

## puppeteer package

### Classes

Class	Description
<a href="#">Accessibility</a>	The Accessibility class provides methods for inspecting Chromium's accessibility tree. The accessibility tree is used by assistive technology such as <a href="#">screen readers</a> or <a href="#">switches</a> .
<a href="#">Browser</a>	A Browser is created when Puppeteer connects to a Chromium instance, either through <a href="#">PuppeteerNode.launch()</a> or <a href="#">Puppeteer.connect()</a> .
<a href="#">BrowserContext</a>	BrowserContexts provide a way to operate multiple independent browser sessions. When a browser is launched, it has a single BrowserContext used by default. The method <a href="#">Browser.newPage</a> creates a page in the default browser context.
<a href="#">BrowserFetcher</a>	BrowserFetcher can download and manage different versions of Chromium and Firefox.
<a href="#">CDPSession</a>	The <code>CDPSession</code> instances are used to talk raw Chrome Devtools Protocol.
<a href="#">Connection</a>	
<a href="#">ConsoleMessage</a>	ConsoleMessage objects are dispatched by page via the 'console' event.
<a href="#">Coverage</a>	The Coverage class provides methods to gathers information about parts of JavaScript and CSS that were used by the page.
<a href="#">CSSCoverage</a>	
<a href="#">CustomError</a>	
<a href="#">Dialog</a>	Dialog instances are dispatched by the <a href="#">Page</a> via the <code>dialog</code> event.
<a href="#">ElementHandle</a>	ElementHandle represents an in-page DOM element.
<a href="#">EventEmitter</a>	The EventEmitter class that many Puppeteer classes extend.
<a href="#">ExecutionContext</a>	This class represents a context for JavaScript execution. A [Page] might have many execution contexts: - each <a href="#">frame</a> has "default" execution context that is always created after frame is attached to DOM. This context is returned by the <a href="#">Frame.executionContext()</a> method. - <a href="#">Extension</a> 's content scripts create additional execution contexts. Besides pages, execution contexts can be found in <a href="#">workers</a> .
<a href="#">FileChooser</a>	File choosers let you react to the page requesting for a file.
<a href="#">Frame</a>	At every point of time, page exposes its current frame tree via the <a href="#">page.mainFrame</a> and <a href="#">frame.childFrames</a> methods.
<a href="#">HTTPRequest</a>	Represents an HTTP request sent by a page.
<a href="#">HTTPResponse</a>	The HTTPResponse class represents responses which are received by the <a href="#">Page</a> class.
<a href="#">JSCoverage</a>	
<a href="#">JSHandle</a>	Represents an in-page JavaScript object. JSHandles can be created with the

	<a href="#">page.evaluateHandle</a> method.
<a href="#">Keyboard</a>	Keyboard provides an api for managing a virtual keyboard. The high level api is <a href="#">Keyboard.type()</a> , which takes raw characters and generates proper keydown, keypress/input, and keyup events on your page.
<a href="#">Mouse</a>	The Mouse class operates in main-frame CSS pixels relative to the top-left corner of the viewport.
<a href="#">Page</a>	Page provides methods to interact with a single tab or <a href="#">extension background page</a> in Chromium.
<a href="#">Puppeteer</a>	The main Puppeteer class.IMPORTANT: if you are using Puppeteer in a Node environment, you will get an instance of <a href="#">PuppeteerNode</a> when you import or require <code>puppeteer</code> . That class extends <code>Puppeteer</code> , so has all the methods documented below as well as all that are defined on <a href="#">PuppeteerNode</a> .
<a href="#">PuppeteerNode</a>	Extends the main <a href="#">Puppeteer</a> class with Node specific behaviour for fetching and downloading browsers.If you're using Puppeteer in a Node environment, this is the class you'll get when you run <code>require('puppeteer')</code> (or the equivalent ES <code>import</code> ).
<a href="#">SecurityDetails</a>	The SecurityDetails class represents the security details of a response that was received over a secure connection.
<a href="#">Target</a>	
<a href="#">TimeoutError</a>	TimeoutError is emitted whenever certain operations are terminated due to timeout.
<a href="#">Touchscreen</a>	The Touchscreen class exposes touchscreen events.
<a href="#">Tracing</a>	The Tracing class exposes the tracing audit interface.
<a href="#">WebWorker</a>	The WebWorker class represents a <a href="#">WebWorker</a> .

## Enumerations

Enumeration	Description
<a href="#">BrowserContextEmittedEvents</a>	
<a href="#">BrowserEmittedEvents</a>	All the events a <a href="#">browser instance</a> may emit.
<a href="#">PageEmittedEvents</a>	All the events that a page instance may emit.

## Functions

Function	Description
<a href="#">clearCustomQueryHandlers()</a>	Clears all registered handlers.
<a href="#">connect(options)</a>	This method attaches Puppeteer to an existing browser instance.
<a href="#">customQueryHandlerNames()</a>	
<a href="#">launch(options)</a>	Launches puppeteer and launches a browser instance with given arguments and options when specified.

<a href="#">registerCustomQueryHandler(name, queryHandler)</a>	Registers a <a href="#">custom query handler</a> . After registration, the handler can be used everywhere where a selector is expected by prepending the selection string with <name>/. The name is only allowed to consist of lower- and upper case latin letters.
<a href="#">unregisterCustomQueryHandler(name)</a>	

## Interfaces

Interface	Description
<a href="#">BoundingBox</a>	
<a href="#">BoxModel</a>	
<a href="#">BrowserConnectOptions</a>	Generic browser options that can be passed when launching any browser or when connecting to an existing browser instance.
<a href="#">BrowserFetcherOptions</a>	
<a href="#">BrowserFetcherRevisionInfo</a>	
<a href="#">BrowserLaunchArgumentOptions</a>	Launcher options that only apply to Chrome.
<a href="#">CDPSessionOnMessageObject</a>	
<a href="#">ClickOptions</a>	
<a href="#">CommonEventEmitter</a>	
<a href="#">ConnectionCallback</a>	
<a href="#">ConnectionTransport</a>	
<a href="#">ConnectOptions</a>	
<a href="#">ConsoleMessageLocation</a>	
<a href="#">ContinueRequestOverrides</a>	
<a href="#">CoverageEntry</a>	The CoverageEntry class represents one entry of the coverage report.
<a href="#">Credentials</a>	
<a href="#">CSSCoverageOptions</a>	Set of configurable options for CSS coverage.
<a href="#">CustomQueryHandler</a>	Contains two functions <code>queryOne</code> and <code>queryAll</code> that can be <a href="#">registered</a> as alternative querying strategies. The functions <code>queryOne</code> and <code>queryAll</code> are executed in the page context. <code>queryOne</code> should take an <code>Element</code> and a selector string as argument and return a single <code>Element</code> or <code>null</code> if no element is found. <code>queryAll</code> takes the same arguments but should instead return a <code>NodeListOf&lt;Element&gt;</code> or <code>Array&lt;Element&gt;</code> with all the elements that match the given query selector.
<a href="#">Device</a>	
<a href="#">FrameAddScriptTagOptions</a>	

<a href="#">FrameAddStyleTagOptions</a>	
<a href="#">FrameWaitForFunctionOptions</a>	
<a href="#">GeolocationOptions</a>	
<a href="#">InternalNetworkConditions</a>	
<a href="#">JSCoverageOptions</a>	Set of configurable options for JS coverage.
<a href="#">JSONObject</a>	
<a href="#">LaunchOptions</a>	Generic launch options that can be passed when launching any browser.
<a href="#">MediaFeature</a>	
<a href="#">Metrics</a>	
<a href="#">MouseOptions</a>	
<a href="#">MouseWheelOptions</a>	
<a href="#">NetworkConditions</a>	
<a href="#">PageEventObject</a>	Denotes the objects received by callback functions for page events. See <a href="#">PageEmittedEvents</a> for more detail on the events and when they are emitted.
<a href="#">PDFMargin</a>	
<a href="#">PDFOptions</a>	Valid options to configure PDF generation via <a href="#">Page.pdf()</a> .
<a href="#">Point</a>	
<a href="#">PressOptions</a>	
<a href="#">ProductLauncher</a>	Describes a launcher - a class that is able to create and launch a browser instance.
<a href="#">PuppeteerEventListener</a>	
<a href="#">RemoteAddress</a>	
<a href="#">ResponseForRequest</a>	Required response data to fulfill a request with.
<a href="#">ScreenshotClip</a>	
<a href="#">ScreenshotOptions</a>	
<a href="#">SerializedAXNode</a>	Represents a Node and the properties of it that are relevant to Accessibility.
<a href="#">SnapshotOptions</a>	
<a href="#">TracingOptions</a>	
<a href="#">Viewport</a>	Sets the viewport of the page.
<a href="#">WaitForOptions</a>	

<a href="#">WaitForSelectorOptions</a>	
<a href="#">WaitForTargetOptions</a>	
<a href="#">WaitTimeoutOptions</a>	

## Variables

Variable	Description
<a href="#">devices</a>	
<a href="#">errors</a>	
<a href="#">EVALUATION_SCRIPT_URL</a>	
<a href="#">networkConditions</a>	
<a href="#">puppeteerErrors</a>	

## Type Aliases

Type Alias	Description
<a href="#">ActionResult</a>	
<a href="#">ChromeReleaseChannel</a>	
<a href="#">ConsoleMessageType</a>	The supported types for console messages.
<a href="#">DevicesMap</a>	
<a href="#">ErrorCode</a>	
<a href="#">EvaluateFn</a>	
<a href="#">EvaluateFnReturnType</a>	
<a href="#">EvaluateHandleFn</a>	
<a href="#">EventType</a>	
<a href="#">Handler</a>	
<a href="#">InterceptResolutionStrategy</a>	
<a href="#">JSONArray</a>	
<a href="#">KeyInput</a>	All the valid keys that can be passed to functions that take user input, such as <a href="#">keyboard.press</a>
<a href="#">MouseButton</a>	
<a href="#">PaperFormat</a>	All the valid paper format types when printing a PDF.
<a href="#">Permission</a>	
<a href="#">Platform</a>	Supported platforms.

<a href="#">PredefinedNetworkConditions</a>	
<a href="#">Product</a>	Supported products.
<a href="#">ProtocolLifeCycleEvent</a>	
<a href="#">PuppeteerErrors</a>	
<a href="#">PuppeteerLifeCycleEvent</a>	
<a href="#">PuppeteerNodeLaunchOptions</a>	Utility type exposed to enable users to define options that can be passed to <code>puppeteer.launch</code> without having to list the set of all types.
<a href="#">ResourceType</a>	Resource types for HTTPRequests as perceived by the rendering engine.
<a href="#">Serializable</a>	
<a href="#">SerializableOrJSHandle</a>	
<a href="#">TargetFilterCallback</a>	
<a href="#">UnwrapElementHandle</a>	Unwraps a DOM element out of an ElementHandle instance
<a href="#">UnwrapPromiseLike</a>	
<a href="#">WrapElementHandle</a>	Wraps a DOM element into an ElementHandle instance