

> Itty bitty little widdle twinkie pinkie [ES2015 Promise](https://people.mozilla.org/~jorendorff/es6-draft.html#sec-promise-objects) implementation [![Build Status](https://travis-ci.org/floatdrop/pinkie.svg?branch=master)](https://travis-ci.org/floatdrop/pinkie) [![Coverage Status](https://coveralls.io/repos/floatdrop/pinkie/badge.svg?branch=master&service=github)](https://coveralls.io/github/floatdrop/pinkie?branch=master) There are [tons of Promise implementations](https://github.com/promises-aplus/promises-spec/blob/master/implementations.md#standalone) out there, but all of them focus on browser compatibility and are often bloated with functionality. This module is an exact Promise specification polyfill (like [native-promise-only](https://github.com/getify/native-promise-only)), but in Node.js land (it should be browserify-able though). ## Install ``` $ npm install --save pinkie ``` ## Usage ```js var fs = require('fs'); var Promise = require('pinkie'); new Promise(function (resolve, reject) { fs.readFile('foo.json', 'utf8', function (err, data) { if (err) { reject(err); return; } resolve(data); }); }); //=> Promise ``` ### API `pinkie` exports bare [ES2015 Promise](https://people.mozilla.org/~jorendorff/es6-draft.html#sec-promise-objects) implementation and polyfills [Node.js rejection events](https://nodejs.org/api/process.html#process\_event\_unhandledrejection). In case you forgot: #### new Promise(executor) Returns new instance of `Promise`. ##### executor \*Required\* Type: `function` Function with two arguments `resolve` and `reject`. The first argument fulfills the promise, the second argument rejects it. #### pinkie.all(promises) Returns a promise that resolves when all of the promises in the `promises` Array argument have resolved. #### pinkie.race(promises) Returns a promise that resolves or rejects as soon as one of the promises in the `promises` Array resolves or rejects, with the value or reason from that promise. #### pinkie.reject(reason) Returns a Promise object that is rejected with the given `reason`. #### pinkie.resolve(value) Returns a Promise object that is resolved with the given `value`. If the `value` is a thenable (i.e. has a then method), the returned promise will "follow" that thenable, adopting its eventual state; otherwise the returned promise will be fulfilled with the `value`. ## Related - [pinkie-promise](https://github.com/floatdrop/pinkie-promise) - Returns the native Promise or this module ## License MIT © [Vsevolod Strukchinsky](http://github.com/floatdrop)