
















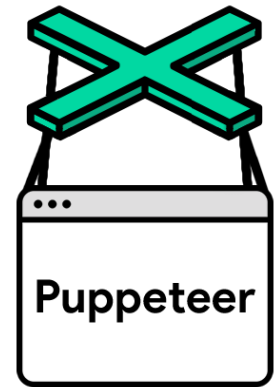
API

- Outline
- Release Notes
- Overview
- puppeteer vs puppeteer-core
- Environment Variables
- Working with Chrome
- Extensions
- interface:
- CustomQueryHandler
-  Puppeteer
-  BrowserFetcher
-  Browser
-  BrowserContext
-  Page
-  WebWorker
-  Accessibility
-  Keyboard
-  Mouse
-  Touchscreen
-  Tracing
-  FileChooser
-  Dialog
-  ConsoleMessage
-  Frame

Puppeteer

 run-checks  failing  npm  v6.0.0

[API](#) | [FAQ](#) | [Contributing](#) | [Troubleshooting](#)



Puppeteer is a Node library which provides a high-level API to control Chrome or Chromium over the [DevTools Protocol](#). Puppeteer runs [headless](#) by default, but can be configured to run full (non-headless) Chrome or Chromium.

What can I do?

Most things that you can do manually in the browser can be done using Puppeteer! Here are a few examples to get you started:

- Generate screenshots and PDFs of pages.
- Crawl a SPA (Single-Page Application) and generate pre-rendered content (i.e. "SSR" (Server-Side Rendering)).
- Automate form submission, UI testing, keyboard input, etc.
- Create an up-to-date, automated testing environment. Run your tests directly in the latest version of Chrome using the latest JavaScript and browser features.
- Capture a [timeline trace](#) of your site to help diagnose performance issues.
- Test Chrome Extensions.

Give it a spin: <https://try-puppeteer.appspot.com/>

Getting Started

Installation

To use Puppeteer in your project, run:

```
npm i puppeteer
# or "yarn add puppeteer"
```