Frame class

At every point of time, page exposes its current frame tree via the [page.mainFrame](#) and [frame.childFrames](#) methods.

Signature:

```
export declare class Frame
```

Remarks

`Frame` object lifecycles are controlled by three events that are all dispatched on the page object:

- [PageEmittedEvents.FrameAttached](#)

- [PageEmittedEvents.FrameNavigated](#)

- [PageEmittedEvents.FrameDetached](#)

The constructor for this class is marked as internal. Third-party code should not call the constructor directly or create subclasses that extend the `Frame` class.

Example 1

An example of dumping frame tree:

```
const puppeteer = require('puppeteer');

(async () => {
  const browser = await puppeteer.launch();
  const page = await browser.newPage();
  await page.goto('https://www.google.com/chrome/browser/canary.html');
  dumpFrameTree(page.mainFrame(), '');
  await browser.close();

  function dumpFrameTree(frame, indent) {
    console.log(indent + frame.url());
    for (const child of frame.childFrames()) {
    dumpFrameTree(child, indent + '  ');
    }
  }
})();
```

Example 2

An example of getting text from an iframe element:

```
const frame = page.frames().find(frame => frame.name() === 'myframe');
const text = await frame.$eval('.selector', element => element.textContent);
```

```
console.log(text);
```

Methods

Method	Modifiers	Description
[$(selector)](#)		This method queries the frame for the given selector.
[$$(selector)](#)		This runs `document.querySelectorAll` in the frame and returns the result.
[$$eval(selector, pageFunction, args)](#)		
[$eval(selector, pageFunction, args)](#)		
[$x(expression)](#)		This method evaluates the given XPath expression and returns the results.
[addScriptTag(options)](#)		Adds a `<script>` tag into the page with the desired url or content.
[addStyleTag(options)](#)		Adds a `<link rel="stylesheet">` tag into the page with the desired url or a `<style type="text/css">` tag with the content.
[childFrames()](#)		
[click(selector, options)](#)		This method clicks the first element found that matches `selector`.
[content()](#)		
[evaluate(pageFunction, args)](#)		
[evaluateHandle(pageFunction, args)](#)		
[executionContext()](#)		
[focus(selector)](#)		This method fetches an element with `selector` and focuses it.
[goto(url, options)](#)		
[hover(selector)](#)		This method fetches an element with `selector`, scrolls it into view if needed, and then uses [Page.mouse](#) to hover over the center of the element.
[isDetached()](#)		
[name()](#)		
[parentFrame()](#)		
[select(selector, values)](#)		Triggers a `change` and `input` event once all the provided options have been selected.
[setContent(html, options)](#)		Set the content of the frame.

[tap(selector)](#)		This method fetches an element with `selector`, scrolls it into view if needed, and then uses [Page.touchscreen](#) to tap in the center of the element.
[title()](#)		
[type(selector, text, options)](#)		Sends a `keydown`, `keypress`/`input`, and `keyup` event for each character in the text.
[url()](#)		
[waitFor(selectorOrFunctionOrTimeout, options, args)](#)		
[waitForFunction(pageFunction, options, args)](#)		
[waitForNavigation(options)](#)		
[waitForSelector(selector, options)](#)		
[waitForTimeout(milliseconds)](#)		Causes your script to wait for the given number of milliseconds.
[waitForXPath(xpath, options)](#)		