

# readme.md for @uptime.link/detector

a detector for answering network questions locally. It does not rely on any online services.

## Availability and Links

- [npmjs.org](https://npmjs.org) (npm package)
- [gitlab.com](https://gitlab.com) (source)
- [github.com](https://github.com) (source mirror)
- [docs](#) (typedoc)

## Installation

```
npm install @uptime.link/detector
```

## Usage

Use TypeScript for best in class intellisense

## Basic Port Detection

```
import { Detector } from '@uptime.link/detector';

const detector = new Detector();

// Check if a port is active (returns detailed result)
const result = await detector.isActive('https://example.com');
```

```
console.log(result.isActive); // true

// Simple boolean check (backward compatible)
const isActive = await detector.isActiveSimple('https://example.com');
console.log(isActive); // true
```

## Service Type Detection

The detector can identify what service is running on a port:

```
// Detect service type along with port check
const result = await detector.isActive('https://github.com', {
  detectServiceType: true,
});

console.log(result.isActive); // true
console.log(result.serviceType); // 'https'

// Direct service type detection
const serviceType = await detector.detectType('ssh://github.com:22');
console.log(serviceType); // 'ssh'
```

## Supported Service Types

The detector can identify the following services:

- **HTTP** (port 80)
- **HTTPS** (port 443)
- **SSH** (port 22)
- **FTP** (port 21)
- **SMTP** (port 25)
- **POP3** (port 110)
- **IMAP** (port 143)
- **MySQL** (port 3306)
- **PostgreSQL** (port 5432)
- **MongoDB** (port 27017)
- **Redis** (port 6379)
- **Unknown** (for unidentified services)

# Advanced Examples

```
// Check localhost ports
const localResult = await detector.isActive('http://localhost:3000', {
  detectServiceType: true,
});

if (localResult.isActive) {
  console.log(`Local service detected: ${localResult.serviceType}`);
}

// Check multiple services
const urls = ['https://api.github.com', 'http://example.com', 'ssh://gitlab.com:22'];

for (const url of urls) {
  const result = await detector.isActive(url, { detectServiceType: true });
  console.log(`${url}: ${result.isActive ? result.serviceType : 'offline'}`);
}

// Handle different URL schemes with automatic port detection
const sshDetection = await detector.detectType('ssh://github.com');
// Automatically uses port 22 for SSH

const mysqlDetection = await detector.detectType('mysql://localhost');
// Automatically uses port 3306 for MySQL
```

## API Reference

### Detector

The main class for network detection.

### Methods

- **isActive(url: string, options?: IDetectorOptions): Promise<IDetectorResult>**
  - Checks if a port is active and optionally detects the service type
  - Returns detailed information about the port status and service
- **isActiveSimple(url: string): Promise<boolean>**
  - Simple boolean check for backward compatibility
  - Returns `true` if port is active, `false` otherwise

- `detectType(url: string): Promise<ServiceType>`
  - Detects the service type running on the specified URL
  - Uses protocol-specific detection methods and banner grabbing

## Interfaces

```
interface IDetectorResult {
  isActive: boolean;
  serviceType?: ServiceType;
  protocol?: 'tcp' | 'udp';
  responseTime?: number;
  tlsVersion?: string;
  serviceBanner?: string;
  error?: string;
}

interface IDetectorOptions {
  timeout?: number;
  includeNetworkDiagnostics?: boolean;
  detectServiceType?: boolean;
}

enum ServiceType {
  HTTP = 'http',
  HTTPS = 'https',
  SSH = 'ssh',
  FTP = 'ftp',
  SMTP = 'smtp',
  POP3 = 'pop3',
  IMAP = 'imap',
  MYSQL = 'mysql',
  POSTGRESQL = 'postgresql',
  MONGODB = 'mongodb',
  REDIS = 'redis',
  UNKNOWN = 'unknown',
}
```

## Features

- **Local Network Detection** - Check ports on localhost without external services
- **Remote Port Detection** - Verify if remote ports are accessible
- **Service Identification** - Detect what service is running on a port
- **Protocol Detection** - Identify HTTP/HTTPS and other protocols
- **Banner Grabbing** - Extract service information from response banners
- **TypeScript Support** - Full type definitions included
- **ESM Module** - Modern ES module support
- **No External Dependencies** - All detection happens locally

# Contribution

We are always happy for code contributions. If you are not the code contributing type that is ok. Still, maintaining Open Source repositories takes considerable time and thought. If you like the quality of what we do and our modules are useful to you we would appreciate a little monthly contribution: You can [contribute one time](#) or [contribute monthly](#). :)

For further information read the linked docs at the top of this readme.

“ MIT licensed | © [Lossless GmbH](#) | By using this npm module you agree to our [privacy policy](#)

repo-footer

---

Revision #3

Created 2026-03-28 11:15:30 UTC by foss.global Team

Updated 2026-03-28 12:21:56 UTC by foss.global Team