

@push.rocks/smartxml

A package for creating and parsing XML formatted files.

- [readme.md for @push.rocks/smartxml](#)
- [changelog.md for @push.rocks/smartxml](#)

readme.md for @push.rocks/smartxml

a package for creating and parsing XML formatted files

Install

To start using @push.rocks/smartxml in your project, you'll first need to install it. This package is distributed via npm, so you can install it by running the following command in your project's root directory:

```
npm install @push.rocks/smartxml --save
```

Usage

Getting Started

First, ensure that your project is set up to use TypeScript and ESM syntax. Then, you can import `SmartXml` from @push.rocks/smartxml as follows:

```
import { SmartXml } from '@push.rocks/smartxml';
```

`SmartXml` provides two main functionalities: parsing XML formatted strings into JavaScript objects and vice versa, creating XML formatted strings from JavaScript objects.

Parsing XML String to Object

To parse an XML string to a JavaScript object, you can use the `parseXmlToObject` method. This method takes an XML string as its argument and returns a JavaScript object.

Here's a simple example:

```
const xmlString = `  
<note>  
  <to>Tove</to>  
  <from>Jani</from>  
  <heading>Reminder</heading>  
  <body>Don't forget me this weekend!</body>  
</note>  
`;  
  
const smartXml = new SmartXml();  
const resultObject = smartXml.parseXmlToObject(xmlString);  
  
console.log(resultObject);
```

Creating XML String from Object

Conversely, if you have a JavaScript object that you want to convert into an XML formatted string, you can use the `createXmlFromObject` method. This method accepts a JavaScript object and returns a string containing the XML representation of the object.

Here's an example:

```
const noteObject = {  
  note: {  
    to: 'Tove',  
    from: 'Jani',  
    heading: 'Reminder',  
    body: "Don't forget me this weekend!",  
  },  
};  
  
const smartXml = new SmartXml();  
const xmlString = smartXml.createXmlFromObject(noteObject);  
  
console.log(xmlString);
```

Advanced Usage

The `SmartXml` class is built on top of the `fast-xml-parser` library, which provides a lot of flexibility for both parsing and creation. For example, when creating XML from objects, you can control attributes, formatting, and more.

Let's look at a more complex example that includes XML attributes:

```
const complexObject = {
  note: {
    '@_id': '12345',
    '@_priority': 'high',
    to: 'Tove',
    from: { '@_domain': 'personal', '#text': 'Jani' },
    heading: {
      '@_style': 'bold',
      '#text': 'Reminder',
    },
    body: 'This is a special note for the weekend.',
  },
};

const smartXml = new SmartXml();
const complexXmlString = smartXml.createXmlFromObject(complexObject, {
  format: true,
});

console.log(complexXmlString);
```

This example demonstrates how to add attributes (using `"@_attributeName": "value"`) and mixed content (using `"#text": "text value"`) to your objects before converting them to XML.

Conclusion

The `@push.rocks/smartxml` package offers a straightforward and powerful means to work with XML in your JavaScript or TypeScript projects. By integrating it, you can easily convert between XML strings and JavaScript objects, allowing for efficient data processing and transmission in scenarios where XML is the preferred format. Whether you're building web services, APIs, or need to handle configuration files in XML, `SmartXml` offers the necessary tools to get the job done with minimal hassle.

License and Legal Information

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

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 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, and any usage must be approved in writing by Task Venture Capital GmbH.

Company Information

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

For any legal inquiries or if you require 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.

changelog.md for @push.rocks/smartxml

2025-11-19 - 2.0.0 - BREAKING CHANGE(core)

Implement custom XmlBuilder, remove xmlbuilder2, upgrade fast-xml-parser, update SmartXml API, tests and CI

- Add a new chainable XmlBuilder implementation (create(), ele(), txt(), att(), up(), com(), end(), toString()) and export it from the package
- Remove xmlbuilder2 dependency and its export from smartxml.plugins.ts
- Upgrade fast-xml-parser dependency to ^5.3.2 and use it for parsing and building XML
- Update SmartXml to use XmlBuilder.create() as the create factory and export XmlBuilder for direct use
- Extend and update tests to cover the new XmlBuilder chainable API and XML serialization behavior
- Update CI workflows: switch container IMAGE registry, adjust NPMCI_COMPUTED_REPOURL and change pnpm install target for npmci package name
- Bump several devDependencies, add pnpm packageManager/pnpm metadata, and apply small tsconfig/.gitignore cleanups

2025-11-19 - 1.2.0 - feat(core)

Implement custom XmlBuilder with chainable API and upgrade dependencies

- **BREAKING:** Removed xmlbuilder2 dependency (replaced with custom implementation)
- Upgraded fast-xml-parser from 4.5.1 to 5.3.2
- Implemented custom XmlBuilder class with fluent/chainable API
 - `.ele(name, attrs)` - Add elements
 - `.txt(content)` - Add text content
 - `.att(name, value)` - Add attributes
 - `.up()` - Navigate to parent

- `.end(options)` - Serialize to XML
- Added comprehensive tests for XmlBuilder functionality
- Maintains API compatibility: `smartxml.create()` continues to work
- Browser-compatible implementation (no Node.js dependencies)
- Export XmlBuilder class for direct usage

2024-12-30 - 1.1.1 - fix(dependencies)

Added missing xmlbuilder2 dependency and relevant exports

- Added xmlbuilder2 to dependencies in package.json
- Export xmlbuilder2 functionality from smartxml.plugins.ts and ts/index.ts

2024-12-30 - 1.1.0 - feat(core)

Upgrade dependencies and enhance XML parsing and building

- Updated fast-xml-parser dependency to version ^4.5.1
- Enhanced XML parsing to preserve order and handle attributes
- Refactored createXmlFromObject method with a new XMLBuilder implementation
- Improved test coverage for XML string creation and parsing

2024-05-29 - 1.0.8 - Various Updates

Minor updates and improvements to configuration and documentation.

- Updated project description
- Modified `tsconfig` file
- Updated `npmextra.json`: githost

2023-10-20 - 1.0.6 to 1.0.8 - Core Fixes and Updates

Maintenance and core updates with improvements in project configuration.

- Core functionality fixes and updates
- Configuration updates
- Preparation for versioning to 1.0.8

2020-10-24 - 1.0.1 to 1.0.6 - Core Fixes

Multiple fixes and enhancements in core components.

- Series of core updates to improve stability and performance