

Worker

The Worker class represents a [WebWorker](#). `worker` event is emitted on the page object to signal a worker creation. `close` event is emitted on the worker object when the worker is gone.

```
def handle_worker(worker):
    print("worker created: " + worker.url)
    worker.on("close", lambda: print("worker destroyed: " + worker.url))

page.on('worker', handle_worker)

print("current workers:")
for worker in page.workers:
    print("    " + worker.url)
```

Methods

evaluate

Added in: v1.8

Returns the return value of `expression`.

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()` returns `undefined`. Playwright also supports transferring some additional values that are not serializable by `JSON`: `-0`, `NaN`, `Infinity`, `-Infinity`.

Usage

```
worker.evaluate(expression)
worker.evaluate(expression, **kwargs)
```

Arguments

- `expression` `str`

JavaScript expression to be evaluated in the browser context. If the expression evaluates to a function, the function is automatically invoked.

- `arg` `EvaluationArgument` (*optional*)

Optional argument to pass to `expression`.

Returns

- `Serializable`

evaluate_handle

Added in: v1.8

Returns the return value of `expression` as a `JSHandle`.

The only difference between `worker.evaluate()` and `worker.evaluate_handle()` is that `worker.evaluate_handle()` returns `JSHandle`.

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

Usage

```
worker.evaluate_handle(expression)
worker.evaluate_handle(expression, **kwargs)
```

Arguments

- `expression` `str`

JavaScript expression to be evaluated in the browser context. If the expression evaluates to a function, the function is automatically invoked.

- `arg` `EvaluationArgument` (*optional*)

Optional argument to pass to `expression`.

Returns

- [JSHandle](#)

Properties

url

Added in: v1.8

Usage

```
worker.url
```

Returns

- [str](#)

Events

on("close")

Added in: v1.8

Emitted when this dedicated [WebWorker](#) is terminated.

Usage

```
worker.on("close", handler)
```

Event data

- [Worker](#)