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 `worker.evaluate` returns a Promise, then `worker.evaluate` would wait for the promise to resolve and return its value. If the function passed to the

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