

readme.md for @push.rocks/webdetector

detect different environments within the browser

Install

To install `@push.rocks/webdetector`, you can use npm or yarn. Run the following command in your terminal:

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

or if you prefer using yarn:

```
yarn add @push.rocks/webdetector
```

This package is designed to work in a browser environment.

Usage

The `@push.rocks/webdetector` module provides a powerful toolset to detect various aspects of the user's browser environment, such as online status, platform type, and PWA (Progressive Web App) installation status. It leverages modern web technologies and observables for real-time detection and updates.

Importing the Module

First, you need to import the module in your project using ESM syntax:

```
import { WebDetector } from '@push.rocks/webdetector';
```

Initializing WebDetector

To start utilizing WebDetector, you need to create an instance of it by passing configuration options. The most crucial option is the `checkOnlineUrl`, which is used to verify internet connectivity.

```
const webDetector = new WebDetector({
  checkOnlineUrl: 'https://example.com', // URL to check online status
});
```

Checking Online Status

WebDetector can check if the browser can reach the internet. This is not a mere online/offline status provided by the browser but a more reliable check that attempts to fetch a predetermined URL.

```
async function checkOnlineStatus() {
  const isOnline = await webDetector.isOnline();
  console.log(`Online status: ${isOnline}`);
}
checkOnlineStatus();
```

Observing Real-Time Online Status

WebDetector allows you to observe changes in online status in real-time, which is useful for applications that need to react to connectivity changes promptly.

```
webDetector.startPeriodicChecks();
webDetector.onlineObservable.subscribe((state) => {
  console.log(`Current state: ${state}`);
});

// Remember to stop the periodic checks when they are no longer needed
webDetector.stopPeriodicChecks();
```

Detecting the Platform

You can easily detect the user's platform (e.g., iOS, Android, Windows, etc.) using WebDetector.

```
const platform = webDetector.platform.detectPlatform();
console.log(`Platform detected: ${platform}`);
```

Checking PWA Installation Status

For Progressive Web Apps, it's vital to know if the user has installed the app. WebDetector provides a straightforward way to check this.

```
const isPwaInstalled = webDetector.pwa.isInstalled();
console.log(`Is PWA Installed: ${isPwaInstalled}`);
```

Comprehensive Observables Integration

`@push.rocks/webdetector` integrates seamlessly with observables provided by RxJS through `@pushrocks/smartrx`, making it easy to integrate with frameworks that use observables for state management.

Conclusion

`@push.rocks/webdetector` is a versatile module that helps detect various aspects of the browser environment, offering tools for checking connectivity, platform type, and PWA installation status. By utilizing observables for real-time updates, it fits well into modern web application architectures.

Remember, the accurate detection and reaction to the user's environment enhance the user experience significantly, making `@push.rocks/webdetector` a valuable addition to your web development toolkit.

For more advanced use cases, refer to the official documentation and source code available on GitHub and GitLab. Your contributions and feedback are welcome to help improve this project further.

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.

Revision #3

Created 2026-03-28 11:13:29 UTC by foss.global Team

Updated 2026-03-28 12:20:14 UTC by foss.global Team