

Platform Adapters Guide

This document provides comprehensive guidance on using the platform adapters in the Blog Writer SDK, including the newly added Shopify and Webflow adapters.

Overview

The Blog Writer SDK now supports **5 major platforms** for multi-platform content publishing:

Platform	Description	Scheduling	Analytics	Key Features
WordPress	WordPress.com and self-hosted sites	✓	✗	Categories, Tags, Media, Revisions
Medium	Medium publishing platform	✗	✓	Publications, Tags, Analytics
LinkedIn	Professional network articles	✗	✓	Professional audience, Analytics
Shopify	E-commerce store blogs	✓	✓	Product integration, Store branding
Webflow	Design-focused CMS	✗	✗	Rich design, Custom fields, Visual editor

Quick Start

Installation

```
import {
  ShopifyAdapter,
  WebflowAdapter,
  MultiPlatformPublisher,
  createPlatformCredentials
} from '@vercel/ai-blog-writer';
```

Basic Usage

```
// Initialize adapters
const shopifyAdapter = new ShopifyAdapter({
  storeUrl: 'mystore.myshopify.com',
  apiVersion: '2024-01'
});

const webflowAdapter = new WebflowAdapter({
  siteId: 'your-site-id',
  apiVersion: 'v1'
});

// Set up credentials
const shopifyCredentials = createPlatformCredentials('shopify', {
  type: 'private_app',
  storeUrl: 'mystore.myshopify.com',
  accessToken: 'your-access-token'
});

// Authenticate and publish
await shopifyAdapter.authenticate(shopifyCredentials);
const publishResult = await shopifyAdapter.publish(formattedContent);
```

Platform-Specific Guides

Shopify Adapter

Features

- **Blog Publishing:** Publish to Shopify store blogs
- **Product Integration:** Link products within blog content using `[product:handle]` syntax
- **Store Branding:** Automatically inherit store branding and theme
- **SEO Optimization:** Built-in SEO fields and meta tag management
- **Content Scheduling:** Schedule posts for future publication

Authentication Options

Private App (Recommended)

```
const credentials: ShopifyCredentials = {
  type: 'private_app',
  storeUrl: 'mystore.myshopify.com',
  accessToken: 'shpat_your_private_app_token',
  scopes: ['write_content', 'read_content', 'read_products']
};
```

OAuth2

```
const credentials: ShopifyCredentials = {
  type: 'oauth2',
  storeUrl: 'mystore.myshopify.com',
  accessToken: 'your_oauth_token',
  scopes: ['write_content', 'read_content']
};
```

Shopify-Specific Features

Product Linking

```
const blogPost: BlogPost = {
  title: "Summer Collection Highlights",
  content: `
    Check out our featured products:
    - [product:summer-dress] - Perfect for beach days
    - [product:sandals-leather] - Comfortable and stylish

    Shop the full collection now!
  `
};
```

Blog Management

```
// Get available tags
const tags = await shopifyAdapter.getTags();

// Check blog analytics
const analytics = await shopifyAdapter.getAnalytics({
  startDate: new Date('2024-01-01'),
  endDate: new Date()
});
```

Configuration Options

```
const shopifyConfig: ShopifyConfig = {
  storeUrl: 'mystore.myshopify.com',
  apiVersion: '2024-01', // Shopify API version
  maxRetries: 3, // Request retry attempts
  retryDelay: 1000, // Delay between retries (ms)
  timeout: 30000 // Request timeout (ms)
};
```

Webflow Adapter

Features

- **CMS Publishing:** Publish to Webflow CMS collections
- **Rich Text Support:** Advanced rich text formatting and styling
- **Asset Management:** Upload and manage images and media files
- **Custom Fields:** Support for custom CMS fields and data types
- **Site Publishing:** Automatically publish site changes after content updates
- **Design Integration:** Seamless integration with Webflow's visual editor

Authentication Options

API Token (Recommended for simple setups)

```
const credentials: WebflowCredentials = {
  type: 'api_token',
  apiToken: 'your_webflow_api_token',
  siteId: 'your_site_id',
  collectionId: 'optional_blog_collection_id'
};
```

OAuth2 (Recommended for apps)

```
const credentials: WebflowCredentials = {
  type: 'oauth2',
  accessToken: 'your_oauth_access_token',
  siteId: 'your_site_id'
};
```

Webflow-Specific Features

Rich Text Content

```
const blogPost: BlogPost = {
  title: "Design System Fundamentals",
  content: `
    <div class="hero-section">
      <h1>Building Scalable Design Systems</h1>
      <p class="lead">Create consistent, maintainable design experiences</p>
    </div>

    <div class="content-grid">
      <div class="feature-card">
        <h3>Typography</h3>
        <p>Establish clear typographic hierarchy and spacing rules.</p>
      </div>
    </div>
  `,
  format: 'html' // Webflow supports rich HTML
};
```

Custom Fields Integration

```
// Get available fields in the collection
const customFields = await webflowAdapter.getCustomFields();

// Publish with custom field data
await webflowAdapter.publish(formattedContent, {
  status: 'published',
  customFields: {
    'featured': true,
    'category': 'design-theory',
    'difficulty-level': 'intermediate',
    'reading-time': '8 min'
  }
});
```

Asset Upload

```
// Images are automatically processed and uploaded to Webflow
const blogPost: BlogPost = {
  title: "Visual Design Trends",
  images: [
    {
      filename: 'hero-image.jpg',
      url: 'https://i.pinimg.com/736x/03/fc/2a/03fc2a27982625f2ae3490a3bff461a8.jpg',
      altText: 'Design trends visualization',
      caption: 'Modern design approaches'
    }
  ]
};
```

Configuration Options

```
const webflowConfig: WebflowConfig = {
  siteId: 'your_webflow_site_id',
  apiVersion: 'v1', // Webflow API version
  maxRetries: 3, // Request retry attempts
  retryDelay: 1000, // Delay between retries (ms)
  timeout: 30000 // Request timeout (ms)
};
```

Multi-Platform Publishing

Publishing to Multiple Platforms

```
const multiPublisher = new MultiPlatformPublisher();

// Add all platform adapters
await multiPublisher.addPlatform(shopifyAdapter, shopifyCredentials);
await multiPublisher.addPlatform(webflowAdapter, webflowCredentials);
await multiPublisher.addPlatform(wordpressAdapter, wordpressCredentials);

// Publish to selected platforms
const result = await multiPublisher.publishToSelected(
  blogPost,
  ['shopify', 'webflow', 'wordpress'],
  {
    adaptContentPerPlatform: true,
    platformSpecificOptions: {
      shopify: {
        tags: ['product-guide', 'ecommerce'],
        status: 'published'
      },
      webflow: {
        customFields: {
          featured: true,
          category: 'business'
        }
      }
    }
  }
);
```

Platform-Specific Optimization

```
const publishOptions: MultiPlatformPublishOptions = {
  adaptContentPerPlatform: true,
  publishOrder: ['webflow', 'shopify', 'wordpress'],

  adaptationRules: {
    shopify: {
      targetWordCount: 800, // Optimize for e-commerce attention spans
      includeImages: true, // Include product images
      preserveFormatting: false // Allow HTML to Markdown conversion
    },
    webflow: {
      preserveFormatting: true, // Maintain rich text formatting
      includeImages: true, // Upload images to Webflow
      adaptForSEO: true // Optimize for Webflow SEO fields
    }
  }
};
```

Analytics and Monitoring

Platform Health Monitoring

```
// Check health of all platforms
const healthReport = await multiPublisher.checkPlatformHealth();

console.log('Overall health:', healthReport.overall);
healthReport.platforms.forEach((platform, health) => {
  console.log(`${platform}: ${health.status} (${health.responseTime}ms)`);
});
```

Aggregated Analytics

```
// Get cross-platform analytics
const analytics = await multiPublisher.getAggregatedAnalytics(
  ['shopify', 'webflow', 'wordpress'],
  {
    startDate: new Date('2024-01-01'),
    endDate: new Date()
  }
);

console.log('Total views:', analytics.totalViews);
console.log('Total engagements:', analytics.totalEngagements);
console.log('Best performing platform:', analytics.topPerformingPlatform);
```

Platform Comparison

```
const comparative = await multiPublisher.getComparativeAnalytics([
  'shopify', 'webflow'],
  { startDate: monthAgo, endDate: now }
);

comparative.comparisons.forEach(comparison => {
  console.log(`${comparison.metric}: ${comparison.winner} leads by ${comparison.difference}%`);
});
```

Content Optimization

SEO Best Practices

Shopify SEO

```
const shopifyOptimizedPost = {
  title: "Best Summer Products 2024 - Free Shipping",
  seoTitle: "Top Summer Products 2024 | Free Shipping | MyStore",
  seoDescription: "Discover the best summer products with free shipping. Shop our curated collection of trending items.",
  tags: ['summer', 'trending', 'free-shipping'],
  content: `
    // Include product recommendations
    [product:summer-essentials-bundle]

    // Add clear call-to-actions
    Shop now and get free shipping on orders over $50!
  `;
};
```

Webflow SEO

```
const webflowOptimizedPost = {
  title: "Complete Guide to Responsive Design",
  seoTitle: "Responsive Web Design Guide 2024 | Best Practices",
  seoDescription: "Learn responsive design principles, techniques, and best practices for creating mobile-friendly websites.",
  content: `
    <div class="seo-optimized-content">
      <h1>Responsive Design Mastery</h1>
      // Properly structured HTML with semantic elements
      // Optimized for Webflow's design capabilities
    </div>
  `;
};
```

Content Adaptation Rules

```
const adaptationConfig = {
  shopify: {
    // Optimize for e-commerce
    focusOnProducts: true,
    includeCallToActions: true,
    optimizeForConversion: true
  },
  webflow: {
    // Optimize for design showcase
    preserveVisualHierarchy: true,
    enhanceTypography: true,
    maintainBrandConsistency: true
  }
};
```

Error Handling and Troubleshooting

Common Authentication Issues

Shopify

```
// Validate credentials
const validation = CredentialValidator.validateShopifyCredentials(credentials);
if (!validation.isValid) {
  console.error('Shopify auth errors:', validation.errors);
}

// Check store URL format
if (!credentials.storeUrl.includes('.myshopify.com')) {
  console.warn('Store URL should be in format: store.myshopify.com');
}
```

Webflow

```
// Check site permissions
const connectionResult = await webflowAdapter.validateConnection();
if (!connectionResult.isValid) {
  console.error('Webflow connection error:', connectionResult.error);
}

// Verify collection access
if (!credentials.collectionId) {
  console.info('Collection ID not specified - will auto-detect blog collection');
}
```


Rate Limit Management

```
// Check rate limits before publishing
const shopifyRateLimit = await shopifyAdapter.getRateLimit();
if (shopifyRateLimit.remaining < 5) {
  console.warn('Shopify rate limit low, waiting...');
  await new Promise(resolve => setTimeout(resolve, 2000));
}

const webflowRateLimit = await webflowAdapter.getRateLimit();
if (webflowRateLimit.remaining < 10) {
  console.warn('Webflow rate limit low, delaying requests...');
}
```

Content Validation

```
// Validate content before publishing
const validation = await adapter.validateContent(formattedContent);

if (!validation.isValid) {
  console.error('Validation errors:');
  validation.errors.forEach(error => {
    console.error(`- ${error.field}: ${error.message}`);
  });
}

if (validation.warnings.length > 0) {
  console.warn('Validation warnings:');
  validation.warnings.forEach(warning => {
    console.warn(`- ${warning.field}: ${warning.message}`);
  });
}
```

Best Practices

1. Authentication Management

- Use environment variables for API tokens and credentials
- Implement token refresh for OAuth2 flows
- Monitor authentication status and handle expiration gracefully
- Use the most appropriate authentication method for each platform

2. Content Strategy

- Adapt content for each platform's audience and strengths
- Use Shopify for product-focused content with clear CTAs
- Use Webflow for design-heavy content with rich visual elements
- Maintain consistent branding while optimizing for platform-specific features

3. Performance Optimization

- Monitor rate limits and implement backoff strategies
- Use bulk operations when supported by the platform
- Implement proper error handling and retry logic
- Cache platform configuration and metadata when possible

4. SEO Optimization

- Customize meta titles and descriptions for each platform
- Use platform-specific SEO features (Shopify collections, Webflow custom fields)
- Implement proper heading structure and semantic markup
- Optimize images with appropriate alt text and captions

5. Analytics and Monitoring

- Set up regular health checks for all platforms
- Monitor publishing success rates and error patterns
- Track cross-platform performance metrics
- Use comparative analytics to optimize platform strategy

Advanced Examples

See the following example files for comprehensive usage patterns:

- [examples/shopify-adapter-demo.ts](#) (./examples/shopify-adapter-demo.ts) - Complete Shopify integration examples
- [examples/webflow-adapter-demo.ts](#) (./examples/webflow-adapter-demo.ts) - Webflow CMS publishing examples
- [examples/multi-platform-integration-demo.ts](#) (./examples/multi-platform-integration-demo.ts) - Cross-platform publishing workflows

Support and Contributing

For issues, feature requests, or contributions related to the platform adapters:

1. Check the existing issues and documentation
2. Test with the provided example code
3. Verify your platform credentials and permissions
4. Submit detailed bug reports with reproduction steps

The platform adapters are actively maintained and new platforms can be added following the established `BasePlatformAdapter` pattern.