider Not Connecting

**Symptoms:** ProviderError: API key not configured

**Solution:**

1. Check `.env` file has correct provider settings
2. Verify API key/token is valid
3. Restart service: `./restart.sh`
4. Check logs: `tail -f logs/service.log`

## Authentication Failed

**Symptoms:** ProviderError: Authentication failed

**Solution:**

- **Ayrshare:** Verify API key at <https://app.ayrshare.com/account>
- **Buffer:** Regenerate access token at <https://buffer.com/developers/apps>
- Check API key format (no extra spaces)

## Profiles Not Found

**Symptoms:** Empty profiles list

**Solution:**

1. Connect social accounts in provider dashboard
2. Wait 1-2 minutes for sync
3. Call `/api/v1/social-accounts/{id}/sync-provider`
4. Verify account permissions

## Post Scheduling Failed

**Symptoms:** ProviderError: Failed to schedule post

**Common Causes:**

- Invalid profile IDs
- Scheduled time in the past
- Media URLs not accessible
- Platform-specific content restrictions

**Solution:**

1. Verify profile IDs exist
2. Check scheduled time is in future
3. Test media URLs are publicly accessible
4. Review platform guidelines

## Rate Limits

**Symptoms:** ProviderError: Rate limit exceeded

**Solution:**

- **Ayrshare:** 1000 requests/hour (default)
- **Buffer:** 60 requests/hour (default)
- Implement exponential backoff
- Consider upgrading plan
- Cache profile data

## Best Practices

---

### 1. Environment Configuration

 **DO:**

- Use environment variables for API keys
- Keep different keys for dev/staging/prod
- Use `.env.local` for local overrides
- Never commit API keys to git

## DON'T:

- Hard-code API keys in code
- Share API keys across environments
- Commit `.env` files
- Use production keys in development

## 2. Error Handling

```
from app.services.providers.base_provider import ProviderError

try:
    provider = get_provider()
    post = await provider.create_post(...)
except ProviderError as e:
    logger.error(f"Provider error: {e.message}")
    # Handle gracefully
    # Maybe retry or notify user
```

## 3. Testing

```
# Use mock provider for tests
from unittest.mock import Mock

mock_provider = Mock(spec=SocialMediaProvider)
mock_provider.get_profiles.return_value = [
    {'id': '1', 'platform': 'facebook', 'username': 'test'}
]

# Inject into service
service = SocialAccountService(db)
service.provider = mock_provider
```

## 4. Performance

- Cache provider instances (singleton pattern)
- Cache profile data (15-minute TTL)
- Batch operations when possible
- Use `async/await` consistently
- Monitor API usage

## Migration Checklist

When switching providers:

- [ ] Update `.env` with new provider settings
- [ ] Test authentication: `provider.authenticate()`
- [ ] Verify profiles: `provider.get_profiles()`
- [ ] Test post creation: `provider.create_post(...)`
- [ ] Check analytics: `provider.get_post_analytics(...)`
- [ ] Update monitoring/alerts
- [ ] Document provider-specific features
- [ ] Train team on new dashboard

- [ ] Monitor error rates for 24 hours
- [ ] Update client applications (if needed)

## Support

---

### Ayrshare Support

- Documentation: <https://www.ayrshare.com/docs>
- Email: support@ayrshare.com
- Community: <https://community.ayrshare.com>

### Buffer Support

- Documentation: <https://buffer.com/developers/api>
- Email: hello@buffer.com
- Status: <https://status.buffer.com>

### Service Issues

- GitHub Issues: <https://github.com/mission-engadi/social-media-service/issues>
  - Internal Slack: #social-media-service
  - On-call: Check PagerDuty rotation
- 

**Last Updated:** December 24, 2025

**Version:** 1.0.0

**Maintainer:** Engineering Team