# uid-safe

[NPM Version](https://npmjs.org/package/uid-safe) [NPM Downloads](https://npmjs.org/package/uid-safe) [Node.js Version](https://nodejs.org/en/download/) [Build Status](https://travis-ci.org/crypto-utils/uid-safe) [Test Coverage](https://coveralls.io/r/crypto-utils/uid-safe?branch=master)

URL and cookie safe UIDs

Create cryptographically secure UIDs safe for both cookie and URL usage. This is in contrast to modules such as [rand-token](https://www.npmjs.com/package/rand-token) and [uid2](https://www.npmjs.com/package/uid2) whose UIDs are actually skewed due to the use of % and unnecessarily truncate the UID. Use this if you could still use UIDs with - and \_ in them.

## Installation

$ npm install uid-safe

## API

var uid = require('uid-safe')

### uid(byteLength, callback)

Asynchronously create a UID with a specific byte length. Because base64 encoding is used underneath, this is not the string length. For example, to create a UID of length 24, you want a byte length of 18.

uid(18, function (err, string) {

if (err) throw err

// do something with the string

})

### uid(byteLength)

Asynchronously create a UID with a specific byte length and return a Promise.

**Note**: To use promises in Node.js *prior to 0.12*, promises must be "polyfilled" using global.Promise = require('bluebird').

uid(18).then(function (string) {

// do something with the string

})

### uid.sync(byteLength)

A synchronous version of above.

var string = uid.sync(18)

## License

[MIT](http://license)