

Instantly share code, notes, and snippets.



ricardocasares / README.md

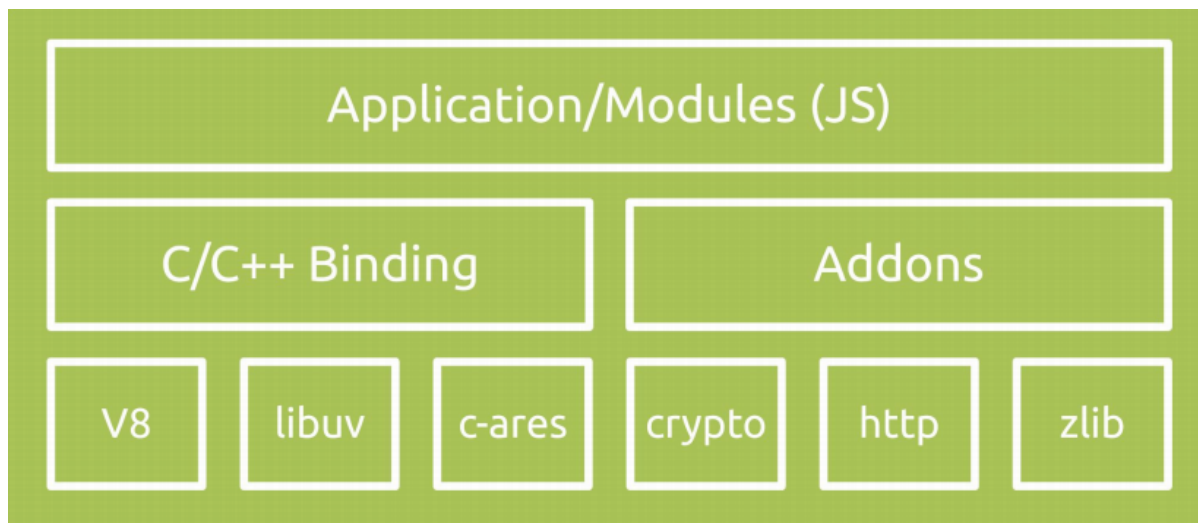
Last active 8 months ago

Node.js interview questions

README.md

Node.js interview questions

What is the architecture of node?



Application & Modules

This is where you write your application & installed modules live. Usually they are written in plain Javascript.

C/C++ bindings

Various glue code and add-ons, it provides Javascript \leftrightarrow C/C++ bridges

Addons

Node.js Addons are dynamically-linked shared objects, written in C++, that can be loaded into Node.js using the `require()` function, and used just as if they were an ordinary Node.js module.

V8

V8 is Google's open source high-performance JavaScript engine, written in C++ and used in Google Chrome, the open source browser from Google, and in Node.js

libuv

libuv is a multi-platform support library with a focus on asynchronous I/O. It was primarily developed for use by Node.js, but it's also used by Luvit, Julia, pyuv, and others.

What are the Node.js globals?

- Buffer
- __dirname
- __filename
- clearImmediate(immediateObject)
- clearInterval(intervalObject)
- clearTimeout(timeoutObject)
- console
- exports
- global
- module
- process
- require()
- setImmediate(callback[, ...args])
- setInterval(callback, delay[, ...args])
- setTimeout(callback, delay[, ...args])

When would you use `child_process` ?

Node is asynchronous non-blocking, which is very effective for high concurrency. However, we have other common needs, such as interacting with operating system shell commands, calling executables, creating child processes for blocking access, or high CPU computations, and so `child_process` is built to meet those needs. As the name suggests, `child_process` is the Node.js way to create sub-processes.

How to interact between proceses?

Via messages:

```
// fork-parent.js
var cp = require('child_process');
var child = cp.fork('./fork-child.js');

child.on('message', function(msg){
    console.log('from child:', msg);
});

child.send('ping!');
```

```
// fork-child.js
process.on('message', function(msg){
    console.log("message from parent:", msg);
    process.send("pong! ");
});
```