

@push.rocks/smartrouter

A JavaScript library providing routing capabilities for web applications.

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

readme.md for @push.rocks/smartrouter @push.rocks/smartrouter

A JavaScript library providing routing capabilities for web applications with support for path parameters, query parameters, and programmatic navigation

Install

To install `@push.rocks/smartrouter`, run the following command in your project directory:

```
pnpm install @push.rocks/smartrouter --save
```

This will add `@push.rocks/smartrouter` to your project's dependencies and enable you to use it within your application.

Usage

`@push.rocks/smartrouter` provides a versatile routing solution for websites, leveraging modern Web APIs to manipulate browser history and handle URL paths intelligently. Below are examples demonstrating how to use `@push.rocks/smartrouter` effectively in a TypeScript project, taking advantage of ESM syntax.

Basic Setup

First, ensure you've installed the package as described in the Install section above. Next, import `SmartRouter` from `@push.rocks/smartrouter` in your application's entry point or any module where routing is required.

```
import { SmartRouter } from '@push.rocks/smartrouter';
```

Initialize the Router

Create an instance of `SmartRouter` and optionally provide configuration options. If your application requires debugging information, `debug` can be set to `true`.

```
const router = new SmartRouter({  
  debug: true, // Enables debugging. Optional and false by default.  
});
```

Define Routes

Define your application routes using the `on` method, which takes a URL pattern and a handler function. The handler function will be called when the application navigates to a URL that matches the pattern.

```
router.on('/home', async (routeInfo) => {  
  console.log('Home route accessed', routeInfo);  
  // Handle the home route  
  // You can load a page component, change document title, etc.  
});  
  
router.on('/about', async (routeInfo) => {  
  console.log('About route accessed', routeInfo);  
  // Handle the about route  
});
```

Path Parameters

`@push.rocks/smartrouter` supports dynamic path parameters. Define path parameters within your route strings using the `:` prefix, and access their values from the `routeInfo.params` object in your handler function.

```
router.on('/user/:userId', async (routeInfo) => {  
  console.log(`User Profile for ID: ${routeInfo.params.userId}`, routeInfo);  
  // Load and display user profile based on userId  
});
```

Query Parameters

Query parameters can be accessed through the `routeInfo.queryParams` object, making it easy to handle complex routing scenarios with optional parameters.

```
router.on('/search', async (routeInfo) => {
  console.log('Search Query:', routeInfo.queryParams.query);
  // Perform a search operation using the provided query parameter
});
```

Programmatic Navigation

Navigate programmatically using the `pushUrl` method. This method allows you to change the URL without reloading the page, and optionally pass state information.

```
// Navigate to the about page
router.pushUrl('/about');

// Navigate to a user profile with URL parameters
router.pushUrl('/user/12345');

// Navigate with state data
router.pushUrl('/dashboard', { previousPage: 'home' });
```

Managing Query Parameters

`@push.rocks/smarterouter` provides methods for managing URL query parameters, enabling dynamic URL manipulation for filter settings, pagination, and other use cases.

```
// Set a query parameter (replace by default)
router.queryParams.setQueryParam('key', 'value');

// Set a query parameter with push (adds to history)
router.queryParams.setQueryParam('key', 'value', 'push');

// Get a query parameter
const value = router.queryParams.getQueryParam('key');
```

```
// Get all query parameters as an object
const allParams = router.queryParams.getAllAsObject();

// Delete a query parameter
router.queryParams.deleteQueryParam('key');

// Delete with push to history
router.queryParams.deleteQueryParam('key', 'push');
```

Sub-Routers

Create sub-routers with a specific base path for modular routing:

```
// Create a sub-router for admin routes
const adminRouter = router.createSubRouter('/admin');

// Routes will be prefixed with /admin
adminRouter.on('/dashboard', async (routeInfo) => {
  // This handles /admin/dashboard
});

adminRouter.on('/users', async (routeInfo) => {
  // This handles /admin/users
});
```

Route Management

The `on` method returns a function that can be used to remove the route:

```
// Add a route
const removeRoute = router.on('/temporary', async (routeInfo) => {
  console.log('Temporary route accessed');
});

// Later, remove the route
removeRoute();
```

Cleanup

When you're done with a router instance, clean it up properly:

```
// Destroy the router, removing all event listeners and routes
router.destroy();
```

This module enables complex routing scenarios, simplifying the handling of navigational logic in modern web applications. By leveraging `@push.rocks/smartrouter`, developers can implement detailed routing mechanisms, manipulate browser history thoughtfully, and maintain cleaner URL structures, enhancing the user experience and making web apps more accessible.

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/smartrouter

2025-07-25 - 1.3.3 - fix(dependencies)

Updated dependencies and improved documentation

- Updated devDependencies: @git.zone/tsbuild, @git.zone/tsbundle, @git.zone/tstest to latest versions
- Removed @push.rocks/tapbundle dependency and updated test imports to use tapbundle from tstest
- Updated dependencies: @push.rocks/lik and @push.rocks/smartrx to latest versions
- Added packageManager field to package.json for pnpm
- Enhanced documentation with new examples for state data, query parameters, sub-routers, and cleanup
- Added verbose flag to test script

2024-10-06 - 1.3.2 - fix(core)

Fix TypeScript type definition for route match function

- Updated the type definition for the matchFunction in the SmartRouter class to include a generic parameter.

2024-10-06 - 1.3.1 - fix(dependencies)

Updated dependencies to latest versions, resolving compatibility issues and improving performance.

- Updated devDependencies to their latest versions, including tsbuild, tsbundle, tstest, tapbundle, and @types/node
- Updated dependencies to latest versions of lik, smartrx, and path-to-regexp

2024-10-06 - 1.3.0 - feat(smartrouter)

Add destroy method to SmartRouter class.

- Introduced a destroy method to the SmartRouter class for cleaning up event listeners and route references.
- Refactored popstate event listener to be removable, improving resource management and preventing memory leaks.

2024-10-06 - 1.2.1 - fix(core)

Ensure stability and performance improvements

- Improves platform compatibility for modern web applications.
- Enhances stability of the routing logic within SmartRouter class.

2024-10-06 - 1.2.0 - feat(core)

Add support for base paths and sub-routers.

- Added basePath feature to SmartRouter for handling base paths.
- Introduced createSubRouter method to create a sub-router with a specific prefix.

2024-10-06 - 1.1.1 - fix(core)

Remove SelectionDimension functionality

- Removed SelectionDimension class and references
- Deleted smartrouter.classes.selectiondimension.ts and related imports

2024-10-06 - 1.1.0 - feat(core)

Add selection dimensions and route removal functionality

- Introduced the SelectionDimension class for managing stateful selections across routes.
- Enhanced SmartRouter with the ability to create and manage selection dimensions.
- Added route removal functionality in SmartRouter using the 'on' method.

2024-05-29 - 1.0.17 - Maintenance

Minor updates and maintenance changes to the project's configuration and metadata.

- Updated project description.
- Modified tsconfig settings.
- Updated npmextra.json for the githost setting.

2023-09-11 to 2023-04-11 - 1.0.16 - Fixes and Organizational Improvements

A series of minor bug fixes and organizational updates.

- Fixed core component issues.
- Switched to a new organizational scheme.

2023-03-18 to 2022-12-31 - 1.0.13 to 1.0.15 - Bug Fixes

Regular fixes to address bugs in the core.

- Recurring fixes in core components across versions.

2022-01-22 to 2021-09-08 - 1.0.11 to 1.0.12 - Stability Enhancements

Enhancements aimed at improving stability with core updates.

- Core stability improvements and updates.

2021-09-08 to 2020-11-30 - 1.0.5 to 1.0.10 - Regular Updates

Ongoing updates focused on the core's functionality.

- Multiple core updates to ensure optimal performance.

2020-11-30 to 2020-06-01 - 1.0.1 to 1.0.4 - Initial Fixes

Initial series of fixes and updates following project kickoff.

- Addressed initial core component issues.