

# N-API

Simplifying Node.js Development and Deployment

Arunesh Chandra, *Microsoft*  
Michael Dawson, *IBM*

# Arunesh Chandra, Microsoft

Sr. Program Manager, Microsoft Edge

N-API

Node-ChakraCore Project

Time-Travel Debugging

Contact info:

[arunesh.chandra@microsoft.com](mailto:arunesh.chandra@microsoft.com)

Twitter: @AruneshC

Github: @aruneshchandra



# Michael Dawson, IBM

Senior Software Developer @ IBM  
IBM Node.js community lead/IBM Runtime Technologies

Node.js collaborator, TSC member and Chair , CommComm Member  
Active in Diagnostics, Build, Security, Benchmarking, N-API and  
user-feedback among other Node.js teams and working groups

Contact me:

[michael\\_dawson@ca.ibm.com](mailto:michael_dawson@ca.ibm.com)

Twitter: @mhawson1

Github: @mhawson

Linkedin: <https://www.linkedin.com/in/michael-dawson-6051282>



# Node.js & the NPM module ecosystem

Vast module ecosystem makes Node.js more powerful

Future growth depends on the stability of this ecosystem

Need to preserve and strengthen this ecosystem

Today the module ecosystem is fragile!

```
node app.js
/home/user1/napi-demo-no-napi/napi_demo/node_modules/bindings/bindings.js:88
    throw e
^
```

Error: The module '/home/user1/napi-demo-no-napi/napi\_demo/node\_modules/leveldown/build/Release/leveldown.node' was compiled against a different Node.js version using  
NODE\_MODULE\_VERSION 57. This version of Node.js requires  
NODE\_MODULE\_VERSION 64. Please try re-compiling or re-installing  
the module (for instance, using 'npm rebuild' or 'npm install').

**Error: The module '/home/user1/napi-demo-no-napi/napi\_demo/node\_m  
was compiled against a different Node.js version using**

```
at require (internal/modules/cjs/helpers.js:20:18)
at bindings (/home/user1/napi-demo-no-napi/napi_demo/node_modules/bindings/bindings.js:81:44)
at Object.<anonymous> (/home/user1/napi-demo-no-napi/napi_demo/node_modules/leveldown/leveldown.js:4:46)
at Module._compile (internal/modules/cjs/loader.js:689:30)
at Object.Module._extensions..js (internal/modules/cjs/loader.js:700:10)
```



addaleax commented 6 days ago

Member



...

The removal of the deprecated versions of V8 methods breaks a *lot* of addons, and generates a large number of failing CITGM entries... Is there something we want to do about it?



Dan Shaw  
@dshaw

Following



Replying to [@dshaw](#) [@AruneshC](#) and 4 others

Mission critical native code dependencies is the #1 reason I've been hearing for folks who haven't upgraded to node v4+.

12:05 PM - 3 Apr 2017

# Imagine...

(Node.js App Developers)

- Upgrading Node.js was less work
- Easily swap back and forth between Node versions
- More high performance native modules
- All modules were portable
- Binaries were compatible
- Leverage Innovation in different Node.js “flavors”

# Imagine...

(Module Developers)

- No code update to support new Node.js release
- No need to create new binaries to support a new Node.js release
- It is easier to write an native module
- The API's that are safe to use are well defined and documented
- Your modules will run on other Node.js flavors

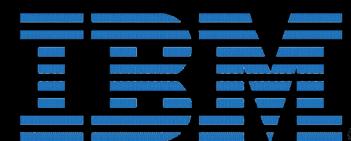
# What's up ?

N-API at the end of the rainbow ☺  
History and Context  
How you can help

# What is N-API

N-API is a stable **Node API layer** for native modules, that **provides ABI compatibility** guarantees across different Node versions & flavors. N-API enables **native modules to just work** across different versions and flavors of Node.js **without recompilations!**

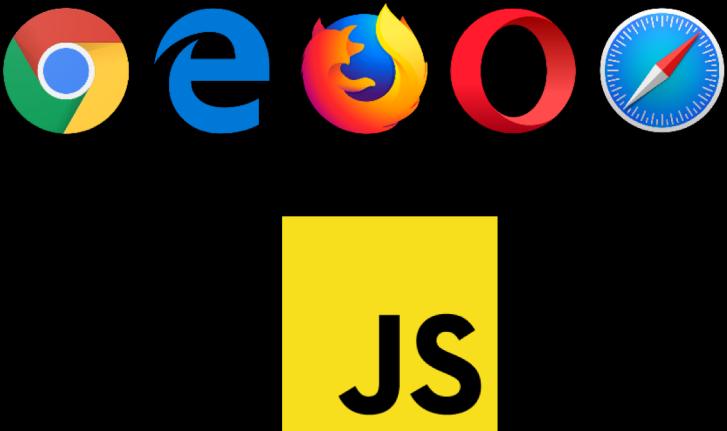
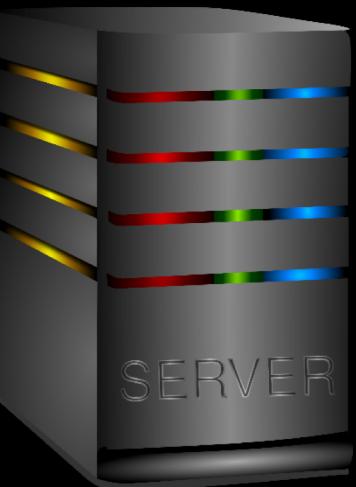
# Contributors



Some context ...

# JavaScript escaped the browser sandbox

---



# Enter Native Addon Modules

## Native addon use cases

Access native APIs on the platform  
unavailable via JavaScript

Derive better performance for  
compute bound code using C/C++

Use of legacy code/logic available in  
C/C++ via JavaScript

# Node.js native add-ons

node-sass

bcrypt

fibers

zmq

canvas

serialport

sqlite3

leveldown

ref

fsevents

websocket

pty.js

microtime

node-hid

libxmljs

hiredis

iconv

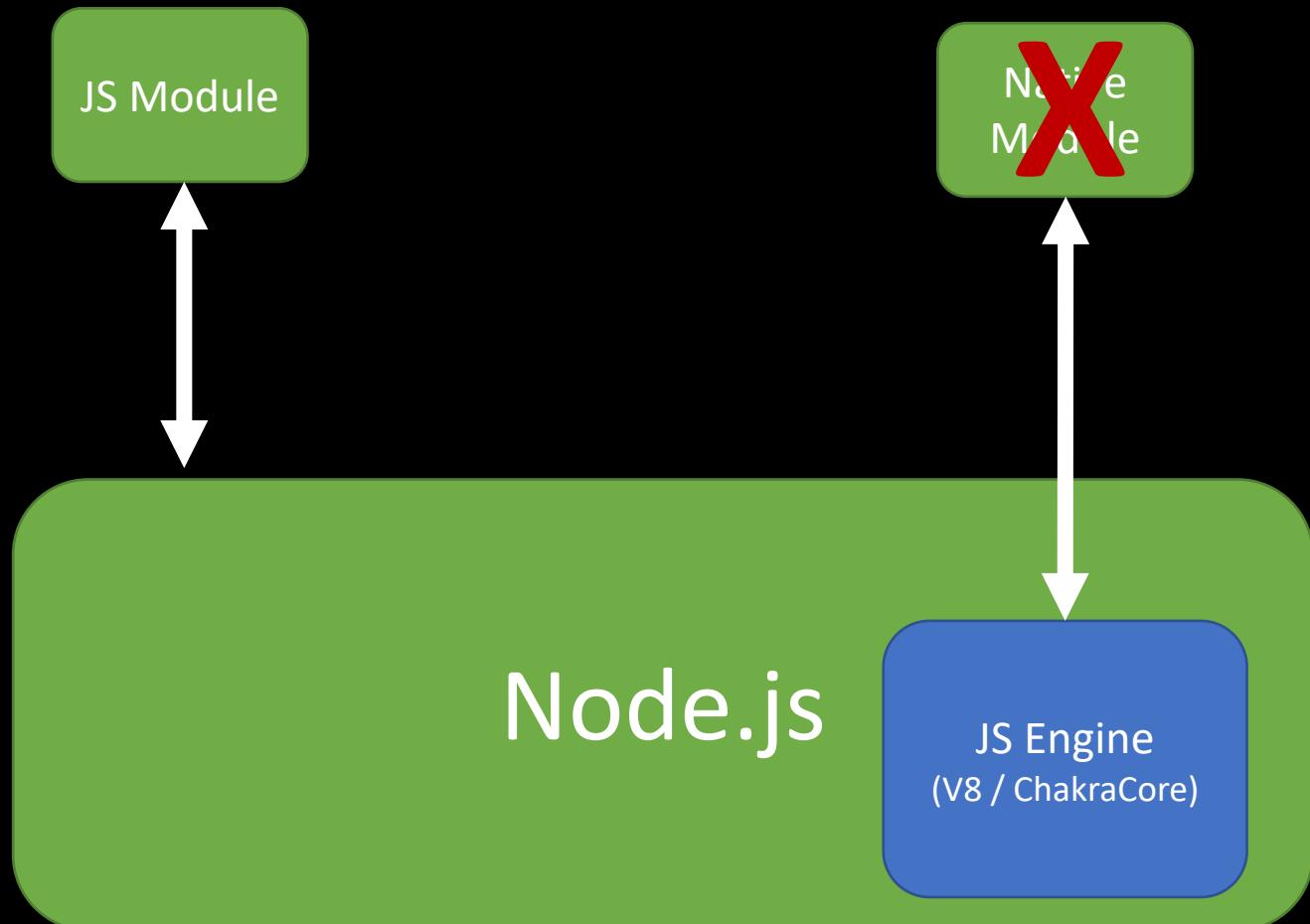
nanomsg

IoTivity

~30%

estimated ecosystem  
dependency

# Before N-API

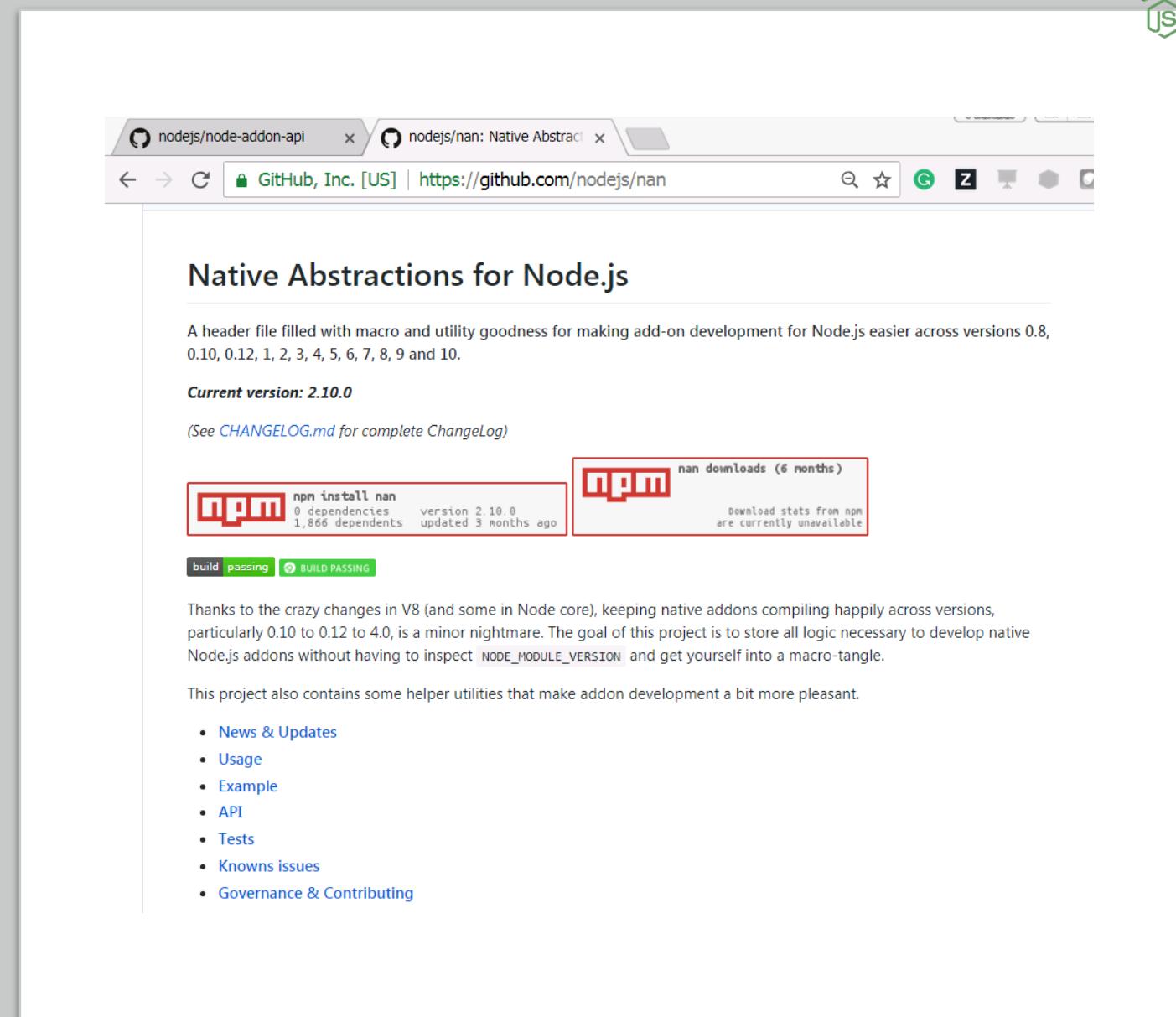


**Module maintainers**  
have to update their  
modules to support  
new Node.js versions

**Module consumers**  
have to recompile  
their NAN modules  
OR  
have to wait for  
updated modules to  
be available

## First attempt to solve the problem - NAN

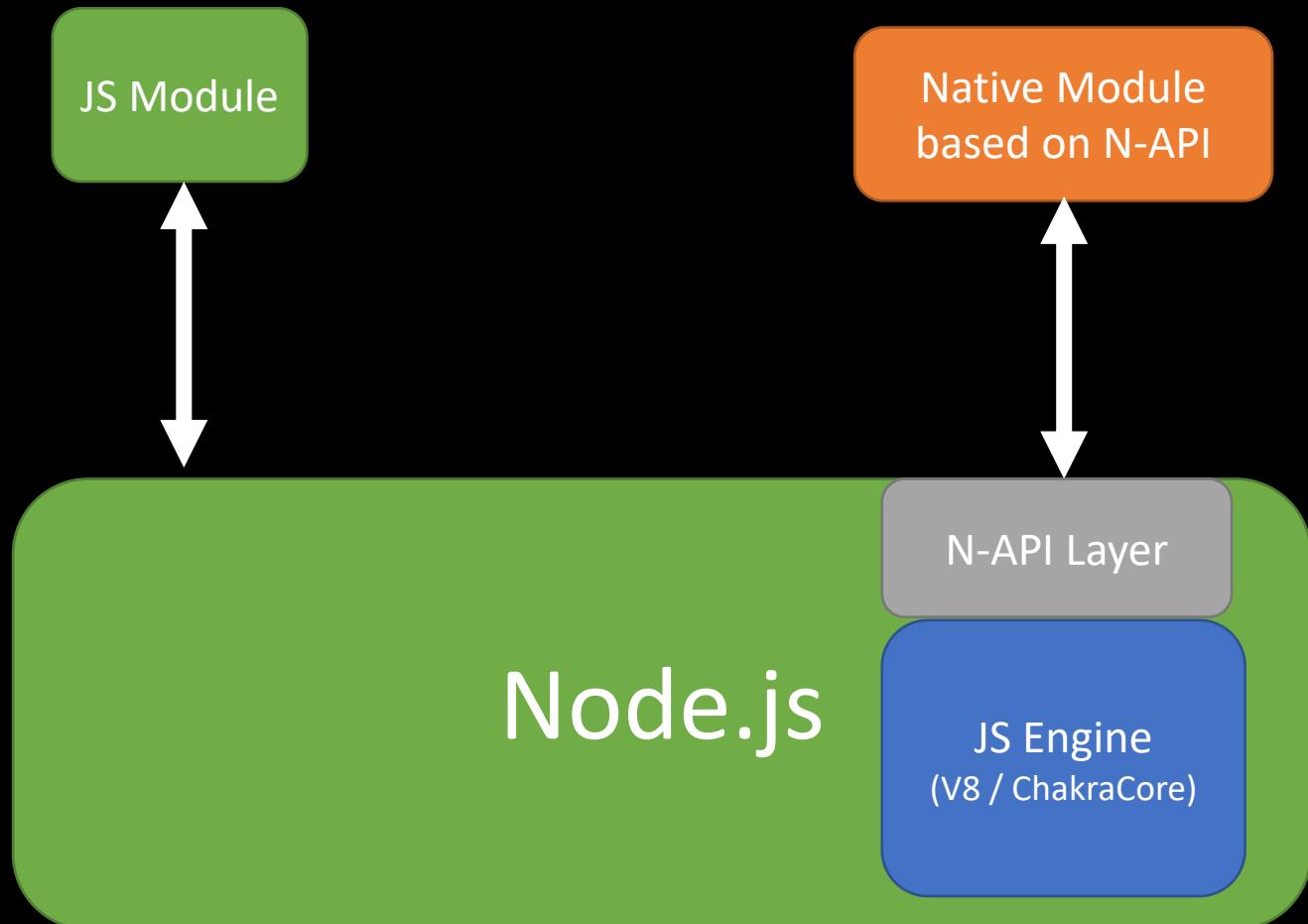
- Available since 0.8 !
- NAN can only provide so much isolation
- Does not address deployment issue
- Needed new approach



The screenshot shows a web browser window with two tabs open. The active tab is titled "nodejs/nan: Native Abstractions for Node.js" and displays the GitHub repository page for the project. The page title is "Native Abstractions for Node.js". A brief description follows: "A header file filled with macro and utility goodness for making add-on development for Node.js easier across versions 0.8, 0.10, 0.12, 1, 2, 3, 4, 5, 6, 7, 8, 9 and 10." The current version is listed as "2.10.0". Below this, a note says "(See [CHANGELOG.md](#) for complete ChangeLog)". Two npm package cards are shown: one for "nan" and one for "nan downloads (6 months)". The "nan" card shows "npm install nan" with "0 dependencies" and "1,866 dependents", "version 2.10.0", and "updated 3 months ago". The "nan downloads (6 months)" card shows a message: "Download stats from npm are currently unavailable". Below the cards, a note states: "Thanks to the crazy changes in V8 (and some in Node core), keeping native addons compiling happily across versions, particularly 0.10 to 0.12 to 4.0, is a minor nightmare. The goal of this project is to store all logic necessary to develop native Node.js addons without having to inspect `NODE_MODULE_VERSION` and get yourself into a macro-tangle." A final note says: "This project also contains some helper utilities that make addon development a bit more pleasant." A bulleted list of links follows:

- [News & Updates](#)
- [Usage](#)
- [Example](#)
- [API](#)
- [Tests](#)
- [Knowns issues](#)
- [Governance & Contributing](#)

# N-API Modules



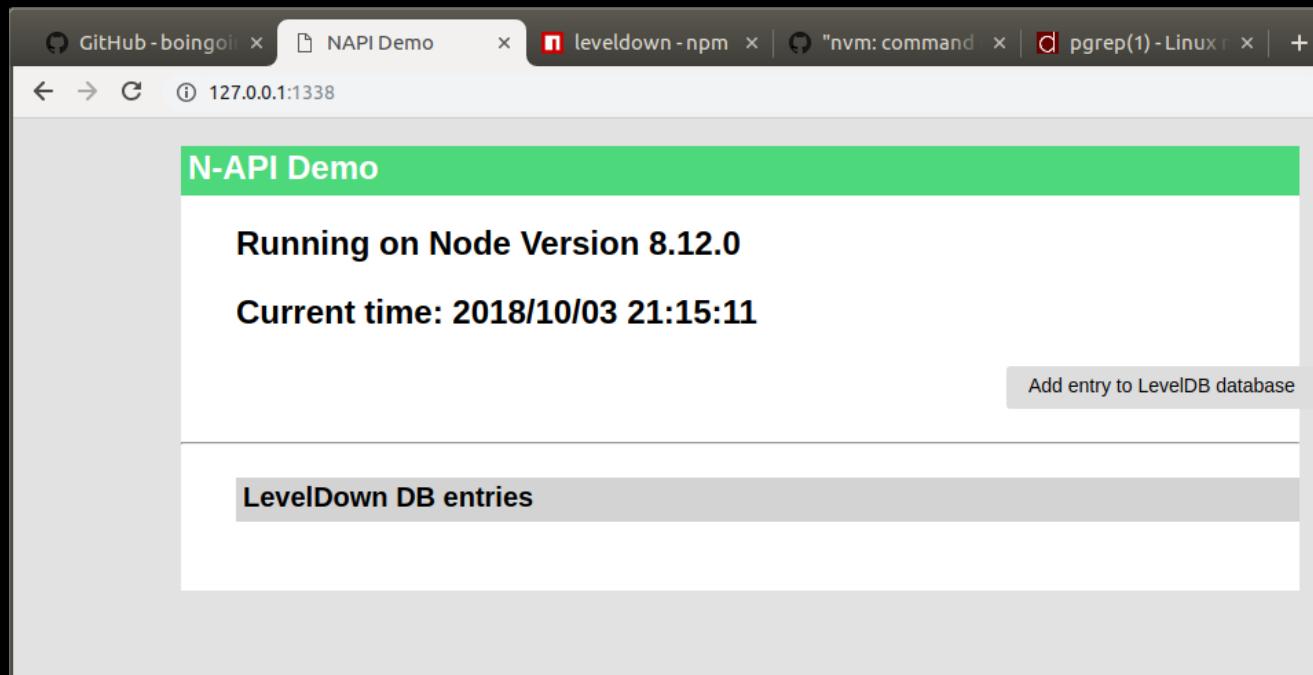
**Module maintainers**  
**don't** have to update  
their modules to  
support new Node.js  
versions

**Module consumers**  
**don't** have to  
recompile their N-API  
modules

OR  
**don't** have to wait for  
updated modules to  
be available

# Demo

- [https://github.com/boingoing/napi\\_demo](https://github.com/boingoing/napi_demo)





# Where Can I use N-API ?

Available  
across all LTS  
release lines!

All the latest LTS release lines provide the same set of APIs for N-API.

- 10.x — Stable
- 8.x — Stable
- 6.x — Experimental (Doc only, looking for feedback)

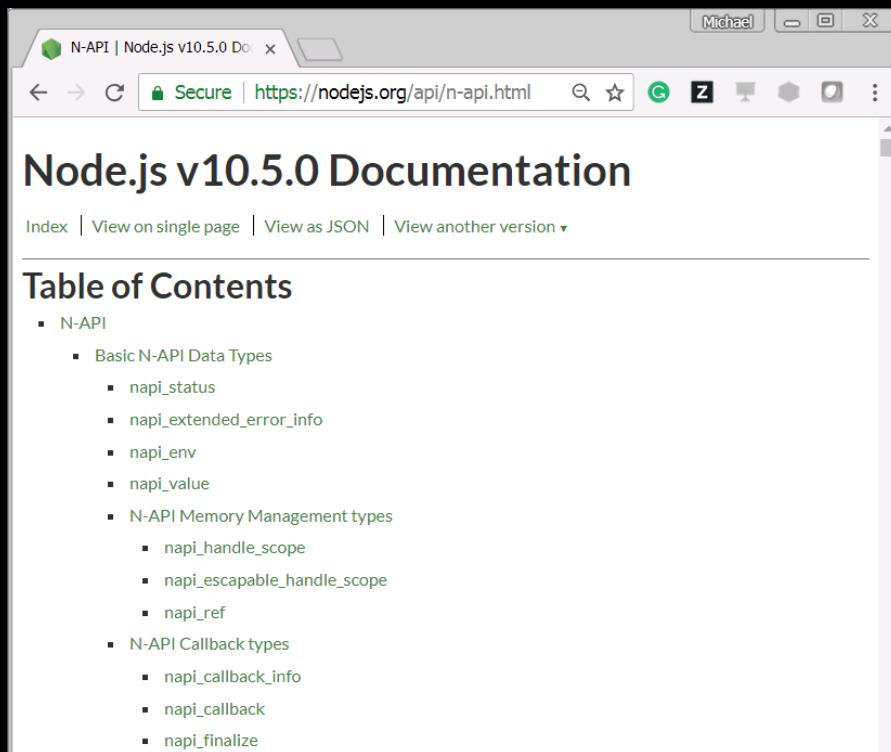
# How Do I use N-API?

C

N-API built into Node.js 6.x, 8.x, 10.x etc.

<https://nodejs.org/api/n-api.html>

```
#include "node_api.h"
```



Node.js v10.5.0 Documentation

Index | View on single page | View as JSON | View another version ▾

## Table of Contents

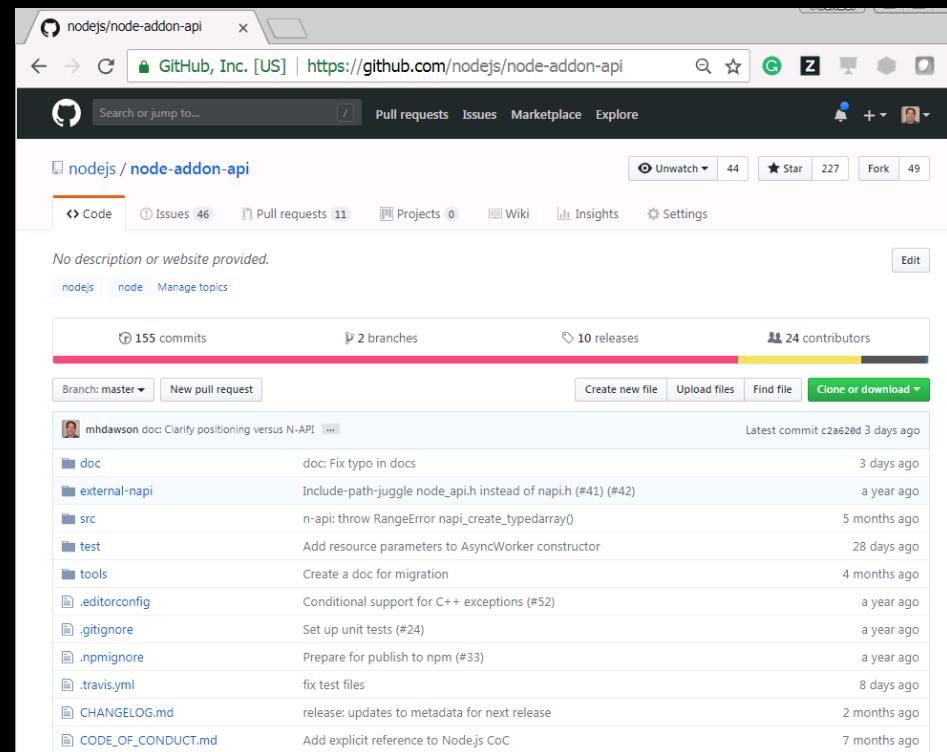
- N-API
  - Basic N-API Data Types
    - napi\_status
    - napi\_extended\_error\_info
    - napi\_env
    - napi\_value
  - N-API Memory Management types
    - napi\_handle\_scope
    - napi escapable\_handle\_scope
    - napi\_ref
  - N-API Callback types
    - napi\_callback\_info
    - napi\_callback
    - napi\_finalize

C++

`npm install node-addon-api`

<https://github.com/nodejs/node-addon-api>

```
#include "napi.h"
```



nodejs/node-addon-api

Search or jump to... Pull requests Issues Marketplace Explore

nodejs / node-addon-api

Code Issues 46 Pull requests 11 Projects 0 Wiki Insights Settings

No description or website provided.

nodejs node Manage topics

155 commits 2 branches 10 releases 24 contributors

Branch: master New pull request Create new file Upload files Find file Clone or download

File	Description	Time Ago
mhdawson doc Clarify positioning versus N-API	Latest commit c2a620d 3 days ago	
doc	doc: Fix typo in docs	3 days ago
external-napi	Include-path-juggle node_api.h instead of napi.h (#41) (#42)	a year ago
src	n-api: throw RangeError napi_create_typedarray()	5 months ago
test	Add resource parameters to AsyncWorker constructor	28 days ago
tools	Create a doc for migration	4 months ago
.editorconfig	Conditional support for C++ exceptions (#52)	a year ago
.gitignore	Set up unit tests (#24)	a year ago
.npmignore	Prepare for publish to npm (#33)	a year ago
.travis.yml	fix test files	8 days ago
CHANGELOG.md	release: updates to metadata for next release	2 months ago
CODE_OF_CONDUCT.md	Add explicit reference to Node.js CoC	7 months ago

# node-addon-api

You Write

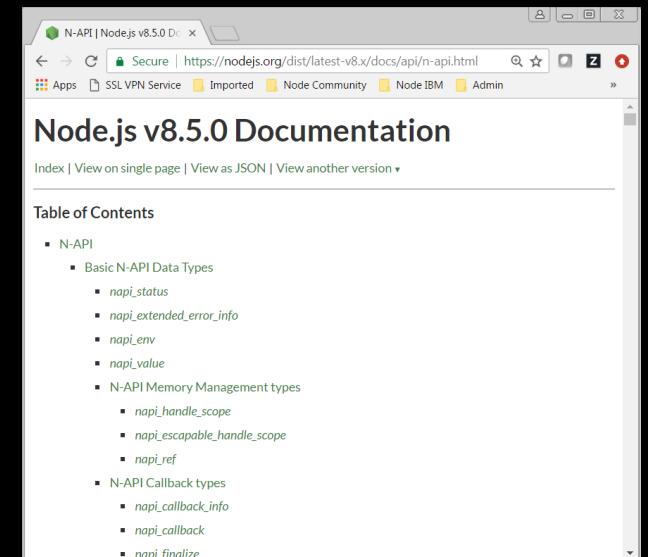
```
1 Object obj = Object::New(env);
2
3 obj["foo"] = String::New(env, "bar");
```

You Get

```
1 napi_status status;
2 napi_value object, string;
3 status = napi_create_object(env, &object);
4 if (status != napi_ok) {
5     napi_throw_error(env, ...);
6     return;
7 }
8 status = napi_create_string_utf8(env, "bar", NAPI_AUTO_LENGTH, &string);
9 if (status != napi_ok) {
10     napi_throw_error(env, ...);
11     return;
12 }
13 status = napi_set_named_property(env, object, "foo", string);
14 if (status != napi_ok) {
15     napi_throw_error(env, ...);
16     return;
17 }
```

# API Shape

- Collection of C APIs available natively in Node.js 6.x, 8.x, 10.x etc.
  - API Docs -  
<https://nodejs.org/dist/latest-v8.x/docs/api/n-api.html>
  - `../src/node_api.h`



```
napi_status napi_create_array(napi_env env, napi_value* result);
napi_status napi_get_last_error_info(napi_env e, const napi_extended_error_info** result);
napi_status napi_is_exception_pending(napi_env e, bool* result);
napi_status napi_get_and_clear_last_exception(napi_env e, napi_value* result);
napi_status napi_throw(napi_env e, napi_value error);
```

# Examples – C- API

## 3\_callbacks

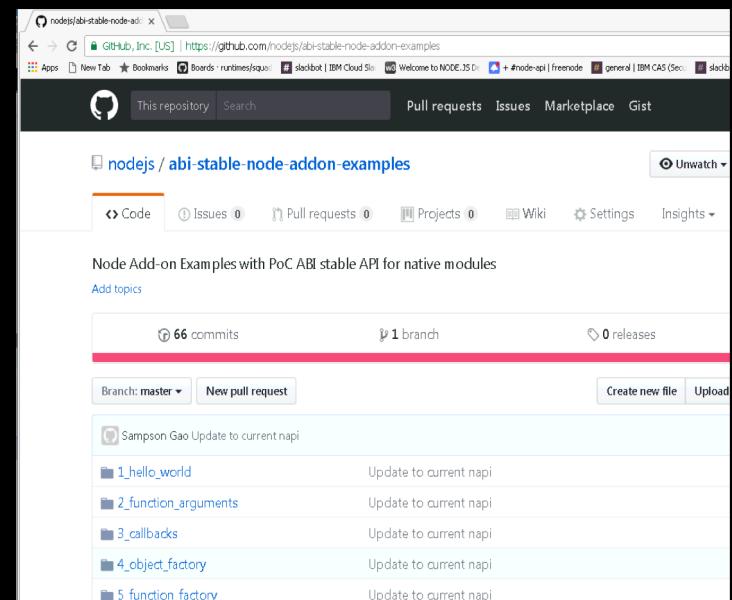
```
#include <node_api.h>
napi_value RunCallback(napi_env env, const napi_callback_info info) {
    ...
}

#define DECLARE_NAPI_METHOD(name, func) \
{ name, 0, func, 0, 0, 0, napi_default, 0 }

napi_value Init(napi_env env, napi_value exports) {
    napi_value new_exports;
    napi_status status =
        napi_create_function(env, "", NAPI_AUTO_LENGTH, RunCallback, nullptr, &new_exports);
    assert(status == napi_ok);
    return new_exports;
}

NAPI_MODULE(addon, Init);
```

<https://github.com/nodejs/node-addon-examples>



# Examples – C- API

```
#include <node_api.h>

napi_value RunCallback(napi_env env, const napi_callback_info info) {
    napi_status status;

    size_t argc = 1;
    napi_value args[1];
    status = napi_get_cb_info(env, info, &argc, args, nullptr, nullptr);

    napi_value cb = args[0];

    napi_value argv[1];
    status = napi_create_string_utf8(env, "hello world", NAPI_AUTO_LENGTH, argv);

    napi_value global;
    status = napi_get_global(env, &global);

    napi_value result;
    status = napi_call_function(env, global, cb, 1, argv, &result);
    return nullptr;
}
```

# Example – C++ Wrapper Example

**NaN**

```
#include <nan.h>

void RunCallback(const Nan::FunctionCallbackInfo<v8::Value>& info) {
    v8::Local<v8::Function> cb = info[0].As<v8::Function>();
    const unsigned argc = 1;
    v8::Local<v8::Value> argv[argc] =
        { Nan::New("hello world").ToLocalChecked() };
    Nan::MakeCallback(Nan::GetCurrentContext()->Global(), cb, argc, argv);
}

void Init(v8::Local<v8::Object> exports,
          v8::Local<v8::Object> module) {
    Nan::SetMethod(module, "exports", RunCallback);
}

NODE_MODULE(addon, Init)
```

**node-addon-api**

```
#include <napi.h>

void RunCallback(const Napi::CallbackInfo& info) {
    Napi::Env env = info.Env();
    Napi::Function cb = info[0].As<Napi::Function>();
    cb.MakeCallback(env.Global(),
                    Napi::String::New(env, "hello world"))
}

Napi::Object Init(Napi::Env env, Napi::Object exports) {
    return Napi::Function::New(env, RunCallback);
}

NODE_API_MODULE(addon, Init)
```

[https://github.com/nodejs/node-addon-examples/tree/master/3\\_callbacks/node-addon-api](https://github.com/nodejs/node-addon-examples/tree/master/3_callbacks/node-addon-api)

<https://github.com/nodejs/node-addon-api/blob/master/tools/conversion.js>

# What are people saying about it ...



**Mikeal Rogers**  
@mikeal

**Following** ▾

Pretty big deal, new native layer for Node.js has landed



**Ingvar Stepanyan**  
@RReverser

**Following** ▾

@nodejs N-API is the kind of API that I wish JS engines would have had for a long time



**Joyee Cheung**  
@JoyeeCheung

**Follow**

Oh wow the C++ wrappers of N-API are much easier to use than NAN/the V8 API. Kudos to everyone who worked on the initiative!

4:32 PM - 24 Jun 2018



**Hubrix**  
@hubrixco

**Follow** ▾

Yowza! #NodeJS, already a very #API-friendly language, gets downright API-amorous with its N-API feature, which gets tagged stable with full support starting in Node 10.

# llnode – Now built on N-API



Joyee Cheung

@JoyeeCheung

Follow

The first version of the JS API of llnode (as a normal Node.js addon) just landed!  
[github.com/nodejs/llnode/...](https://github.com/nodejs/llnode/)

Thanks to the the reviewers in each iteration and @rnchamberlain for the original work, also h/t to the N-API team:  
**it got much simpler after being rewritten with N-API!**

# Log++

- Native logging framework
- Built on N-API
- Compute-intensive formatting is implemented in high performance C++
- The formatting task is offloaded to a background processing thread
- 4 – 7 times faster than any existing logging solutions.

**logpp**

0.0.6 • Public • Published 2 months ago

Readme 2 Dependencies 0 Depen

## Log++ -- A Logging Framework for Modern Development

Log++ is designed to minimize logging overhead on your application so you can use it to provide rich/useful information about your application without impacting your users. As a very rough benchmark we have a comparison with three other popular loggers -- [Bunyan](#), [Debug](#), and [Pino](#) ([Extreme](#)). The first 3 benchmarks are taken from [Pino](#) and the last one is from us. Each message is written 100k times and these timings are from an Intel Core i7-5600 running Node-V8 10.0 on Windows 10.

```

• Basic: info('hello world -- logger')
• String: info('hello %s', 'world')
• Multi: info('hello %s %j %d', 'world', { obj: true }, 4)
• Complex: info('hello at %j from %s with %j %n -- %s', new Date(), app,
  ['iter', { f: i, g: i.toString() }], i - 5, (i % 2 === 0 ? 'ok' :
  'skip'))

```

The results in the table below show a representative timing taken for each logger framework on each benchmark and the speedup of Log++ over the next best performing framework.

Logger	Basic	String	Multi	Complex
Bunyan	690 ms	779 ms	1106 ms	1578 ms
Debug	266 ms	385 ms	515 ms	842 ms
Pino	141 ms	211 ms	378 ms	887 ms
Log++	71 ms	109 ms	253 ms	438 ms
Speedup	1.98x	1.93x	1.49x	1.92x

<https://github.com/mrkmarron/logpp>

<https://www.npmjs.com/package/logpp>

# How you can help – Node.js App Developer

- Use N-API version of the native modules your app depends on
  - Look for Badges when choosing new modules
  - Run script to identify non-NAPI modules
    - <https://github.com/nodejs/node-addon-api/blob/master/doc/checker-tool.md>
    - <https://github.com/nodejs/node-addon-api/blob/master/tools/check-napi.js>
  - Raise issue on module to ask for N-API version, @mention N-API team
- Port your fav native module to N-API

# How you can help - Native Module maintainer

- Port your native module to N-API
- Review documentation for gaps and provide feedback
- Support others who are trying to port your module to N-API

*Join the N-API Working Group at <https://github.com/nodejs/abi-stable-node>*

# Useful Links

- node-addon-api Documentation (C++ Wrapper)  
*<https://github.com/nodejs/node-addon-api>*
- N-API Documentation (Flat C apis)  
*<https://nodejs.org/api/n-api.html>*
- Conversion tool for migrating existing NAN modules to N-API  
*<https://github.com/nodejs/node-addon-api/blob/master/tools/conversion.js>*
- Yeoman generator for building N-API module from scratch  
*<https://github.com/digitalinfinity/generator-napi-module> (C++ Wrapper)*
- Guidance for publishing N-API versions of your modules  
*<https://nodejs.org/en/docs/guides/publishing-napi-modules/>*
- Online Tutorials  
*<https://napi.inspiredware.com/getting-started/first.html>*

@aruneshc  
@mhdawson1

Thank You!

*Join the N-API Working Group at <https://github.com/nodejs/abi-stable-node>*

# Copyright and Trademarks

© IBM Corporation and Microsoft Corporation 2018. All Rights Reserved

IBM, the IBM logo, ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies.

A current list of IBM trademarks is available on the Web at  
“Copyright and trademark information” at  
[www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml)

Microsoft is a trademark of Microsoft Corporation in the United States, other countries, or both.

Node.js is an official trademark of Joyent. IBM SDK for Node.js is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

Java, JavaScript and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

npm is a trademark of npm, Inc.

Intel and the Intel logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States or other countries.

Other trademarks or logos are owned by their respective owners.