

# deno\_runtime crate

crates.io v0.74.0 docs failing

This is a slim version of the Deno CLI which removes typescript integration and various tooling (like lint and doc). Basically only JavaScript execution with Deno's operating system bindings (ops).

## Stability

This crate is built using battle-tested modules that were originally in `deno` crate, however the API of this crate is subject to rapid and breaking changes.

### MainWorker

The main API of this crate is `MainWorker`. `MainWorker` is a structure encapsulating `deno_core::JsRuntime` with a set of ops used to implement `Deno` namespace.

When creating a `MainWorker` implementors must call `MainWorker::bootstrap` to prepare JS runtime for use.

`MainWorker` is highly configurable and allows to customize many of the runtime's properties:

- module loading implementation
- error formatting
- support for source maps
- support for V8 inspector and Chrome Devtools debugger
- HTTP client user agent, CA certificate
- random number generator seed

### Worker Web API

`deno_runtime` comes with support for `Worker` Web API. The `Worker` API is implemented using `WebWorker` structure.

When creating a new instance of `MainWorker` implementors must provide a callback function that is used when creating a new instance of `Worker`.

All `WebWorker` instances are descendents of `MainWorker` which is responsible for setting up communication with child worker. Each `WebWorker` spawns a new OS thread that is dedicated solely to that worker.