



# N-API: Next Generation Node API for native modules

Arunesh Chandra, Microsoft

Michael Dawson, IBM

# Arunesh Chandra, Microsoft

Senior Program Manager @ Microsoft

Developer Tools and Services for Node.js  
*Time-Travel Debugging*

Node.js Community  
*Node-ChakraCore, N-API WG*

Contact me:

[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, CTC and TSC member

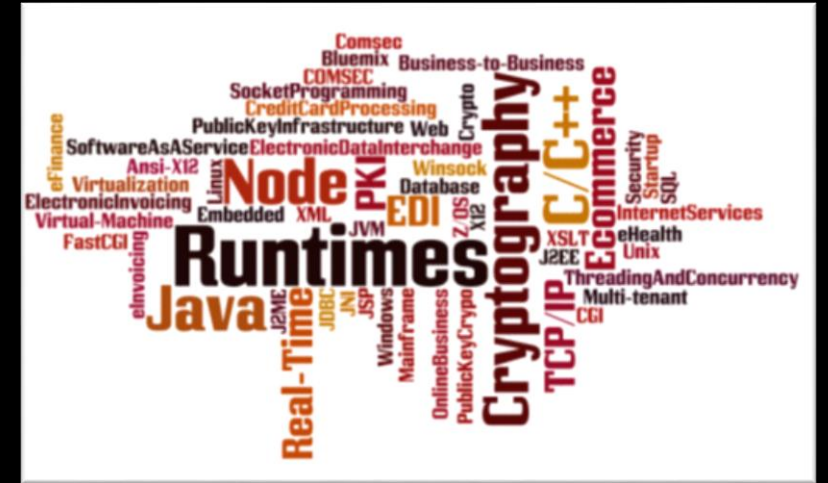
Active in LTS, build, benchmarking, api and post-mortem working groups

## Contact me:

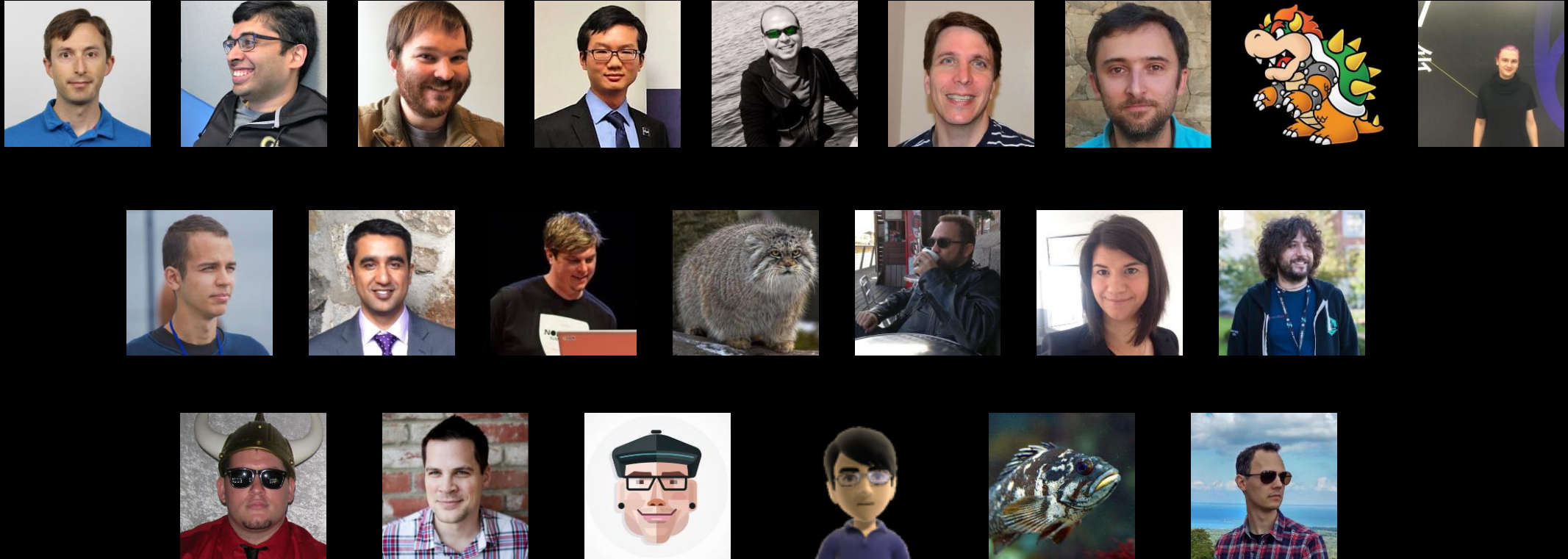
Twitter: @mhdawson1

Github: @mhdawson

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



# Contributors





# 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.

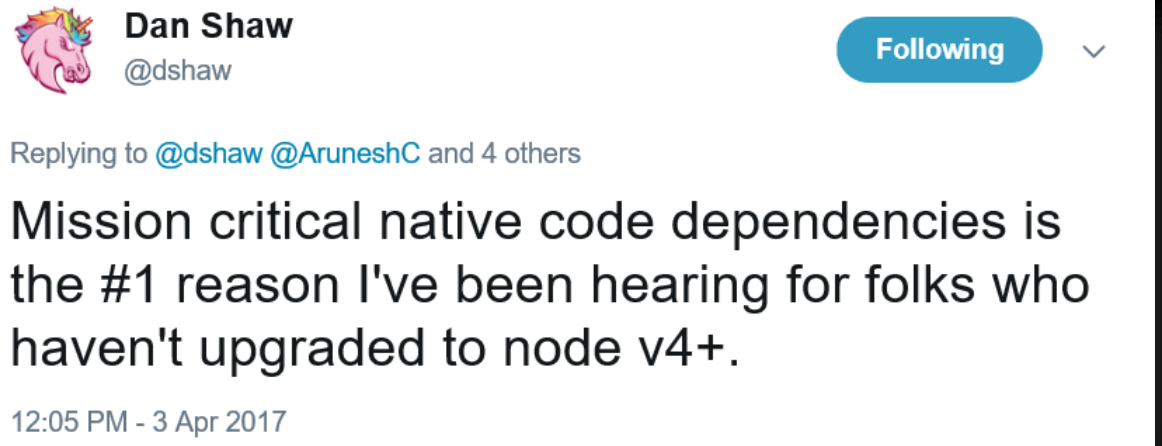
N-API enables native modules to just work across different versions and flavors of Node.js **without recompilations!**

*Available as Experimental in Node.js 8*



# Why do I care about N-API ?

Reduces friction from upgrading to newer Node.js versions in production deployments



Reduces maintenance cost for Native module maintainers

# What are people saying about it ...



**Mikeal Rogers**

@mikeal

Following



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



**Ingvar Stepanyan**

@RRverser

Following



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



# Is it the new NaN ?

- N-API has more complete isolation from V8
- Compile once/run multiple versions and flavors of Node
- Both C and C++ usage supported
- N-API expected to replace NaN usage





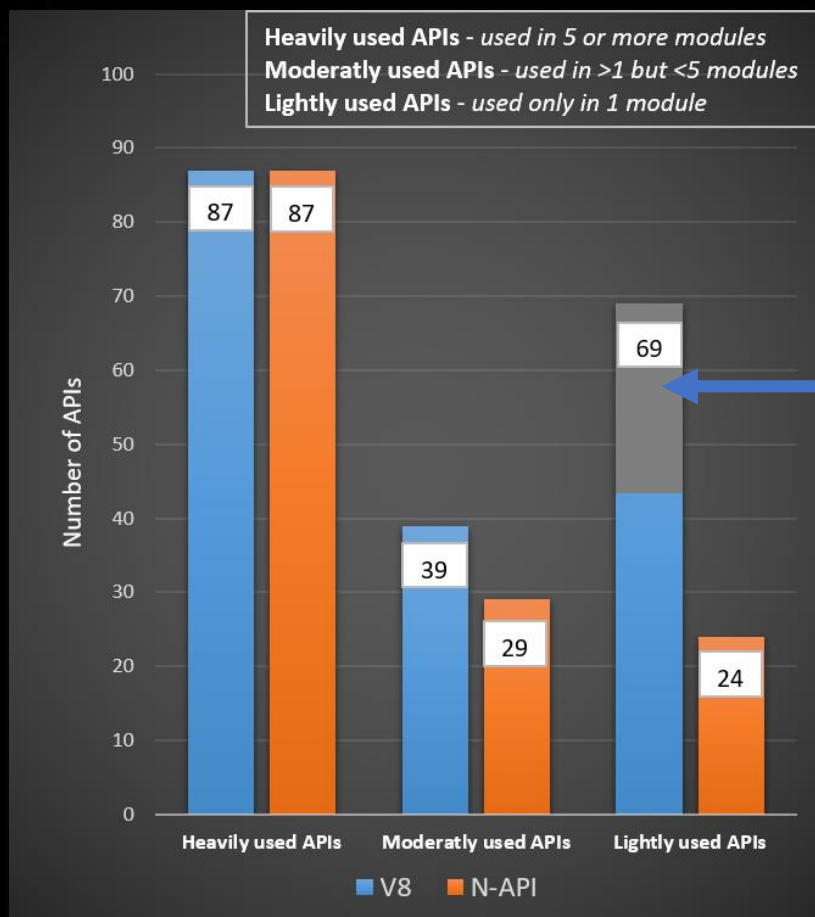
# Ported Modules

- Node-Sass
- Canvas
- SQLite
- LevelDown
- Nanomsg
- IoTivity



# N-API Coverage

*Data based on Top 30 depended on native modules*



v8: Isolate APIs  
no N-API equivalence planned

195  
140/195  
83/195

*Total V8 APIs used in 30 modules*

*N-API equivalent exists*

*Exercised by 5 ported modules*



# N-API Demo

---

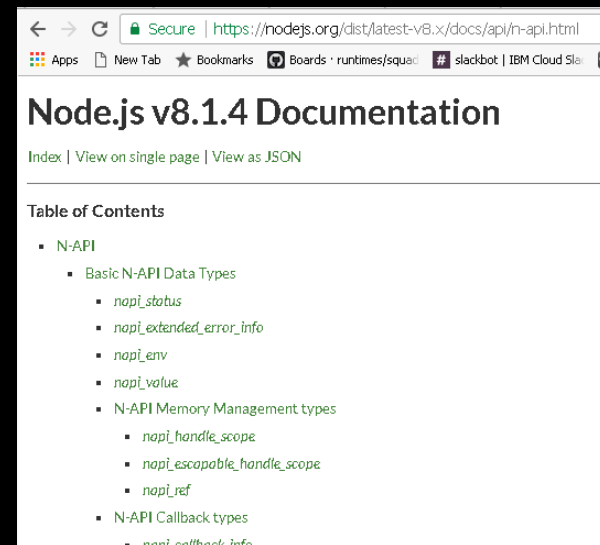
github: [https://github.com/boingoing/napi\\_demo](https://github.com/boingoing/napi_demo)

YouTube: <https://www.youtube.com/watch?v=nmXhJ88nZsk>

# API Shape



- Collection of C APIs available natively in Node.js 8.0
  - 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



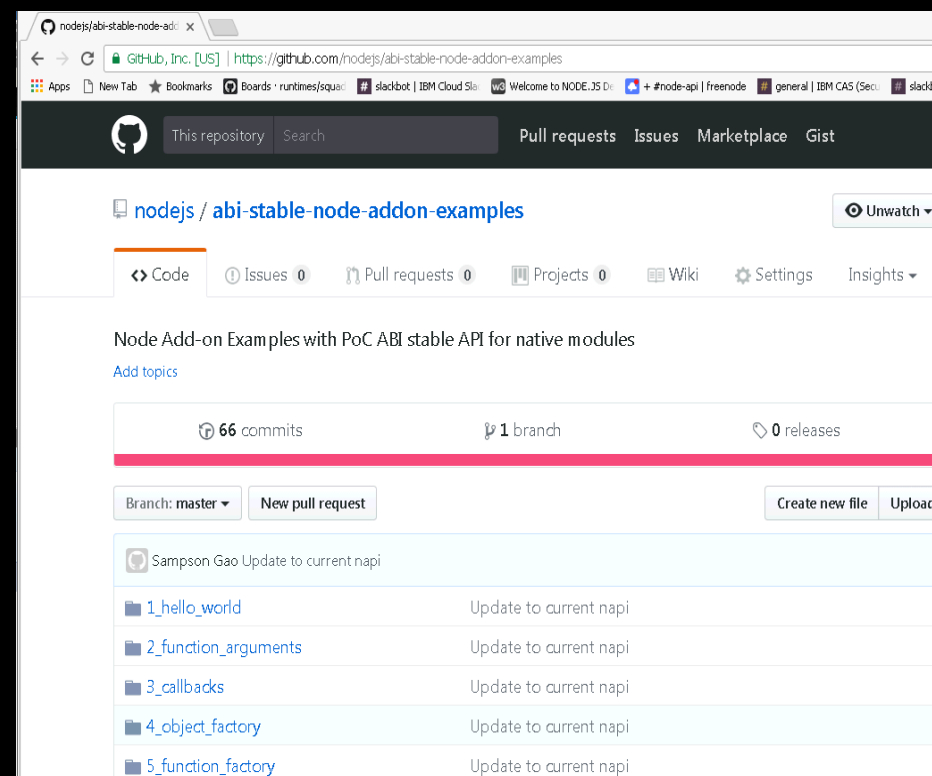
```
#include <node_api.h>
napi_value RunCallback(napi_env env, const napi_callback_info info) {
    ...
}

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

void Init(napi_env env, napi_value exports, napi_value module, void* priv) {
    napi_status status;
    napi_property_descriptor desc = DECLARE_NAPI_METHOD("exports", RunCallback);
    status = napi_define_properties(env, module, 1, &desc);
    assert(status == napi_ok);
}

NAPI_MODULE(addon, Init)
```

<https://github.com/nodejs/abi-stable-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, NULL, NULL);
```

optional 'this'

optional 'data pointer'

```
    napi_value cb = args[0];
```

optional length

```
    napi_value argv[1];
```

```
    status = napi_create_string_utf8(env, "hello world", -1, 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;
```

```
}
```

```
NODE_BUILD=/home/mhdawson/newpull/io.js  
export PATH=$NODE_BUILD:$NODE_BUILD/./abi-stable-node-addon-  
examples/node_modules/node-gyp/bin/:$PATH  
export NPM=$NODE_BUILD/deps/npm/bin/npm-cli.js  
alias npm=$NPM  
alias node-gyp=node-gyp.js  
export npm_config_nodedir=$NODE_BUILD  
export NODE_OPTIONS=--napi-modules
```

# Example – C++ Wrapper Module



The screenshot shows the npm package page for `node-addon-api`. It includes the package name, version, and a description: "Node.js API (N-API) Package". Below the description, there is a section for "API Documentation" with links to "ABI-Stable C APIs in Node.js" and "C++ APIs in this package". A "Getting Started" section is also visible, stating "To use N-API in a native module:".

<https://www.npmjs.com/package/node-addon-api>

The screenshot shows the GitHub repository page for `nodejs/node-addon-api`. It displays the repository name, a search bar, and navigation links for "Code", "Issues", "Pull requests", "Projects", "Wiki", "Settings", and "Insights". Below these links, it shows the repository statistics: "56 commits", "2 branches", and "0 releases". A list of recent commits is visible, including one by `langxiong` and `jasongin` titled "Reference: fix constructor miss some member variables initialize. (#74)".

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

The screenshot shows the "Napi Namespace Reference" page. It lists various classes and their descriptions, including `Array`, `ArrayBuffer`, `AsyncWorker`, `Boolean`, `Buffer`, `CallbackInfo`, `ClassPropertyDescriptor`, `Env`, `Error`, `EscapableHandleScope`, `External`, `Function`, `FunctionReference`, `HandleScope`, `Name`, `Number`, `Object`, `ObjectReference`, `ObjectWrap`, `PropertyDescriptor`, `RangeError`, `Reference`, `String`, `Symbol`, `TypedArray`, `TypedArrayOf`, `TypeError`, and `Value`.

[https://nodejs.github.io/node-addon-api/namespace\\_napi.html](https://nodejs.github.io/node-addon-api/namespace_napi.html)



# 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::Function cb = info[0].As<Napi::Function>();
    cb.MakeCallback(info.Env().Global(), { Napi::String::New(info.Env(),
"hello world") });
}
```

```
void Init(Napi::Env env, Napi::Object exports, Napi::Object module) {
    module.DefineProperty(Napi::PropertyDescriptor::Function("exports",
RunCallback));
}
```

```
NODE_API_MODULE(addon, Init)
```

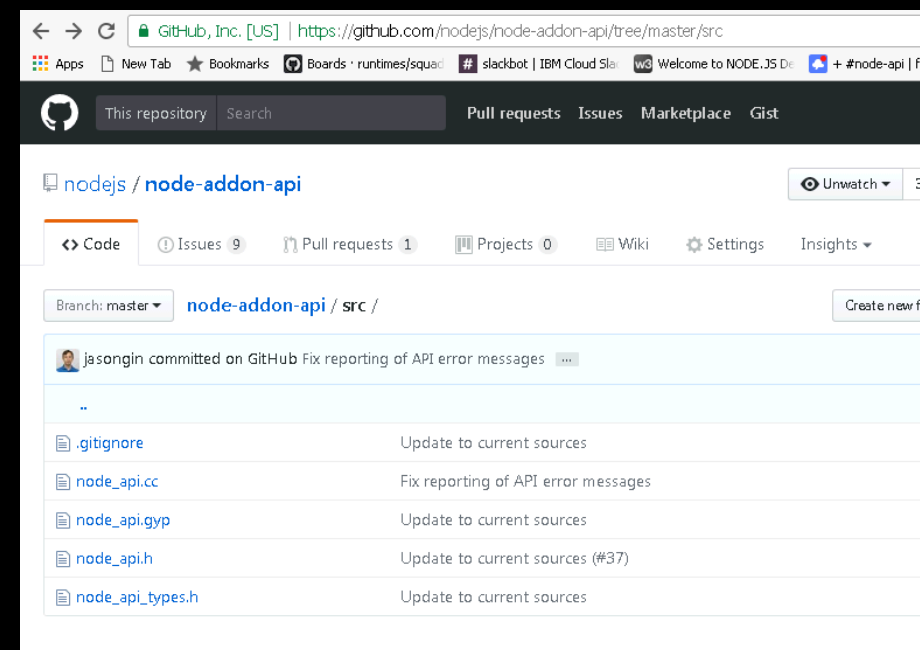
<https://github.com/nodejs/node-addon-api/blob/master/tools/conversion.js>





# Backward Compatibility

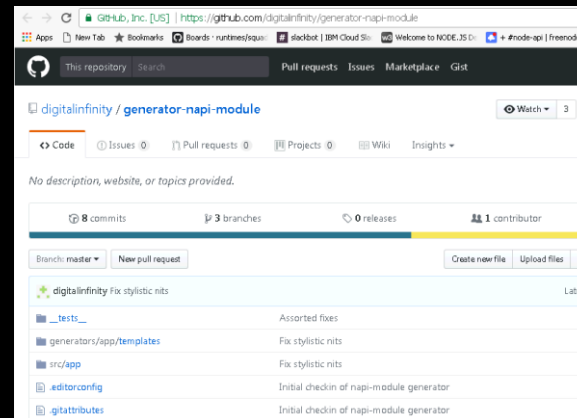
- Available in 8.x
- Plan to backport to 6.x
- For 4.x
  - Copy of API built into node-addon-api module
  - Do **NOT** get build once/run any version
  - Can maintain module with single code base





# Tools for getting started

- From scratch
  - <https://github.com/digitalinfinity/generator-napi-module> (C++ Wrapper)



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



# How you can help

- Port a module
- Try out one of the ported modules
- Help improve the documentation
- Improve test coverage
- Join the N-API Working Group

# Calling all native module maintainers



- Feedback is key to success
  - Earlier the Better
  - Helps overall community
  - API additions prioritized by need
- We are reaching out
  - Need leading modules to help out
  - We understand it will take time
- Guidance for publishing n-api versions
  - <https://nodejs.org/en/docs/guides/publishing-napi-modules/>

<https://github.com/nodejs/abi-stable-node/milestone/5>

- |                          |  |
|--------------------------|--|
| <input type="checkbox"/> | 📌 Document electron issues with different v8 versions.<br>#247 opened on Jun 1 by mhdawson |
| <input type="checkbox"/> | 📌 Reach out to owners of ported modules<br>#246 opened on Jun 1 by mhdawson                |
| <input type="checkbox"/> | 📌 Reach out to top 30 modules owners<br>#245 opened on Jun 1 by mhdawson                   |

# Copyright and Trademarks

© IBM Corporation and Microsoft Corporation, 2017. 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.