

## babel-runtime

Source code of released version | Source code of development version \*\*\*

Meteor maintains a version of the runtime helpers needed by Babel-transpiled code. In most cases, the code is copied from Babel's helper implementations, though we have also made some changes.

Benefits of maintaining our own package include:

- IE 8 support. Babel's helpers target IE 9 and do not work in IE 8, but generally IE 8 support can be achieved with only minor changes.
- Backwards-compatibility. When the Babel compiler changes, the helpers sometimes change. Our Babel package can keep old helpers for back-compat. (If we change over to publishing original ES6 code in packages instead of transpiled code, this becomes less important.)
- Client-side code size. We've made the helpers file as small as possible.

## Helpers

Helpers needed for each transform **as of Babel v5.6.15**:

- es3.propertyLiterals: None
- es3.memberExpressionLiterals: None
- es6.arrowFunctions: None
- es6.templateLiterals
  - taggedTemplateLiteralLoose
- es6.classes
  - inherits
  - classCallCheck
  - createClass (only for getter/setters)
  - Excluded because only for decorator support(2): createDecoratedClass, defineDecoratedPropertyDescriptor
- es6.constants: None
- es6.blockScoping: None
  - Excluded because only for spec mode(1): temporalUndefined, temporalAssertDefined
- es6.properties.shorthand: None
- es6.properties.computed: None
  - Excluded because only for non-loose mode(1): defineProperty
- es6.parameters: None
- es6.spread
  - bind (for new A(...b))
- es6.forOf: None
- es7.objectRestSpread
  - \_extends
  - Everything in es6.destructuring

- es6.destructuring
  - `objectWithoutProperties`
  - `objectDestructuringEmpty`
- es7.trailingFunctionCommas: None
- flow: None

Footnotes:

1. A transform can be run in “loose,” normal, or “spec” mode, with “loose” providing the fastest, most lightweight, and usually most browser-compatible transpilation, while “spec” mode tries extra hard to be spec-compliant at the expense of those things. We’ve found that “loose” mode is the best mode for production code for every transform we’ve looked at.
2. Decorators are still a Stage 1 proposal and are only implemented in Babel as an experiment.