estree / **estree**

The ESTree Spec

estree javascript ast parsing specification

168 commits | **4** branches | **0** releases | **29** contributors

Branch: **master** | New pull request | Find file

Clone or download

mysticatea and **RReverser** Merge optional-catch-binding into es2019

Latest commit `df2290b` May 24, 2018

experimental	Merge optional-catch-binding into es2019	May 24, 2018
extensions	Rename spec.md to es5.md	Aug 23, 2016
.gitmodules	add export extensions as a submodule	Jan 21, 2016
README.md	Add note in readme about syntax (#156)	Feb 28, 2017
deprecated.md	Perform destructuring of specs for easier navigation from TOC.	Oct 25, 2015
es2015.md	Include missing SpreadElement in NewExpression arguments. (#182)	Jan 22, 2018
es2016.md	change references to spec versions to use yearly naming scheme	Aug 22, 2016
es2017.md	change references to spec versions to use yearly naming scheme	Aug 22, 2016
es2018.md	fix description	May 23, 2018
es2019.md	Merge optional-catch-binding into es2019	May 24, 2018
es5.md	Make Directive | Statement comments meaningful (#173)	Oct 5, 2017

README.md

The ESTree Spec

Once upon a time, an unsuspecting Mozilla engineer created an API in Firefox that exposed the SpiderMonkey engine's JavaScript parser as a JavaScript API. Said engineer documented the format it produced, and this format caught on as a lingua franca for tools that manipulate JavaScript source code.

Meanwhile JavaScript is evolving, notably with the upcoming release of ES2015. This site will serve as a community standard for people involved in building and using these tools to help evolve this format to keep up with the evolution of the JavaScript language.

Discussion

We've started the process of bringing together various communities using this format to move it forward into the ES2015 era and beyond. Feel free to join us! We'll be discussing in the issue tracker and in `#esprima` on Freenode.

AST Descriptor Syntax

The spec uses a custom syntax to describe its structures. For example, at the time of writing, 'es2015.md' contained a description of `Program` as seen below

```
extend interface Program {
    sourceType: "script" | "module";
    body: [ Statement | ModuleDeclaration ];
}
```

Participating Members

- Dave Herman (Mozilla)
- Ingvar Stepanyan (Acorn)
- Nicholas C. Zakas (ESLint)
- Mike Sherov, Ariya Hidayat (Esprima)
- Michael Ficarra (@michaelficarra)
- Henry Zhu, Logan Smyth, Daniel Tschinder (Babel)

Inactive Members

- Sebastian McKenzie (Babel)
- Kyle Simpson (@getify)

Philosophy

Suggested additions and modifications must follow these guidelines:

1. **Backwards compatible:** Non-additive modifications to existing constructs will not be considered unless immense support is in favor of such changes. (eg. #65)
2. **Contextless:** Nodes should not retain any information about their parent. ie. a `FunctionExpression` should not be aware of if it's a concise method. (eg. #5)
3. **Unique:** Information should not be duplicated. ie. a `kind` property should not be present on `Literal` if the type can be discerned from the `value`. (eg. #61)
4. **Extensible:** New nodes should be specced to easily allow future spec additions. This means expanding the coverage of node types. ie. `MetaProperty` over `NewTarget` to cover future meta properties. (eg. #32)