

readme.md for @push.rocks/smartsitemap

“`push` A comprehensive TypeScript sitemap library with a chainable builder API — supporting standard, news, image, video, and hreflang sitemaps with auto-splitting, streaming, validation, and RSS feed integration.

Issue Reporting and Security

For reporting bugs, issues, or security vulnerabilities, please visit community.foss.global/. This is the central community hub for all issue reporting. Developers who sign and comply with our contribution agreement and go through identification can also get a code.foss.global/ account to submit Pull Requests directly.

Install

```
pnpm install @push.rocks/smartsitemap
```

📦 Features

- `push` **Chainable Builder API** — Fluent, composable API where every method returns `this`
- `push` **News Sitemaps** — Google News-compatible with proper namespace handling
- `push` **Image Sitemaps** — Full `image:image` extension support
- `push` **Video Sitemaps** — Full `video:video` extension with all fields
- `push` **hreflang / i18n** — `xhtml:link` alternate language annotations
- `push` **Sitemap Index** — Automatic splitting at 50K URLs with index generation
- `push` **Streaming** — Node.js Readable stream for million-URL sitemaps
- `push` **Validation** — URL validation, size limits, spec compliance checks
- `push` **Statistics** — URL counts, image/video/news counts, size estimates

- **☐ RSS/Atom Feed Import** — Convert feeds to sitemaps (unique feature!)
- **☐ YAML Config** — Declarative sitemap definition from YAML
- **☐ Multi-Format Output** — XML, TXT, JSON, gzipped buffer
- **☐ XSL Stylesheets** — Browser-viewable sitemaps
- **☐ Bidirectional Parsing** — Parse existing sitemaps back into structured data
- **☐ Full TypeScript** — Complete type safety with exported interfaces

Quick Start

```
import { SmartSitemap } from '@push.rocks/smartsitemap';

// 3 lines to a valid sitemap ☐☐
const xml = SmartSitemap.create()
  .addUrl('https://example.com/')
  .addUrl('https://example.com/about')
  .addUrl('https://example.com/blog')
  .toXml();
```

Output:

```
<?xml version="1.0" encoding="UTF-8"?>
<urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9">
  <url>
    <loc>https://example.com/</loc>
  </url>
  <url>
    <loc>https://example.com/about</loc>
  </url>
  <url>
    <loc>https://example.com/blog</loc>
  </url>
</urlset>
```

Usage

☐☐ Standard Sitemap with Full Control

```
import { SmartSitemap } from '@push.rocks/smartsitemap';

const xml = SmartSitemap.create({ baseUrl: 'https://example.com' })
  .setDefaultChangeFreq('weekly')
  .setDefaultPriority(0.5)
  .setXslUrl('/sitemap.xml')
  .add({
    loc: 'https://example.com/',
    changefreq: 'daily',
    priority: 1.0,
    lastmod: new Date(),
  })
  .add({
    loc: 'https://example.com/products',
    changefreq: 'daily',
    priority: 0.9,
    images: [
      { loc: 'https://example.com/img/product1.jpg', title: 'Product 1' },
    ],
  })
  .add({
    loc: 'https://example.com/blog/post-1',
    lastmod: '2025-01-15',
    alternates: [
      { hreflang: 'de', href: 'https://example.com/de/blog/post-1' },
      { hreflang: 'fr', href: 'https://example.com/fr/blog/post-1' },
    ],
  })
  .toXml();
```

☐☐ Builder from a URL Array

```
const builder = SmartSitemap.fromUrls([
  'https://example.com/',
  'https://example.com/about',
  'https://example.com/contact',
]);
```

```
const xml = builder
  .setDefaultChangeFreq('monthly')
  .toXml();
```

📄 News Sitemap

```
const xml = SmartSitemap.createNews({
  publicationName: 'The Daily Tech',
  publicationLanguage: 'en',
})
.addNewsUrl(
  'https://example.com/news/breaking-story',
  'Breaking: TypeScript 6.0 Released!',
  new Date(),
  ['typescript', 'programming'],
)
.addNewsUrl(
  'https://example.com/news/another-story',
  'Node.js Gets Even Faster',
  new Date(),
)
.toXml();
```

📄 News Sitemap from RSS Feed

This is smartsitemap's killer feature — no other sitemap library does this:

```
// From a feed URL
const builder = SmartSitemap.createNews({
  publicationName: 'The Daily Tech',
  publicationLanguage: 'en',
});
await builder.importFromFeedUrl('https://thedailytech.com/rss/');
const xml = builder.toXml();

// Or as a one-liner with the static factory
const feedBuilder = await SmartSitemap.fromFeedUrl('https://example.com/rss/');
const feedXml = feedBuilder.toXml();
```

📄 News Sitemap from Articles

Works seamlessly with `@tsclass/tsclass` `IArticle` objects from your CMS:

```
import type { content } from '@tsclass/tsclass';

const articles: content.IArticle[] = [/* from your CMS or database */];

const xml = SmartSitemap.fromArticles(articles, {
  publicationName: 'My Publication',
  publicationLanguage: 'en',
}).toXml();
```

📄 Image Sitemap

```
const xml = SmartSitemap.create()
  .add({
    loc: 'https://example.com/gallery',
    images: [
      { loc: 'https://example.com/img/photo1.jpg', title: 'Sunset' },
      { loc: 'https://example.com/img/photo2.jpg', caption: 'Mountain view' },
    ],
  })
  .toXml();
```

📄 Video Sitemap

```
const xml = SmartSitemap.create()
  .add({
    loc: 'https://example.com/videos/tutorial',
    videos: [
      {
        thumbnailLoc: 'https://example.com/thumb.jpg',
        title: 'Getting Started with TypeScript',
        description: 'A comprehensive guide to TypeScript for beginners.',
        contentLoc: 'https://example.com/video.mp4',
        duration: 600,
      }
    ]
  })
  .toXml();
```

```
        rating: 4.8,
        publicationDate: new Date(),
        tags: ['typescript', 'tutorial', 'programming'],
    },
],
})
.toXml();
```

☐☐ hreflang / Internationalization

```
const xml = SmartSitemap.create()
    .add({
        loc: 'https://example.com/page',
        alternates: [
            { hreflang: 'en', href: 'https://example.com/page' },
            { hreflang: 'de', href: 'https://example.com/de/page' },
            { hreflang: 'fr', href: 'https://example.com/fr/page' },
            { hreflang: 'x-default', href: 'https://example.com/page' },
        ],
    })
    .toXml();
```

☐☐ Automatic Sitemap Index Splitting

When you exceed 50K URLs, smartsitemap automatically splits into a sitemap index:

```
const builder = SmartSitemap.create({
    baseUrl: 'https://example.com',
    maxUrlsPerSitemap: 45000, // default is 50000
});

// Add hundreds of thousands of URLs
for (const page of allPages) {
    builder.addUrl(page.url, page.lastModified);
}

const set = builder.toSitemapSet();
// set.needsIndex === true
```

```

// set.indexXml → '<?xml ...><sitemapindex>...'
// set.sitemaps → [
//   { filename: 'sitemap-1.xml', xml: '...' },
//   { filename: 'sitemap-2.xml', xml: '...' },
//   { filename: 'sitemap-3.xml', xml: '...' },
// ]

// Or build an index manually
const index = SmartSitemap.createIndex()
  .addSitemap('https://example.com/sitemap-blog.xml')
  .addSitemap('https://example.com/sitemap-products.xml', new Date())
  .toXml();

```

☐☐ Streaming for Large Sitemaps

For sitemaps with millions of URLs that can't fit in memory:

```

import { createWriteStream } from 'fs';
import { createGzip } from 'zlib';
import { SitemapStream } from '@push.rocks/smartsitemap';

const stream = new SitemapStream();
const output = createWriteStream('/var/www/sitemap.xml.gz');
stream.pipe(createGzip()).pipe(output);

// Stream URLs from a database cursor
for await (const page of databaseCursor()) {
  stream.pushUrl({
    loc: page.url,
    lastmod: page.updatedAt,
    changefreq: 'weekly',
  });
}

stream.finish();

```

☐☐ Merge, Dedupe, Filter & Sort

Combine multiple sitemap sources with powerful collection operations:

```
const blogSitemap = SmartSitemap.create()
  .setDefaultChangeFreq('weekly')
  .addFromArray(blogUrls);

const productSitemap = SmartSitemap.create()
  .setDefaultChangeFreq('daily')
  .addFromArray(productUrls);

const xml = SmartSitemap.create()
  .merge(blogSitemap)
  .merge(productSitemap)
  .dedupe()
  .filter(url => !url.loc.includes('/deprecated/'))
  .sort((a, b) => a.loc.localeCompare(b.loc))
  .toXml();
```

☐☐ YAML Configuration

Define sitemaps declaratively:

```
const yaml = `
baseUrl: https://example.com
defaults:
  priority: 0.5
urls:
  daily:
    - /
    - /blog
  weekly:
    - /docs
    - /tutorials
  monthly:
    - /about
    - /contact
  yearly:
    - /privacy
    - /terms
```

```
`;  
  
const builder = await SmartSitemap.fromYaml(yaml);  
const xml = builder.toXml();
```

□ Validation

Catch errors before they reach search engines:

```
const result = SmartSitemap.create()  
  .addUrl('not-a-valid-url')  
  .add({ loc: 'https://example.com/', priority: 1.5 }) // out of range  
  .validate();  
  
console.log(result.valid); // false  
console.log(result.errors);  
// [  
//   { field: 'loc', message: 'Invalid URL: "not-a-valid-url"', url: 'not-a-valid-url' },  
//   { field: 'priority', message: 'Priority must be between 0.0 and 1.0', url:  
//     'https://example.com/' },  
// ]
```

□□ Statistics

Get insight into your sitemap:

```
const stats = SmartSitemap.create()  
  .addUrl('https://example.com/')  
  .add({ loc: 'https://example.com/gallery', images: [{ loc: '/img/1.jpg' }] })  
  .stats();  
  
console.log(stats);  
// {  
//   urlCount: 2,  
//   imageCount: 1,  
//   videoCount: 0,  
//   newsCount: 0,  
//   alternateCount: 0,
```

```
// estimatedSizeBytes: 750,  
// needsIndex: false,  
// }
```

☐☐ Multi-Format Output

```
const builder = SmartSitemap.create()  
  .addUrl('https://example.com/')  
  .addUrl('https://example.com/about');  
  
// XML (default)  
const xml = builder.toXml();  
  
// Plain text (one URL per line)  
const txt = builder.toTxt();  
// "https://example.com/\nhttps://example.com/about"  
  
// JSON  
const json = builder.toJson();  
  
// Gzipped XML buffer (for serving compressed)  
const gzipped = await builder.toGzipBuffer();
```

☐☐ Parse Existing Sitemaps

Read and parse sitemaps back into structured data:

```
// From URL  
const parsed = await SmartSitemap.parseUrl('https://example.com/sitemap.xml');  
console.log(parsed.type); // 'urlset' or 'sitemapindex'  
console.log(parsed.urls); // ISitemapUrl[]  
  
// From XML string  
const result = await SmartSitemap.parse(sitemapXmlString);  
  
// Parse and get a pre-populated builder for modification  
const builder = await SitemapParser.toBuilder(existingSitemapXml);
```

```
builder
  .addUrl('https://example.com/new-page')
  .filter(url => !url.loc.includes('/old/'))
  .toXml();

// Detect type without full parsing
SitemapParser.detectType('<urlset ...>'); // 'urlset'
SitemapParser.detectType('<sitemapindex ...>'); // 'sitemapindex'
```

☐☐ Real-World Integration Examples

Express.js / Hono / Fastify Server

```
import { SmartSitemap } from '@push.rocks/smartsitemap';

// Serve dynamic sitemap
app.get('/sitemap.xml', async (req, res) => {
  const xml = SmartSitemap.create()
    .setDefaultChangeFreq('weekly')
    .addFromArray(await getUrlsFromDatabase())
    .toXml();

  res.header('Content-Type', 'application/xml');
  res.send(xml);
});

// Serve news sitemap from RSS
app.get('/news-sitemap.xml', async (req, res) => {
  const builder = SmartSitemap.createNews({ publicationName: 'My Site' });
  await builder.importFromFeedUrl('https://mysite.com/rss/');

  res.header('Content-Type', 'application/xml');
  res.send(builder.toXml());
});
```

```
// Auto-split with sitemap index
app.get('/sitemap-index.xml', async (req, res) => {
  const builder = SmartSitemap.create({ baseUrl: 'https://mysite.com' });
  builder.addFromArray(await getAllUrls()); // 200K+ URLs

  const set = builder.toSitemapSet();
  res.header('Content-Type', 'application/xml');
  res.send(set.indexXml ?? set.sitemaps[0].xml);
});
```

Static Site Generator

```
import { SmartSitemap } from '@push.rocks/smartsitemap';
import { writeFileSync } from 'fs';

const xml = SmartSitemap.create()
  .setDefaultChangeFreq('weekly')
  .add({ loc: 'https://mysite.com/', changefreq: 'daily', priority: 1.0 })
  .add({ loc: 'https://mysite.com/about', changefreq: 'monthly' })
  .addFromArray(blogPostUrls)
  .dedupe()
  .toXml();

writeFileSync('./public/sitemap.xml', xml);
```

API Reference

SmartSitemap (Static Factories)

Method	Returns	Description
<code>SmartSitemap.create(options?)</code>	<code>UrlsetBuilder</code>	Create a standard sitemap builder
<code>SmartSitemap.createNews(options)</code>	<code>NewsSitemapBuilder</code>	Create a news sitemap builder
<code>SmartSitemap.createIndex(options?)</code>	<code>SitemapIndexBuilder</code>	Create a sitemap index builder

Method	Returns	Description
<code>SmartSitemap.fromUrls(urls, options?)</code>	<code>UrlsetBuilder</code>	Builder from URL string array
<code>SmartSitemap.fromYaml(yaml)</code>	<code>Promise<UrlsetBuilder></code>	Builder from YAML config
<code>SmartSitemap.fromFeedUrl(url, options?)</code>	<code>Promise<UrlsetBuilder></code>	Builder from RSS/Atom feed URL
<code>SmartSitemap.fromFeedString(xml, options?)</code>	<code>Promise<UrlsetBuilder></code>	Builder from RSS/Atom feed string
<code>SmartSitemap.fromArticles(articles, options)</code>	<code>NewsSitemapBuilder</code>	Builder from IArticle array
<code>SmartSitemap.parse(xml)</code>	<code>Promise<IParsedSitemap></code>	Parse sitemap XML string
<code>SmartSitemap.parseUrl(url)</code>	<code>Promise<IParsedSitemap></code>	Fetch and parse sitemap
<code>SmartSitemap.validate(xml)</code>	<code>Promise<IValidationResult></code>	Validate sitemap XML

UrlsetBuilder (Chainable)

Method	Returns	Description
<code>.add(url)</code>	<code>this</code>	Add a URL with full <code>ISitemapUrl</code> options
<code>.addUrl(loc, lastmod?)</code>	<code>this</code>	Add by URL string
<code>.addUrls(urls)</code>	<code>this</code>	Add multiple <code>ISitemapUrl</code> objects
<code>.addFromArray(locs)</code>	<code>this</code>	Add from plain string array
<code>.merge(other)</code>	<code>this</code>	Merge in another builder's URLs
<code>.filter(predicate)</code>	<code>this</code>	Filter URLs in-place
<code>.map(transform)</code>	<code>this</code>	Transform URLs in-place
<code>.sort(compareFn?)</code>	<code>this</code>	Sort URLs (default: alphabetical)
<code>.dedupe()</code>	<code>this</code>	Remove duplicate URLs by loc
<code>.setDefaultChangeFreq(freq)</code>	<code>this</code>	Set default changefreq
<code>.setDefaultPriority(priority)</code>	<code>this</code>	Set default priority (0.0-1.0)
<code>.setXslUrl(url)</code>	<code>this</code>	Set XSL stylesheet URL
<code>.importFromFeedUrl(url, options?)</code>	<code>Promise<this></code>	Import from RSS/Atom feed URL
<code>.importFromFeedString(xml, options?)</code>	<code>Promise<this></code>	Import from RSS/Atom feed string
<code>.importFromYaml(yaml)</code>	<code>Promise<this></code>	Import from YAML config
<code>.importFromArticles(articles)</code>	<code>this</code>	Import from IArticle array
<code>.toXml()</code>	<code>string</code>	Export as sitemap XML

Method	Returns	Description
<code>.toTxt()</code>	<code>string</code>	Export as plain text
<code>.toJson()</code>	<code>string</code>	Export as JSON
<code>.toGzipBuffer()</code>	<code>Promise<Buffer></code>	Export as gzipped XML
<code>.toSitemapSet()</code>	<code>ISitemapSet</code>	Auto-split with index
<code>.toStream()</code>	<code>SitemapStream</code>	Export as Node.js Readable stream
<code>.validate()</code>	<code>IValidationResult</code>	Validate against spec
<code>.stats()</code>	<code>ISitemapStats</code>	Get statistics
<code>.getUrls()</code>	<code>ISitemapUrl[]</code>	Get the raw URL array
<code>.count</code>	<code>number</code>	Get URL count

NewsSitemapBuilder (extends UrlsetBuilder)

Method	Returns	Description
<code>.addNewsUrl(loc, title, date, keywords?)</code>	<code>this</code>	Add a news article with publication info

SitemapIndexBuilder

Method	Returns	Description
<code>.add(entry)</code>	<code>this</code>	Add a sitemap index entry
<code>.addSitemap(loc, lastmod?)</code>	<code>this</code>	Add by URL string
<code>.addSitemaps(entries)</code>	<code>this</code>	Add multiple entries
<code>SitemapIndexBuilder.fromBuilder(builder, baseUrl)</code>	<code>{index, sitemaps[]}</code>	Auto-split a builder
<code>.toXml()</code>	<code>string</code>	Export as sitemap index XML
<code>.count</code>	<code>number</code>	Get entry count

SitemapStream

Method	Description
--------	-------------

<code>.pushUrl(url)</code>	Push a URL entry to the stream
<code>.finish()</code>	Signal end of stream, writes closing tag
<code>.count</code>	Number of URLs written

Key Types

```
interface ISitemapUrl {
    loc: string; // Required – absolute URL
    lastmod?: Date | string | number; // Date, ISO string, or timestamp (ms)
    changefreq?: TChangeFreq; //
    'always' | 'hourly' | 'daily' | 'weekly' | 'monthly' | 'yearly' | 'never'
    priority?: number; // 0.0 to 1.0
    images?: ISitemapImage[]; // Image extension
    videos?: ISitemapVideo[]; // Video extension
    news?: ISitemapNews; // News extension
    alternates?: ISitemapAlternate[]; // hreflang alternates
}

interface ISitemapOptions {
    baseUrl?: string;
    xslUrl?: string;
    defaultChangeFreq?: TChangeFreq;
    defaultPriority?: number;
    prettyPrint?: boolean; // default: true
    maxUrlsPerSitemap?: number; // default: 50000
    gzip?: boolean;
    validate?: boolean; // default: true
}

interface INewsSitemapOptions extends ISitemapOptions {
    publicationName: string; // Required
    publicationLanguage?: string; // default: 'en'
}
```

License and Legal Information

This repository contains open-source code licensed under the MIT License. A copy of the license can be found in the [LICENSE](#) file.

Please note: The MIT License does not grant permission to use the trade names, trademarks, service marks, or product names of the project, except as required for reasonable and customary use in describing the origin of the work and reproducing the content of the NOTICE file.

Trademarks

This project is owned and maintained by Task Venture Capital GmbH. The names and logos associated with Task Venture Capital GmbH and any related products or services are trademarks of Task Venture Capital GmbH or third parties, and are not included within the scope of the MIT license granted herein.

Use of these trademarks must comply with Task Venture Capital GmbH's Trademark Guidelines or the guidelines of the respective third-party owners, and any usage must be approved in writing. Third-party trademarks used herein are the property of their respective owners and used only in a descriptive manner, e.g. for an implementation of an API or similar.

Company Information

Task Venture Capital GmbH Registered at District Court Bremen HRB 35230 HB, Germany

For any legal inquiries or further information, please contact us via email at hello@task.vc.

By using this repository, you acknowledge that you have read this section, agree to comply with its terms, and understand that the licensing of the code does not imply endorsement by Task Venture Capital GmbH of any derivative works.

Revision #3

Created 2026-03-28 11:12:33 UTC by foss.global Team

Updated 2026-03-28 12:19:18 UTC by foss.global Team