

```

addRequire: false, // allows
  `require()`
experimentalModuleApi: false, //
  uses Module#_compile instead of
  `vm` (you probably don't want
  this and it is bypassed by
  default when vm.Module is
  supported)
};

await vm.getContext({}, options);

```

## Changelog

- v6.0.0 Changes import and require to be project relative (not relative to this package on the file system).
- v5.0.0 Removes sync API, swap to async-only. Better compatibility with --experimental-vm-modules Node flag.
- v4.0.0 Swap to use Module.\_compile as a workaround for #2 (Node regression with experimental modules API in Node v20.10+)
- v3.0.0 ESM-only package. Node 16+

## node-retrieve-globals

Execute a string of JavaScript using Node.js and return the global variable values and functions.

- Supported on Node.js 16 and newer.
- Uses var, let, const, function, Array and Object [destructuring assignment](#).
- Async-only as of v5.0.
- Can return any valid JS data type (including functions).
- Can provide an external data object as context to the local execution scope
- Transforms ESM import statements to work with current CommonJS limitations in Node's vm.
- Uses [Node's vm module to execute JavaScript](#)
  - ⚠️ The node:vm module is not a security mechanism. Do not use it to run untrusted code.
  - codeGeneration (e.g. eval) is disabled by default; use setCreateContextOptions({codeGeneration: { strings: true, wasm: true } }) to re-enable.
  - Works with or without --experimental-vm-modules flag (for vm.Module support). (v5.0.0 and newer)
  - Future-friendly feature tests for when vm.Module is stable and --experimental-vm-modules is no longer necessary. (v5.0.0 and newer)
- In use on:
  - [JavaScript in Eleventy Front Matter](#) (and [Demo](#))

## Advanced options

U sage

npm install node-retrieve-globals

Available on [upm](#)

## Installation

- WebC's <SCF1> WebC:setup>