# OAuth Providers Setup Guide

Complete guide for setting up OAuth authentication providers for the Bedtime Blog system.

## Table of Contents



## Overview

The blog supports three OAuth providers:

- **Google** - Most reliable and widely used
- **Facebook** - Good for social media integration
- **Twitter/X** - For Twitter users and cross-posting

Each provider requires setting up a developer account and creating an application to get the necessary credentials.

## Google OAuth Setup

### Step 1: Create Google Cloud Project

1. Go to Google Cloud Console
2. Create a new project or select existing one
3. Name it something like "Bedtime Blog OAuth"

### Step 2: Enable Google+ API

1. Navigate to **APIs & Services** > **Library**
2. Search for "Google+ API" and enable it
3. Also enable "People API" for better user profile access

### Step 3: Create OAuth 2.0 Credentials

1. Go to **APIs & Services** > **Credentials**
2. Click **Create Credentials** > **OAuth 2.0 Client IDs**
3. Configure the consent screen first if prompted:
    - **Application name**: "Bedtime Blog"
    - **User support email**: Your email
    - **Developer contact email**: Your email
    - **Authorized domains**: Add your domain (e.g., `blog.ingasti.com`)

### Step 4: Configure OAuth Client

1. **Application type**: Web application
2. **Name**: "Bedtime Blog Web Client"
3. **Authorized JavaScript origins**:

```
https://blog.ingasti.com
http://localhost:3000 (for development)
```

4. **Authorized redirect URIs**:

```
https://bapi.ingasti.com/api/auth/google/callback
http://localhost:5000/api/auth/google/callback (for development)
```

### Step 5: Get Credentials

1. After creation, you'll get:
   - **Client ID**: `your-google-client-id.apps.googleusercontent.com`
   - **Client Secret**: `your-google-client-secret`
2. Download the JSON file for backup

### Example Google Environment Variables

```
GOOGLE_CLIENT_ID=123456789-abcdef.apps.googleusercontent.com
GOOGLE_CLIENT_SECRET=GOCSPX-your-secret-here
```

## Facebook OAuth Setup

### Step 1: Create Facebook Developer Account

1. Go to Facebook for Developers
2. Log in with your Facebook account
3. Create a developer account if needed

### Step 2: Create Facebook App

1. Click **Create App**
2. Choose **Build Connected Experiences**
3. **App name**: "Bedtime Blog"
4. **App contact email**: Your email
5. Select **Business** as the app type

### Step 3: Configure Facebook Login

1. In your app dashboard, go to **Products**

2. Find **Facebook Login** and click **Set Up**
3. Choose **Web** platform

### Step 4: Configure OAuth Settings

1. Go to **Facebook Login** > **Settings**
2. **Valid OAuth Redirect URIs**:

```
https://bapi.ingasti.com/api/auth/facebook/callback
http://localhost:5000/api/auth/facebook/callback
```

3. **Valid OAuth Redirect URIs for Mobile**:

```
https://blog.ingasti.com/
```

### Step 5: Get App Credentials

1. Go to **Settings** > **Basic**
2. Copy:
   - **App ID**: Your Facebook App ID
   - **App Secret**: Click **Show** to reveal

### Step 6: App Review (For Production)

For production use, you'll need to submit your app for review:

1. Go to **App Review**
2. Request permissions for `email` and `public_profile`
3. Provide app details and privacy policy

### Example Facebook Environment Variables

```
FACEBOOK_APP_ID=1234567890123456
FACEBOOK_APP_SECRET=your-facebook-app-secret-here
```

## Twitter/X OAuth Setup

### Step 1: Create Twitter Developer Account

1. Go to Twitter Developer Platform
2. Apply for a developer account
3. Complete the application process (may take 1-2 days)

### Step 2: Create Twitter App

1. Go to **Developer Portal**
2. Click **Create Project**
3. **Project name**: "Bedtime Blog"
4. **Use case**: "Building tools for Twitter users"
5. **Project description**: Describe your blog's purpose

### Step 3: Create App within Project

1. **App name**: "Bedtime Blog Auth"
2. Get your **API Key** and **API Secret Key**

### Step 4: Configure App Permissions

1. Go to your app settings
2. **App permissions**: Read and write
3. **Type of app**: Web App

### Step 5: Configure Authentication Settings

1. Enable **3-legged OAuth**
2. **Callback URLs**:

```
https://bapi.ingasti.com/api/auth/twitter/callback
http://localhost:5000/api/auth/twitter/callback
```

3. **Website URL**: `https://blog.ingasti.com`
4. **Terms of Service**: Link to your terms
5. **Privacy Policy**: Link to your privacy policy

### Step 6: Get Credentials

1. **API Key**: Your consumer key
2. **API Secret Key**: Your consumer secret
3. **Bearer Token**: For API access

### Example Twitter Environment Variables

```
TWITTER_CONSUMER_KEY=your-twitter-api-key
TWITTER_CONSUMER_SECRET=your-twitter-api-secret
```

## Environment Configuration

### Development Environment (.env.local)

Create a `.env.local` file in your project root:

```
# Google OAuth
GOOGLE_CLIENT_ID=your-google-client-id.apps.googleusercontent.com
GOOGLE_CLIENT_SECRET=GOCSPX-your-google-secret

# Facebook OAuth
FACEBOOK_APP_ID=1234567890123456
FACEBOOK_APP_SECRET=your-facebook-app-secret

# Twitter OAuth
TWITTER_CONSUMER_KEY=your-twitter-api-key
TWITTER_CONSUMER_SECRET=your-twitter-api-secret

# OAuth URLs (development)
GOOGLE_CALLBACK_URL=http://localhost:5000/api/auth/google/callback
FACEBOOK_CALLBACK_URL=http://localhost:5000/api/auth/facebook/callback
TWITTER_CALLBACK_URL=http://localhost:5000/api/auth/twitter/callback

# Frontend URL for redirects
FRONTEND_URL=http://localhost:3000
```

### Production Environment

For production, use your deployment platform's environment variable system:

```
# Production OAuth URLs
GOOGLE_CALLBACK_URL=https://bapi.ingasti.com/api/auth/google/callback
FACEBOOK_CALLBACK_URL=https://bapi.ingasti.com/api/auth/facebook/callback
TWITTER_CALLBACK_URL=https://bapi.ingasti.com/api/auth/twitter/callback

# Production Frontend URL
FRONTEND_URL=https://blog.ingasti.com
```

### Kubernetes Secrets

For Kubernetes deployment, create secrets:

```
apiVersion: v1
kind: Secret
metadata:
  name: oauth-secrets
type: Opaque
stringData:
  GOOGLE_CLIENT_ID: "your-google-client-id"
  GOOGLE_CLIENT_SECRET: "your-google-secret"
  FACEBOOK_APP_ID: "your-facebook-app-id"
  FACEBOOK_APP_SECRET: "your-facebook-secret"
  TWITTER_CONSUMER_KEY: "your-twitter-key"
  TWITTER_CONSUMER_SECRET: "your-twitter-secret"
```

# Testing OAuth Integration

## Using the Operations Panel

1. Go to your blog's Operations Panel: `https://blog.ingasti.com/ops`
2. Navigate to **OAuth Config** tab
3. You should see configuration forms for each provider
4. Test each provider's connection

## Manual Testing

**Test Google OAuth:**

```
curl -X GET "https://bapi.ingasti.com/api/auth/google"
# Should redirect to Google login
```

**Test Facebook OAuth:**

```
curl -X GET "https://bapi.ingasti.com/api/auth/facebook"
# Should redirect to Facebook login
```

**Test Twitter OAuth:**

```
curl -X GET "https://bapi.ingasti.com/api/auth/twitter"
# Should redirect to Twitter login
```

## Frontend Integration

The frontend OAuth buttons should be available in:

- Login page: `/login`
- User registration
- Comment system (if implemented)

# Troubleshooting

## Common Issues

### 1. "Invalid OAuth Redirect URI" Error

**Problem**: The callback URL doesn't match what's configured in the provider.

**Solution**:

- Check that your callback URLs in the provider settings exactly match your environment variables
- Ensure you're using the correct protocol (http vs https)
- Verify domain spelling

**2. "App Not Yet Available" (Facebook)**

**Problem**: Facebook app is in development mode.

**Solution**:

- Add test users in Facebook Developer Console
- Submit app for review for production use
- Ensure privacy policy and terms of service are accessible

**3. "Invalid Consumer Key" (Twitter)**

**Problem**: Twitter API credentials are incorrect or expired.

**Solution**:

- Regenerate API keys in Twitter Developer Portal
- Ensure the app has correct permissions
- Check that callback URLs are properly configured

**4. "Access Denied" Errors**

**Problem**: Missing required permissions or scopes.

**Solution**:

- Review required scopes for each provider
- Ensure your app requests appropriate permissions
- Check that user has granted necessary permissions

## Debug Mode

Enable debug logging in your environment:

```
DEBUG=oauth:*
NODE_ENV=development
```

## Check OAuth Status

Use the admin panel to verify OAuth configuration:

1. Go to Operations Panel > OAuth Config
2. Each provider should show "Connected" or "Configured"
3. Test buttons should return success responses

### Security Considerations

1. **Never commit OAuth secrets to version control**
2. **Use different apps for development and production**
3. **Regularly rotate OAuth secrets**
4. **Monitor OAuth usage in provider dashboards**
5. **Implement proper error handling for OAuth failures**

### Provider-Specific Notes

#### Google

- Google+ API is deprecated but still works for basic authentication
- Consider migrating to People API for new implementations
- Google has strict requirements for production OAuth consent screens

#### Facebook

- Facebook login requires HTTPS in production
- App review is required for public access
- Privacy policy must be accessible and comprehensive

#### Twitter/X

- Twitter's OAuth 1.0a is more complex than OAuth 2.0
- Rate limiting is more restrictive than other providers
- Developer account approval can take time

## Support and Resources

### Official Documentation

- Google OAuth 2.0
- Facebook Login
- Twitter OAuth

### Blog-Specific Support

- Check Operations Panel > OAuth Config for real-time status
- Review API logs for detailed error messages
- Contact support if configuration issues persist

---

**Last Updated**: August 9, 2025 **Version**: 2.0 **Maintainer**: Blog Admin Team