

WebWorker class

The WebWorker class represents a [WebWorker](#).

Signature:

```
export declare class WebWorker extends EventEmitter
```

Extends: [EventEmitter](#)

Remarks

The events `workercreated` and `workerdestroyed` are emitted on the page object to signal the worker lifecycle.

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

Example

```
page.on('workercreated', worker => console.log('Worker created: ' + worker.url()));
page.on('workerdestroyed', worker => console.log('Worker destroyed: ' +
worker.url()));

console.log('Current workers:');
for (const worker of page.workers()) {
  console.log('  ' + worker.url());
}
```

Properties

Property	Modifiers	Type	Description
_client		CDPSession	
_executionContextCallback		(value: ExecutionContext) => void	
_executionContextPromise		Promise< ExecutionContext >	
_url		string	

Methods

Method	Modifiers	Description
evaluate(pageFunction, args)		If the function passed to the <code>worker.evaluate</code> returns a Promise, then <code>worker.evaluate</code> would wait for the promise to resolve and return its value. If the function passed to the

		<p><code>worker.evaluate</code> returns a non-serializable value, then <code>worker.evaluate</code> resolves to <code>undefined</code>. DevTools Protocol also supports transferring some additional values that are not serializable by JSON: <code>-0</code>, <code>NaN</code>, <code>Infinity</code>, <code>-Infinity</code>, and bigint literals. Shortcut for <code>await</code></p> <pre>worker.executionContext().evaluate(pageFunction, ...args).</pre>
evaluateHandle(pageFunction, args)		<p>The only difference between <code>worker.evaluate</code> and <code>worker.evaluateHandle</code> is that <code>worker.evaluateHandle</code> returns in-page object (JSHandle). If the function passed to the <code>worker.evaluateHandle</code> returns a <code>Promise</code>, then <code>worker.evaluateHandle</code> would wait for the promise to resolve and return its value. Shortcut for <code>await</code></p> <pre>worker.executionContext().evaluateHandle(pageFunction, ...args)</pre>
executionContext()		Returns the <code>ExecutionContext</code> the <code>WebWorker</code> runs in
url()		