Mouse class

The Mouse class operates in main-frame CSS pixels relative to the top-left corner of the viewport.

Signature:

```
export declare class Mouse
```

Remarks

Every `page` object has its own Mouse, accessible with [`page.mouse`](#pagemouse).

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

Example 1

```
// Using 'page.mouse' to trace a 100x100 square.
await page.mouse.move(0, 0);
await page.mouse.down();
await page.mouse.move(0, 100);
await page.mouse.move(100, 100);
await page.mouse.move(100, 0);
await page.mouse.move(0, 0);
await page.mouse.up();
```

Note: The mouse events trigger synthetic `MouseEvent` s. This means that it does not fully replicate the functionality of what a normal user would be able to do with their mouse.

For example, dragging and selecting text is not possible using `page.mouse` . Instead, you can use the `DocumentOrShadowRoot.getSelection()` functionality implemented in the platform.

Example 2

For example, if you want to select all content between nodes:

```
await page.evaluate((from, to) => {
  const selection = from.getRootNode().getSelection();
  const range = document.createRange();
  range.setStartBefore(from);
  range.setEndAfter(to);
  selection.removeAllRanges();
  selection.addRange(range);
}, fromJSHandle, toJSHandle);
```

If you then would want to copy-paste your selection, you can use the clipboard api:

```
// The clipboard api does not allow you to copy, unless the tab is focused.
await page.bringToFront();
await page.evaluate(() => {
  // Copy the selected content to the clipboard
  document.execCommand('copy');
  // Obtain the content of the clipboard as a string
  return navigator.clipboard.readText();
});
```

Note: If you want access to the clipboard API, you have to give it permission to do so:

```
await browser.defaultBrowserContext().overridePermissions(
  '<your origin>', ['clipboard-read', 'clipboard-write']
);
```

Methods

Method	Modifiers	Description
click(x, y, options)		Shortcut for `mouse.move`, `mouse.down` and `mouse.up`.
down(options)		Dispatches a `mousedown` event.
drag(start, target)		Dispatches a `drag` event.
dragAndDrop(start, target, options)		Performs a drag, dragenter, dragover, and drop in sequence.
dragEnter(target, data)		Dispatches a `dragenter` event.
dragOver(target, data)		Dispatches a `dragover` event.
drop(target, data)		Performs a dragenter, dragover, and drop in sequence.
move(x, y, options)		Dispatches a `mousemove` event.
up(options)		Dispatches a `mouseup` event.
wheel(options)		Dispatches a `mousewheel` event.