



Status

Changes

Console Output

Edit Build Information

Polling Log

Timings

Pipeline Overview

Thread Dump

Pause/resume

Replay

Pipeline Steps

Workspaces

Previous Build



Console Output

Download

Copy

[View as plain text](#)

```
Started by GitHub push by raghumn1999
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/jenkins_home/workspace/CI_nodejs_app
[Pipeline] {
[Pipeline] ws
Running in /var/jenkins_home/nodejs-app
[Pipeline] {
```

```
[Pipeline] stage
[Pipeline] { (Checkout)
[Pipeline] checkout
The recommended git tool is: NONE
No credentials specified
> git rev-parse --resolve-git-dir /var/jenkins_home/nodejs-app/.git #
timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url
https://github.com/raghumn1999/vananam\_assignment.git # timeout=10
Fetching upstream changes from
https://github.com/raghumn1999/vananam\_assignment.git
> git --version # timeout=10
> git --version # 'git version 2.39.5'
> git fetch --tags --force --progress --
https://github.com/raghumn1999/vananam\_assignment.git
+refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision 1b24e8d0d92e0dd4a6e18dd96bc26aaba096ea2a
(refs/remotes/origin/main)
> git config core.sparsecheckout # timeout=10
> git checkout -f 1b24e8d0d92e0dd4a6e18dd96bc26aaba096ea2a # timeout=10
Commit message: "Updated readme file"
> git rev-list --no-walk 0be8203220c6c7b6581a56a02ba0034842c0b9fc #
timeout=10
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Install Dependencies)
[Pipeline] dir
Running in /var/jenkins_home/nodejs-app/express
[Pipeline] {
[Pipeline] sh
+ set +x
Now using node v24.5.0 (npm v11.5.1)

up to date, audited 387 packages in 4s

66 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

```
[Pipeline] }  
[Pipeline] // dir  
[Pipeline] }  
[Pipeline] // stage  
[Pipeline] stage  
[Pipeline] { (Run Lint)  
[Pipeline] dir  
Running in /var/jenkins_home/nodejs-app/express  
[Pipeline] {  
[Pipeline] sh  
+ set +x  
Now using node v24.5.0 (npm v11.5.1)
```

```
> express@5.1.0 lint  
> eslint .
```

```
[Pipeline] }  
[Pipeline] // dir  
[Pipeline] }  
[Pipeline] // stage  
[Pipeline] stage  
[Pipeline] { (Run Tests)  
[Pipeline] dir  
Running in /var/jenkins_home/nodejs-app/express  
[Pipeline] {  
[Pipeline] sh  
+ set +x  
Now using node v24.5.0 (npm v11.5.1)
```

```
> express@5.1.0 test  
> mocha --require test/support/env --reporter spec --check-leaks test/  
test/acceptance/
```

Route

- ✓ should work without handlers
- ✓ should not stack overflow with a large sync stack (179ms)
- .all
 - ✓ should add handler
 - ✓ should handle VERBS
 - ✓ should stack

`.VERB`

- ✓ should support `.get`
- ✓ should limit to just `.VERB`
- ✓ should allow `fallthrough`

errors

- ✓ should handle errors via arity 4 functions
- ✓ should handle `throw`
- ✓ should handle throwing inside error handlers
- ✓ should handle `throw` in `.all`
- ✓ should handle single error handler

Router

- ✓ should return a function with router methods
- ✓ should support `.use` of other routers
- ✓ should support dynamic routes
- ✓ should handle blank URL
- ✓ should handle missing URL
- ✓ handle missing method
- ✓ should not stack overflow with many registered routes (588ms)
- ✓ should not stack overflow with a large sync route stack (350ms)
- ✓ should not stack overflow with a large sync middleware stack (127ms)

`.handle`

- ✓ should dispatch

`.multiple callbacks`

- ✓ should throw if a callback is null
- ✓ should throw if a callback is undefined
- ✓ should throw if a callback is not a function
- ✓ should not throw if all callbacks are functions

error

- ✓ should skip non error middleware
- ✓ should handle throwing inside routes with params
- ✓ should handle throwing in handler after async param
- ✓ should handle throwing inside error handlers

FQDN

- ✓ should not obscure FQDNs
- ✓ should ignore FQDN in search
- ✓ should ignore FQDN in path
- ✓ should adjust FQDN `req.url`
- ✓ should adjust FQDN `req.url` with multiple handlers
- ✓ should adjust FQDN `req.url` with multiple routed handlers

`.all`

- ✓ should support using `.all` to capture all http verbs

.use

- ✓ should require middleware
- ✓ should reject string as middleware
- ✓ should reject number as middleware
- ✓ should reject null as middleware
- ✓ should reject Date as middleware
- ✓ should be called for any URL
- ✓ should accept array of middleware

.param

- ✓ should require function
- ✓ should reject non-function
- ✓ should call param function when routing VERBS
- ✓ should call param function when routing middleware
- ✓ should only call once per request
- ✓ should call when values differ

parallel requests

- ✓ should not mix requests (51ms)

app.all()

- ✓ should add a router per method (63ms)
- ✓ should run the callback for a method just once

app

.engine(ext, fn)

- ✓ should map a template engine
- ✓ should throw when the callback is missing
- ✓ should work without leading "."
- ✓ should work "view engine" setting
- ✓ should work "view engine" with leading "."

HEAD

- ✓ should default to GET
- ✓ should output the same headers as GET requests

app.head()

- ✓ should override

app

- ✓ should inherit from event emitter
- ✓ should be callable
- ✓ should 404 without routes

`app.parent`

- ✓ should return the parent when mounted

`app.mountpath`

- ✓ should return the mounted path

`app.path()`

- ✓ should return the canonical

`in development`

- ✓ should disable "view cache"

`in production`

- ✓ should enable "view cache"

`without NODE_ENV`

- ✓ should default to development

`app.listen()`

- ✓ should wrap with an HTTP server
- ✓ should callback on HTTP server errors
- ✓ accepts port + hostname + backlog + callback
- ✓ accepts just a callback (no args)
- ✓ `server.address()` gives a { address, port, family } object

`app`

`.locals`

- ✓ should default object with null prototype

`.settings`

- ✓ should contain app settings

`OPTIONS`

- ✓ should default to the routes defined
- ✓ should only include each method once
- ✓ should not be affected by `app.all`
- ✓ should not respond if the path is not defined
- ✓ should forward requests down the middleware chain when error occurs in response handler
 - ✓ should pass error to callback

`app.options()`

- ✓ should override the default behavior

app

.param(names, fn)

- ✓ should map the array

.param(name, fn)

- ✓ should map logic for a single param
- ✓ should only call once per request
- ✓ should call when values differ
- ✓ should support altering req.params across routes
- ✓ should not invoke without route handler
- ✓ should work with encoded values
- ✓ should catch thrown error
- ✓ should catch thrown secondary error
- ✓ should defer to next route
- ✓ should defer all the param routes
- ✓ should not call when values differ on error
- ✓ should call when values differ when using "next"

app

.render(name, fn)

- ✓ should support absolute paths
 - ✓ should support absolute paths with "view engine"
 - ✓ should expose app.locals
 - ✓ should support index.<engine>
 - ✓ should handle render error throws
- when the file does not exist
- ✓ should provide a helpful error
- when an error occurs
- ✓ should invoke the callback
- when an extension is given
- ✓ should render the template
- when "view engine" is given
- ✓ should render the template
- when "views" is given
- ✓ should lookup the file in the path
- when array of paths
- ✓ should lookup the file in the path
 - ✓ should lookup in later paths until found
 - ✓ should error if file does not exist
- when a "view" constructor is given
- ✓ should create an instance of it
- caching

- ✓ should always lookup view without cache

- ✓ should cache with "view cache" setting

.render(name, options, fn)

- ✓ should render the template

- ✓ should expose app.locals

- ✓ should give precedence to app.render() locals caching

- ✓ should cache with cache option

app

.request

- ✓ should extend the request prototype

- ✓ should only extend for the referenced app

- ✓ should inherit to sub apps

- ✓ should allow sub app to override

- ✓ should not pollute parent app

app

.response

- ✓ should extend the response prototype

- ✓ should only extend for the referenced app

- ✓ should inherit to sub apps

- ✓ should allow sub app to override

- ✓ should not pollute parent app

app.route

- ✓ should return a new route

- ✓ should all .VERB after .all

- ✓ should support dynamic routes

- ✓ should not error on empty routes

promise support

- ✓ should pass rejected promise value

- ✓ should pass rejected promise without value

- ✓ should ignore resolved promise

error handling

- ✓ should pass rejected promise value

- ✓ should pass rejected promise without value

- ✓ should ignore resolved promise

app.router

- ✓ should restore req.params after leaving router

- ✓ should be .use()able

- ✓ should allow literal "."
- ✓ should allow rewriting of the url
- ✓ should run in order added
- ✓ should be chainable
- ✓ should not use disposed router/middleware

methods

- ✓ should include ACL
- ✓ should reject numbers for app.acl
- ✓ should include BIND
- ✓ should reject numbers for app.bind
- ✓ should include CHECKOUT
- ✓ should reject numbers for app.checkout
- ✓ should include COPY
- ✓ should reject numbers for app.copy
- ✓ should include DELETE
- ✓ should reject numbers for app.delete
- ✓ should include GET
- ✓ should reject numbers for app.get
- ✓ should include HEAD
- ✓ should reject numbers for app.head
- ✓ should include LINK
- ✓ should reject numbers for app.link
- ✓ should include LOCK
- ✓ should reject numbers for app.lock
- ✓ should include M-SEARCH
- ✓ should reject numbers for app.m-search
- ✓ should include MERGE
- ✓ should reject numbers for app.merge
- ✓ should include MKACTION
- ✓ should reject numbers for app.mkactivity
- ✓ should include MKCALENDAR
- ✓ should reject numbers for app.mkcalendar
- ✓ should include MKCOL
- ✓ should reject numbers for app.mkcol
- ✓ should include MOVE
- ✓ should reject numbers for app.move
- ✓ should include NOTIFY
- ✓ should reject numbers for app.notify
- ✓ should include OPTIONS
- ✓ should reject numbers for app.options
- ✓ should include PATCH
- ✓ should reject numbers for app.patch

- ✓ should include POST
- ✓ should reject numbers for app.post
- ✓ should include PROPFIND
- ✓ should reject numbers for app.propfind
- ✓ should include PROPPATCH
- ✓ should reject numbers for app.proppatch
- ✓ should include PURGE
- ✓ should reject numbers for app.purge
- ✓ should include PUT
- ✓ should reject numbers for app.put
- ✓ should include QUERY
- ✓ should reject numbers for app.query
- ✓ should include REBIND
- ✓ should reject numbers for app.rebind
- ✓ should include REPORT
- ✓ should reject numbers for app.report
- ✓ should include SEARCH
- ✓ should reject numbers for app.search
- ✓ should include SOURCE
- ✓ should reject numbers for app.source
- ✓ should include SUBSCRIBE
- ✓ should reject numbers for app.subscribe
- ✓ should include TRACE
- ✓ should reject numbers for app.trace
- ✓ should include UNBIND
- ✓ should reject numbers for app.unbind
- ✓ should include UNLINK
- ✓ should reject numbers for app.unlink
- ✓ should include UNLOCK
- ✓ should reject numbers for app.unlock
- ✓ should include UNSUBSCRIBE
- ✓ should reject numbers for app.unsubscribe
- ✓ should re-route when method is altered

decode params

- ✓ should decode correct params
- ✓ should not accept params in malformed paths
- ✓ should not decode spaces
- ✓ should work with unicode

when given a regexp

- ✓ should match the pathname only
- ✓ should populate req.params with the captures
- ✓ should populate req.params with named captures

- ✓ should ensure regexp matches path prefix

case sensitivity

- ✓ should be disabled by default
- when "case sensitive routing" is enabled
 - ✓ should match identical casing
 - ✓ should not match otherwise

params

- ✓ should overwrite existing req.params by default
- ✓ should allow merging existing req.params
- ✓ should use params from router
- ✓ should merge numeric indices req.params
- ✓ should merge numeric indices req.params when more in parent
- ✓ should merge numeric indices req.params when parent has same number
- ✓ should ignore invalid incoming req.params
- ✓ should restore req.params

trailing slashes

- ✓ should be optional by default
- when "strict routing" is enabled
 - ✓ should match trailing slashes
 - ✓ should pass-through middleware
 - ✓ should pass-through mounted middleware
 - ✓ should match no slashes
 - ✓ should match middleware when omitting the trailing slash
 - ✓ should match middleware
 - ✓ should match middleware when adding the trailing slash
 - ✓ should fail when omitting the trailing slash
 - ✓ should fail when adding the trailing slash

:name

- ✓ should denote a capture group
- ✓ should match a single segment only
- ✓ should allow several capture groups
- ✓ should work following a partial capture group
- ✓ should work inside literal parenthesis
- ✓ should work in array of paths

:name?

- ✓ should denote an optional capture group
- ✓ should populate the capture group

:name*

- ✓ should match one segment
- ✓ should match many segments
- ✓ should match zero segments

:name+

- ✓ should match one segment
- ✓ should match many segments
- ✓ should not match zero segments

.:name

- ✓ should denote a format

.:name?

- ✓ should denote an optional format

when next() is called

- ✓ should continue lookup

when next("route") is called

- ✓ should jump to next route

when next("router") is called

- ✓ should jump out of router

when next(err) is called

- ✓ should break out of app.router
- ✓ should call handler in same route, if exists

promise support

- ✓ should pass rejected promise value
- ✓ should pass rejected promise without value
- ✓ should ignore resolved promise

error handling

- ✓ should pass rejected promise value
- ✓ should pass rejected promise without value
- ✓ should ignore resolved promise

app

.VERB()

- ✓ should not get invoked without error handler on error
- ✓ should only call an error handling routing callback when an error is

propagated

app

- ✓ should emit "mount" when mounted

.use(app)

- ✓ should mount the app
- ✓ should support mount-points
- ✓ should set the child's .parent
- ✓ should support dynamic routes
- ✓ should support mounted app anywhere

.use(middleware)

- ✓ should accept multiple arguments
- ✓ should invoke middleware for all requests

- ✓ should accept array of middleware
- ✓ should accept multiple arrays of middleware
- ✓ should accept nested arrays of middleware
- .use(path, middleware)
 - ✓ should require middleware
 - ✓ should reject string as middleware
 - ✓ should reject number as middleware
 - ✓ should reject null as middleware
 - ✓ should reject Date as middleware
 - ✓ should strip path from req.url
 - ✓ should accept multiple arguments
 - ✓ should invoke middleware for all requests starting with path
 - ✓ should work if path has trailing slash
 - ✓ should accept array of middleware
 - ✓ should accept multiple arrays of middleware
 - ✓ should accept nested arrays of middleware
 - ✓ should support array of paths
 - ✓ should support array of paths with middleware array
 - ✓ should support regexp path
 - ✓ should support empty string path

config

- .set()
 - ✓ should set a value
 - ✓ should set prototype values
 - ✓ should return the app
 - ✓ should return the app when undefined
 - ✓ should return set value
 - ✓ should return undefined for prototype values
- "etag"
 - ✓ should throw on bad value
 - ✓ should set "etag fn"
- "trust proxy"
 - ✓ should set "trust proxy fn"
- .get()
 - ✓ should return undefined when unset
 - ✓ should return undefined for prototype values
 - ✓ should otherwise return the value
- when mounted
 - ✓ should default to the parent app
 - ✓ should given precedence to the child
 - ✓ should inherit "trust proxy" setting

- ✓ should prefer child "trust proxy" setting
- .enable()
 - ✓ should set the value to true
 - ✓ should set prototype values
- .disable()
 - ✓ should set the value to false
 - ✓ should set prototype values
- .enabled()
 - ✓ should default to false
 - ✓ should return true when set
 - ✓ should default to false for prototype values
- .disabled()
 - ✓ should default to true
 - ✓ should return false when set
 - ✓ should default to true for prototype values

exports

- ✓ should expose Router
- ✓ should expose json middleware
- ✓ should expose raw middleware
- ✓ should expose static middleware
- ✓ should expose text middleware
- ✓ should expose urlencoded middleware
- ✓ should expose the application prototype
- ✓ should expose the request prototype
- ✓ should expose the response prototype
- ✓ should permit modifying the .application prototype
- ✓ should permit modifying the .request prototype
- ✓ should permit modifying the .response prototype

express.json()

- ✓ should parse JSON
- ✓ should handle Content-Length: 0
- ✓ should handle empty message-body
- ✓ should handle no message-body
- ✓ should 400 when only whitespace
- ✓ should 400 when invalid content-length
- ✓ should handle duplicated middleware
- when JSON is invalid
 - ✓ should 400 for bad token
 - ✓ should 400 for incomplete
 - ✓ should include original body on error object

```
with limit option
  ✓ should 413 when over limit with Content-Length
  ✓ should 413 when over limit with chunked encoding
  ✓ should 413 when inflated body over limit
  ✓ should accept number of bytes
  ✓ should not change when options altered
  ✓ should not hang response
  ✓ should not error when inflating
with inflate option
  when false
    ✓ should not accept content-encoding
  when true
    ✓ should accept content-encoding
with strict option
  when undefined
    ✓ should 400 on primitives
  when false
    ✓ should parse primitives
  when true
    ✓ should not parse primitives
    ✓ should not parse primitives with leading whitespaces
    ✓ should allow leading whitespaces in JSON
    ✓ should include correct message in stack trace
with type option
  when "application/vnd.api+json"
    ✓ should parse JSON for custom type
    ✓ should ignore standard type
  when ["application/json", "application/vnd.api+json"]
    ✓ should parse JSON for "application/json"
    ✓ should parse JSON for "application/vnd.api+json"
    ✓ should ignore "application/x-json"
  when a function
    ✓ should parse when truthy value returned
    ✓ should work without content-type
    ✓ should not invoke without a body
with verify option
  ✓ should assert value if function
  ✓ should error from verify
  ✓ should allow custom codes
  ✓ should allow custom type
  ✓ should include original body on error object
  ✓ should allow pass-through
```

- ✓ should work with different charsets
- ✓ should 415 on unknown charset prior to verify

async local storage

- ✓ should persist store
- ✓ should persist store when unmatched content-type
- ✓ should persist store when inflated
- ✓ should persist store when inflate error
- ✓ should persist store when parse error
- ✓ should persist store when limit exceeded

charset

- ✓ should parse utf-8
- ✓ should parse utf-16
- ✓ should parse when content-length != char length
- ✓ should default to utf-8
- ✓ should fail on unknown charset

encoding

- ✓ should parse without encoding
- ✓ should support identity encoding
- ✓ should support gzip encoding
- ✓ should support deflate encoding
- ✓ should be case-insensitive
- ✓ should 415 on unknown encoding
- ✓ should 400 on malformed encoding
- ✓ should 413 when inflated value exceeds limit

express.raw()

- ✓ should parse application/octet-stream
- ✓ should 400 when invalid content-length
- ✓ should handle Content-Length: 0
- ✓ should handle empty message-body
- ✓ should handle duplicated middleware

with limit option

- ✓ should 413 when over limit with Content-Length
- ✓ should 413 when over limit with chunked encoding
- ✓ should 413 when inflated body over limit
- ✓ should accept number of bytes
- ✓ should not change when options altered
- ✓ should not hang response
- ✓ should not error when inflating

with inflate option

when false

- ✓ should not accept content-encoding


```
when true
  ✓ should accept content-encoding
with type option
  when "application/vnd+octets"
    ✓ should parse for custom type
    ✓ should ignore standard type
  when ["application/octet-stream", "application/vnd+octets"]
    ✓ should parse "application/octet-stream"
    ✓ should parse "application/vnd+octets"
    ✓ should ignore "application/x-foo"
  when a function
    ✓ should parse when truthy value returned
    ✓ should work without content-type
    ✓ should not invoke without a body
with verify option
  ✓ should assert value is function
  ✓ should error from verify
  ✓ should allow custom codes
  ✓ should allow pass-through
async local storage
  ✓ should persist store
  ✓ should persist store when unmatched content-type
  ✓ should persist store when inflated
  ✓ should persist store when inflate error
  ✓ should persist store when limit exceeded
charset
  ✓ should ignore charset
encoding
  ✓ should parse without encoding
  ✓ should support identity encoding
  ✓ should support gzip encoding
  ✓ should support deflate encoding
  ✓ should be case-insensitive
  ✓ should 415 on unknown encoding

express.static()
  basic operations
    ✓ should require root path
    ✓ should require root path to be string
    ✓ should serve static files
    ✓ should support nesting
    ✓ should set Content-Type
```

- ✓ should set Last-Modified
- ✓ should default max-age=0
- ✓ should support urlencoded pathnames
- ✓ should not choke on auth-looking URL
- ✓ should support index.html
- ✓ should support ../
- ✓ should support HEAD
- ✓ should skip POST requests
- ✓ should support conditional requests
- ✓ should support precondition checks
- ✓ should serve zero-length files
- ✓ should ignore hidden files

current dir

- ✓ should be served with "."

acceptRanges

when false

- ✓ should not include Accept-Ranges
- ✓ should ignore Range request header

when true

- ✓ should include Accept-Ranges
- ✓ should obey Range request header

cacheControl

when false

- ✓ should not include Cache-Control
- ✓ should ignore maxAge

when true

- ✓ should include Cache-Control

extensions

- ✓ should be not be enabled by default
- ✓ should be configurable
- ✓ should support disabling extensions
- ✓ should support fallbacks
- ✓ should 404 if nothing found

fallthrough

- ✓ should default to true

when true

- ✓ should fall-through when OPTIONS request
- ✓ should fall-through when URL malformed
- ✓ should fall-through when traversing past root
- ✓ should fall-through when URL too long

with redirect: true

- ✓ should fall-through when directory

- ✓ should redirect when directory without slash

with redirect: false

- ✓ should fall-through when directory
- ✓ should fall-through when directory without slash

when false

- ✓ should 405 when OPTIONS request
- ✓ should 400 when URL malformed
- ✓ should 403 when traversing past root
- ✓ should 404 when URL too long

with redirect: true

- ✓ should 404 when directory
- ✓ should redirect when directory without slash

with redirect: false

- ✓ should 404 when directory
- ✓ should 404 when directory without slash

hidden files

- ✓ should be served when dotfiles: "allow" is given

immutable

- ✓ should default to false
- ✓ should set immutable directive in Cache-Control

lastModified

when false

- ✓ should not include Last-Modified

when true

- ✓ should include Last-Modified

maxAge

- ✓ should accept string
- ✓ should be reasonable when infinite

redirect

- ✓ should redirect directories
- ✓ should include HTML link
- ✓ should redirect directories with query string
- ✓ should not redirect to protocol-relative locations
- ✓ should ensure redirect URL is properly encoded
- ✓ should respond with default Content-Security-Policy
- ✓ should not redirect incorrectly

when false

- ✓ should disable redirect

setHeaders

- ✓ should reject non-functions
- ✓ should get called when sending file
- ✓ should not get called on 404

- ✓ should not get called on redirect

when traversing past root

- ✓ should catch urlencoded ../

- ✓ should not allow root path disclosure

when request has "Range" header

- ✓ should support byte ranges

- ✓ should be inclusive

- ✓ should set Content-Range

- ✓ should support -n

- ✓ should support n-

- ✓ should respond with 206 "Partial Content"

- ✓ should set Content-Length to the # of octets transferred

when last-byte-pos of the range is greater than current length

- ✓ is taken to be equal to one less than the current length

- ✓ should adapt the Content-Length accordingly

when the first- byte-pos of the range is greater than the current

length

- ✓ should respond with 416

- ✓ should include a Content-Range header of complete length

when syntactically invalid

- ✓ should respond with 200 and the entire contents

when index at mount point

- ✓ should redirect correctly

when mounted

- ✓ should redirect relative to the originalUrl

- ✓ should not choke on auth-looking URL

when mounted "root" as a file

- ✓ should load the file when on trailing slash

- ✓ should 404 when trailing slash

when responding non-2xx or 304

- ✓ should not alter the status

when index file serving disabled

- ✓ should next() on directory

- ✓ should redirect to trailing slash

- ✓ should next() on mount point

- ✓ should redirect to trailing slash mount point

express.text()

- ✓ should parse text/plain

- ✓ should 400 when invalid content-length

- ✓ should handle Content-Length: 0

- ✓ should handle empty message-body

- ✓ should handle duplicated middleware
- with defaultCharset option
 - ✓ should change default charset
 - ✓ should honor content-type charset
- with limit option
 - ✓ should 413 when over limit with Content-Length
 - ✓ should 413 when over limit with chunked encoding
 - ✓ should 413 when inflated body over limit
 - ✓ should accept number of bytes
 - ✓ should not change when options altered
 - ✓ should not hang response
 - ✓ should not error when inflating (104ms)
- with inflate option
 - when false
 - ✓ should not accept content-encoding
 - when true
 - ✓ should accept content-encoding
- with type option
 - when "text/html"
 - ✓ should parse for custom type
 - ✓ should ignore standard type
 - when ["text/html", "text/plain"]
 - ✓ should parse "text/html"
 - ✓ should parse "text/plain"
 - ✓ should ignore "text/xml"
 - when a function
 - ✓ should parse when truthy value returned
 - ✓ should work without content-type
 - ✓ should not invoke without a body
- with verify option
 - ✓ should assert value is function
 - ✓ should error from verify
 - ✓ should allow custom codes
 - ✓ should allow pass-through
 - ✓ should 415 on unknown charset prior to verify
- async local storage
 - ✓ should persist store
 - ✓ should persist store when unmatched content-type
 - ✓ should persist store when inflated
 - ✓ should persist store when inflate error
 - ✓ should persist store when limit exceeded

charset

- ✓ should parse utf-8
- ✓ should parse codepage charsets
- ✓ should parse when content-length != char length
- ✓ should default to utf-8
- ✓ should 415 on unknown charset

encoding

- ✓ should parse without encoding
- ✓ should support identity encoding
- ✓ should support gzip encoding
- ✓ should support deflate encoding
- ✓ should be case-insensitive
- ✓ should 415 on unknown encoding

express.urlencoded()

- ✓ should parse x-www-form-urlencoded
- ✓ should 400 when invalid content-length
- ✓ should handle Content-Length: 0
- ✓ should handle empty message-body
- ✓ should handle duplicated middleware
- ✓ should not parse extended syntax

with extended option

when false

- ✓ should not parse extended syntax
- ✓ should parse multiple key instances

when true

- ✓ should parse multiple key instances
- ✓ should parse extended syntax
- ✓ should parse parameters with dots
- ✓ should parse fully-encoded extended syntax
- ✓ should parse array index notation
- ✓ should parse array index notation with large array
- ✓ should parse array of objects syntax
- ✓ should parse deep object

with inflate option

when false

- ✓ should not accept content-encoding

when true

- ✓ should accept content-encoding

with limit option

- ✓ should 413 when over limit with Content-Length
- ✓ should 413 when over limit with chunked encoding

- ✓ should 413 when inflated body over limit
- ✓ should accept number of bytes
- ✓ should not change when options altered
- ✓ should not hang response
- ✓ should not error when inflating

with parameterLimit option

with extended: false

- ✓ should reject 0
- ✓ should reject string
- ✓ should 413 if over limit
- ✓ should work when at the limit
- ✓ should work if number is floating point
- ✓ should work with large limit (149ms)
- ✓ should work with Infinity limit (164ms)

with extended: true

- ✓ should reject 0
- ✓ should reject string
- ✓ should 413 if over limit
- ✓ should work when at the limit
- ✓ should work if number is floating point
- ✓ should work with large limit (56ms)
- ✓ should work with Infinity limit (217ms)

with type option

when "application/vnd.x-www-form-urlencoded"

- ✓ should parse for custom type
- ✓ should ignore standard type

when ["urlencoded", "application/x-pairs"]

- ✓ should parse "application/x-www-form-urlencoded"
- ✓ should parse "application/x-pairs"
- ✓ should ignore application/x-foo

when a function

- ✓ should parse when truthy value returned
- ✓ should work without content-type
- ✓ should not invoke without a body

with verify option

- ✓ should assert value if function
- ✓ should error from verify
- ✓ should allow custom codes
- ✓ should allow custom type
- ✓ should allow pass-through
- ✓ should 415 on unknown charset prior to verify

async local storage

- ✓ should persist store
- ✓ should persist store when unmatched content-type
- ✓ should persist store when inflated
- ✓ should persist store when inflate error
- ✓ should persist store when limit exceeded

charset

- ✓ should parse utf-8
- ✓ should parse when content-length != char length
- ✓ should default to utf-8
- ✓ should fail on unknown charset

encoding

- ✓ should parse without encoding
- ✓ should support identity encoding
- ✓ should support gzip encoding
- ✓ should support deflate encoding
- ✓ should be case-insensitive
- ✓ should 415 on unknown encoding

middleware

.next()

- ✓ should behave like connect

throw after .end()

- ✓ should fail gracefully

req

- ✓ should accept an argument list of type names
- .accepts(type)
 - ✓ should return true when Accept is not present
 - ✓ should return true when present
 - ✓ should return false otherwise
- .accepts(types)
 - ✓ should return the first when Accept is not present
 - ✓ should return the first acceptable type
 - ✓ should return false when no match is made
 - ✓ should take quality into account
 - ✓ should return the first acceptable type with canonical mime types

req

.acceptsCharsets(type)

- when Accept-Charset is not present
 - ✓ should return true

when Accept-Charset is present

- ✓ should return true
- ✓ should return false otherwise

req

.acceptsEncodings

- ✓ should return encoding if accepted
- ✓ should be false if encoding not accepted

req

.acceptsLanguages

- ✓ should return language if accepted
- ✓ should be false if language not accepted

when Accept-Language is not present

- ✓ should always return language

req

.baseUrl

- ✓ should be empty for top-level route
- ✓ should contain lower path
- ✓ should contain full lower path
- ✓ should travel through routers correctly

req

.fresh

- ✓ should return true when the resource is not modified
- ✓ should return false when the resource is modified
- ✓ should return false without response headers
- ✓ should ignore "If-Modified-Since" when "If-None-Match" is present

req

.get(field)

- ✓ should return the header field value
- ✓ should special-case Referer
- ✓ should throw missing header name
- ✓ should throw for non-string header name

req

.host

- ✓ should return the Host when present
- ✓ should strip port number
- ✓ should return undefined otherwise

- ✓ should work with IPv6 Host
- ✓ should work with IPv6 Host and port
- when "trust proxy" is enabled
 - ✓ should respect X-Forwarded-Host
 - ✓ should ignore X-Forwarded-Host if socket addr not trusted
 - ✓ should default to Host
- when trusting hop count
 - ✓ should respect X-Forwarded-Host
- when "trust proxy" is disabled
 - ✓ should ignore X-Forwarded-Host

req

.hostname

- ✓ should return the Host when present
- ✓ should strip port number
- ✓ should return undefined otherwise
- ✓ should work with IPv6 Host
- ✓ should work with IPv6 Host and port
- when "trust proxy" is enabled
 - ✓ should respect X-Forwarded-Host
 - ✓ should ignore X-Forwarded-Host if socket addr not trusted
 - ✓ should default to Host
- when multiple X-Forwarded-Host
 - ✓ should use the first value
 - ✓ should remove OWS around comma
 - ✓ should strip port number
- when "trust proxy" is disabled
 - ✓ should ignore X-Forwarded-Host

req

.ip

- when X-Forwarded-For is present
 - when "trust proxy" is enabled
 - ✓ should return the client addr
 - ✓ should return the addr after trusted proxy based on count
 - ✓ should return the addr after trusted proxy based on list
 - ✓ should return the addr after trusted proxy, from sub app
 - when "trust proxy" is disabled
 - ✓ should return the remote address
- when X-Forwarded-For is not present
 - ✓ should return the remote address

req

.ips

when X-Forwarded-For is present

when "trust proxy" is enabled

- ✓ should return an array of the specified addresses

- ✓ should stop at first untrusted

when "trust proxy" is disabled

- ✓ should return an empty array

when X-Forwarded-For is not present

- ✓ should return []

req.is()

when given a mime type

- ✓ should return the type when matching

- ✓ should return false when not matching

- ✓ should ignore charset

when content-type is not present

- ✓ should return false

when given an extension

- ✓ should lookup the mime type

when given */subtype

- ✓ should return the full type when matching

- ✓ should return false when not matching

- ✓ should ignore charset

when given type/*

- ✓ should return the full type when matching

- ✓ should return false when not matching

- ✓ should ignore charset

req

.path

- ✓ should return the parsed pathname

req

.protocol

- ✓ should return the protocol string

when "trust proxy" is enabled

- ✓ should respect X-Forwarded-Proto

- ✓ should default to the socket addr if X-Forwarded-Proto not present

- ✓ should ignore X-Forwarded-Proto if socket addr not trusted

- ✓ should default to http

when trusting hop count

- ✓ should respect X-Forwarded-Proto
- when "trust proxy" is disabled
- ✓ should ignore X-Forwarded-Proto

req

- .query
 - ✓ should default to {}
 - ✓ should default to parse simple keys
- when "query parser" is extended
 - ✓ should parse complex keys
 - ✓ should parse parameters with dots
- when "query parser" is simple
 - ✓ should not parse complex keys
- when "query parser" is a function
 - ✓ should parse using function
- when "query parser" disabled
 - ✓ should not parse query
- when "query parser" enabled
 - ✓ should not parse complex keys
- when "query parser" an unknown value
 - ✓ should throw

req

- .range(size)
 - ✓ should return parsed ranges
 - ✓ should cap to the given size
 - ✓ should cap to the given size when open-ended
 - ✓ should have a .type
 - ✓ should accept any type
 - ✓ should return undefined if no range
- .range(size, options)
 - with "combine: true" option
 - ✓ should return combined ranges

req

- .route
 - ✓ should be the executed Route

req

- .secure
 - when X-Forwarded-Proto is missing

- ✓ should return false when http
- .secure
 - when X-Forwarded-Proto is present
 - ✓ should return false when http
 - ✓ should return true when "trust proxy" is enabled
 - ✓ should return false when initial proxy is http
 - ✓ should return true when initial proxy is https
 - when "trust proxy" trusting hop count
 - ✓ should respect X-Forwarded-Proto
- req
 - .signedCookies
 - ✓ should return a signed JSON cookie
 - req
 - .stale
 - ✓ should return false when the resource is not modified
 - ✓ should return true when the resource is modified
 - ✓ should return true without response headers
- req
 - .subdomains
 - when present
 - ✓ should return an array
 - ✓ should work with IPv4 address
 - ✓ should work with IPv6 address
 - otherwise
 - ✓ should return an empty array
 - with no host
 - ✓ should return an empty array
 - with trusted X-Forwarded-Host
 - ✓ should return an array
 - when subdomain offset is set
 - when subdomain offset is zero
 - ✓ should return an array with the whole domain
 - ✓ should return an array with the whole IPv4
 - ✓ should return an array with the whole IPv6
 - when present
 - ✓ should return an array
 - otherwise
 - ✓ should return an empty array

req

.xhr

- ✓ should return true when X-Requested-With is xmlhttprequest
- ✓ should case-insensitive
- ✓ should return false otherwise
- ✓ should return false when not present

res

.append(field, val)

- ✓ should append multiple headers
- ✓ should accept array of values
- ✓ should get reset by res.set(field, val)
- ✓ should work with res.set(field, val) first
- ✓ should work together with res.cookie

res

.attachment()

- ✓ should Content-Disposition to attachment

.attachment(filename)

- ✓ should add the filename param
- ✓ should set the Content-Type

.attachment(utf8filename)

- ✓ should add the filename and filename* params
- ✓ should set the Content-Type

res

.clearCookie(name)

- ✓ should set a cookie passed expiry

.clearCookie(name, options)

- ✓ should set the given params
- ✓ should ignore maxAge
- ✓ should ignore user supplied expires param

res

.cookie(name, object)

- ✓ should generate a JSON cookie

.cookie(name, string)

- ✓ should set a cookie
- ✓ should allow multiple calls

.cookie(name, string, options)

- ✓ should set params

expires

- ✓ should throw on invalid date

partitioned

- ✓ should set partitioned

maxAge

- ✓ should set relative expires
- ✓ should set max-age
- ✓ should not mutate the options object
- ✓ should not throw on null
- ✓ should not throw on undefined
- ✓ should throw an error with invalid maxAge

priority

- ✓ should set low priority
- ✓ should set medium priority
- ✓ should set high priority
- ✓ should throw with invalid priority

signed

- ✓ should generate a signed JSON cookie

signed without secret

- ✓ should throw an error

.signedCookie(name, string)

- ✓ should set a signed cookie

res

.download(path)

- ✓ should transfer as an attachment
- ✓ should accept range requests
- ✓ should respond with requested byte range

.download(path, filename)

- ✓ should provide an alternate filename

.download(path, fn)

- ✓ should invoke the callback

async local storage

- ✓ should persist store
- ✓ should persist store on error

.download(path, options)

- ✓ should allow options to res.sendFile()

with "headers" option

- ✓ should set headers on response
- ✓ should use last header when duplicated
- ✓ should override Content-Type
- ✓ should not set headers on 404

when headers contains Content-Disposition

- ✓ should be ignored
- ✓ should be ignored case-insensitively

with "root" option

- ✓ should allow relative path
- ✓ should allow up within root
- ✓ should reject up outside root
- ✓ should reject reading outside root

.download(path, filename, fn)

- ✓ should invoke the callback

.download(path, filename, options, fn)

- ✓ should invoke the callback
- ✓ should allow options to res.sendFile()

when options.headers contains Content-Disposition

- ✓ should be ignored
- ✓ should be ignored case-insensitively

on failure

- ✓ should invoke the callback
- ✓ should remove Content-Disposition

res

.format(obj)

with canonicalized mime types

- ✓ should utilize qvalues in negotiation
- ✓ should allow wildcard type/subtypes
- ✓ should default the Content-Type
- ✓ should set the correct charset for the Content-Type
- ✓ should Vary: Accept

when Accept is not present

- ✓ should invoke the first callback

when no match is made

- ✓ should respond with 406 not acceptable

with extnames

- ✓ should utilize qvalues in negotiation
- ✓ should allow wildcard type/subtypes
- ✓ should default the Content-Type
- ✓ should set the correct charset for the Content-Type
- ✓ should Vary: Accept

when Accept is not present

- ✓ should invoke the first callback

when no match is made

- ✓ should respond with 406 not acceptable

with parameters

- ✓ should utilize qvalues in negotiation
- ✓ should allow wildcard type/subtypes
- ✓ should default the Content-Type
- ✓ should set the correct charset for the Content-Type
- ✓ should Vary: Accept

when Accept is not present

- ✓ should invoke the first callback

when no match is made

- ✓ should respond with 406 not acceptable

given .default

- ✓ should be invoked instead of auto-responding
- ✓ should work when only .default is provided
- ✓ should be able to invoke other formatter

in router

- ✓ should utilize qvalues in negotiation
- ✓ should allow wildcard type/subtypes
- ✓ should default the Content-Type
- ✓ should set the correct charset for the Content-Type
- ✓ should Vary: Accept

when Accept is not present

- ✓ should invoke the first callback

when no match is made

- ✓ should respond with 406 not acceptable

in router

- ✓ should utilize qvalues in negotiation
- ✓ should allow wildcard type/subtypes
- ✓ should default the Content-Type
- ✓ should set the correct charset for the Content-Type
- ✓ should Vary: Accept

when Accept is not present

- ✓ should invoke the first callback

when no match is made

- ✓ should respond with 406 not acceptable

res

.get(field)

- ✓ should get the response header field

res

.json(object)

- ✓ should not support jsonp callbacks
- ✓ should not override previous Content-Types

when given primitives

- ✓ should respond with json for null
- ✓ should respond with json for Number
- ✓ should respond with json for String

when given an array

- ✓ should respond with json

when given an object

- ✓ should respond with json

"json escape" setting

- ✓ should be undefined by default
- ✓ should unicode escape HTML-sniffing characters
- ✓ should not break undefined escape

"json replacer" setting

- ✓ should be passed to JSON.stringify()

"json spaces" setting

- ✓ should be undefined by default
- ✓ should be passed to JSON.stringify()

res

- ✓ should not override previous Content-Types

.jsonp(object)

- ✓ should respond with jsonp
- ✓ should use first callback parameter with jsonp
- ✓ should ignore object callback parameter with jsonp
- ✓ should allow renaming callback
- ✓ should allow []
- ✓ should disallow arbitrary js
- ✓ should escape utf whitespace
- ✓ should not escape utf whitespace for json fallback
- ✓ should include security header and prologue
- ✓ should not override previous Content-Types with no callback
- ✓ should override previous Content-Types with callback

when given undefined

- ✓ should invoke callback with no arguments

when given null

- ✓ should invoke callback with null

when given a string

- ✓ should invoke callback with a string

when given a number

- ✓ should invoke callback with a number

when given an array

- ✓ should invoke callback with an array

when given an object

- ✓ should invoke callback with an object

"json escape" setting

- ✓ should be undefined by default
- ✓ should unicode escape HTML-sniffing characters
- ✓ should not break undefined escape

"json replacer" setting

- ✓ should be passed to JSON.stringify()

"json spaces" setting

- ✓ should be undefined by default
- ✓ should be passed to JSON.stringify()

res

.links(obj)

- ✓ should set Link header field
- ✓ should set Link header field for multiple calls
- ✓ should set multiple links for single rel

res

- ✓ should work when mounted

.locals

- ✓ should be empty by default

res

.location(url)

- ✓ should set the header
- ✓ should preserve trailing slashes when not present
- ✓ should encode "url"
- ✓ should encode data uri1
- ✓ should encode data uri2
- ✓ should consistently handle non-string input: boolean
- ✓ should consistently handle non-string inputs: object
- ✓ should consistently handle non-string inputs: array
- ✓ should consistently handle empty string input
- ✓ should accept an instance of URL

location header encoding

- ✓ should not touch already-encoded sequences in "url"
- ✓ should consistently handle relative urls
- ✓ should not encode urls in such a way that they can bypass redirect

allow lists

- ✓ should not be case sensitive
- ✓ should work with https

- ✓ should correctly encode schemaless paths
- ✓ should keep backslashes in the path
- ✓ should escape header splitting for old node versions
- ✓ should encode unicode correctly
- ✓ should encode unicode correctly even with a bad host
- ✓ should work correctly despite using deprecated url.parse
- ✓ should encode file uri path

res

.redirect(url)

- ✓ should default to a 302 redirect
- ✓ should encode "url"
- ✓ should not touch already-encoded sequences in "url"

.redirect(status, url)

- ✓ should set the response status

when the request method is HEAD

- ✓ should ignore the body

when accepting html

- ✓ should respond with html
- ✓ should escape the url
- ✓ should not render evil javascript links in anchor href (prevent XSS)
- ✓ should include the redirect type

when accepting text

- ✓ should respond with text
- ✓ should encode the url
- ✓ should include the redirect type

when accepting neither text or html

- ✓ should respond with an empty body

res

.render(name)

- ✓ should support absolute paths
- ✓ should support absolute paths with "view engine"
- ✓ should error without "view engine" set and file extension to a non-

engine module

- ✓ should error without "view engine" set and no file extension
- ✓ should expose app.locals
- ✓ should expose app.locals with `name` property
- ✓ should support index.<engine>

when an error occurs

- ✓ should next(err)

when "view engine" is given

- ✓ should render the template

when "views" is given

- ✓ should lookup the file in the path

when array of paths

- ✓ should lookup the file in the path
- ✓ should lookup in later paths until found

.render(name, option)

- ✓ should render the template
- ✓ should expose app.locals
- ✓ should expose res.locals
- ✓ should give precedence to res.locals over app.locals
- ✓ should give precedence to res.render() locals over res.locals
- ✓ should give precedence to res.render() locals over app.locals

.render(name, options, fn)

- ✓ should pass the resulting string

.render(name, fn)

- ✓ should pass the resulting string

when an error occurs

- ✓ should pass it to the callback

res

- ✓ should always check regardless of length
- ✓ should respond with 304 Not Modified when fresh
- ✓ should not perform freshness check unless 2xx or 304
- ✓ should not support jsonp callbacks
- ✓ should be chainable

.send()

- ✓ should set body to ""

.send(null)

- ✓ should set body to ""

.send(undefined)

- ✓ should set body to ""

.send(Number)

- ✓ should send as application/json

.send(String)

- ✓ should send as html
- ✓ should set ETag
- ✓ should not override Content-Type
- ✓ should override charset in Content-Type
- ✓ should keep charset in Content-Type for Buffers

.send(Buffer)

- ✓ should send as octet-stream

- ✓ should set ETag
- ✓ should not override Content-Type
- ✓ should accept Uint8Array
- ✓ should not override ETag

.send(Object)

- ✓ should send as application/json

when the request method is HEAD

- ✓ should ignore the body

when .statusCode is 204

- ✓ should strip Content-* fields, Transfer-Encoding field, and body

when .statusCode is 205

- ✓ should strip Transfer-Encoding field and body, set Content-Length

when .statusCode is 304

- ✓ should strip Content-* fields, Transfer-Encoding field, and body

"etag" setting

when enabled

- ✓ should send ETag
- ✓ should send ETag in response to ACL request
- ✓ should send ETag in response to BIND request
- ✓ should send ETag in response to CHECKOUT request
- ✓ should send ETag in response to COPY request
- ✓ should send ETag in response to DELETE request
- ✓ should send ETag in response to GET request
- ✓ should send ETag in response to HEAD request
- ✓ should send ETag in response to LINK request
- ✓ should send ETag in response to LOCK request
- ✓ should send ETag in response to M-SEARCH request
- ✓ should send ETag in response to MERGE request
- ✓ should send ETag in response to MKACTIVITY request
- ✓ should send ETag in response to MKCALENDAR request
- ✓ should send ETag in response to MKCOL request
- ✓ should send ETag in response to MOVE request
- ✓ should send ETag in response to NOTIFY request
- ✓ should send ETag in response to OPTIONS request
- ✓ should send ETag in response to PATCH request
- ✓ should send ETag in response to POST request
- ✓ should send ETag in response to PROPFIND request
- ✓ should send ETag in response to PROPPATCH request
- ✓ should send ETag in response to PURGE request
- ✓ should send ETag in response to PUT request
- ✓ should send ETag in response to QUERY request
- ✓ should send ETag in response to REBIND request

- ✓ should send ETag in response to REPORT request
- ✓ should send ETag in response to SEARCH request
- ✓ should send ETag in response to SOURCE request
- ✓ should send ETag in response to SUBSCRIBE request
- ✓ should send ETag in response to TRACE request
- ✓ should send ETag in response to UNBIND request
- ✓ should send ETag in response to UNLINK request
- ✓ should send ETag in response to UNLOCK request
- ✓ should send ETag in response to UNSUBSCRIBE request
- ✓ should send ETag for empty string response
- ✓ should send ETag for long response
- ✓ should not override ETag when manually set
- ✓ should not send ETag for res.send()

when disabled

- ✓ should send no ETag
- ✓ should send ETag when manually set

when "strong"

- ✓ should send strong ETag

when "weak"

- ✓ should send weak ETag

when a function

- ✓ should send custom ETag
- ✓ should not send falsy ETag

res

.sendFile(path)

- ✓ should error missing path
- ✓ should error for non-string path
- ✓ should error for non-absolute path
- ✓ should transfer a file
- ✓ should transfer a file with special characters in string
- ✓ should include ETag
- ✓ should 304 when ETag matches
- ✓ should disable the ETag function if requested
- ✓ should 404 for directory
- ✓ should 404 when not found
- ✓ should send cache-control by default
- ✓ should not serve dotfiles by default
- ✓ should not override manual content-types
- ✓ should not error if the client aborts

.sendFile(path, fn)

- ✓ should invoke the callback when complete

- ✓ should invoke the callback when client aborts
- ✓ should invoke the callback when client already aborted
- ✓ should invoke the callback without error when HEAD
- ✓ should invoke the callback without error when 304
- ✓ should invoke the callback on 404

async local storage

- ✓ should persist store
- ✓ should persist store on error

.sendFile(path, options)

- ✓ should pass options to send module

with "acceptRanges" option

when true

- ✓ should advertise byte range accepted
- ✓ should respond to range request

when false

- ✓ should not advertise accept-ranges
- ✓ should not honor range requests

with "cacheControl" option

when true

- ✓ should send cache-control header

when false

- ✓ should not send cache-control header

with "dotfiles" option

when "allow"

- ✓ should allow dotfiles

when "deny"

- ✓ should deny dotfiles

when "ignore"

- ✓ should ignore dotfiles

with "headers" option

- ✓ should set headers on response
- ✓ should use last header when duplicated
- ✓ should override Content-Type
- ✓ should not set headers on 404

with "immutable" option

when true

- ✓ should send cache-control header with immutable

when false

- ✓ should not send cache-control header with immutable

with "lastModified" option

when true

- ✓ should send last-modified header

- ✓ should conditionally respond with if-modified-since when false

- ✓ should not have last-modified header

- ✓ should not honor if-modified-since

with "maxAge" option

- ✓ should set cache-control max-age to milliseconds

- ✓ should cap cache-control max-age to 1 year

- ✓ should min cache-control max-age to 0

- ✓ should floor cache-control max-age

when cacheControl: false

- ✓ should not send cache-control

when string

- ✓ should accept plain number as milliseconds

- ✓ should accept suffix "s" for seconds

- ✓ should accept suffix "m" for minutes

- ✓ should accept suffix "d" for days

with "root" option

- ✓ should allow relative path

- ✓ should allow up within root

- ✓ should reject up outside root

- ✓ should reject reading outside root

res

.setStatus(statusCode)

- ✓ should send the status code and message as body

- ✓ should work with unknown code

- ✓ should raise error for invalid status code

res

.set(field, value)

- ✓ should set the response header field

- ✓ should coerce to a string

.set(field, values)

- ✓ should set multiple response header fields

- ✓ should coerce to an array of strings

- ✓ should not set a charset if one is already set

- ✓ should throw when Content-Type is an array

.set(object)

- ✓ should set multiple fields

- ✓ should coerce to a string

res

`.status(code)`

- ✓ should set the status code when valid

accept valid ranges

- ✓ should set the response status code to 101
- ✓ should set the response status code to 201
- ✓ should set the response status code to 302
- ✓ should set the response status code to 403
- ✓ should set the response status code to 501
- ✓ should set the response status code to 700
- ✓ should set the response status code to 800
- ✓ should set the response status code to 900

invalid status codes

- ✓ should raise error for status code below 100
- ✓ should raise error for status code above 999
- ✓ should raise error for non-integer status codes
- ✓ should raise error for undefined status code
- ✓ should raise error for null status code
- ✓ should raise error for string status code
- ✓ should raise error for NaN status code

`res`

`.type(str)`

- ✓ should set the Content-Type based on a filename
- ✓ should default to application/octet-stream
- ✓ should set the Content-Type with type/subtype

`res.vary()`

with no arguments

- ✓ should throw error

with an empty array

- ✓ should not set Vary

with an array

- ✓ should set the values

with a string

- ✓ should set the value

when the value is present

- ✓ should not add it again

`utils.etag(body, encoding)`

- ✓ should support strings
- ✓ should support utf8 strings
- ✓ should support buffer

- ✓ should support empty string

utils.normalizeType acceptParams method

- ✓ should handle a type with a malformed parameter and break the loop in acceptParams

utils.setCharset(type, charset)

- ✓ should do anything without type
- ✓ should return type if not given charset
- ✓ should keep charset if not given charset
- ✓ should set charset
- ✓ should override charset

utils.wetag(body, encoding)

- ✓ should support strings
- ✓ should support utf8 strings
- ✓ should support buffer
- ✓ should support empty string

auth

GET /

- ✓ should redirect to /login

GET /login

- ✓ should render login form
- ✓ should display login error for bad user
- ✓ should display login error for bad password (56ms)

GET /logout

- ✓ should redirect to /

GET /restricted

- ✓ should redirect to /login without cookie
- ✓ should succeed with proper cookie (50ms)

POST /login

- ✓ should fail without proper username
- ✓ should fail without proper password (46ms)
- ✓ should succeed with proper credentials (45ms)

content-negotiation

GET /

- ✓ should default to text/html
- ✓ should accept to text/plain
- ✓ should accept to application/json

GET /users

- ✓ should default to text/html
- ✓ should accept to text/plain
- ✓ should accept to application/json

cookie-sessions

GET /

- ✓ should display no views
- ✓ should set a session cookie
- ✓ should display 1 view on revisit

cookies

GET /

- ✓ should have a form
- ✓ should respond with no cookies
- ✓ should respond to cookie

GET /forget

- ✓ should clear cookie

POST /

- ✓ should set a cookie
- ✓ should no set cookie w/o reminder

downloads

GET /

- ✓ should have a link to amazing.txt

GET /files/notes/groceries.txt

- ✓ should have a download header

GET /files/amazing.txt

- ✓ should have a download header

GET /files/missing.txt

- ✓ should respond with 404

GET /files/../../index.js

- ✓ should respond with 403

ejs

GET /

- ✓ should respond with html

error-pages

GET /

- ✓ should respond with page list

Accept: text/html

GET /403

- ✓ should respond with 403

GET /404

- ✓ should respond with 404

GET /500

- ✓ should respond with 500

Accept: application/json

GET /403

- ✓ should respond with 403

GET /404

- ✓ should respond with 404

GET /500

- ✓ should respond with 500

Accept: text/plain

GET /403

- ✓ should respond with 403

GET /404

- ✓ should respond with 404

GET /500

- ✓ should respond with 500

error

GET /

- ✓ should respond with 500

GET /next

- ✓ should respond with 500

GET /missing

- ✓ should respond with 404

hello-world

GET /

- ✓ should respond with hello world

GET /missing

- ✓ should respond with 404

markdown

GET /

- ✓ should respond with html (41ms)

GET /fail

- ✓ should respond with an error

multi-router

GET /

- ✓ should respond with root handler

GET /api/v1/

- ✓ should respond with APIv1 root handler

GET /api/v1/users

- ✓ should respond with users from APIv1

GET /api/v2/

- ✓ should respond with APIv2 root handler

GET /api/v2/users

- ✓ should respond with users from APIv2

mvc

GET /

- ✓ should redirect to /users

GET /pet/0

- ✓ should get pet

GET /pet/0/edit

- ✓ should get pet edit page

PUT /pet/2

- ✓ should update the pet

GET /users

- ✓ should display a list of users (128ms)

GET /user/:id

when present

- ✓ should display the user

- ✓ should display the users pets

when not present

- ✓ should 404

GET /user/:id/edit

- ✓ should display the edit form

PUT /user/:id

- ✓ should 500 on error

- ✓ should update the user

POST /user/:id/pet

- ✓ should create a pet for user

params

GET /

- ✓ should respond with instructions

GET /user/0

- ✓ should respond with a user

GET /user/9

- ✓ should fail to find user

```
GET /users/0-2
  ✓ should respond with three users
GET /users/foo-bar
  ✓ should fail integer parsing
```

resource

```
GET /
  ✓ should respond with instructions
GET /users
  ✓ should respond with all users
GET /users/1
  ✓ should respond with user 1
GET /users/9
  ✓ should respond with error
GET /users/1..3
  ✓ should respond with users 1 through 3
DELETE /users/1
  ✓ should delete user 1
DELETE /users/9
  ✓ should fail
GET /users/1..3.json
  ✓ should respond with users 2 and 3 as json
```

route-map

```
GET /users
  ✓ should respond with users
DELETE /users
  ✓ should delete users
GET /users/:id
  ✓ should get a user
GET /users/:id/pets
  ✓ should get a users pets
GET /users/:id/pets/:pid
  ✓ should get a users pet
```

route-separation

```
GET /
  ✓ should respond with index
GET /users
  ✓ should list users
GET /user/:id
  ✓ should get a user
```

- ✓ should 404 on missing user

GET /user/:id/view

- ✓ should get a user
- ✓ should 404 on missing user

GET /user/:id/edit

- ✓ should get a user to edit

PUT /user/:id/edit

- ✓ should edit a user

POST /user/:id/edit?_method=PUT

- ✓ should edit a user

GET /posts

Dashboard CI_nodejs_app #40

vhost

example.com

GET /

- ✓ should say hello

GET /foo

- ✓ should say foo

foo.example.com

GET /

- ✓ should redirect to /foo

bar.example.com

GET /

- ✓ should redirect to /bar

web-service

GET /api/users

without an api key

- ✓ should respond with 400 bad request

with an invalid api key

- ✓ should respond with 401 unauthorized

with a valid api key

- ✓ should respond users json

GET /api/repos

without an api key

- ✓ should respond with 400 bad request

with an invalid api key

- ✓ should respond with 401 unauthorized

with a valid api key

- ✓ should respond repos json

GET /api/user/:name/repos

without an api key

- ✓ should respond with 400 bad request

with an invalid api key

- ✓ should respond with 401 unauthorized

with a valid api key

- ✓ should respond user repos json
- ✓ should 404 with unknown user

when requesting an invalid route

- ✓ should respond with 404 json

1238 passing (12s)

```
[Pipeline] }
[Pipeline] // dir
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Building docker image and pushing image to my registry)
[Pipeline] script
[Pipeline] {
[Pipeline] withEnv
[Pipeline] {
[Pipeline] withDockerRegistry
$ docker login -u raghumn -p ***** https://index.docker.io/v1/
WARNING! Using --password via the CLI is insecure. Use --password-stdin.
```

WARNING! Your credentials are stored unencrypted in
'/var/jenkins_home/nodejs-app@tmp/fbf246f7-dcda-43ce-8776-659d91290b53/config.json'.

Configure a credential helper to remove this warning. See
<https://docs.docker.com/go/credential-store/>

Login Succeeded

```
[Pipeline] {
[Pipeline] isUnix
[Pipeline] withEnv
[Pipeline] {
[Pipeline] sh
+ docker build -t raghumn/nodejsapp:40 .
```

DEPRECATED: The legacy builder is deprecated and will be removed in a future release.

Install the buildx component to build images with BuildKit:

<https://docs.docker.com/go/buildx/>

Sending build context to Docker daemon 49.24MB

Step 1/9 : FROM node:24.5.0-alpine

---> 0b691ce23fac

Step 2/9 : RUN mkdir /apps

---> Using cache

---> 66797407e19b

Step 3/9 : WORKDIR /apps

---> Using cache

---> c782b80cb042

Step 4/9 : COPY express .

---> Using cache

---> 8e285b2af425

Step 5/9 : RUN chown -R node:node /apps

---> Using cache

---> a54c46120668

Step 6/9 : USER node

---> Using cache

---> 4215946aa6ad

Step 7/9 : RUN npm install

---> Using cache

---> c1b7cdd21310

Step 8/9 : EXPOSE 3000

---> Using cache

---> 767fb090976a

Step 9/9 : CMD ["node","examples/hello-world/"]

---> Using cache

---> ebd2b5fd29e4

Successfully built ebd2b5fd29e4

Successfully tagged raghumn/nodejsapp:40

[Pipeline] }

[Pipeline] // withEnv

[Pipeline] isUnix

[Pipeline] withEnv

[Pipeline] {

[Pipeline] sh

+ docker tag raghumn/nodejsapp:40 index.docker.io/raghumn/nodejsapp:40

[Pipeline] }

```
[Pipeline] // withEnv
[Pipeline] isUnix
[Pipeline] withEnv
[Pipeline] {
[Pipeline] sh
+ docker push index.docker.io/raghumn/nodejsapp:40
The push refers to repository [docker.io/raghumn/nodejsapp]
53cffa8011b2: Preparing
4f1d43df96e0: Preparing
1fe22ac45d58: Preparing
b1847263dcf4: Preparing
7a33222fb068: Preparing
12f4d6be233e: Preparing
d0596b02df10: Preparing
418dccb7d85a: Preparing
12f4d6be233e: Waiting
d0596b02df10: Waiting
418dccb7d85a: Waiting
b1847263dcf4: Layer already exists
4f1d43df96e0: Layer already exists
1fe22ac45d58: Layer already exists
53cffa8011b2: Layer already exists
7a33222fb068: Layer already exists
12f4d6be233e: Layer already exists
418dccb7d85a: Layer already exists
d0596b02df10: Layer already exists
40: digest:
sha256:f946210e5607b301580487667bb77b0532acda65bfe49916ea0297d9eff1e575 size:
1996
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // withDockerRegistry
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // script
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Getting previous image tag)
```

```
[Pipeline] script
[Pipeline] {
[Pipeline] echo
Previous build number for rollback if deployment fails: 39
[Pipeline] }
[Pipeline] // script
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Declarative: Post Actions)
[Pipeline] script
[Pipeline] {
[Pipeline] }
[Pipeline] // script
[Pipeline] echo
CI passed. Triggering CD job...
[Pipeline] build (Scheduling CD_nodejs_app)
Scheduling project: CD\_nodejs\_app
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // ws
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```