

— MODERNIZING —

WINSTON

— FOR NODE@4 —

AN EXERCISE IN OPEN SOURCE, STREAMS, ES6, AND
KEEPING CODE ALIVE.

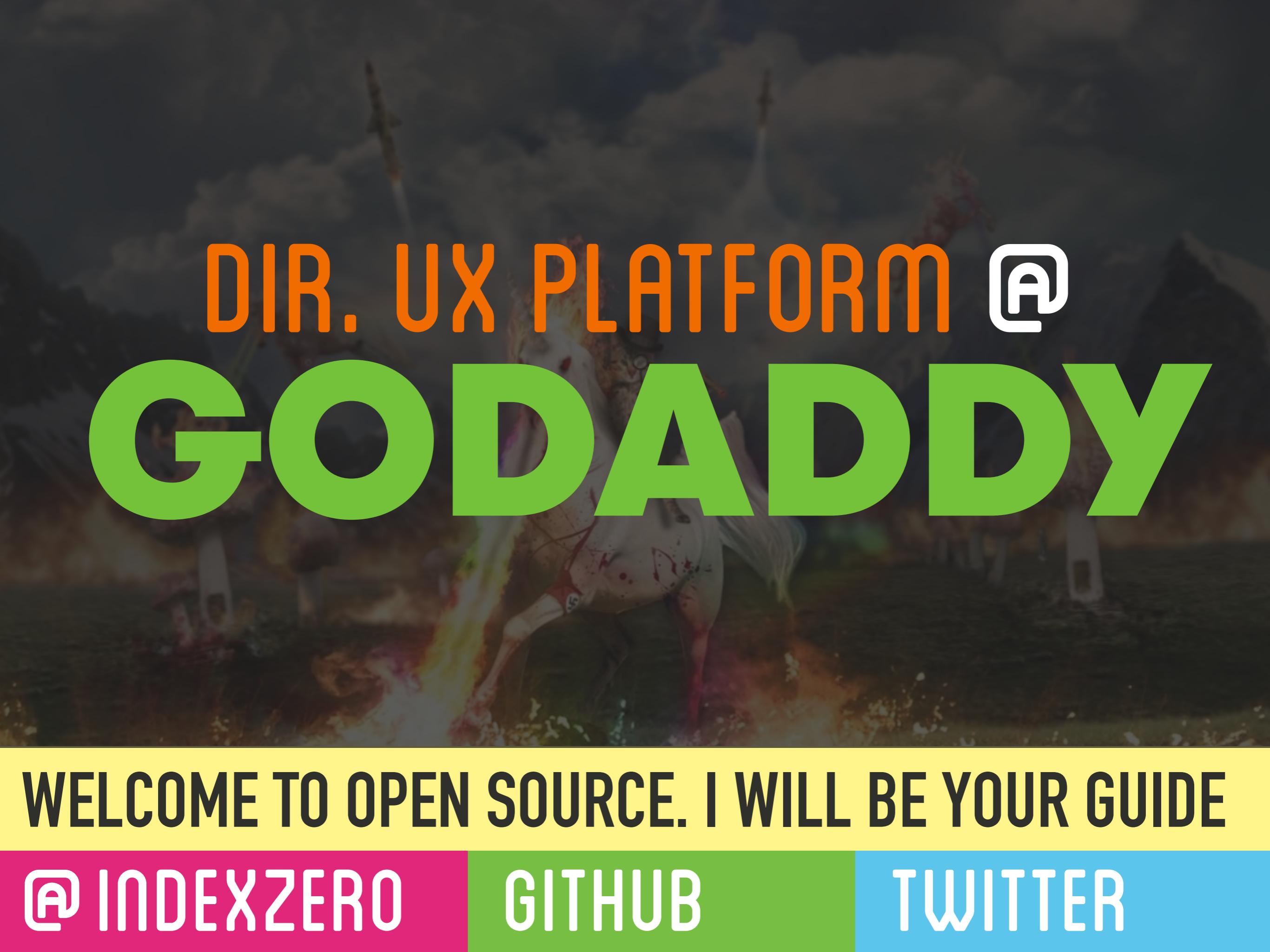


WHY, HELLO THERE



WELCOME TO OPEN SOURCE. I WILL BE YOUR GUIDE

@INDEXZERO



DIR. UX PLATFORM @
GODADDY

WELCOME TO OPEN SOURCE. I WILL BE YOUR GUIDE

@INDEXZERO

GITHUB

TWITTER



Easy
Carrot
Reci



Now. Let me tell you
A STORY...





A LONG TIME AGO IN A GALAXY FAR
FAR AWAY...

STAR
WARS

— ACTUALLY —

IT'S A STORY

— ABOUT —

OPEN SOURCE



nodejitsu

Intelligent, scalable Node.js clouds.
Host applications. Keep them up.

extended
private beta

THIS SUMMER

Claim your username and sign up for our invite list:

username

& e-mail address

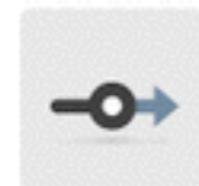
Sign up

Learn more...



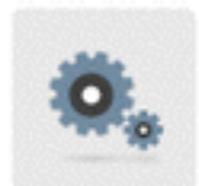
[The platform](#) »

Reliable Node.js cloud
services & technology.



[How to get started](#) »

Build your applications,
deploy and keep them up.



[Technology](#) »

Powered by open source
software & community.

Home Docs Products API Examples

Sign in

NODEJITSU



— STARTED IN —

Intelligent, scalable Node.js clouds.

Host applications. Keep them up.

MID 2010

— AND BACK THEN —

Learn more...

NODE.JS WAS LIKE



A car is driving on a dirt road in a desert landscape. In the background, there are large mesas and buttes under a clear blue sky. The car is positioned in the lower half of the frame, moving from left to right.

THE WILD WEST



THE WILD WEST

A black and white photograph of a car driving on a dirt road in a desert landscape. In the background, there are large mesas and buttes under a cloudy sky. The car is positioned in the lower half of the frame, angled towards the left.

BUT REALLY...

A large, dimly lit warehouse with high ceilings and metal beams. The floor is covered with numerous stacks of cardboard boxes, neatly piled in rows. The lighting is coming from overhead, creating a dramatic play of light and shadow on the boxes.

THE
MODULES
WE NEEDED WERE LIKE...

404 NOT FOUND



404 NOT FOUND

SO, LIKE MANY OTHERS

WE WROTE THEM

OURSELVES



— CERTAINLY ONE OF —
THE BEST FEELINGS
— EVER. FOR SERIOUS.

The text is overlaid on a dark, out-of-focus background image of a cat's face and fur. The words 'CERTAINLY ONE OF' are in white, 'THE BEST FEELINGS' is in large yellow capital letters, and 'EVER. FOR SERIOUS.' is in large green capital letters. A small white circle is positioned on the horizontal line between 'THE BEST FEELINGS' and 'EVER. FOR SERIOUS.'

STAR **THE**
EMPIRE
STRIKES BACK WALT DISNEY



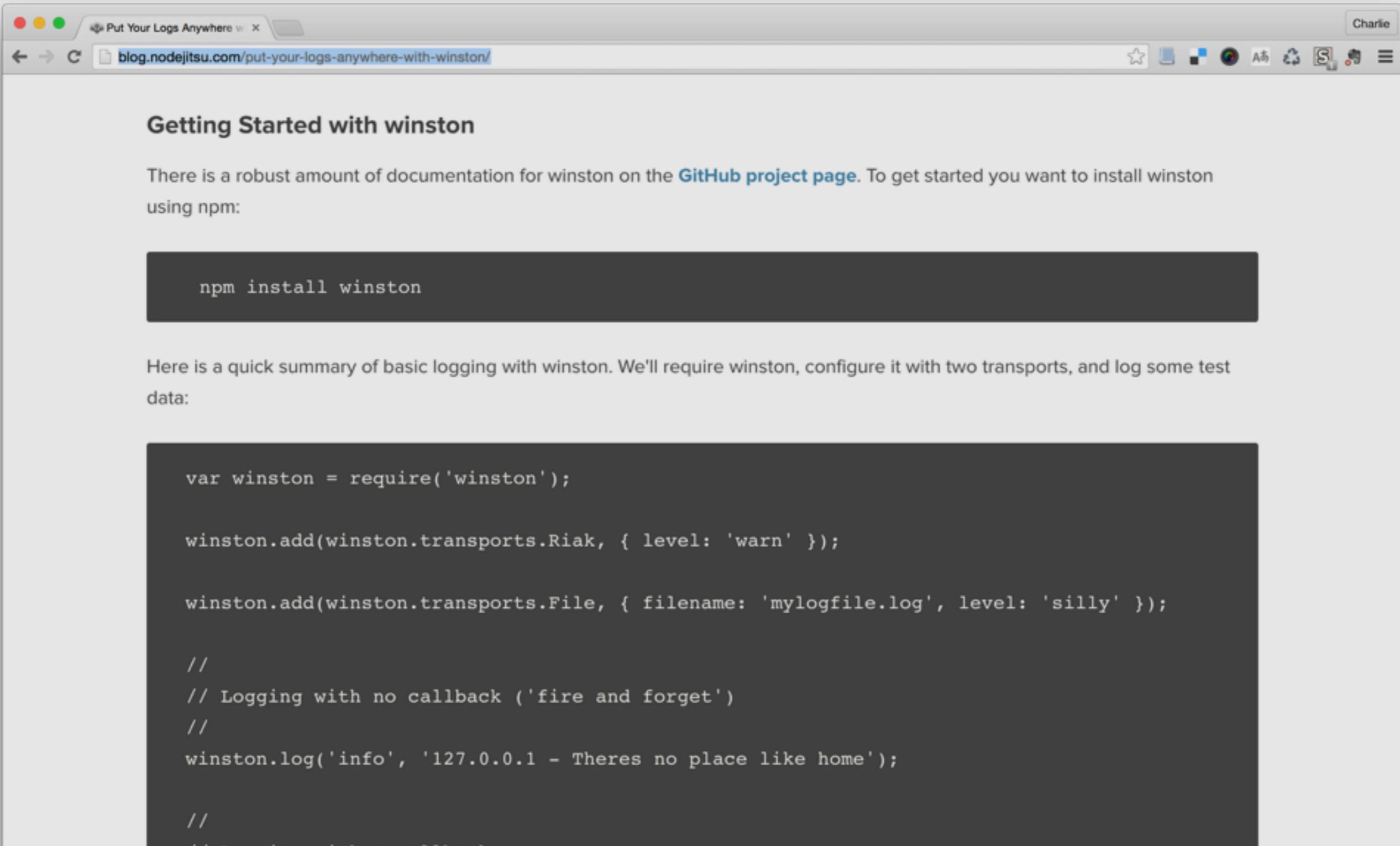


STAR THE
**EMPIRE
STRIKES BACK**

BUT

BACK TO OUR STORY...

REMEMBER THAT



The screenshot shows a web browser window with the following details:

- Address Bar:** blog.nodejitsu.com/put-your-logs-anywhere-with-winston/
- Page Title:** Put Your Logs Anywhere with Winston
- Section Header:** Getting Started with winston
- Text Content:** A paragraph explaining that there is a robust amount of documentation for winston on the [GitHub project page](#). It instructs the user to get started by installing winston using npm.
- Code Block:** A dark box containing the command `npm install winston`.
- Text Content (below code):** A quick summary of basic logging with winston, mentioning the requirement to install winston, configure it with two transports, and log some test data.
- Code Block (example code):** A dark box containing sample JavaScript code for setting up winston with two transports (Riak and File) and logging an info message.

THIS IS 2010



THIS IS 2010



node.conf

BEFORE THE FIRST ONE OF THESE

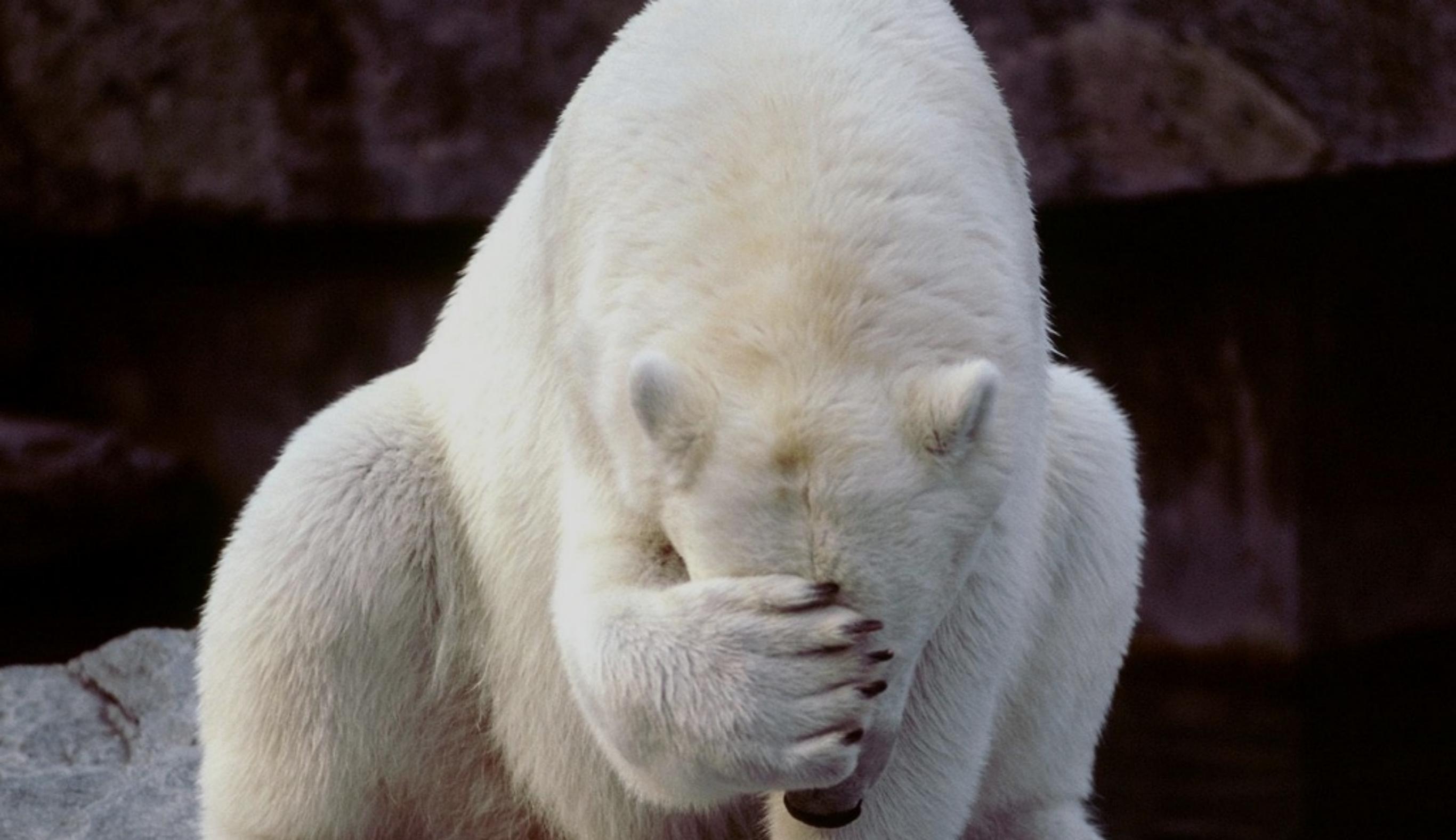




NO STREAMS API



ES5 IS STILL NEW

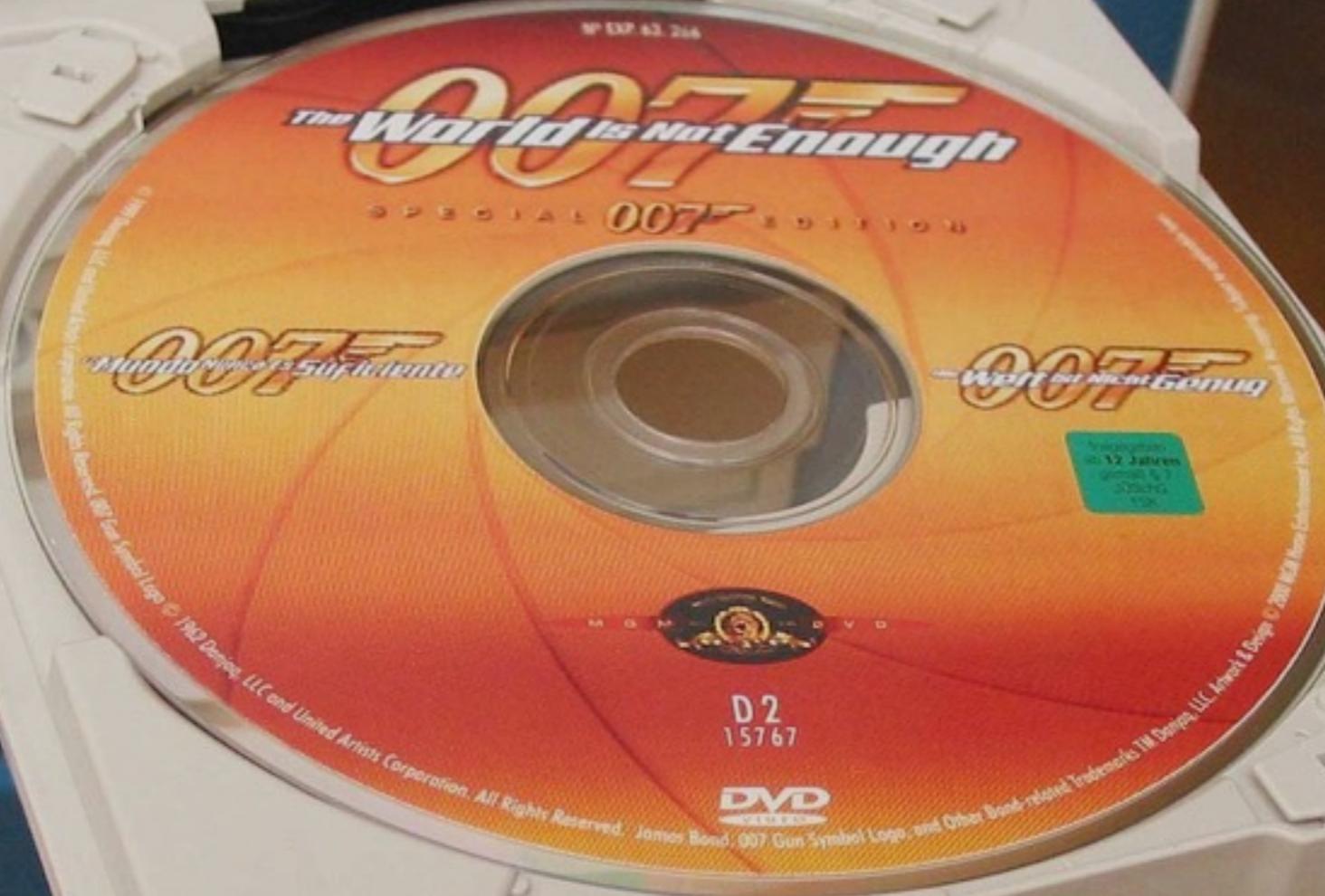


LIMITED CODE COVERAGE



BUT IT WAS STILL

A PRETTY GOOD API



A stack of James Bond 007 DVDs, with the top one clearly showing the title "The World is Not Enough" and the "DVD" logo.

ABOUT
2.6 MILLION
MONTHLY DOWNLOADS

BUNYAN: ~0.56M



— OVER —
700 PACKAGES
— IN THE —
**WINSTON
ECOSYSTEM**



A man and a woman are in a car. The man, on the left, is looking back over his shoulder with a shocked expression, his mouth slightly open. The woman, on the right, is looking towards the man. The interior of the car is visible, including the headrests and a red object in the center console.

THERE WERE
SOME
DARK TIMES

OVER 200+ OPEN ISSUES



OVER 200+ OPEN ISSUES

USER NEGATIVITY



OVER 200+ OPEN ISSUES

USER NEGATIVITY

IS THIS PROJECT
ABANDONED?



— IT TURNS OUT —

TAKING “ABUSE”



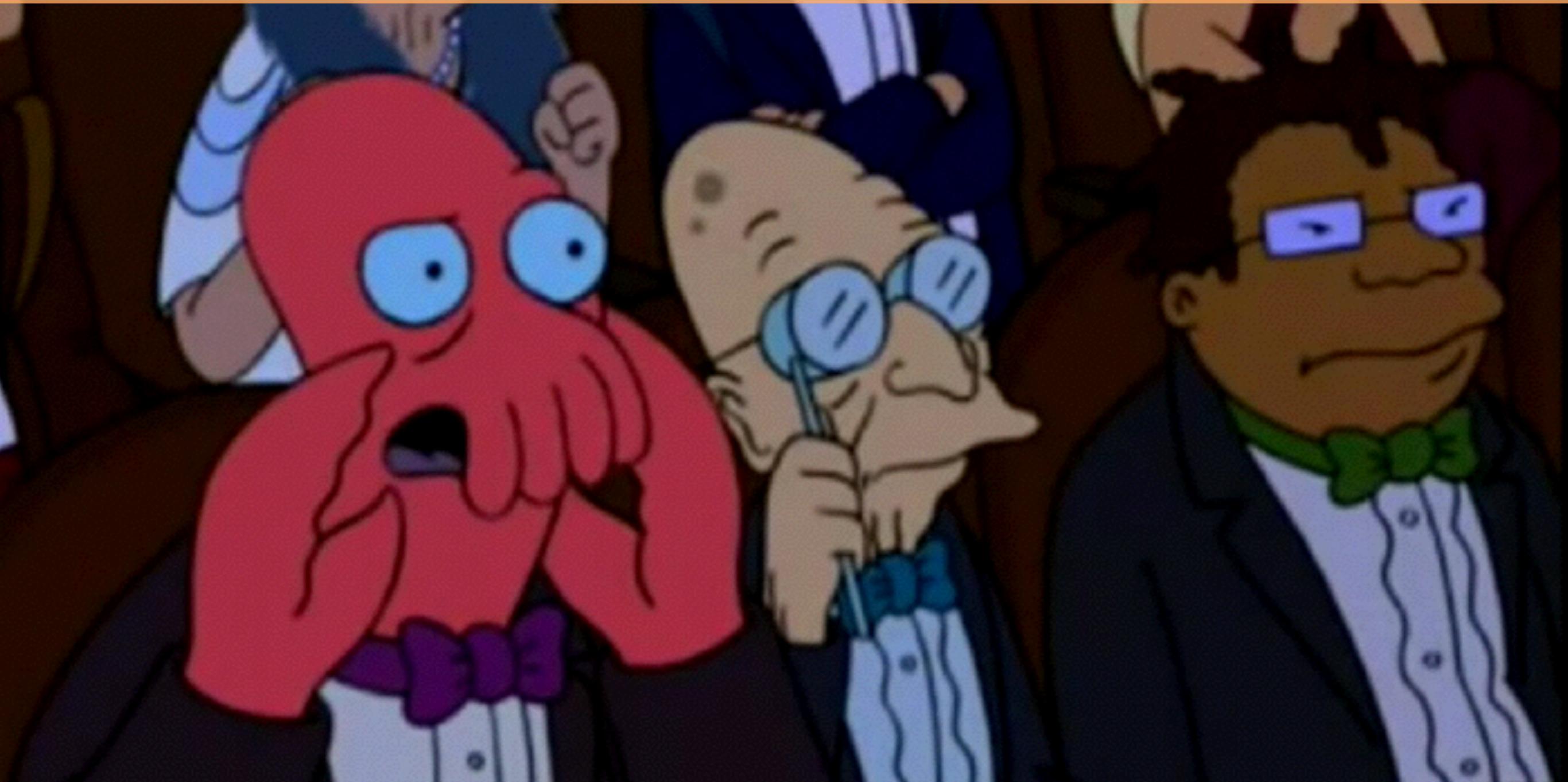
FROM YOUR USERS



JUST

TOTALLY SUCKS

YOUR MODULE'S BAD



AND YOU SHOULD FEEL BAD

YOUR MODULE'S BAD

TRUST ME

..... | DID

AND YOU SHOULD FEEL BAD



PLEASE. BE

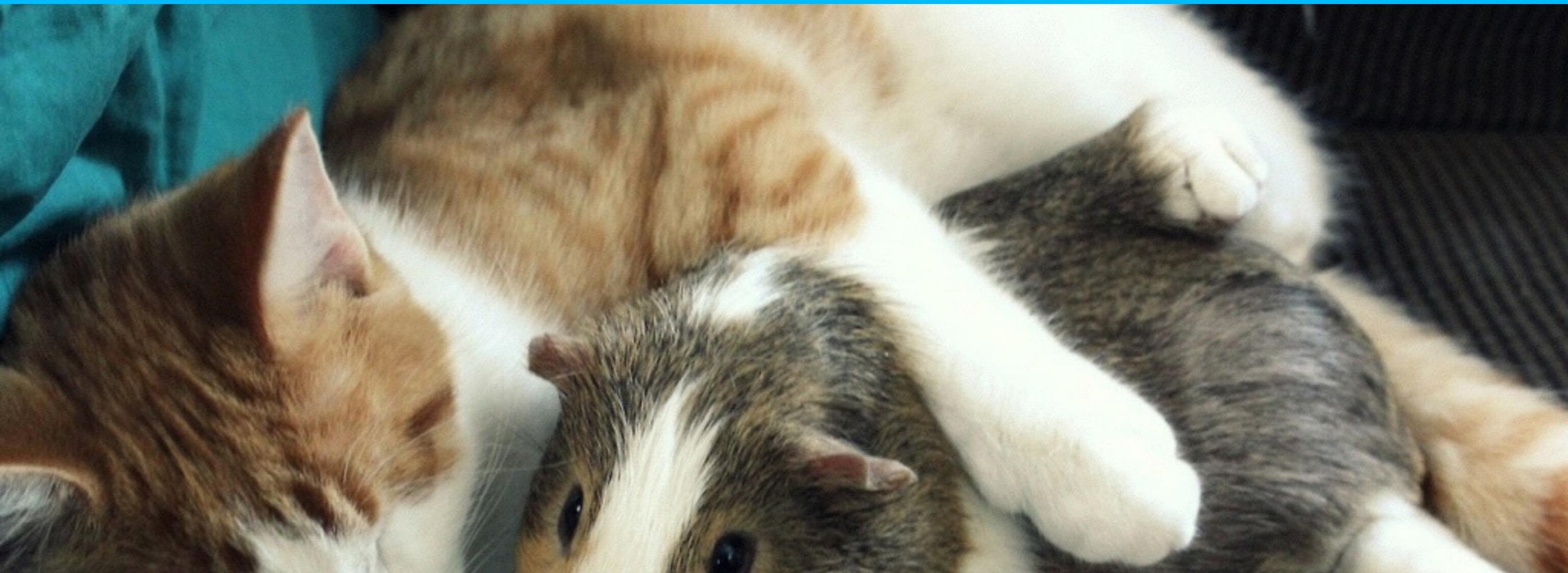
#\$%@!NG

NICE

CULTIVATE PERSONAL INTERACTION



CULTIVATE PERSONAL INTERACTION



WHEN IN DOUBT, FOLLOW-UP PRIVATELY

THINK OF IT AS DISCRETION.

... OR HAVING SOME CLASS.

THE SAGA CONTINUES

STAR WARS

RETURN OF THE JEDI

U



Starring

MARK HAMILL • HARRISON FORD • CARRIE FISHER
BILLY-DEE WILLIAMS • ANTHONY DANIELS

Co-starring DAVID PROWSE • KENNY BAKER • PETER MAYHEW • FRANK OZ

Directed by RICHARD MARQUAND Produced by HOWARD KAZANJIAN
Story by GEORGE LUCAS Screenplay by LAWRENCE KASDAN and GEORGE LUCAS
Executive Producer GEORGE LUCAS Music by JOHN WILLIAMS



20TH CENTURY FOX FILM CORPORATION
© 1983 Lucasfilm Ltd. All Rights Reserved

DOLBY STEREO
IN SELECTED THEATRES

Original Soundtrack by JOHN WILLIAMS
A Lucasfilm Ltd. Production • A Twentieth Century Fox Release

IN COLOUR

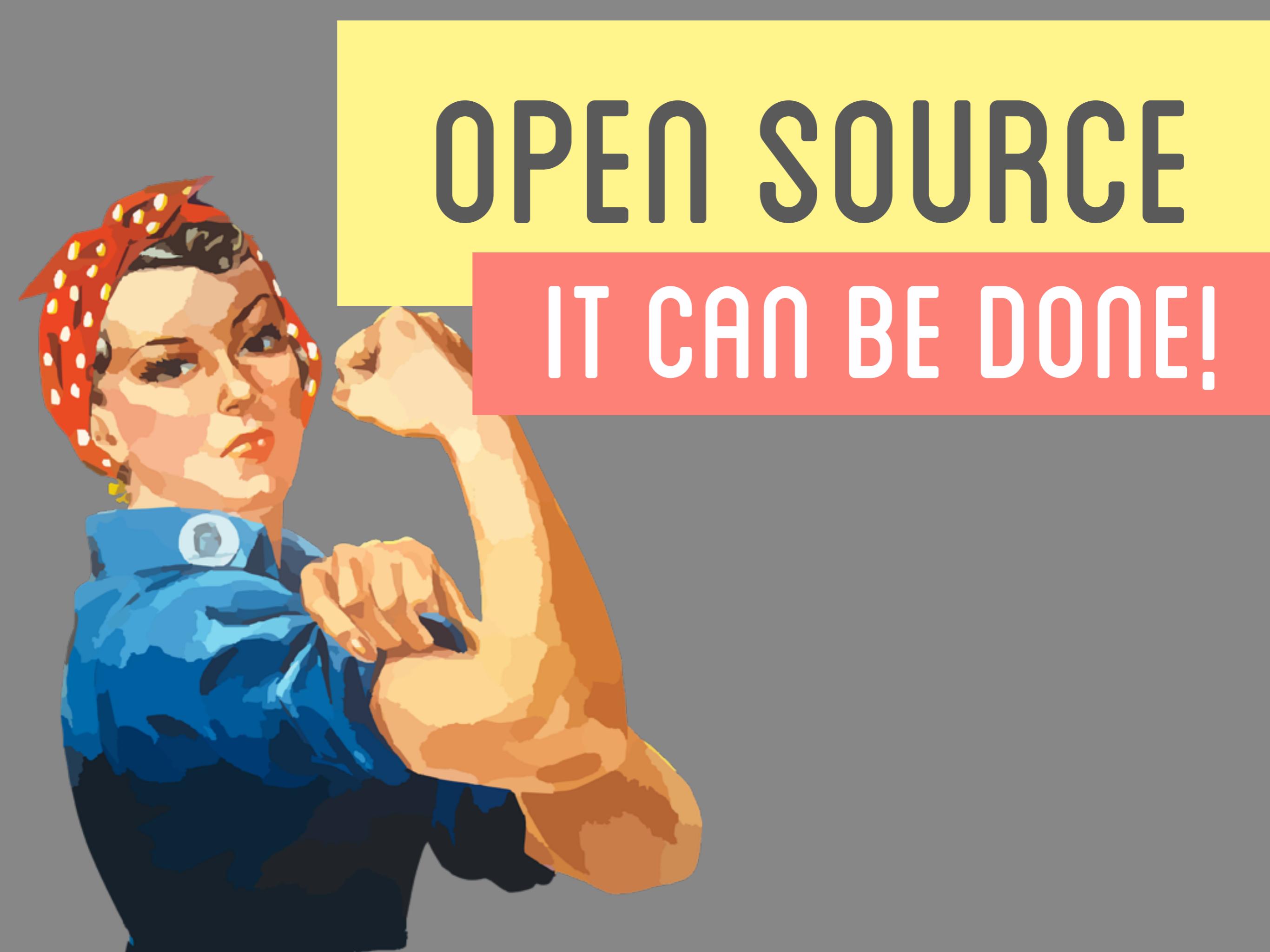
Revised through 20th Century Fox

ONCE AGAIN



BACK TO OUR STORY...

AFTER CAREFUL THOUGHT, MEDITATION,
AND THE IOJS RELEASE I DECIDED...



OPEN SOURCE
IT CAN BE DONE!



YOU ARE PROBABLY THINKING

BUT HOW?



— SCIENCE CAT SAYS —
“GOOD QUESTION”



WRITING CODE

— IS —

ABSOLUTELY NOT

THE FIRST ANSWER



DO HAVE

A LONG TERM

PROJECT GOAL

FIND ANYONE

— WHO IS —

WILLING TO HELP

— AND START —

WEEKLY MEETINGS



YES! WE WANT YOU!

TO JOIN WINSTON

TRIAGE ALL THE THINGS!



SIGNAL CORPS

TRIAGE ALL THE THINGS!

ISSUES, FIRST.

THEN PULL REQUESTS.







FIND
— THE —
HIGH LEVEL PROBLEMS
— AND —
TRIAGE INTO BUCKETS

NEXT-PATCH

NEXT-MINOR

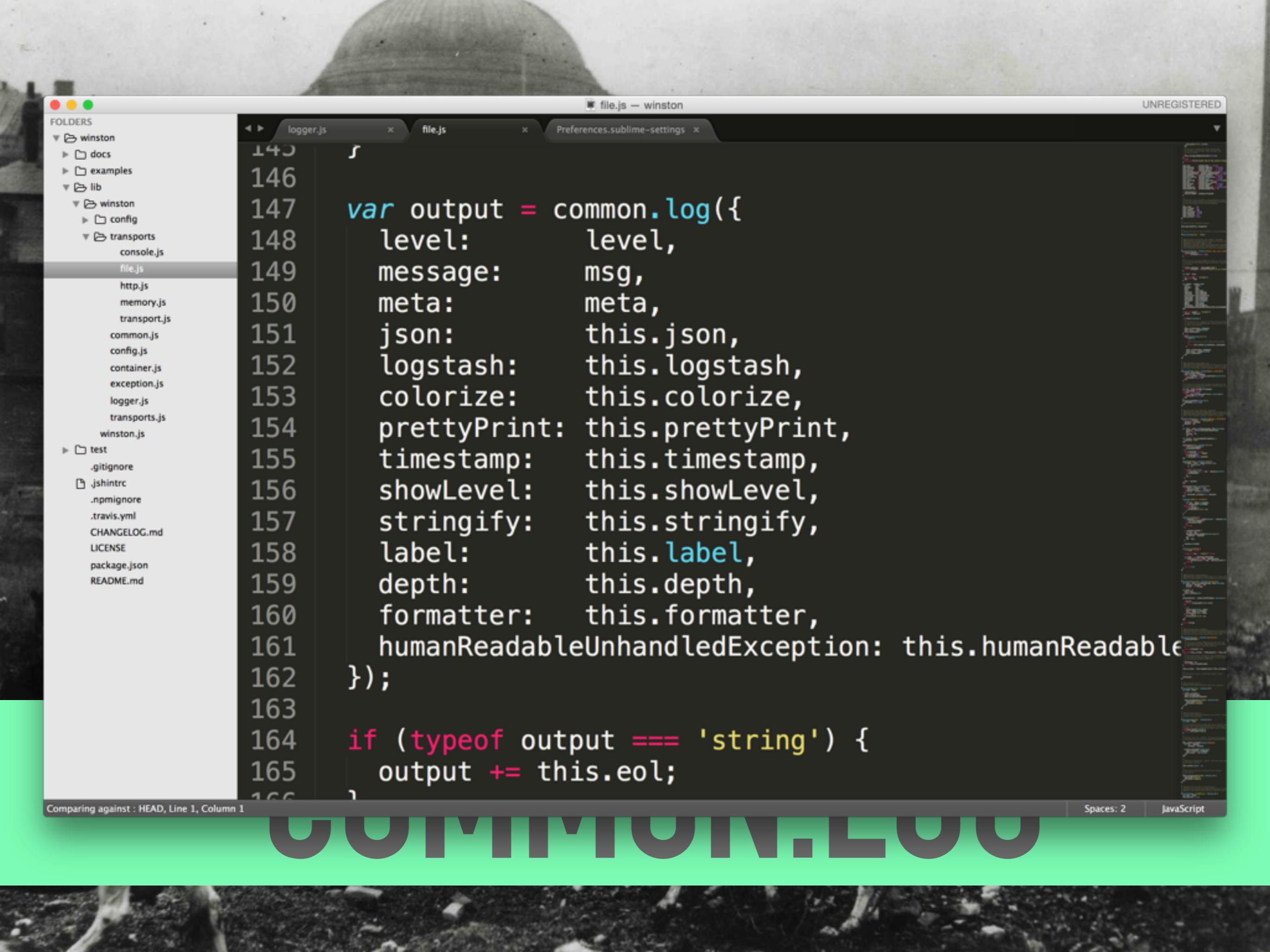
NEXT-MAJOR



REVERSED LEVELS



COMMON.LOG

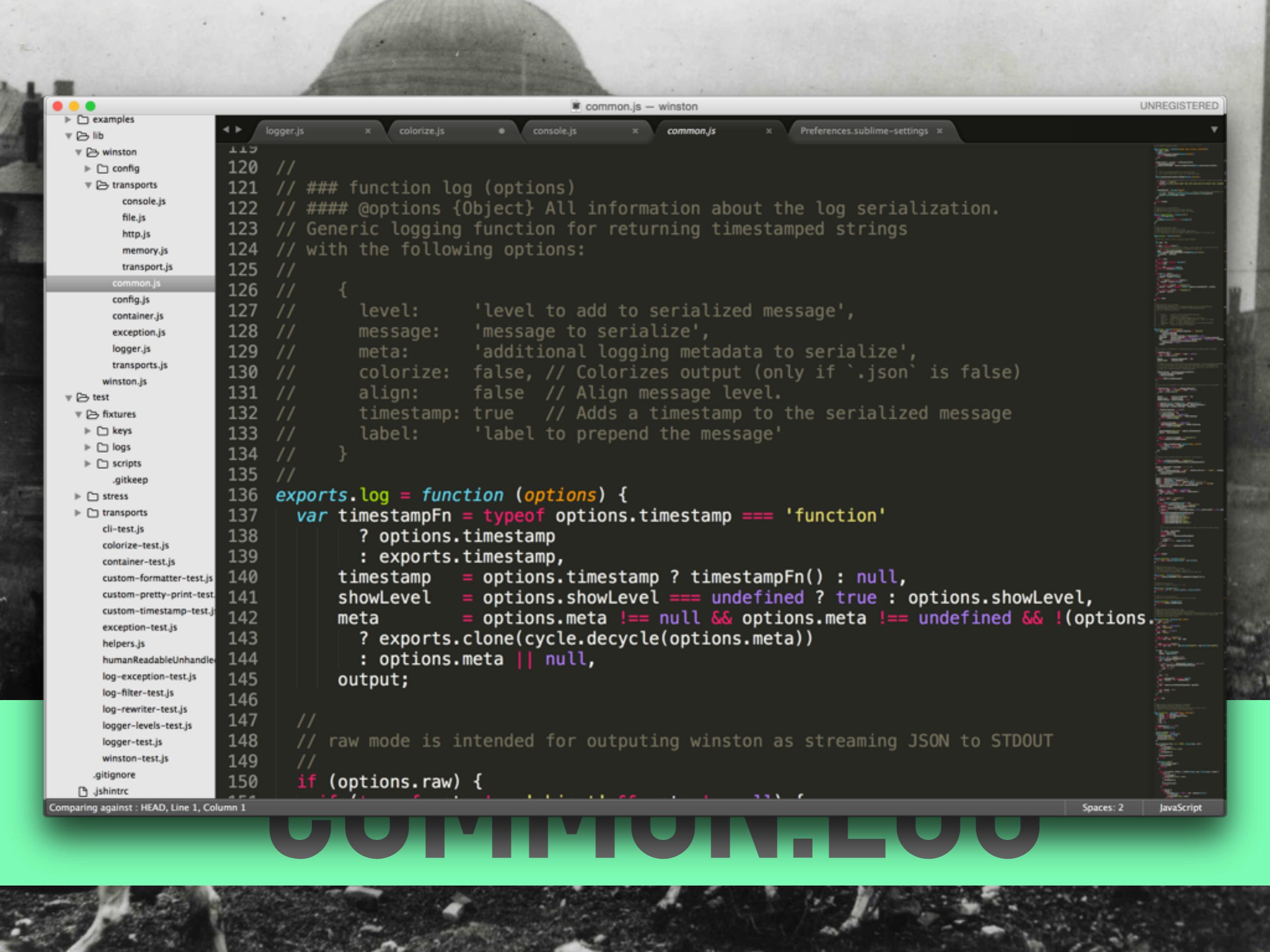


A close-up of a person's face, with their eyes closed, serves as the background for a Sublime Text window. The window shows a file named 'file.js' with the title 'file.js — winston'. The code in the file is a constructor for a logger object, using commonjs module syntax. The code is as follows:

```
145
146
147 var output = common.log({
148     level:           level,
149     message:        msg,
150     meta:           meta,
151     json:           this.json,
152     logstash:       this.logstash,
153     colorize:       this.colorize,
154     prettyPrint:    this.prettyPrint,
155     timestamp:      this.timestamp,
156     showLevel:      this.showLevel,
157     stringify:     this.stringify,
158     label:          this.label,
159     depth:          this.depth,
160     formatter:      this.formatter,
161     humanReadableUnhandledException: this.humanReadableUnhandledException
162 });
163
164 if (typeof output === 'string') {
165     output += this.eol;
166 }
```

The Sublime Text interface includes a sidebar with a 'FOLDERS' list containing 'winston', 'docs', 'examples', 'lib', 'test', and several configuration files. The status bar at the bottom shows 'Comparing against : HEAD, Line 1, Column 1', 'Spaces: 2', and 'JavaScript'.

COMMITTON.LU



A screenshot of a Sublime Text editor window showing the file `common.js` from the `winston` library. The editor has a dark theme and displays code in a light-colored font. The code is a JavaScript function that handles logging options. The file is part of a larger project structure visible in the sidebar.

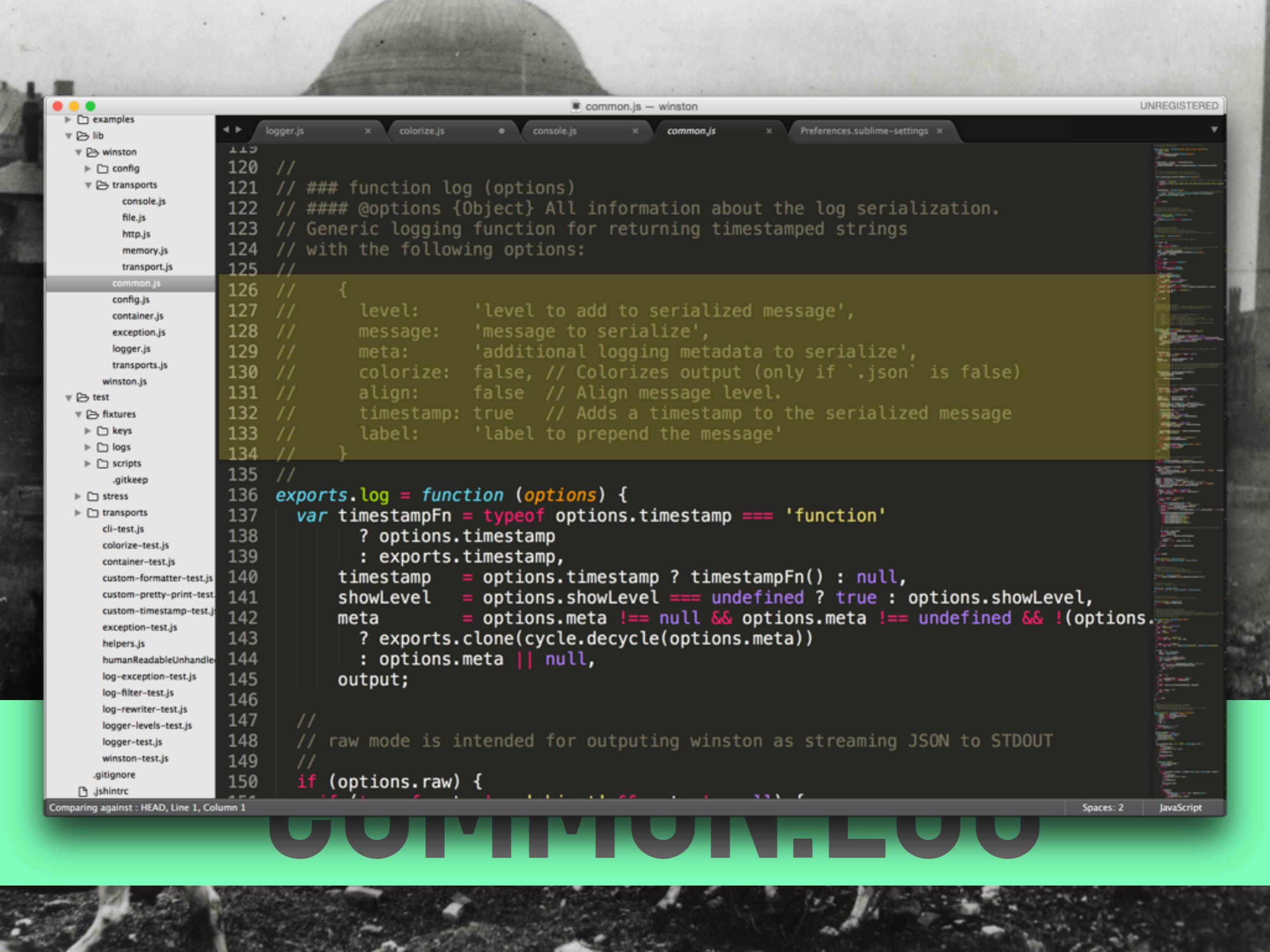
```
common.js — winston UNREGISTERED

examples
lib
  winston
    config
    transports
      console.js
      file.js
      http.js
      memory.js
      transport.js
  common.js
  config.js
  container.js
  exception.js
  logger.js
  transports.js
  winston.js
test
  fixtures
    keys
    logs
    scripts
    .gitkeep
stress
transports
  cli-test.js
  colorize-test.js
  container-test.js
  custom-formatter-test.js
  custom-pretty-print-test
  custom-timestamp-test.js
  exception-test.js
  helpers.js
  humanReadableUnhandle
  log-exception-test.js
  log-filter-test.js
  log-rewriter-test.js
  logger-levels-test.js
  logger-test.js
  winston-test.js
.gitignore
jshintrc

Comparing against : HEAD, Line 1, Column 1
Spaces: 2
JavaScript
```

```
120  //
121  // ### function log (options)
122  // ##### @options {Object} All information about the log serialization.
123  // Generic logging function for returning timestamped strings
124  // with the following options:
125  //
126  // {
127  //   level:      'level to add to serialized message',
128  //   message:    'message to serialize',
129  //   meta:       'additional logging metadata to serialize',
130  //   colorize:   false, // Colorizes output (only if `json` is false)
131  //   align:      false // Align message level.
132  //   timestamp: true // Adds a timestamp to the serialized message
133  //   label:      'label to prepend the message'
134  // }
135  //
136  exports.log = function (options) {
137    var timestampFn = typeof options.timestamp === 'function'
138      ? options.timestamp
139      : exports.timestamp,
140    timestamp = options.timestamp ? timestampFn() : null,
141    showLevel = options.showLevel === undefined ? true : options.showLevel,
142    meta      = options.meta !== null && options.meta !== undefined && !(options.
143      ? exports.clone(cycle.decycle(options.meta))
144      : options.meta || null),
145    output;
146
147  //
148  // raw mode is intended for outputing winston as streaming JSON to STDOUT
149  //
150  if (options.raw) {
```

COMMITON.EU



A person's face is visible in the background, looking directly at the camera. The laptop screen in the foreground displays a code editor with the file 'common.js' open. The code is a part of the Winston library, specifically the 'logger.js' module. The code defines a 'log' function that takes an 'options' object and returns a timestamped string. The options object includes fields for level, message, meta, colorize, align, timestamp, and label. The code uses exports.log = function (options) {} to define the function. The file is part of a larger project structure with subfolders like 'examples', 'lib', 'test', and 'stress'.

```
common.js — winston
UNREGISTERED

examples
lib
  winston
    config
    transports
      console.js
      file.js
      http.js
      memory.js
      transport.js
  common.js
  config.js
  container.js
  exception.js
  logger.js
  transports.js
  winston.js
test
  fixtures
    keys
    logs
    scripts
    .gitkeep
stress
transports
  cli-test.js
  colorize-test.js
  container-test.js
  custom-formatter-test.js
  custom-pretty-print-test
  custom-timestamp-test.js
  exception-test.js
  helpers.js
  humanReadableUnhandle
  log-exception-test.js
  log-filter-test.js
  log-rewriter-test.js
  logger-levels-test.js
  logger-test.js
  winston-test.js
.gitignore
jshintrc

120  //
121  // ### function log (options)
122  // ##### @options {Object} All information about the log serialization.
123  // Generic logging function for returning timestamped strings
124  // with the following options:
125  //
126  // {
127  //   level:      'level to add to serialized message',
128  //   message:    'message to serialize',
129  //   meta:       'additional logging metadata to serialize',
130  //   colorize:   false, // Colorizes output (only if '.json' is false)
131  //   align:      false // Align message level.
132  //   timestamp: true // Adds a timestamp to the serialized message
133  //   label:      'label to prepend the message'
134  // }
135  //
136  exports.log = function (options) {
137    var timestampFn = typeof options.timestamp === 'function'
138      ? options.timestamp
139      : exports.timestamp,
140    timestamp = options.timestamp ? timestampFn() : null,
141    showLevel = options.showLevel === undefined ? true : options.showLevel,
142    meta = options.meta !== null && options.meta !== undefined && !(options.
143      ? exports.clone(cycle.decycle(options.meta))
144      : options.meta || null,
145    output;
146
147  //
148  // raw mode is intended for outputing winston as streaming JSON to STDOUT
149  //
150  if (options.raw) {
```

Comparing against : HEAD, Line 1, Column 1

Spaces: 2

JavaScript

GOIMMUN.EU





HERE BE

DRAGONS



THEN YOU

CAN LAND YOUR PRS

THEN YOU

CAN LAND YOUR PRS

DO NOT REFACTOR THE THINGS YOU HATE
YET. FOR SERIOUS.

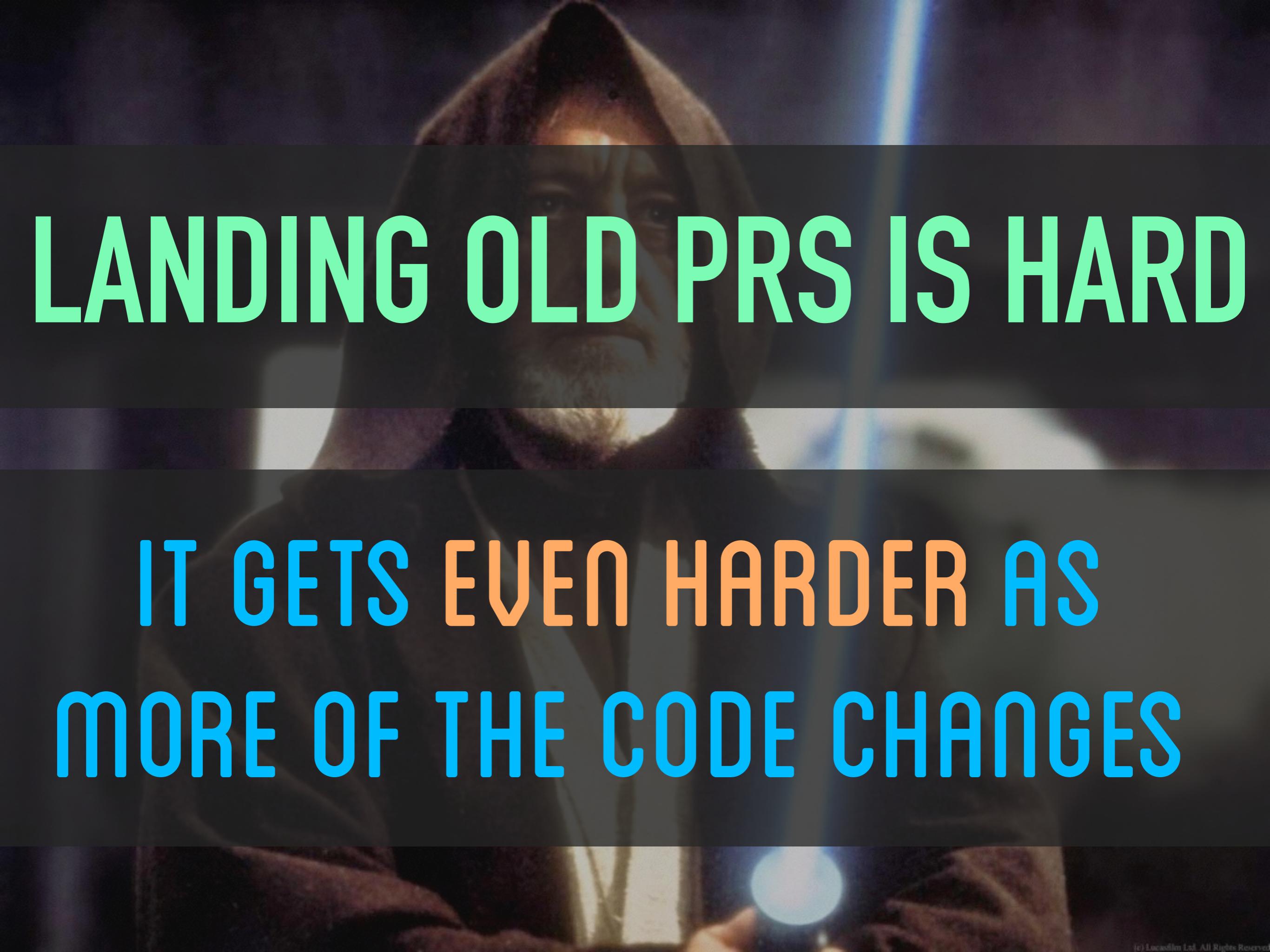


A woman with long brown hair, wearing a brown t-shirt, is shown from the chest up. She is looking down with her head slightly bowed, her right hand resting near her face. The background is a plain, light grey.

BUT I HATE THE THINGS
I HATE ABOUT
my code!

A close-up of a young Obi-Wan Kenobi's face, looking directly at the viewer with a serious expression. He is wearing a brown Jedi robe and holding a blue lightsaber hilt. The background is dark and out of focus.

PATIENCE



LANDING OLD PRS IS HARD

IT GETS EVEN HARDER AS
MORE OF THE CODE CHANGES



A close-up photograph of a man with light brown hair, wearing a dark grey suit jacket, a light blue dress shirt, and a maroon tie with diagonal stripes in blue and gold. He is looking down and to his right with a serious expression. His right hand is raised, holding a black pen. A large, semi-transparent dark red rectangle covers the upper half of the image, containing the text.

JUST ASK THEM TO RESUBMIT

A close-up photograph of a man with short brown hair, wearing a dark suit jacket, a white shirt, and a striped tie. He is looking down at a black smartphone held in his right hand. His expression is focused and slightly weary. The background is a plain, light-colored wall.

DEVELOPERS VERY RARELY COMMIT

THERE IS USUALLY
ONLY ONE CHANCE PER PR.





— PLEASE —

DO NOT BE

— THIS PERSON —

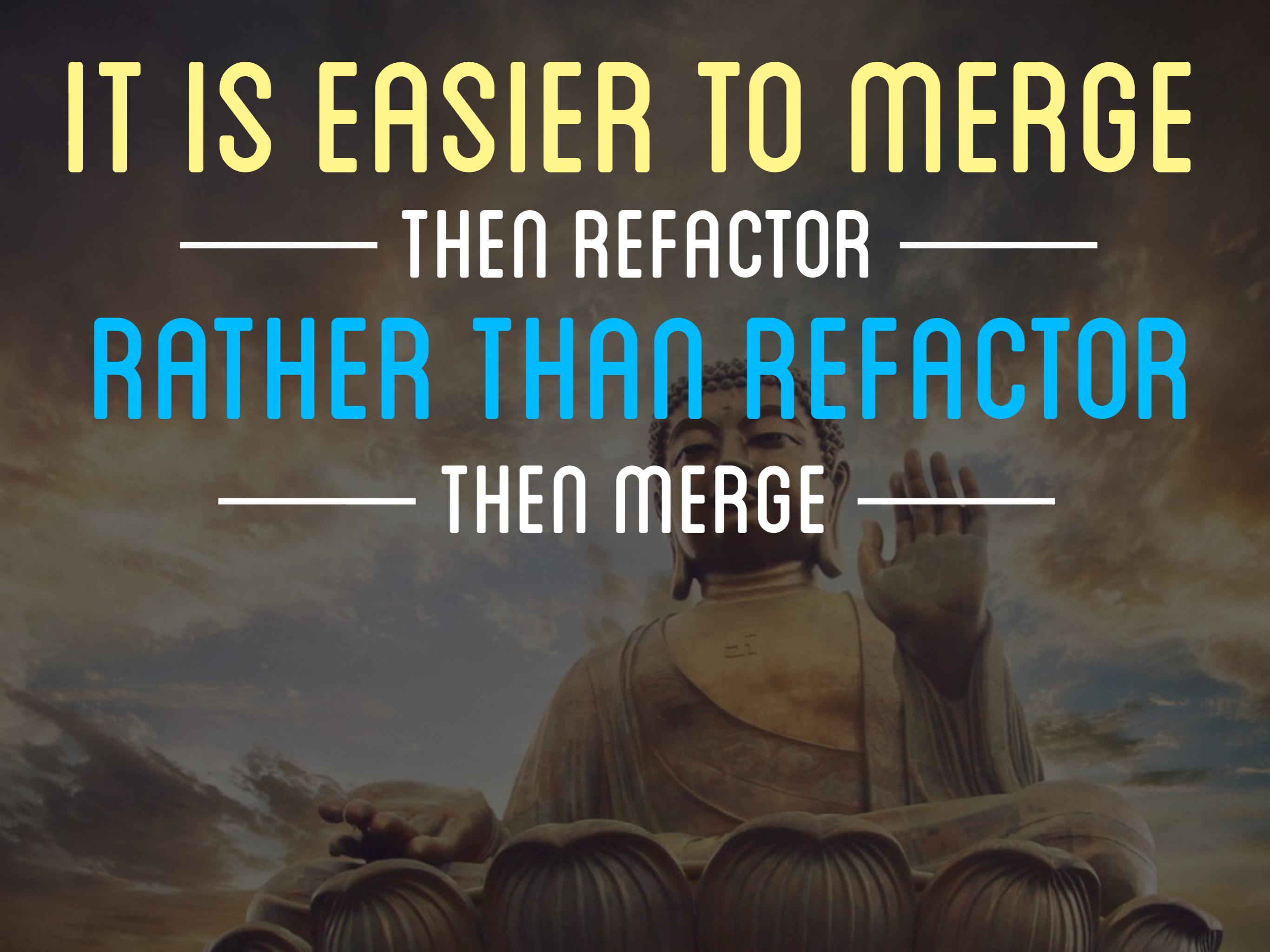
The text is overlaid on a background illustration of a man with glasses, looking at a computer screen with a shocked expression. The background shows a window with a city skyline and some foliage.

IT IS EASIER TO MERGE

— THEN REFACTOR —

RATHER THAN REFACTOR

— THEN MERGE —





A photograph of a person's hands resting on a keyboard. The hands are positioned as if the person is about to start playing. The lighting is low, with a warm glow from the screen of a laptop or monitor in the background, creating a focused and creative atmosphere.

NOW YOU CAN

—FOCUS—

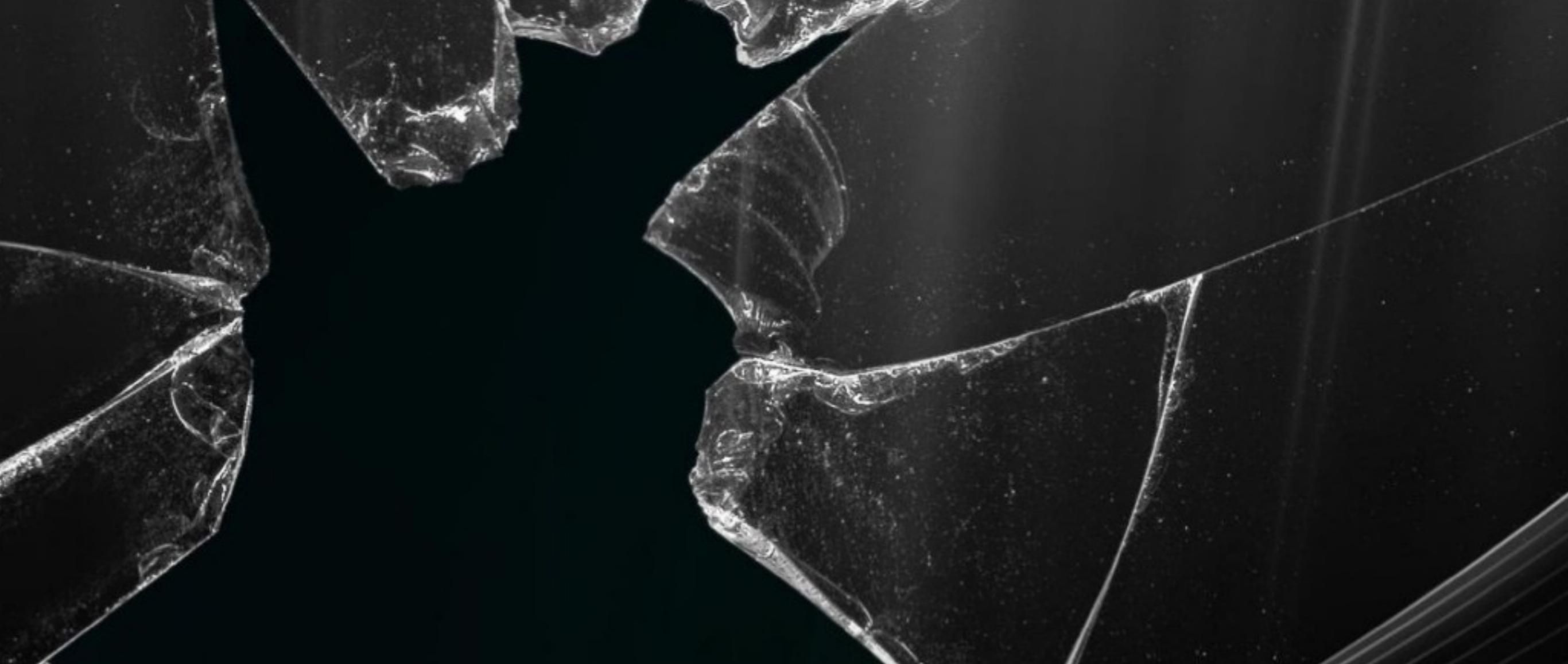
ON THAT LONG TERM

PROJECT GOAL

A dark, cracked screen serves as the background for the entire image. In the center, there is a bright, glowing blue energy core with a complex, branching pattern of light rays emanating from it, resembling a star or a neural network.

WINSTON@1.0.0

SMALL BREAKS



WINSTON@2.0.0

BIGGER BREAKS



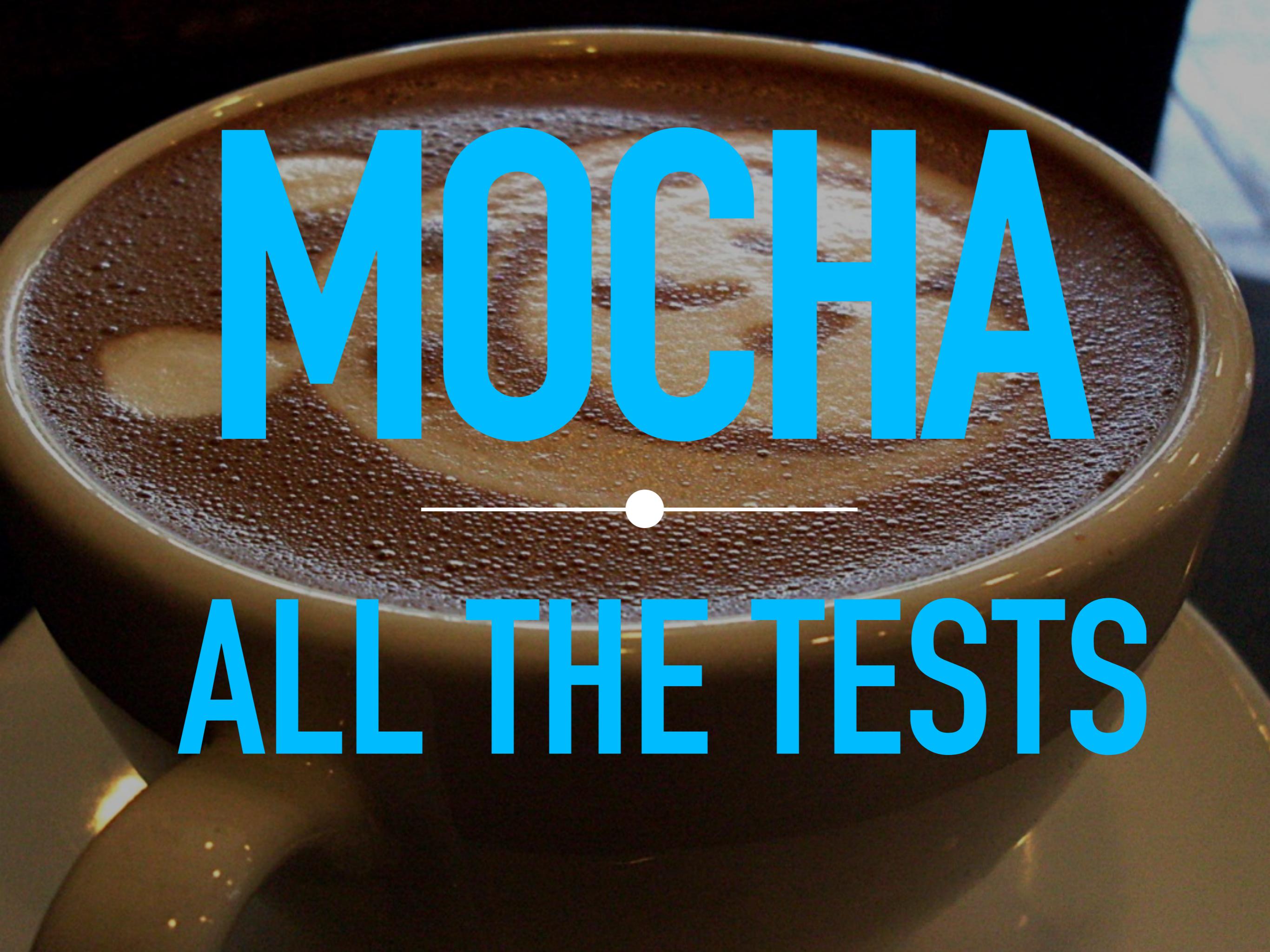
WINSTON@3.0.0

BREAK TO NEW APIS

A photograph of a lush green forest. In the foreground, a small stream flows over rocks, with a small waterfall visible in the middle ground. The background is filled with dense green foliage and trees.

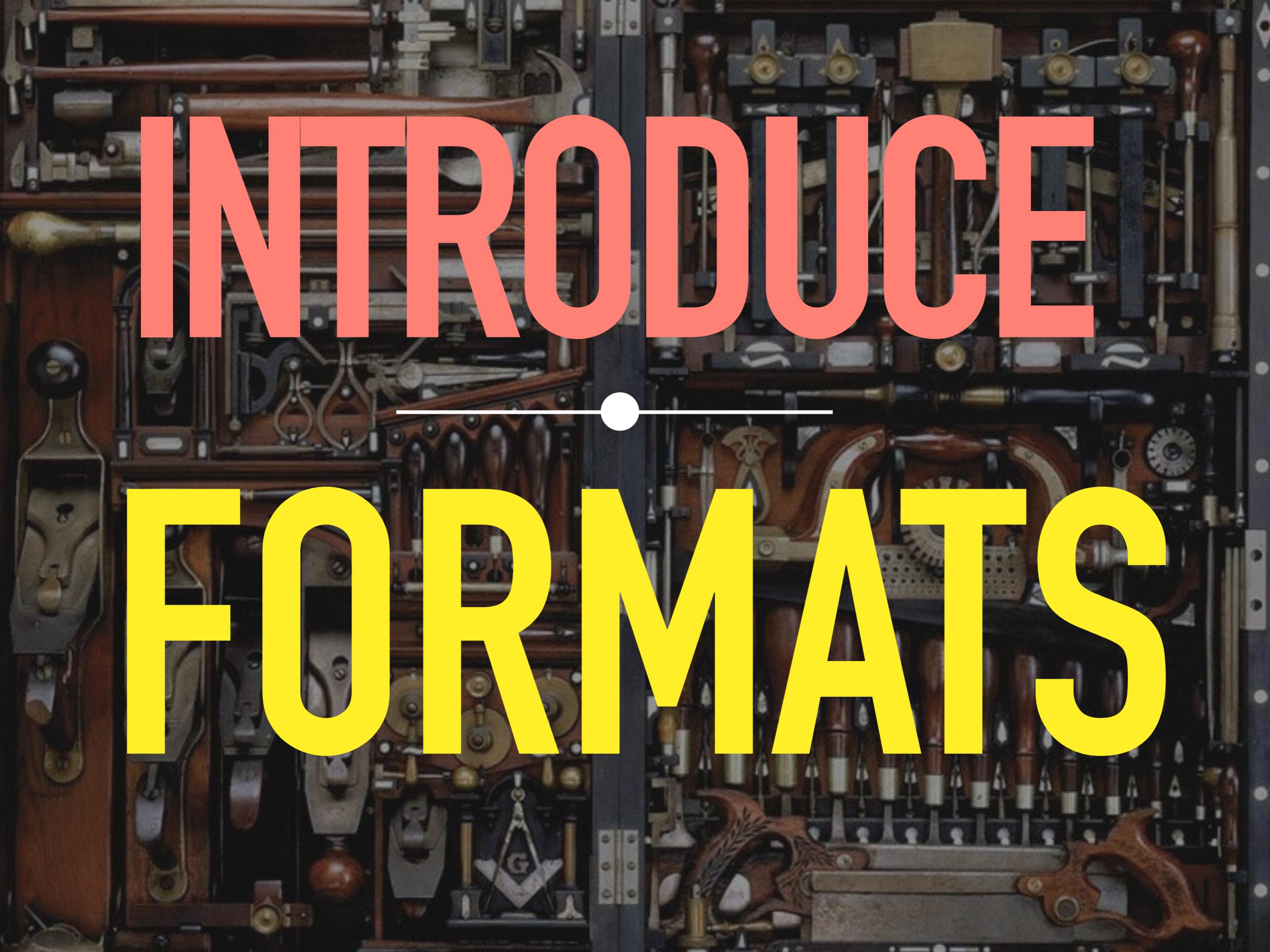
STREAMS

ALL THE THINGS

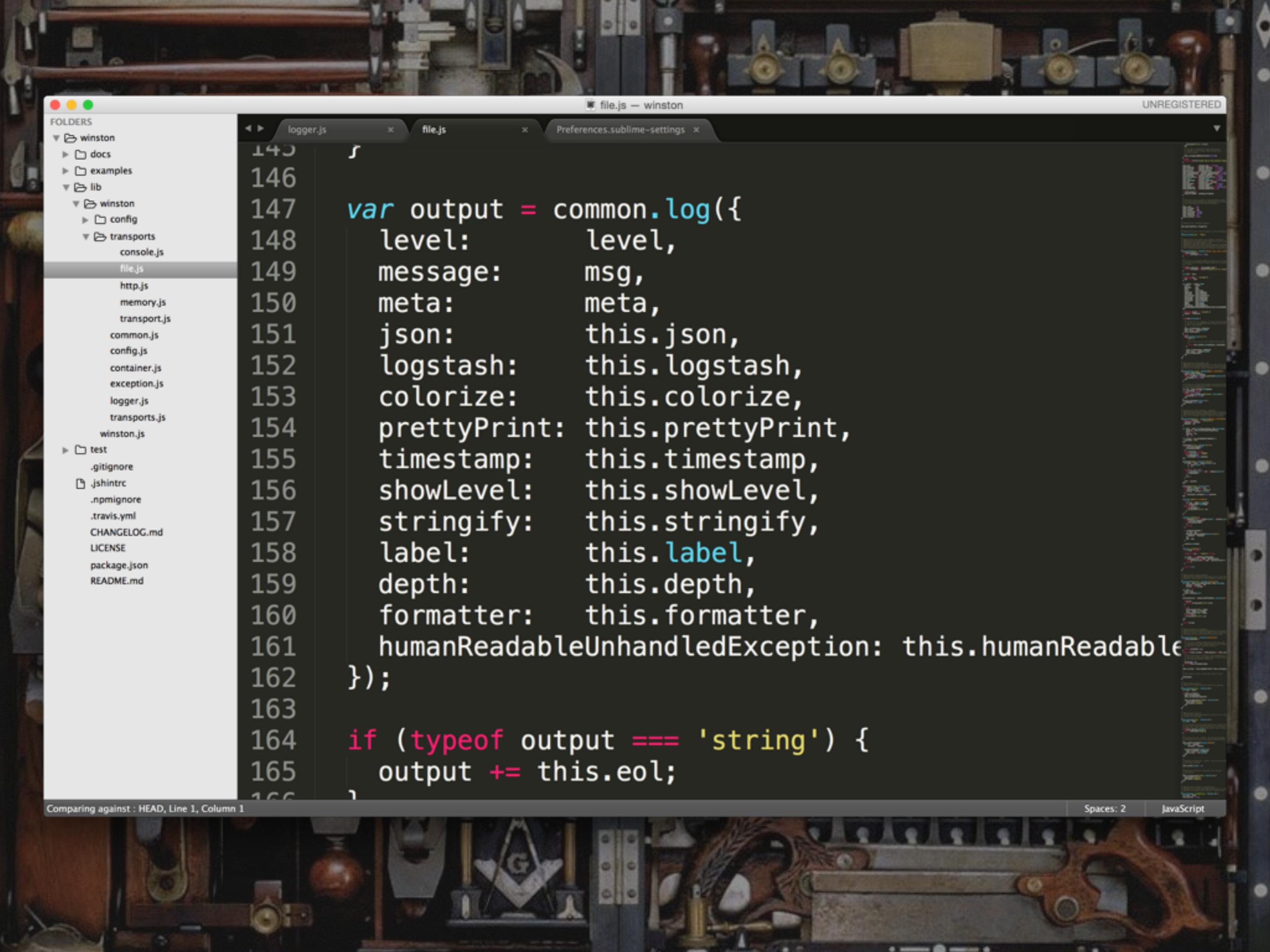


MOCHA

ALL THE TESTS



INTRODUCE FORMATS



file.js — winston

UNREGISTERED

FOLDERS

- winston
 - docs
 - examples
 - lib
 - winston
 - config
 - transports
 - console.js
 - file.js
 - http.js
 - memory.js
 - transport.js
 - common.js
 - config.js
 - container.js
 - exception.js
 - logger.js
 - transports.js
 - winston.js
 - test
 - .gitignore
 - .jshintrc
 - .npmignore
 - .travis.yml
 - CHANGELOG.md
 - LICENSE
 - package.json
 - README.md

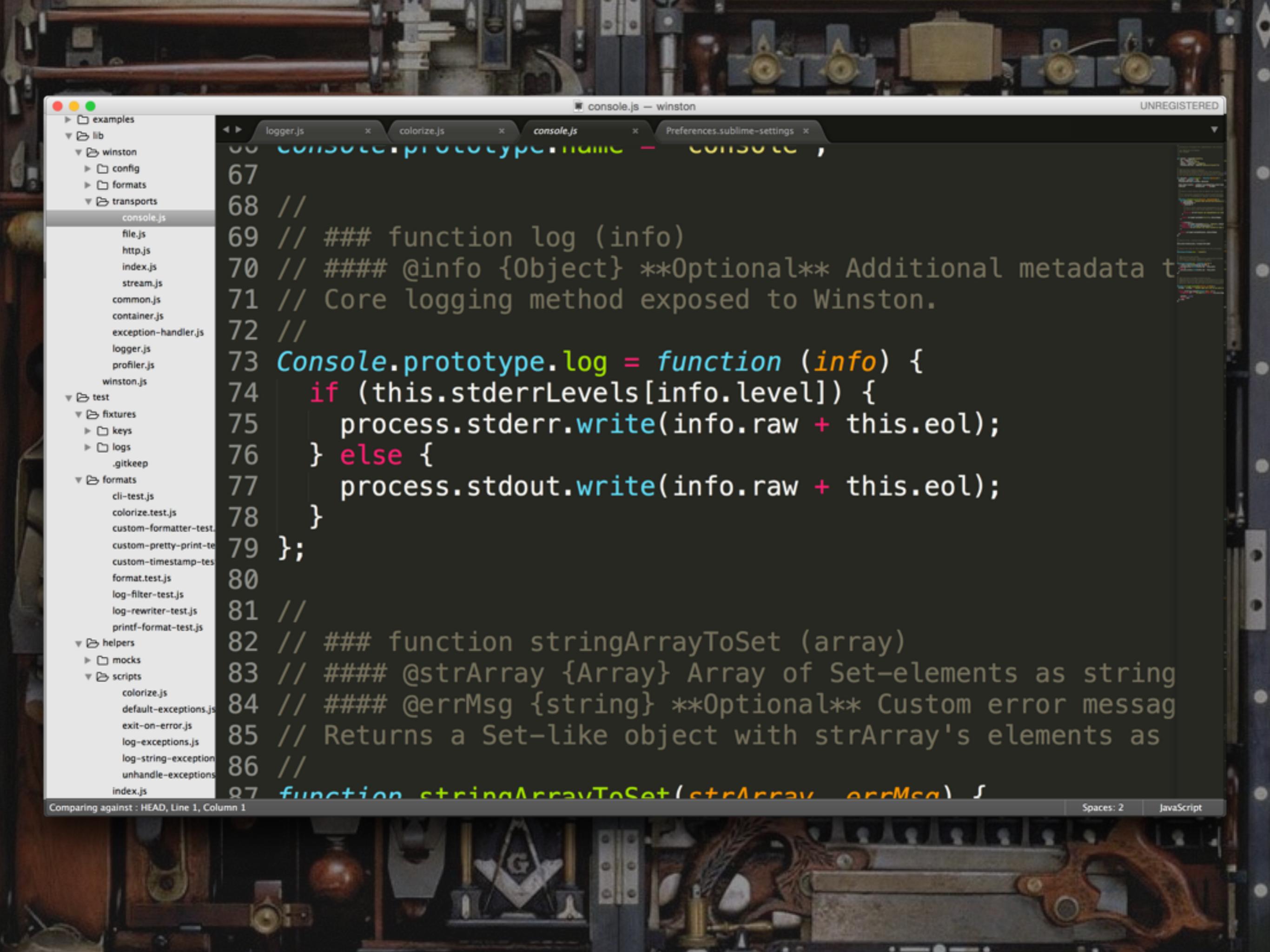
logger.js file.js Preferences.sublime-settings

```
145
146
147 var output = common.log({
148     level:           level,
149     message:         msg,
150     meta:            meta,
151     json:            this.json,
152     logstash:        this.logstash,
153     colorize:        this.colorize,
154     prettyPrint:     this.prettyPrint,
155     timestamp:       this.timestamp,
156     showLevel:        this.showLevel,
157     stringify:       this.stringify,
158     label:            this.label,
159     depth:            this.depth,
160     formatter:        this.formatter,
161     humanReadableUnhandledException: this.humanReadableUnhandledException
162 });
163
164 if (typeof output === 'string') {
165     output += this.eol;
166 }
```

Comparing against : HEAD, Line 1, Column 1

Spaces: 2

JavaScript



console.js — winston UNREGISTERED

```
examples
lib
  winston
    config
    formats
    transports
      console.js
        file.js
        http.js
        index.js
        stream.js
      common.js
      container.js
      exception-handler.js
      logger.js
      profiler.js
      winston.js
test
  fixtures
    keys
    logs
    .gitkeep
formats
  cli-test.js
  colorize.test.js
  custom-formatter-test.js
  custom-pretty-print-test.js
  custom-timestamp-test.js
  format-test.js
  log-filter-test.js
  log-rewriter-test.js
  printf-format-test.js
helpers
  mocks
  scripts
    colorize.js
    default-exceptions.js
    exit-on-error.js
    log-exceptions.js
    log-string-exception.js
    unhandle-exceptions.js
  index.js
```

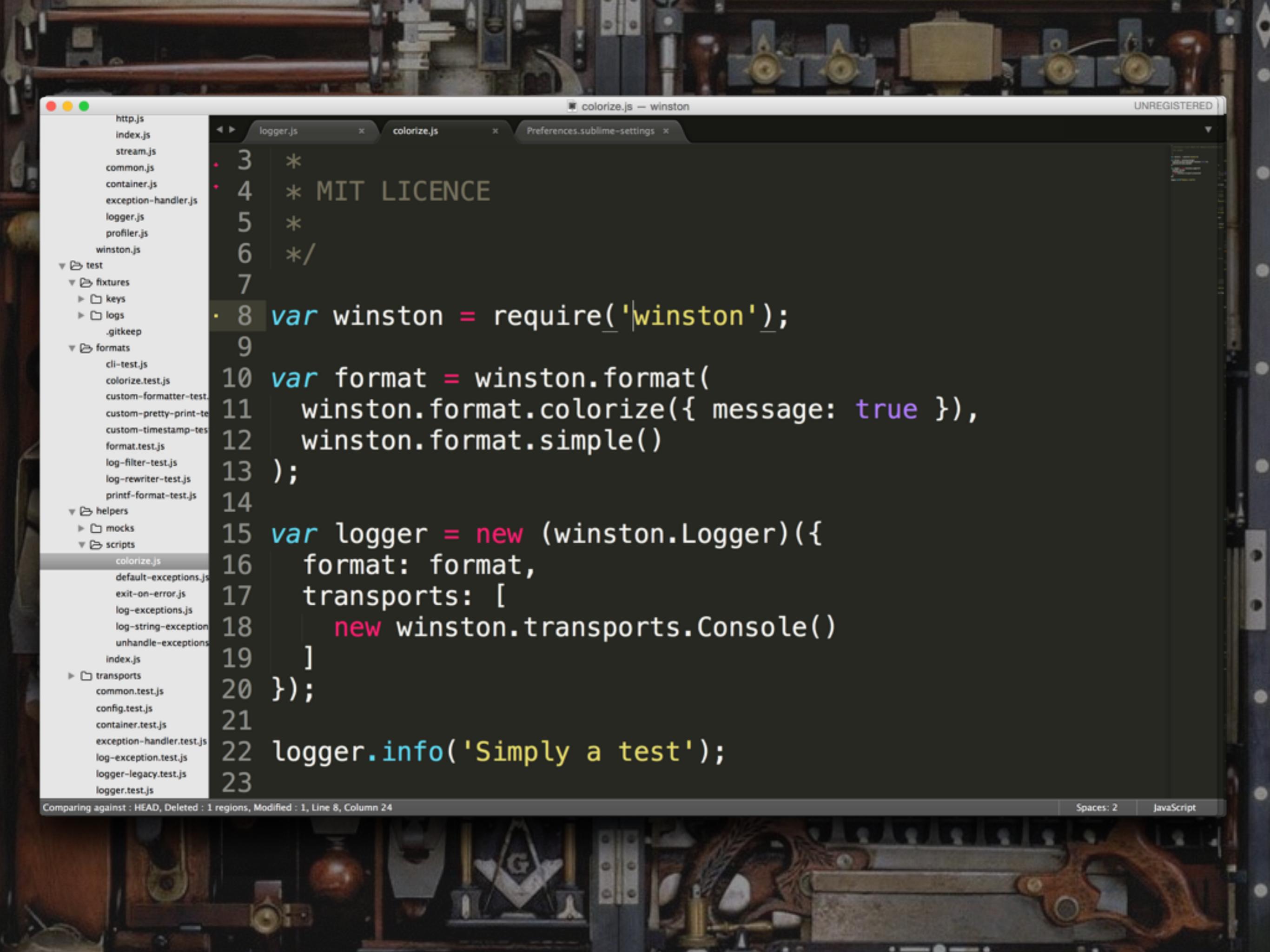
console.js

```
67
68  //
69  // ### function log (info)
70 // ##### @info {Object} **Optional** Additional metadata to
71 // Core logging method exposed to Winston.
72 //
73 Console.prototype.log = function (info) {
74   if (this.stderrLevels[info.level]) {
75     process.stderr.write(info.raw + this.eol);
76   } else {
77     process.stdout.write(info.raw + this.eol);
78   }
79 };
80
81 //
82 // ### function stringArrayToSet (array)
83 // ##### @strArray {Array} Array of Set-elements as string
84 // ##### @errMsg {string} **Optional** Custom error message
85 // Returns a Set-like object with strArray's elements as
86 //
87 function stringArrayToSet(strArray, errMsg) {
```

Comparing against : HEAD, Line 1, Column 1

Spaces: 2

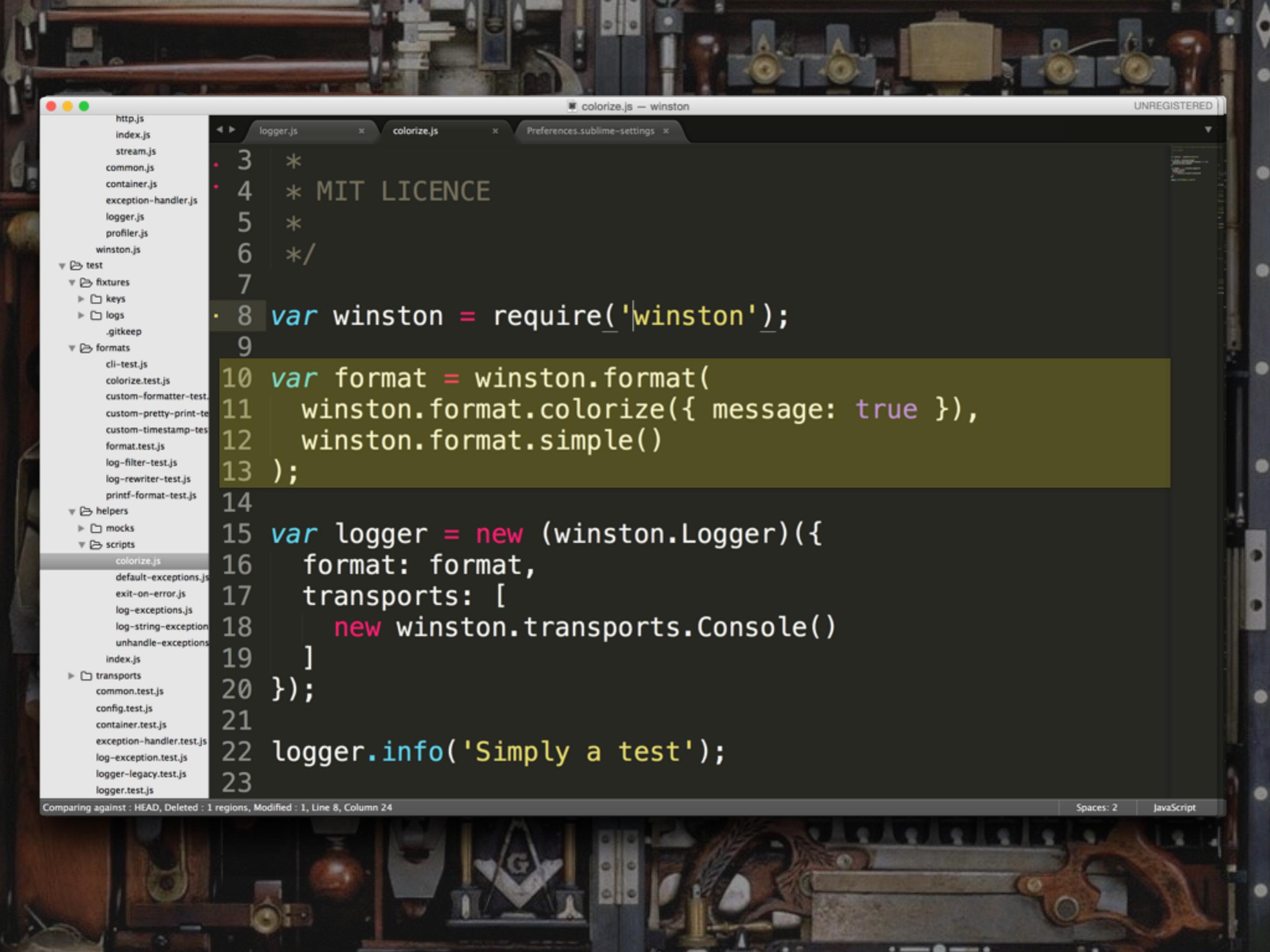
JavaScript



colorize.js — winston UNREGISTERED

```
3 *  
4 * MIT LICENCE  
5 *  
6 */  
7  
8 var winston = require('winston');  
9  
10 var format = winston.format(  
11   winston.format.colorize({ message: true }),  
12   winston.format.simple()  
13 );  
14  
15 var logger = new (winston.Logger)({  
16   format: format,  
17   transports: [  
18     new winston.transports.Console()  
19   ]  
20 });  
21  
22 logger.info('Simply a test');  
23
```

Comparing against : HEAD, Deleted : 1 regions, Modified : 1, Line 8, Column 24 Spaces: 2 JavaScript



colorize.js — winston UNREGISTERED

http.js
index.js
stream.js
common.js
container.js
exception-handler.js
logger.js
profiler.js
winston.js

test

fixtures

keys
logs
.gitkeep

formats

cli-test.js
colorize.test.js
custom-formatter-test.js
custom-pretty-print-test.js
custom-timestamp-test.js
format.test.js
log-filter-test.js
log-rewriter-test.js
printf-format-test.js

helpers

mocks
scripts

colorize.js
default-exceptions.js
exit-on-error.js
log-exceptions.js
log-string-exception.js
unhandle-exceptions.js
index.js

transports

common.test.js
config.test.js
container.test.js
exception-handler.test.js
log-exception.test.js
logger-legacy.test.js
logger.test.js

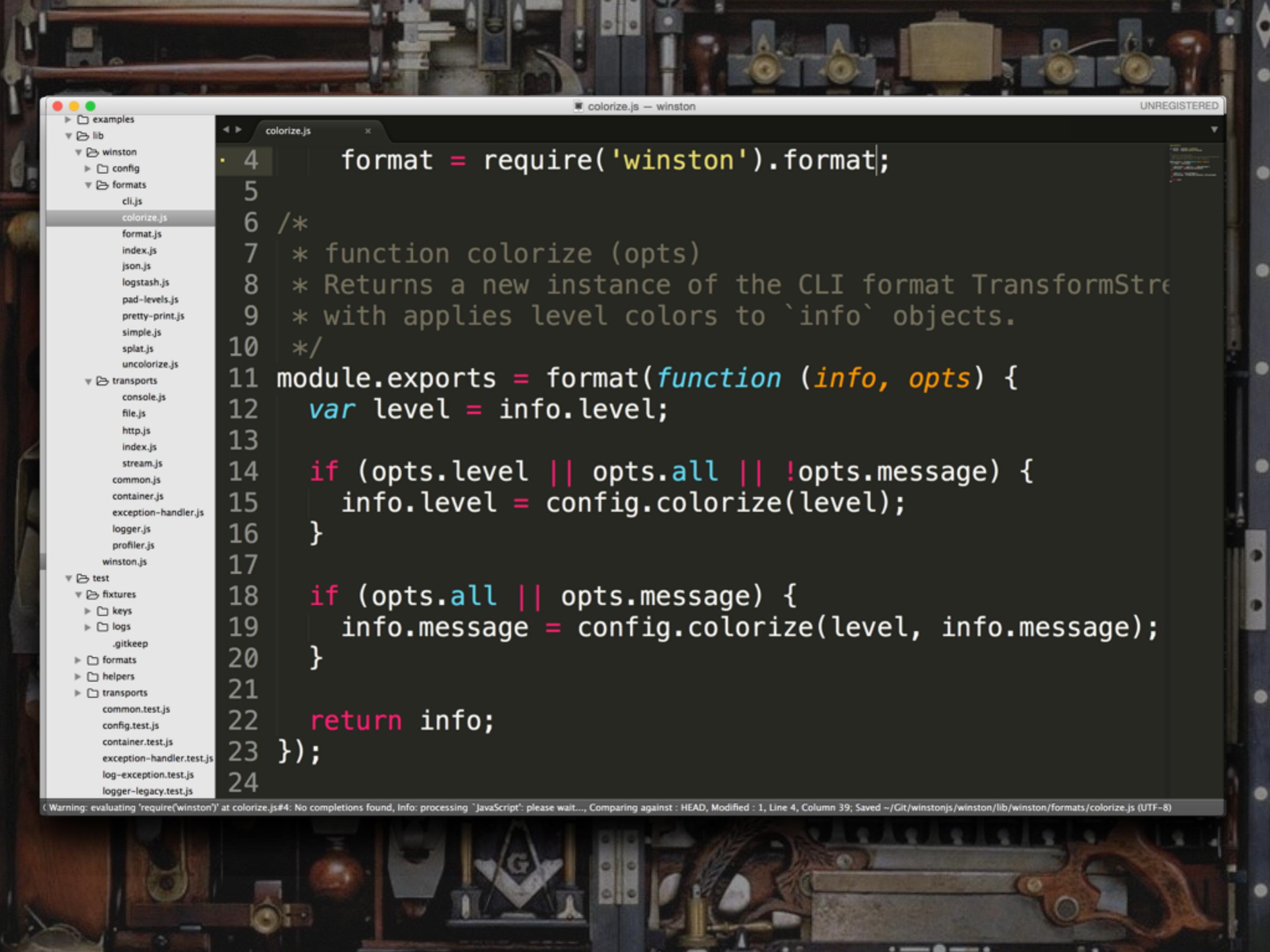
colorize.js

3 *
4 * MIT LICENCE
5 *
6 */
7
8 var winston = require('winston');
9
10 var format = winston.format(
11 winston.format.colorize({ message: true }),
12 winston.format.simple()
13);
14
15 var logger = new (winston.Logger)({
16 format: format,
17 transports: [
18 new winston.transports.Console()
19]
20 });
21
22 logger.info('Simply a test');
23

Comparing against : HEAD, Deleted : 1 regions, Modified : 1, Line 8, Column 24

Spaces: 2

JavaScript



colorize.js – winston UNREGISTERED

```
format = require('winston').format;

/*
 * function colorize (opts)
 * Returns a new instance of the CLI format TransformStream
 * with applies level colors to `info` objects.
 */
module.exports = format(function (info, opts) {
  var level = info.level;

  if (opts.level || opts.all || !opts.message) {
    info.level = config.colorize(level);
  }

  if (opts.all || opts.message) {
    info.message = config.colorize(level, info.message);
  }

  return info;
});
```

Warning: evaluating 'require('winston')' at colorize.js#4: No completions found, Info: processing 'JavaScript': please wait..., Comparing against : HEAD, Modified : 1, Line 4, Column 39; Saved ~/Git/winstonjs/winston/lib/winston/formats/colorize.js (UTF-8)



A black and white photograph of three dogs in a field. In the foreground, a black and white dog sits facing the camera. Behind it, two other dogs are partially visible: a brown and white dog to the left and a dark-colored dog to the right. The background is a soft-focus landscape of green and brown.

— DEFINE —

YOUR OWN

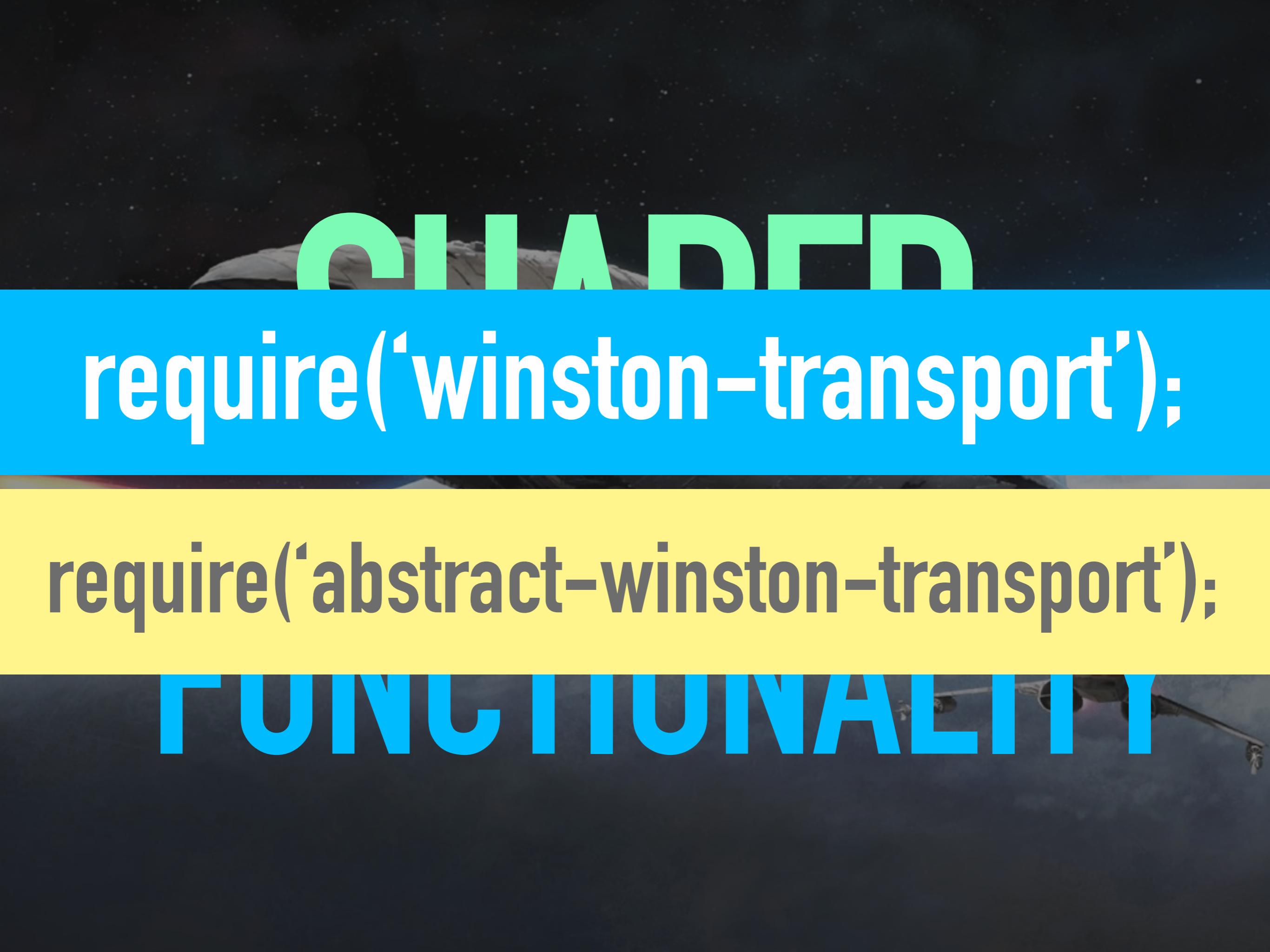
FORMATS

SHARED FUNCTIONALITY

A Star Wars X-wing fighter is shown in flight against a dark, star-filled background. The fighter is angled upwards and to the left. A white horizontal line with a central dot is overlaid on the image, positioned below the text.

GUARDED
require('winston-transport');

FUNCTIONALITY



CLIADEON
require('winston-transport');

require('abstract-winston-transport');

FUNCTIONALITY

BACKWARDS



COMPATABILITY

```
require('winston-compat');
```

BACKWARDS

COMPATABILITY

```
require('winston-compat');
```

common.log

COMPATABILITY

```
require('winston-compat');
```

common.log

winston.transports.Transport

COMPATIBILITY

```
require('winston-compat');
```

common.log

winston.transports.Transport

Old vows-based test macros

require('logmark')



require('logmark')

30% FASTER
THAN **Winston@2**

require('logmark')

30% FASTER
THAN WINSTON@2

50% FASTER THAN BUNYAN

IN BETA THIS WEEK



{github, twitter}.com/indexzero

MAY THE SOURCE BE WITH YOU

SPECIAL THANKS TO

Alberto Pose & Jarrett Cruger



YES! WE WANT YOU!

TO JOIN WINSTON