

1.10.1

Fixes

- Update dependencies (#389) [9f5eedc]

1.10.0

Features

- Add HaveHTTPStatusMatcher (#378) [f335c94]
- Changed matcher for content-type in VerifyJSONRepresenting (#377) [6024f5b]
- Make ghttp usable with x-unit style tests (#376) [c0be499]
- Implement PanicWith matcher (#381) [f8032b4]

1.9.0

Features

- Add ContainElements matcher (#370) [2f57380]
- Output missing and extra elements in ConsistOf failure message [a31eda7]
- Document method LargestMatching [7c5a280]

1.8.1

Fixes

- Fix unexpected MatchError() behaviour (#375) [8ae7b2f]

1.8.0

Features

- Allow optional description to be lazily evaluated function (#364) [bf64010]
- Support wrapped errors (#359) [0a981cb]

1.7.1

Fixes

- Bump go-yaml version to cover fixed ddos heuristic (#362) [95e431e]

1.7.0

Features

- export format property variables (#347) [642e5ba]

Fixes

- minor fix in the documentation of ExpectWithOffset (#358) [beea727]

1.6.0

Features

- Display special chars on error [41e1b26]

- Add BeElementOf matcher [6a48b48]

Fixes

- Remove duplication in XML matcher tests [cc1a6cb]
- Remove unnecessary conversions (#357) [7bf756a]
- Fixed import order (#353) [2e3b965]
- Added missing error handling in test (#355) [c98d3eb]
- Simplify code (#356) [0001ed9]
- Simplify code (#354) [0d9100e]
- Fixed typos (#352) [3f647c4]
- Add failure message tests to BeElementOf matcher [efe19c3]
- Update go-testcov untested sections [37ee382]
- Mark all uncovered files so go-testcov ./... works [53b150e]
- Reenable gotip in travis [5c249dc]
- Fix the typo of comment (#345) [f0e010e]
- Optimize contain_element_matcher [abeb93d]

1.5.0

Features

- Added MatchKeys matchers [8b909fc]

Fixes and Minor Improvements

- Add type aliases to remove stuttering [03b0461]
- Don't run session_test.go on windows (#324) [5533ce8]

1.4.3

Fixes:

- ensure file name and line numbers are correctly reported for XUnit [6fff58f]
- Fixed matcher for content-type (#305) [69d9b43]

1.4.2

Fixes:

- Add go.mod and go.sum files to define the gomega go module [f3de367, a085d30]
- Work around go vet issue with Go v1.11 (#300) [40dd6ad]
- Better output when using with go XUnit-style tests, fixes #255 (#297) [29a4b97]
- Fix MatchJSON fail to parse json.RawMessage (#298) [ae19f1b]
- show threshold in failure message of BeNumericallyMatcher (#293) [4bbecc8]

1.4.1

Fixes:

- Update documentation formatting and examples (#289) [9be8410]
- allow 'Receive' matcher to be used with concrete types (#286) [41673fd]
- Fix data race in ghttp server (#283) [7ac6b01]
- Travis badge should only show master [cc102ab]

1.4.0

Features

- Make string pretty diff user configurable (#273) [eb112ce, 649b44d]

Fixes

- Use `httputil.DumpRequest` to pretty-print unhandled requests (#278) [a4ff0fc, b7d1a52]
- fix typo `float32 > float32` (#272) [041ae3b, 6e33911]
- Fix link to documentation on adding your own matchers (#270) [bb2c830, fcebc62]
- Use setters and getters to avoid race condition (#262) [13057c3, a9c79f1]
- Avoid sending a signal if the process is not alive (#259) [b8043e5, 4fc1762]
- Improve message from `AssignableToTypeOf` when expected value is nil (#281) [9c1fb20]

1.3.0

Improvements:

- The `Equal` matcher matches byte slices more performantly.
- Improved how `MatchError` matches error strings.
- `MatchXML` ignores the order of xml node attributes.
- Improve support for XUnit style golang tests. ([#254](#))

Bug Fixes:

- Diff generation now handles multi-byte sequences correctly.
- Multiple goroutines can now call `gexec.Build` concurrently.

1.2.0

Improvements:

- Added `BeSent` which attempts to send a value down a channel and fails if the attempt blocks. Can be paired with `Eventually` to safely send a value down a channel with a timeout.
- `Ω`, `Expect`, `Eventually`, and `Consistently` now immediately `panic` if there is no registered fail handler. This is always a mistake that can hide failing tests.
- `Receive()` no longer errors when passed a closed channel, it's perfectly fine to attempt to read from a closed channel so `Ω(c).Should(Receive())` always fails and `Ω(c).ShouldNot(Receive())` always passes with a closed channel.
- Added `HavePrefix` and `HaveSuffix` matchers.
- `ghhttp` can now handle concurrent requests.
- Added `Succeed` which allows one to write `Ω(MyFunction()).Should(Succeed())`.
- Improved `ghhttp`'s behavior around failing assertions and panics:
 - If a registered handler makes a failing assertion `ghhttp` will return `500`.
 - If a registered handler panics, `ghhttp` will return `500` and fail the test. This is new behavior that may cause existing code to break. This code is almost certainly incorrect and creating a false positive.
- `ghhttp` servers can take an `io.Writer`. `ghhttp` will write a line to the writer when each request arrives.
- Added `WithTransform` matcher to allow munging input data before feeding into the relevant matcher
- Added boolean `And`, `Or`, and `Not` matchers to allow creating composite matchers

- Added `gbytes.TimeoutCloser` , `gbytes.TimeoutReader` , and `gbytes.TimeoutWriter` - these are convenience wrappers that timeout if the underlying Closer/Reader/Writer does not return within the allotted time.
- Added `gbytes.BufferedReader` - this constructs a `gbytes.Buffer` that asynchronously reads the passed-in `io.Reader` into its buffer.

Bug Fixes:

- `gexec`: `session.Wait` now uses `EventuallyWithOffset` to get the right line number in the failure.
- `ContainElement` no longer bails if a passed-in matcher errors.

1.0 (8/2/2014)

No changes. Dropping "beta" from the version number.

1.0.0-beta (7/8/2014)

Breaking Changes:

- Changed `OmegaMatcher` interface. Instead of having `Match` return failure messages, two new methods `FailureMessage` and `NegatedFailureMessage` are called instead.
- Moved and renamed `OmegaFailHandler` to `types.GomegaFailHandler` and `OmegaMatcher` to `types.GomegaMatcher`. Any references to `OmegaMatcher` in any custom matchers will need to be changed to point to `types.GomegaMatcher`

New Test-Support Features:

- `ghhttp` : supports testing http clients
 - Provides a flexible fake http server
 - Provides a collection of chainable http handlers that perform assertions.
- `gbytes` : supports making ordered assertions against streams of data
 - Provides a `gbytes.Buffer`
 - Provides a `Say` matcher to perform ordered assertions against output data
- `gexec` : supports testing external processes
 - Provides support for building Go binaries
 - Wraps and starts `exec.Cmd` commands
 - Makes it easy to assert against stdout and stderr
 - Makes it easy to send signals and wait for processes to exit
 - Provides an `Exit` matcher to assert against exit code.

DSL Changes:

- `Eventually` and `Consistently` can accept `time.Duration` interval and polling inputs.
- The default timeouts for `Eventually` and `Consistently` are now configurable.

New Matchers:

- `ConsistOf` : order-independent assertion against the elements of an array/slice or keys of a map.
- `BeTemporally` : like `BeNumerically` but for `time.Time`
- `HaveKeyWithValue` : asserts a map has a given key with the given value.

Updated Matchers:

- `Receive` matcher can take a matcher as an argument and passes only if the channel under test receives an object that satisfies the passed-in matcher.
- Matchers that implement `MatchMayChangeInTheFuture(actual interface{}) bool` can inform `Eventually` and/or `Consistently` when a match has no chance of changing status in the future. For example, `Receive` returns `false` when a channel is closed.

Misc:

- Start using semantic versioning
- Start maintaining changelog

Major refactor:

- Pull out Gomega's internal to `internal`