

Contributing to Node.js Core

2024, OSSCA, Node.js - Code and Learn

Daeyeon Jeong

Index

- Node.js PR Workflow
- Contribution ideas

Node.js PR Workflow

Identify an Issue

- 이슈 선정
 - PR은 먼저 시작하는 사람이 우선. 다른 방식의 PR은 리뷰에 따라 한쪽이 결정
 - cookie licking 하지 않기
 - 이슈 history를 추적하여 분석 필요
- 큰 변화나 코드 수정이 많은 경우 '사전 의견' 청취가 필요한지 판단
 - 이러한 PR은 Issue 발행 후 충분한 논의를 거쳐 생성
 - 실제 코드로 논의하는 것이 더 효과적일때는 PR로 시작
 - 🧑‍🔬 Issue와 PR을 같이 준비한 경우, 바로 올리지 않으면 누군가 먼저 PR을 올릴 수 있음

Node.js PR Workflow

Make Changes

- 코드 작성에 앞서서 최신 코드 기반으로 코드 작성 시작하기

```
$ git remote add upstream https://github.com/nodejs/node.git
```

```
...
```

```
$ git checkout -b ...
```

```
$ git fetch upstream HEAD && git rebase FETCH_HEAD
```

- 일반적으로 작업한 commit을 squashing 후 제안 (각 commit 개별 적용 필요 없을 경우)

Node.js PR Workflow

Make Changes

- Coding convention 확인
- 문서화 되지 않은 coding convention이 존재하므로 전체적으로 맞춰서 작성
 - 예: destructuring assignment 시 알파벳 순서, const, symbol의 네이밍 규칙, private property 등
- primordials 적용
 - JavaScript 엔진의 원시 값과 객체를 다루는 개념. 전역 객체의 본 상태 오염을 방지하여 보안성 향상
 - 가독성과 성능 이슈. 성능 hot path에 적용 시 주의가 필요하며 사용예 검증 이니셔티브가 진행 중
- 문서 수정
 - Documentation style guide 확인 (doc/README.md)
 - 예: 성 중립 복수 대명사 (they, their, them) 사용. I, we, you 사용을 피함
 - 예: word-wrapped at 80 characters. Use serial commas.

Node.js PR Workflow

Make Changes

```
subsystem: explain the commit in one line
```

The body of the commit message should be one or more paragraphs, explaining things in more detail. Please word-wrap to keep columns to 72 characters or less.

Fixes: <https://github.com/nodejs/node/issues/1337>

Refs: <https://eslint.org/docs/rules/space-in-parens.html>

- Commit message convention
 1. subsystem prefix
 2. imperative verb로 시작
 3. 72자 이하로 작성
- 여러 개의 연관된 subsystem을 추가로 기술할 수 있음: `benchmark, test, lib: XXX`
- Linking keywords와 Refs 활용, 👍 URL은 commit message 대신 PR description에 작성 권장

Node.js PR Workflow

Write Tests

- 버그 수정
 - ✓ 버그 해결을 입증할 테스트 필요 시 commit에 추가
 - 변경 전 실패, 변경 후 성공 케이스가 직관적으로 보이도록 작성
 - 새로 생성, 또는 기존 관련 TC에 추가 가능
- 성능 개선
 - 성능 개선을 입증할 정보 필요 시 테스트 작성
 - benchmark 파일을 추가하거나 PR 설명에 성능 정보를 포함
 - `needs-benchmark-ci` : CI 인프라 사용 권한 필요

Node.js PR Workflow

Write Tests

- TC 작성 방법 확인 (doc/contributing/writing-tests.md, test/common/README.md)

```
1 // Flags: --expose-internals --no-warnings --experimental-shadow-realm
2 'use strict';
3 const common = require('../common');
4 ...
5 setTimeout(..., common.platformTimeout(4000))
6 server.listen(0, () => {
7   http.get({
8     }, common.mustCall((res) => {
9     ...
10   }));
11 });
```

- Must-know details

- `Flags:`
- `require('../common')`: `common` 모듈의 기능을 사용하지 않더라도 반드시 포함해야 함.
- `common.mustCall`, `common.isWindows`, `setTimeout(..., common.platformTimeout(4000))`; 등

Node.js PR Workflow

Write Benchmark Tests

- Benchmark 측정 방법 확인 (doc/contributing/writing-and-running-benchmarks.md)

```
# node benchmark/compare.js --old ./node-main --new ./node-pr-XXXX string_decoder > compare-pr-XXXX.csv
$ npx node-benchmark-compare compare-pr-XXXX.csv # or cat compare-pr-XXXX.csv | Rscript benchmark/compare.R
```

	confidence	improvement	accuracy (*)	(**)	(***)
string-decoder.js ... inLen=128 encoding='ascii'		3.76 %	±1.36%	±1.82%	±2.40%
string-decoder.js ... inLen=128 encoding='utf8'	*	0.81 %	±0.53%	±0.71%	±0.93%
string-decoder.js ... inLen=64 encoding='ascii'		2.70 %	±0.83%	±1.11%	±1.45%
string-decoder.js ... inLen=32 encoding='base64-ascii'		-1.57 %	±0.83%	±1.21%	±1.46%
string-decoder.js ... inLen=32 encoding='ascii'	***	-2.70 %	±0.83%	±1.11%	±1.45%
...					
0.05 false positives, when considering a 5% risk acceptance (*, **, ***),					
0.01 false positives, when considering a 1% risk acceptance (**, ***),					
0.00 false positives, when considering a 0.1% risk acceptance (***)					

- 🙄 tl;dr: 결과에서 confidence column에 별표(*, ** 또는 ***)가 없는 모든 줄은 무시

Node.js PR Workflow

Run Tests Locally

- commit을 push 하기 전 로컬 테스트 수행
- 수정 범위에 따라 적절한 linter 및 formatter 수행

```
$ make format-cpp
$ make lint-cpp
$ make lint-js
$ make lint-md
$ make format-md
# $ make test-doc -j
$ npx -q core-validate-commit --no-validate-metadata --tap
```

- 정확한 테스트 내용은 `Makefile` 와 `.github/workflows/*.yaml` 를 참조


Node.js PR Workflow

Open a Pull Request

- Target branch 지정
 - `main` or `vY.x-staging` (backporting)
 - branch 구분: `main`, `vY.x-staging`, `vY.x` (예: `v22.x-staging`, `v22.x`)
- Review 요청 (Optional)
- Linking keywords 추가 (Optional)
- Labeling (Important, Optional)
 - Submodule
 - Merge method (`commit-queue-squash` / `commit-queue-rebase`)
 - Breaking change (`semver-major` / `semver-minor`)
 - Backport info (`dont-land-on-vY.x`)

Node.js Versioning

- 'semver-major' release
 - 매 6개월마다 main에서 분기
 - 4월/10월에 출시, 홀수 버전 출시 시 짝수 버전은 LTS로 전환
 - Active 상태 : 안정적 판단 시 신규 기능 추가 및 중요한 버그 수정 (LTS 전환부터 12개월 동안)
 - Maintenance 상태 : 중요한 버그 수정 (이후 18개월간)
- Semantic versioning (Semver)
 - MAJOR.MINOR.PATCH
 - MAJOR : ! Breaking change (backward-incompatible change)
 - MINOR : Non-breaking feature addition

 module: __cjsModule on ESM CJS wrapper and in require ESM ✓ esm loaders needs-ci semver-major
#53848 opened 2 weeks ago by guybedford • Approved

Node.js PR Workflow

Open a Pull Request

- CI Test 결과 확인
 - GitHub Action CI
 - 실패시 링크 참조, 또는 `.github/workflows/*.yaml` 참조. 로컬 테스트 및 수정
 - 신규 Contributor는 GitHub Action CI 자동 실행이 안됨
 - Jenkins CI
 - `needs-ci` 실패시
 - ✓ flaky 여부 확인 (<https://github.com/nodejs/reliability/issues>)
 - `needs-citgm` : Node.js 생태계에 미치는 영향 여부 확인 (<https://github.com/nodejs/citgm>)
 - ! Node.js는 많은 undocumented behaviors를 'public'으로 간주함.
 - Triagers 이상 멤버가 실행 가능

Node.js PR Workflow

Request Reviews

■ Approvals

- `Normal` : 최소 2명의 collaborators 승인 필요, 48시간 대기 (다른 reviewer의 참여를 위함)
- `Fast-track` : `fix regressions`, `doc only` 등에 대해 Upvote로 대기 없이 merge 가능
- `Semver-major` : 2명의 TSC voting members 승인 필요

■ Change requests

- 👍 리뷰 반영은 `force-push` 보다 `fixup commit` 작성 추천 (rebase 필요 제외)
 - `git commit --fixup`
- 이전에 landing 조건이 되었어도 코드 변경이 발생하면 재승인 필요
- Objection은 hard objection만 인정되며, 수정 요청에 대한 합의가 안 되면 TSC 안건으로 요청 가능

Node.js PR Workflow

Landing

- Merging pull requests
 - Label 기반 자동화 : 5분 주기로 `commit-queue` labeled PR을 확인, 조건을 판단하여 merge 수행
 - GitHub Merge pull request 버튼을 이용하면 메타데이터 추가가 어렵기 때문
 - [@node-core/utils](#) 사용



근데, PR 뭘하지...

Contribution ideas

- `good first issue` 도 쉽지 않구나... ☹️
- `confirmed-bug` : 내 환경에서 재현 가능한 이슈 찾기
- `TODO` : 남겨진 이슈 찾아보기
- `분석/리뷰` : 관심 있는 주제에 대한 다른 사람의 PR 분석하며 배우고 리뷰 참여하기
- `문서 개선` : 정의된 API가 실제 코드와 일치하는지 확인하고, 예제 문제점 찾기, WG 참여
- `관련 프로젝트 참여` : core 외에 관심 있는 다른 프로젝트 (deps 모듈, doc tool, 웹사이트 개선 등)
- `WPT` (Web Platform Test) : 테스트와 결과가 이미 있는 (TDD) 이슈 수정하기
- `신규 모듈` - 이슈가 많은 신규 모듈 발전에 기여하기
- `성능 개선` - 알고리즘 개선점 확인 및 WG 참여

Idea

Remove duplicate code

- JavaScript task. Look for the 'ERR_INVALID_XXX' pattern and identify duplicate validation code
- Refactor the duplicate code using methods from internal/validators.js.

```
validateInteger(shard.index, 'options.shard.index');  
- if (shard.index <= 0 || shard.total < shard.index) {  
-   throw new ERR_OUT_OF_RANGE('options.shard.index', `>= 1 && <= ${shard.total}`, shard.index);  
- }  
+ validateInteger(shard.index, 'options.shard.index', 1, shard.total);
```

```
/** @type {validateInteger} */  
const validateInteger = hideStackFrames(  
  (value, name, min = NumberMIN_SAFE_INTEGER, max = NumberMAX_SAFE_INTEGER) => {  
    if (typeof value !== 'number')  
      throw new ERR_INVALID_ARG_TYPE(name, 'number', value);  
    if (!NumberIsInteger(value))  
      throw new ERR_OUT_OF_RANGE(name, 'an integer', value);  
    if (value < min || value > max)  
      throw new ERR_OUT_OF_RANGE(name, `>= ${min} && <= ${max}`, value);  
  },  
);
```

Idea

Remove internal errors

- JavaScript task. Investigate occurrences of `ERR_INTERNAL_ASSERTION` in the issues
- Remove the assert statement and replace it with an appropriate error throw (semver-major).

```
- assert(typeof options.checkServerIdentity === 'function');  
+ validateFunction(options.checkServerIdentity, 'options.checkServerIdentity'); // lib/internal/validators.js
```

```
- node:internal/assert:14  
-   throw new ERR_INTERNAL_ASSERTION(message);  
-   ^  
- Error [ERR_INTERNAL_ASSERTION]: This is caused by either a bug in Node.js or incorrect usage of Node.js internals.  
- Please open an issue with this stack trace at https://github.com/nodejs/node/issues  
  
+ node:internal/validators:440  
+   throw new ERR_INVALID_ARG_TYPE(name, 'Function', value);  
+   ^  
+ TypeError [ERR_INVALID_ARG_TYPE]: The "options.checkServerIdentity" property must be of type function. Received ...
```

Idea

Improve typings for bindings

- TypeScript task. Basic Knowledge about C++ bindings
- Identify binding APIs that lack Intellisense and write `.d.ts` files to provide type annotations.
- Example: <https://github.com/nodejs/node/pull/40222>

```
// lib/os.js
const { getUptime, ... } = internalBinding('os');
```

```
// src/node_os.cc
static void GetUptime(const FunctionCallbackInfo<Value>& args) {
    double uptime; ...
    args.GetReturnValue().Set(uptime);
}
...
SetMethod(context, target, "getUptime", GetUptime);
```

```
// typings/internalBinding/os.d.ts
declare function InternalBinding(binding: 'os'): {
    ...
    getUptime(): number; ...
}
```

Idea

Performance improvements

- Investigate specific APIs to improve algorithms or optimize common paths.
- Review the Performance WG issues as well.
- Example: <https://github.com/nodejs/node/pull/29066>

```
function main({ n, bytes, partial }) {  
  const source = Buffer.allocUnsafe(bytes);  
  const target = Buffer.allocUnsafe(bytes);  
  const sourceStart = (partial === 'true' ? Math.floor(bytes / 2) : 0);  
  bench.start();  
  for (let i = 0; i < n; i++) {  
    source.copy(target, 0, sourceStart);  
  }  
  ...  
}
```

buffers/buffer-copy.js	n=6000000	partial='false'	bytes=0	***	580.96 %	±3.05%	±4.06%	±5.30%
buffers/buffer-copy.js	n=6000000	partial='false'	bytes=8	***	179.90 %	±5.71%	±7.69%	±10.20%
buffers/buffer-copy.js	n=6000000	partial='false'	bytes=128	***	174.81 %	±1.17%	±1.56%	±2.05%
buffers/buffer-copy.js	n=6000000	partial='false'	bytes=32768	*	3.62 %	±3.30%	±4.44%	±5.89%
buffers/buffer-copy.js	n=6000000	partial='true'	bytes=0	***	566.67 %	±3.63%	±4.87%	±6.42%
buffers/buffer-copy.js	n=6000000	partial='true'	bytes=8	***	24.66 %	±1.17%	±1.57%	±2.05%

Idea

Improve Web API Compatibility

- JavaScript Task. Check the WPT (Web Platform Test) status and fix any failed tests.
- Update WPT to ensure compatibility with Web APIs.

```
// test/wpt/status/streams.json
"readable-streams/owning-type.any.js": {
  "fail": {
    "note": "Readable streams with type owning are not yet supported",
    "expected": [
      "ReadableStream can be constructed with owning type",
      "ReadableStream of type owning should call start with a ReadableStreamDefaultController",
      "ReadableStream should be able to call enqueue with an empty transfer list",
      "ReadableStream should check transfer parameter",
      "ReadableStream of type owning should transfer enqueued chunks"
    ]
  }
},
```

Idea

Remove unnecessary `Environment::GetCurrent`

- C++ Task. Basic knowledge of C++ and the `node::Environment` class.
- Look for patterns where `env` is only used to retrieve `env->isolate()` and/or `env->context()`.
 - These can be replaced with `args.GetIsolate()` or `args.GetIsolate()->GetCurrentContext()`, as they involve fewer memory reads.

```
void SomeCallback(const FunctionCallbackInfo<Value>& args) {  
    Environment* env = Environment::GetCurrent(context);  
    // Isolate* isolate = args.GetIsolate();  
    ...  
  
    args.GetReturnValue().Set(ArrayBuffer::New(env->isolate(), std::move(store)));  
}
```

Idea

Replace `ToLocalChecked()` uses with `ToLocal()`

- C++ Task. Basic knowledge of C++ and a basic understanding of V8
- Identify patterns where internal aborts may occur due to the use of `ToLocalChecked()`.
- Replace them with the safer `ToLocal()`.
- Replace the `ToLocalChecked()` uses with the safer `ToLocal()`.
- Example: <https://github.com/nodejs/node/pull/53869>

```
- Local<Value> buffer = buffers->Get(env->context(), i).ToLocalChecked();  
+ Local<Value> buffer;  
+ if (!buffers->Get(env->context(), i).ToLocal(&buffer)) return;
```


Thank you

<https://github.com/daeyeon/code-and-learn>
daeyeon.dev@gmail.com