# Change Log

Please refer to [UPGRADING](http://upgrading.md) guide for upgrading to a major version.

## 7.9.2 - 2024-07-24

### Fixed

* Adjusted handler selection to use cURL if its version is 7.21.2 or higher, rather than 7.34.0

## 7.9.1 - 2024-07-19

### Fixed

* Fix TLS 1.3 check for HTTP/2 requests

## 7.9.0 - 2024-07-18

### Changed

* Improve protocol version checks to provide feedback around unsupported protocols
* Only select the cURL handler by default if 7.34.0 or higher is linked
* Improved CurlMultiHandler to avoid busy wait if possible
* Dropped support for EOL guzzlehttp/psr7 v1
* Improved URI user info redaction in errors

## 7.8.2 - 2024-07-18

### Added

* Support for PHP 8.4

## 7.8.1 - 2023-12-03

### Changed

* Updated links in docs to their canonical versions
* Replaced call\_user\_func\* with native calls

## 7.8.0 - 2023-08-27

### Added

* Support for PHP 8.3
* Added automatic closing of handles on CurlFactory object destruction

## 7.7.1 - 2023-08-27

### Changed

* Remove the need for AllowDynamicProperties in CurlMultiHandler

## 7.7.0 - 2023-05-21

### Added

* Support guzzlehttp/promises v2

## 7.6.1 - 2023-05-15

### Fixed

* Fix SetCookie::fromString MaxAge deprecation warning and skip invalid MaxAge values

## 7.6.0 - 2023-05-14

### Added

* Support for setting the minimum TLS version in a unified way
* Apply on request the version set in options parameters

## 7.5.2 - 2023-05-14

### Fixed

* Fixed set cookie constructor validation
* Fixed handling of files with '0' body

### Changed

* Corrected docs and default connect timeout value to 300 seconds

## 7.5.1 - 2023-04-17

### Fixed

* Fixed NO\_PROXY settings so that setting the proxy option to no overrides the env variable

### Changed

* Adjusted guzzlehttp/psr7 version constraint to ^1.9.1 || ^2.4.5

## 7.5.0 - 2022-08-28

### Added

* Support PHP 8.2
* Add request to delay closure params

## 7.4.5 - 2022-06-20

### Fixed

* Fix change in port should be considered a change in origin
* Fix CURLOPT\_HTTPAUTH option not cleared on change of origin

## 7.4.4 - 2022-06-09

### Fixed

* Fix failure to strip Authorization header on HTTP downgrade
* Fix failure to strip the Cookie header on change in host or HTTP downgrade

## 7.4.3 - 2022-05-25

### Fixed

* Fix cross-domain cookie leakage

## 7.4.2 - 2022-03-20

### Fixed

* Remove curl auth on cross-domain redirects to align with the Authorization HTTP header
* Reject non-HTTP schemes in StreamHandler
* Set a default ssl.peer\_name context in StreamHandler to allow force\_ip\_resolve

## 7.4.1 - 2021-12-06

### Changed

* Replaced implicit URI to string coercion [#2946](https://github.com/guzzle/guzzle/pull/2946)
* Allow symfony/deprecation-contracts version 3 [#2961](https://github.com/guzzle/guzzle/pull/2961)

### Fixed

* Only close curl handle if it's done [#2950](https://github.com/guzzle/guzzle/pull/2950)

## 7.4.0 - 2021-10-18

### Added

* Support PHP 8.1 [#2929](https://github.com/guzzle/guzzle/pull/2929), [#2939](https://github.com/guzzle/guzzle/pull/2939)
* Support psr/log version 2 and 3 [#2943](https://github.com/guzzle/guzzle/pull/2943)

### Fixed

* Make sure we always call restore\_error\_handler() [#2915](https://github.com/guzzle/guzzle/pull/2915)
* Fix progress parameter type compatibility between the cURL and stream handlers [#2936](https://github.com/guzzle/guzzle/pull/2936)
* Throw InvalidArgumentException when an incorrect headers array is provided [#2916](https://github.com/guzzle/guzzle/pull/2916), [#2942](https://github.com/guzzle/guzzle/pull/2942)

### Changed

* Be more strict with types [#2914](https://github.com/guzzle/guzzle/pull/2914), [#2917](https://github.com/guzzle/guzzle/pull/2917), [#2919](https://github.com/guzzle/guzzle/pull/2919), [#2945](https://github.com/guzzle/guzzle/pull/2945)

## 7.3.0 - 2021-03-23

### Added

* Support for DER and P12 certificates [#2413](https://github.com/guzzle/guzzle/pull/2413)
* Support the cURL (http://) scheme for StreamHandler proxies [#2850](https://github.com/guzzle/guzzle/pull/2850)
* Support for guzzlehttp/psr7:^2.0 [#2878](https://github.com/guzzle/guzzle/pull/2878)

### Fixed

* Handle exceptions on invalid header consistently between PHP versions and handlers [#2872](https://github.com/guzzle/guzzle/pull/2872)

## 7.2.0 - 2020-10-10

### Added

* Support for PHP 8 [#2712](https://github.com/guzzle/guzzle/pull/2712), [#2715](https://github.com/guzzle/guzzle/pull/2715), [#2789](https://github.com/guzzle/guzzle/pull/2789)
* Support passing a body summarizer to the http errors middleware [#2795](https://github.com/guzzle/guzzle/pull/2795)

### Fixed

* Handle exceptions during response creation [#2591](https://github.com/guzzle/guzzle/pull/2591)
* Fix CURLOPT\_ENCODING not to be overwritten [#2595](https://github.com/guzzle/guzzle/pull/2595)
* Make sure the Request always has a body object [#2804](https://github.com/guzzle/guzzle/pull/2804)

### Changed

* The TooManyRedirectsException has a response [#2660](https://github.com/guzzle/guzzle/pull/2660)
* Avoid "functions" from dependencies [#2712](https://github.com/guzzle/guzzle/pull/2712)

### Deprecated

* Using environment variable GUZZLE\_CURL\_SELECT\_TIMEOUT [#2786](https://github.com/guzzle/guzzle/pull/2786)

## 7.1.1 - 2020-09-30

### Fixed

* Incorrect EOF detection for response body streams on Windows.

### Changed

* We dont connect curl sink on HEAD requests.
* Removed some PHP 5 workarounds

## 7.1.0 - 2020-09-22

### Added

* GuzzleHttp\MessageFormatterInterface

### Fixed

* Fixed issue that caused cookies with no value not to be stored.
* On redirects, we allow all safe methods like GET, HEAD and OPTIONS.
* Fixed logging on empty responses.
* Make sure MessageFormatter::format returns string

### Deprecated

* All functions in GuzzleHttp has been deprecated. Use static methods on Utils instead.
* ClientInterface::getConfig()
* Client::getConfig()
* Client::\_\_call()
* Utils::defaultCaBundle()
* CurlFactory::LOW\_CURL\_VERSION\_NUMBER

## 7.0.1 - 2020-06-27

* Fix multiply defined functions fatal error [#2699](https://github.com/guzzle/guzzle/pull/2699)

## 7.0.0 - 2020-06-27

No changes since 7.0.0-rc1.

## 7.0.0-rc1 - 2020-06-15

### Changed

* Use error level for logging errors in Middleware [#2629](https://github.com/guzzle/guzzle/pull/2629)
* Disabled IDN support by default and require ext-intl to use it [#2675](https://github.com/guzzle/guzzle/pull/2675)

## 7.0.0-beta2 - 2020-05-25

### Added

* Using Utils class instead of functions in the GuzzleHttp namespace. [#2546](https://github.com/guzzle/guzzle/pull/2546)
* ClientInterface::MAJOR\_VERSION [#2583](https://github.com/guzzle/guzzle/pull/2583)

### Changed

* Avoid the getenv function when unsafe [#2531](https://github.com/guzzle/guzzle/pull/2531)
* Added real client methods [#2529](https://github.com/guzzle/guzzle/pull/2529)
* Avoid functions due to global install conflicts [#2546](https://github.com/guzzle/guzzle/pull/2546)
* Use Symfony intl-idn polyfill [#2550](https://github.com/guzzle/guzzle/pull/2550)
* Adding methods for HTTP verbs like Client::get(), Client::head(), Client::patch() etc [#2529](https://github.com/guzzle/guzzle/pull/2529)
* ConnectException extends TransferException [#2541](https://github.com/guzzle/guzzle/pull/2541)
* Updated the default User Agent to "GuzzleHttp/7" [#2654](https://github.com/guzzle/guzzle/pull/2654)

### Fixed

* Various intl icu issues [#2626](https://github.com/guzzle/guzzle/pull/2626)

### Removed

* Pool option pool\_size [#2528](https://github.com/guzzle/guzzle/pull/2528)

## 7.0.0-beta1 - 2019-12-30

The diff might look very big but 95% of Guzzle users will be able to upgrade without modification. Please see [the upgrade document](http://upgrading.md) that describes all BC breaking changes.

### Added

* Implement PSR-18 and dropped PHP 5 support [#2421](https://github.com/guzzle/guzzle/pull/2421) [#2474](https://github.com/guzzle/guzzle/pull/2474)
* PHP 7 types [#2442](https://github.com/guzzle/guzzle/pull/2442) [#2449](https://github.com/guzzle/guzzle/pull/2449) [#2466](https://github.com/guzzle/guzzle/pull/2466) [#2497](https://github.com/guzzle/guzzle/pull/2497) [#2499](https://github.com/guzzle/guzzle/pull/2499)
* IDN support for redirects [2424](https://github.com/guzzle/guzzle/pull/2424)

### Changed

* Dont allow passing null as third argument to BadResponseException::\_\_construct() [#2427](https://github.com/guzzle/guzzle/pull/2427)
* Use SAPI constant instead of method call [#2450](https://github.com/guzzle/guzzle/pull/2450)
* Use native function invocation [#2444](https://github.com/guzzle/guzzle/pull/2444)
* Better defaults for PHP installations with old ICU lib [2454](https://github.com/guzzle/guzzle/pull/2454)
* Added visibility to all constants [#2462](https://github.com/guzzle/guzzle/pull/2462)
* Dont allow passing null as URI to Client::request() and Client::requestAsync() [#2461](https://github.com/guzzle/guzzle/pull/2461)
* Widen the exception argument to throwable [#2495](https://github.com/guzzle/guzzle/pull/2495)

### Fixed

* Logging when Promise rejected with a string [#2311](https://github.com/guzzle/guzzle/pull/2311)

### Removed

* Class SeekException [#2162](https://github.com/guzzle/guzzle/pull/2162)
* RequestException::getResponseBodySummary() [#2425](https://github.com/guzzle/guzzle/pull/2425)
* CookieJar::getCookieValue() [#2433](https://github.com/guzzle/guzzle/pull/2433)
* uri\_template() and UriTemplate [#2440](https://github.com/guzzle/guzzle/pull/2440)
* Request options save\_to and exceptions [#2464](https://github.com/guzzle/guzzle/pull/2464)

## 6.5.2 - 2019-12-23

* idn\_to\_ascii() fix for old PHP versions [#2489](https://github.com/guzzle/guzzle/pull/2489)

## 6.5.1 - 2019-12-21

* Better defaults for PHP installations with old ICU lib [#2454](https://github.com/guzzle/guzzle/pull/2454)
* IDN support for redirects [#2424](https://github.com/guzzle/guzzle/pull/2424)

## 6.5.0 - 2019-12-07

* Improvement: Added support for reset internal queue in MockHandler. [#2143](https://github.com/guzzle/guzzle/pull/2143)
* Improvement: Added support to pass arbitrary options to curl\_multi\_init. [#2287](https://github.com/guzzle/guzzle/pull/2287)
* Fix: Gracefully handle passing null to the header option. [#2132](https://github.com/guzzle/guzzle/pull/2132)
* Fix: RetryMiddleware did not do exponential delay between retires due unit mismatch. [#2132](https://github.com/guzzle/guzzle/pull/2132)
* Fix: Prevent undefined offset when using array for ssl\_key options. [#2348](https://github.com/guzzle/guzzle/pull/2348)
* Deprecated ClientInterface::VERSION

## 6.4.1 - 2019-10-23

* No guzzle.phar was created in 6.4.0 due expired API token. This release will fix that
* Added parent::\_\_construct() to FileCookieJar and SessionCookieJar

## 6.4.0 - 2019-10-23

* Improvement: Improved error messages when using curl < 7.21.2 [#2108](https://github.com/guzzle/guzzle/pull/2108)
* Fix: Test if response is readable before returning a summary in RequestException::getResponseBodySummary() [#2081](https://github.com/guzzle/guzzle/pull/2081)
* Fix: Add support for GUZZLE\_CURL\_SELECT\_TIMEOUT environment variable [#2161](https://github.com/guzzle/guzzle/pull/2161)
* Improvement: Added GuzzleHttp\Exception\InvalidArgumentException [#2163](https://github.com/guzzle/guzzle/pull/2163)
* Improvement: Added GuzzleHttp\\_current\_time() to use hrtime() if that function exists. [#2242](https://github.com/guzzle/guzzle/pull/2242)
* Improvement: Added curl's appconnect\_time in TransferStats [#2284](https://github.com/guzzle/guzzle/pull/2284)
* Improvement: Make GuzzleException extend Throwable wherever it's available [#2273](https://github.com/guzzle/guzzle/pull/2273)
* Fix: Prevent concurrent writes to file when saving CookieJar [#2335](https://github.com/guzzle/guzzle/pull/2335)
* Improvement: Update MockHandler so we can test transfer time [#2362](https://github.com/guzzle/guzzle/pull/2362)

## 6.3.3 - 2018-04-22

* Fix: Default headers when decode\_content is specified

## 6.3.2 - 2018-03-26

* Fix: Release process

## 6.3.1 - 2018-03-26

* Bug fix: Parsing 0 epoch expiry times in cookies [#2014](https://github.com/guzzle/guzzle/pull/2014)
* Improvement: Better ConnectException detection [#2012](https://github.com/guzzle/guzzle/pull/2012)
* Bug fix: Malformed domain that contains a "/" [#1999](https://github.com/guzzle/guzzle/pull/1999)
* Bug fix: Undefined offset when a cookie has no first key-value pair [#1998](https://github.com/guzzle/guzzle/pull/1998)
* Improvement: Support PHPUnit 6 [#1953](https://github.com/guzzle/guzzle/pull/1953)
* Bug fix: Support empty headers [#1915](https://github.com/guzzle/guzzle/pull/1915)
* Bug fix: Ignore case during header modifications [#1916](https://github.com/guzzle/guzzle/pull/1916)
* Minor code cleanups, documentation fixes and clarifications.

## 6.3.0 - 2017-06-22

* Feature: force IP resolution (ipv4 or ipv6) [#1608](https://github.com/guzzle/guzzle/pull/1608), [#1659](https://github.com/guzzle/guzzle/pull/1659)
* Improvement: Don't include summary in exception message when body is empty [#1621](https://github.com/guzzle/guzzle/pull/1621)
* Improvement: Handle on\_headers option in MockHandler [#1580](https://github.com/guzzle/guzzle/pull/1580)
* Improvement: Added SUSE Linux CA path [#1609](https://github.com/guzzle/guzzle/issues/1609)
* Improvement: Use class reference for getting the name of the class instead of using hardcoded strings [#1641](https://github.com/guzzle/guzzle/pull/1641)
* Feature: Added read\_timeout option [#1611](https://github.com/guzzle/guzzle/pull/1611)
* Bug fix: PHP 7.x fixes [#1685](https://github.com/guzzle/guzzle/pull/1685), [#1686](https://github.com/guzzle/guzzle/pull/1686), [#1811](https://github.com/guzzle/guzzle/pull/1811)
* Deprecation: BadResponseException instantiation without a response [#1642](https://github.com/guzzle/guzzle/pull/1642)
* Feature: Added NTLM auth [#1569](https://github.com/guzzle/guzzle/pull/1569)
* Feature: Track redirect HTTP status codes [#1711](https://github.com/guzzle/guzzle/pull/1711)
* Improvement: Check handler type during construction [#1745](https://github.com/guzzle/guzzle/pull/1745)
* Improvement: Always include the Content-Length if there's a body [#1721](https://github.com/guzzle/guzzle/pull/1721)
* Feature: Added convenience method to access a cookie by name [#1318](https://github.com/guzzle/guzzle/pull/1318)
* Bug fix: Fill CURLOPT\_CAPATH and CURLOPT\_CAINFO properly [#1684](https://github.com/guzzle/guzzle/pull/1684)
* Improvement: Use \GuzzleHttp\Promise\rejection\_for function instead of object init [#1827](https://github.com/guzzle/guzzle/pull/1827)
* Minor code cleanups, documentation fixes and clarifications.

## 6.2.3 - 2017-02-28

* Fix deprecations with guzzle/psr7 version 1.4

## 6.2.2 - 2016-10-08

* Allow to pass nullable Response to delay callable
* Only add scheme when host is present
* Fix drain case where content-length is the literal string zero
* Obfuscate in-URL credentials in exceptions

## 6.2.1 - 2016-07-18

* Address HTTP\_PROXY security vulnerability, CVE-2016-5385: <https://httpoxy.org/>
* Fixing timeout bug with StreamHandler: <https://github.com/guzzle/guzzle/pull/1488>
* Only read up to Content-Length in PHP StreamHandler to avoid timeouts when a server does not honor Connection: close.
* Ignore URI fragment when sending requests.

## 6.2.0 - 2016-03-21

* Feature: added GuzzleHttp\json\_encode and GuzzleHttp\json\_decode. <https://github.com/guzzle/guzzle/pull/1389>
* Bug fix: Fix sleep calculation when waiting for delayed requests. <https://github.com/guzzle/guzzle/pull/1324>
* Feature: More flexible history containers. <https://github.com/guzzle/guzzle/pull/1373>
* Bug fix: defer sink stream opening in StreamHandler. <https://github.com/guzzle/guzzle/pull/1377>
* Bug fix: do not attempt to escape cookie values. <https://github.com/guzzle/guzzle/pull/1406>
* Feature: report original content encoding and length on decoded responses. <https://github.com/guzzle/guzzle/pull/1409>
* Bug fix: rewind seekable request bodies before dispatching to cURL. <https://github.com/guzzle/guzzle/pull/1422>
* Bug fix: provide an empty string to http\_build\_query for HHVM workaround. <https://github.com/guzzle/guzzle/pull/1367>

## 6.1.1 - 2015-11-22

* Bug fix: Proxy::wrapSync() now correctly proxies to the appropriate handler <https://github.com/guzzle/guzzle/commit/911bcbc8b434adce64e223a6d1d14e9a8f63e4e4>
* Feature: HandlerStack is now more generic. <https://github.com/guzzle/guzzle/commit/f2102941331cda544745eedd97fc8fd46e1ee33e>
* Bug fix: setting verify to false in the StreamHandler now disables peer verification. <https://github.com/guzzle/guzzle/issues/1256>
* Feature: Middleware now uses an exception factory, including more error context. <https://github.com/guzzle/guzzle/pull/1282>
* Feature: better support for disabled functions. <https://github.com/guzzle/guzzle/pull/1287>
* Bug fix: fixed regression where MockHandler was not using sink. <https://github.com/guzzle/guzzle/pull/1292>

## 6.1.0 - 2015-09-08

* Feature: Added the on\_stats request option to provide access to transfer statistics for requests. <https://github.com/guzzle/guzzle/pull/1202>
* Feature: Added the ability to persist session cookies in CookieJars. <https://github.com/guzzle/guzzle/pull/1195>
* Feature: Some compatibility updates for Google APP Engine <https://github.com/guzzle/guzzle/pull/1216>
* Feature: Added support for NO\_PROXY to prevent the use of a proxy based on a simple set of rules. <https://github.com/guzzle/guzzle/pull/1197>
* Feature: Cookies can now contain square brackets. <https://github.com/guzzle/guzzle/pull/1237>
* Bug fix: Now correctly parsing = inside of quotes in Cookies. <https://github.com/guzzle/guzzle/pull/1232>
* Bug fix: Cusotm cURL options now correctly override curl options of the same name. <https://github.com/guzzle/guzzle/pull/1221>
* Bug fix: Content-Type header is now added when using an explicitly provided multipart body. <https://github.com/guzzle/guzzle/pull/1218>
* Bug fix: Now ignoring Set-Cookie headers that have no name.
* Bug fix: Reason phrase is no longer cast to an int in some cases in the cURL handler. <https://github.com/guzzle/guzzle/pull/1187>
* Bug fix: Remove the Authorization header when redirecting if the Host header changes. <https://github.com/guzzle/guzzle/pull/1207>
* Bug fix: Cookie path matching fixes <https://github.com/guzzle/guzzle/issues/1129>
* Bug fix: Fixing the cURL body\_as\_string setting <https://github.com/guzzle/guzzle/pull/1201>
* Bug fix: quotes are no longer stripped when parsing cookies. <https://github.com/guzzle/guzzle/issues/1172>
* Bug fix: form\_params and query now always uses the & separator. <https://github.com/guzzle/guzzle/pull/1163>
* Bug fix: Adding a Content-Length to PHP stream wrapper requests if not set. <https://github.com/guzzle/guzzle/pull/1189>

## 6.0.2 - 2015-07-04

* Fixed a memory leak in the curl handlers in which references to callbacks were not being removed by curl\_reset.
* Cookies are now extracted properly before redirects.
* Cookies now allow more character ranges.
* Decoded Content-Encoding responses are now modified to correctly reflect their state if the encoding was automatically removed by a handler. This means that the Content-Encoding header may be removed an the Content-Length modified to reflect the message size after removing the encoding.
* Added a more explicit error message when trying to use form\_params and multipart in the same request.
* Several fixes for HHVM support.
* Functions are now conditionally required using an additional level of indirection to help with global Composer installations.

## 6.0.1 - 2015-05-27

* Fixed a bug with serializing the query request option where the & separator was missing.
* Added a better error message for when body is provided as an array. Please use form\_params or multipart instead.
* Various doc fixes.

## 6.0.0 - 2015-05-26

* See the UPGRADING.md document for more information.
* Added multipart and form\_params request options.
* Added synchronous request option.
* Added the on\_headers request option.
* Fixed expect handling.
* No longer adding default middlewares in the client ctor. These need to be present on the provided handler in order to work.
* Requests are no longer initiated when sending async requests with the CurlMultiHandler. This prevents unexpected recursion from requests completing while ticking the cURL loop.
* Removed the semantics of setting default to true. This is no longer required now that the cURL loop is not ticked for async requests.
* Added request and response logging middleware.
* No longer allowing self signed certificates when using the StreamHandler.
* Ensuring that sink is valid if saving to a file.
* Request exceptions now include a "handler context" which provides handler specific contextual information.
* Added GuzzleHttp\RequestOptions to allow request options to be applied using constants.
* $maxHandles has been removed from CurlMultiHandler.
* MultipartPostBody is now part of the guzzlehttp/psr7 package.

## 5.3.0 - 2015-05-19

* Mock now supports save\_to
* Marked AbstractRequestEvent::getTransaction() as public.
* Fixed a bug in which multiple headers using different casing would overwrite previous headers in the associative array.
* Added Utils::getDefaultHandler()
* Marked GuzzleHttp\Client::getDefaultUserAgent as deprecated.
* URL scheme is now always lowercased.

## 6.0.0-beta.1

* Requires PHP >= 5.5
* Updated to use PSR-7
  + Requires immutable messages, which basically means an event based system owned by a request instance is no longer possible.
  + Utilizing the [Guzzle PSR-7 package](https://github.com/guzzle/psr7).
  + Removed the dependency on guzzlehttp/streams. These stream abstractions are available in the guzzlehttp/psr7 package under the GuzzleHttp\Psr7 namespace.
* Added middleware and handler system
  + Replaced the Guzzle event and subscriber system with a middleware system.
  + No longer depends on RingPHP, but rather places the HTTP handlers directly in Guzzle, operating on PSR-7 messages.
  + Retry logic is now encapsulated in GuzzleHttp\Middleware::retry, which means the guzzlehttp/retry-subscriber is now obsolete.
  + Mocking responses is now handled using GuzzleHttp\Handler\MockHandler.
* Asynchronous responses
  + No longer supports the future request option to send an async request. Instead, use one of the \*Async methods of a client (e.g., requestAsync, getAsync, etc.).
  + Utilizing GuzzleHttp\Promise instead of React's promise library to avoid recursion required by chaining and forwarding react promises. See <https://github.com/guzzle/promises>
  + Added requestAsync and sendAsync to send request asynchronously.
  + Added magic methods for getAsync(), postAsync(), etc. to send requests asynchronously.
* Request options
  + POST and form updates
    - Added the form\_fields and form\_files request options.
    - Removed the GuzzleHttp\Post namespace.
    - The body request option no longer accepts an array for POST requests.
  + The exceptions request option has been deprecated in favor of the http\_errors request options.
  + The save\_to request option has been deprecated in favor of sink request option.
* Clients no longer accept an array of URI template string and variables for URI variables. You will need to expand URI templates before passing them into a client constructor or request method.
* Client methods get(), post(), put(), patch(), options(), etc. are now magic methods that will send synchronous requests.
* Replaced Utils.php with plain functions in functions.php.
* Removed GuzzleHttp\Collection.
* Removed GuzzleHttp\BatchResults. Batched pool results are now returned as an array.
* Removed GuzzleHttp\Query. Query string handling is now handled using an associative array passed into the query request option. The query string is serialized using PHP's http\_build\_query. If you need more control, you can pass the query string in as a string.
* GuzzleHttp\QueryParser has been replaced with the GuzzleHttp\Psr7\parse\_query.

## 5.2.0 - 2015-01-27

* Added AppliesHeadersInterface to make applying headers to a request based on the body more generic and not specific to PostBodyInterface.
* Reduced the number of stack frames needed to send requests.
* Nested futures are now resolved in the client rather than the RequestFsm
* Finishing state transitions is now handled in the RequestFsm rather than the RingBridge.
* Added a guard in the Pool class to not use recursion for request retries.

## 5.1.0 - 2014-12-19

* Pool class no longer uses recursion when a request is intercepted.
* The size of a Pool can now be dynamically adjusted using a callback. See <https://github.com/guzzle/guzzle/pull/943>.
* Setting a request option to null when creating a request with a client will ensure that the option is not set. This allows you to overwrite default request options on a per-request basis. See <https://github.com/guzzle/guzzle/pull/937>.
* Added the ability to limit which protocols are allowed for redirects by specifying a protocols array in the allow\_redirects request option.
* Nested futures due to retries are now resolved when waiting for synchronous responses. See <https://github.com/guzzle/guzzle/pull/947>.
* "0" is now an allowed URI path. See <https://github.com/guzzle/guzzle/pull/935>.
* Query no longer typehints on the $query argument in the constructor, allowing for strings and arrays.
* Exceptions thrown in the end event are now correctly wrapped with Guzzle specific exceptions if necessary.

## 5.0.3 - 2014-11-03

This change updates query strings so that they are treated as un-encoded values by default where the value represents an un-encoded value to send over the wire. A Query object then encodes the value before sending over the wire. This means that even value query string values (e.g., ":") are url encoded. This makes the Query class match PHP's http\_build\_query function. However, if you want to send requests over the wire using valid query string characters that do not need to be encoded, then you can provide a string to Url::setQuery() and pass true as the second argument to specify that the query string is a raw string that should not be parsed or encoded (unless a call to getQuery() is subsequently made, forcing the query-string to be converted into a Query object).

## 5.0.2 - 2014-10-30

* Added a trailing \r\n to multipart/form-data payloads. See <https://github.com/guzzle/guzzle/pull/871>
* Added a GuzzleHttp\Pool::send() convenience method to match the docs.
* Status codes are now returned as integers. See <https://github.com/guzzle/guzzle/issues/881>
* No longer overwriting an existing application/x-www-form-urlencoded header when sending POST requests, allowing for customized headers. See <https://github.com/guzzle/guzzle/issues/877>
* Improved path URL serialization.
  + No longer double percent-encoding characters in the path or query string if they are already encoded.
  + Now properly encoding the supplied path to a URL object, instead of only encoding ' ' and '?'.
  + Note: This has been changed in 5.0.3 to now encode query string values by default unless the rawString argument is provided when setting the query string on a URL: Now allowing many more characters to be present in the query string without being percent encoded. See <https://datatracker.ietf.org/doc/html/rfc3986#appendix-A>

## 5.0.1 - 2014-10-16

Bugfix release.

* Fixed an issue where connection errors still returned response object in error and end events event though the response is unusable. This has been corrected so that a response is not returned in the getResponse method of these events if the response did not complete. <https://github.com/guzzle/guzzle/issues/867>
* Fixed an issue where transfer statistics were not being populated in the RingBridge. <https://github.com/guzzle/guzzle/issues/866>

## 5.0.0 - 2014-10-12

Adding support for non-blocking responses and some minor API cleanup.

### New Features

* Added support for non-blocking responses based on guzzlehttp/guzzle-ring.
* Added a public API for creating a default HTTP adapter.
* Updated the redirect plugin to be non-blocking so that redirects are sent concurrently. Other plugins like this can now be updated to be non-blocking.
* Added a "progress" event so that you can get upload and download progress events.
* Added GuzzleHttp\Pool which implements FutureInterface and transfers requests concurrently using a capped pool size as efficiently as possible.
* Added hasListeners() to EmitterInterface.
* Removed GuzzleHttp\ClientInterface::sendAll and marked GuzzleHttp\Client::sendAll as deprecated (it's still there, just not the recommended way).

### Breaking changes

The breaking changes in this release are relatively minor. The biggest thing to look out for is that request and response objects no longer implement fluent interfaces.

* Removed the fluent interfaces (i.e., return $this) from requests, responses, GuzzleHttp\Collection, GuzzleHttp\Url, GuzzleHttp\Query, GuzzleHttp\Post\PostBody, and GuzzleHttp\Cookie\SetCookie. This blog post provides a good outline of why I did this: <https://ocramius.github.io/blog/fluent-interfaces-are-evil/>. This also makes the Guzzle message interfaces compatible with the current PSR-7 message proposal.
* Removed "functions.php", so that Guzzle is truly PSR-4 compliant. Except for the HTTP request functions from function.php, these functions are now implemented in GuzzleHttp\Utils using camelCase. GuzzleHttp\json\_decode moved to GuzzleHttp\Utils::jsonDecode. GuzzleHttp\get\_path moved to GuzzleHttp\Utils::getPath. GuzzleHttp\set\_path moved to GuzzleHttp\Utils::setPath. GuzzleHttp\batch should now be GuzzleHttp\Pool::batch, which returns an objectStorage. Using functions.php caused problems for many users: they aren't PSR-4 compliant, require an explicit include, and needed an if-guard to ensure that the functions are not declared multiple times.
* Rewrote adapter layer.
  + Removing all classes from GuzzleHttp\Adapter, these are now implemented as callables that are stored in GuzzleHttp\Ring\Client.
  + Removed the concept of "parallel adapters". Sending requests serially or concurrently is now handled using a single adapter.
  + Moved GuzzleHttp\Adapter\Transaction to GuzzleHttp\Transaction. The Transaction object now exposes the request, response, and client as public properties. The getters and setters have been removed.
* Removed the "headers" event. This event was only useful for changing the body a response once the headers of the response were known. You can implement a similar behavior in a number of ways. One example might be to use a FnStream that has access to the transaction being sent. For example, when the first byte is written, you could check if the response headers match your expectations, and if so, change the actual stream body that is being written to.
* Removed the asArray parameter from GuzzleHttp\Message\MessageInterface::getHeader. If you want to get a header value as an array, then use the newly added getHeaderAsArray() method of MessageInterface. This change makes the Guzzle interfaces compatible with the PSR-7 interfaces.
* GuzzleHttp\Message\MessageFactory no longer allows subclasses to add custom request options using double-dispatch (this was an implementation detail). Instead, you should now provide an associative array to the constructor which is a mapping of the request option name mapping to a function that applies the option value to a request.
* Removed the concept of "throwImmediately" from exceptions and error events. This control mechanism was used to stop a transfer of concurrent requests from completing. This can now be handled by throwing the exception or by cancelling a pool of requests or each outstanding future request individually.
* Updated to "GuzzleHttp\Streams" 3.0.
  + GuzzleHttp\Stream\StreamInterface::getContents() no longer accepts a maxLen parameter. This update makes the Guzzle streams project compatible with the current PSR-7 proposal.
  + GuzzleHttp\Stream\Stream::\_\_construct, GuzzleHttp\Stream\Stream::factory, and GuzzleHttp\Stream\Utils::create no longer accept a size in the second argument. They now accept an associative array of options, including the "size" key and "metadata" key which can be used to provide custom metadata.

## 4.2.2 - 2014-09-08

* Fixed a memory leak in the CurlAdapter when reusing cURL handles.
* No longer using request\_fulluri in stream adapter proxies.
* Relative redirects are now based on the last response, not the first response.

## 4.2.1 - 2014-08-19

* Ensuring that the StreamAdapter does not always add a Content-Type header
* Adding automated github releases with a phar and zip

## 4.2.0 - 2014-08-17

* Now merging in default options using a case-insensitive comparison. Closes <https://github.com/guzzle/guzzle/issues/767>
* Added the ability to automatically decode Content-Encoding response bodies using the decode\_content request option. This is set to true by default to decode the response body if it comes over the wire with a Content-Encoding. Set this value to false to disable decoding the response content, and pass a string to provide a request Accept-Encoding header and turn on automatic response decoding. This feature now allows you to pass an Accept-Encoding header in the headers of a request but still disable automatic response decoding. Closes <https://github.com/guzzle/guzzle/issues/764>
* Added the ability to throw an exception immediately when transferring requests in parallel. Closes <https://github.com/guzzle/guzzle/issues/760>
* Updating guzzlehttp/streams dependency to ~2.1
* No longer utilizing the now deprecated namespaced methods from the stream package.

## 4.1.8 - 2014-08-14

* Fixed an issue in the CurlFactory that caused setting the stream=false request option to throw an exception. See: <https://github.com/guzzle/guzzle/issues/769>
* TransactionIterator now calls rewind on the inner iterator. See: <https://github.com/guzzle/guzzle/pull/765>
* You can now set the Content-Type header to multipart/form-data when creating POST requests to force multipart bodies. See <https://github.com/guzzle/guzzle/issues/768>

## 4.1.7 - 2014-08-07

* Fixed an error in the HistoryPlugin that caused the same request and response to be logged multiple times when an HTTP protocol error occurs.
* Ensuring that cURL does not add a default Content-Type when no Content-Type has been supplied by the user. This prevents the adapter layer from modifying the request that is sent over the wire after any listeners may have already put the request in a desired state (e.g., signed the request).
* Throwing an exception when you attempt to send requests that have the "stream" set to true in parallel using the MultiAdapter.
* Only calling curl\_multi\_select when there are active cURL handles. This was previously changed and caused performance problems on some systems due to PHP always selecting until the maximum select timeout.
* Fixed a bug where multipart/form-data POST fields were not correctly aggregated (e.g., values with "&").

## 4.1.6 - 2014-08-03

* Added helper methods to make it easier to represent messages as strings, including getting the start line and getting headers as a string.

## 4.1.5 - 2014-08-02

* Automatically retrying cURL "Connection died, retrying a fresh connect" errors when possible.
* cURL implementation cleanup
* Allowing multiple event subscriber listeners to be registered per event by passing an array of arrays of listener configuration.

## 4.1.4 - 2014-07-22

* Fixed a bug that caused multi-part POST requests with more than one field to serialize incorrectly.
* Paths can now be set to "0"
* ResponseInterface::xml now accepts a libxml\_options option and added a missing default argument that was required when parsing XML response bodies.
* A save\_to stream is now created lazily, which means that files are not created on disk unless a request succeeds.

## 4.1.3 - 2014-07-15

* Various fixes to multipart/form-data POST uploads
* Wrapping function.php in an if-statement to ensure Guzzle can be used globally and in a Composer install
* Fixed an issue with generating and merging in events to an event array
* POST headers are only applied before sending a request to allow you to change the query aggregator used before uploading
* Added much more robust query string parsing
* Fixed various parsing and normalization issues with URLs
* Fixing an issue where multi-valued headers were not being utilized correctly in the StreamAdapter

## 4.1.2 - 2014-06-18

* Added support for sending payloads with GET requests

## 4.1.1 - 2014-06-08

* Fixed an issue related to using custom message factory options in subclasses
* Fixed an issue with nested form fields in a multi-part POST
* Fixed an issue with using the json request option for POST requests
* Added ToArrayInterface to GuzzleHttp\Cookie\CookieJar

## 4.1.0 - 2014-05-27

* Added a json request option to easily serialize JSON payloads.
* Added a GuzzleHttp\json\_decode() wrapper to safely parse JSON.
* Added setPort() and getPort() to GuzzleHttp\Message\RequestInterface.
* Added the ability to provide an emitter to a client in the client constructor.
* Added the ability to persist a cookie session using $\_SESSION.
* Added a trait that can be used to add event listeners to an iterator.
* Removed request method constants from RequestInterface.
* Fixed warning when invalid request start-lines are received.
* Updated MessageFactory to work with custom request option methods.
* Updated cacert bundle to latest build.

## 4.0.2 (2014-04-16)

* Proxy requests using the StreamAdapter now properly use request\_fulluri (#632)
* Added the ability to set scalars as POST fields (#628)

## 4.0.1 - 2014-04-04

* The HTTP status code of a response is now set as the exception code of RequestException objects.
* 303 redirects will now correctly switch from POST to GET requests.
* The default parallel adapter of a client now correctly uses the MultiAdapter.
* HasDataTrait now initializes the internal data array as an empty array so that the toArray() method always returns an array.

## 4.0.0 - 2014-03-29

* For information on changes and upgrading, see: <https://github.com/guzzle/guzzle/blob/master/UPGRADING.md#3x-to-40>
* Added GuzzleHttp\batch() as a convenience function for sending requests in parallel without needing to write asynchronous code.
* Restructured how events are added to GuzzleHttp\ClientInterface::sendAll(). You can now pass a callable or an array of associative arrays where each associative array contains the "fn", "priority", and "once" keys.

## 4.0.0.rc-2 - 2014-03-25

* Removed getConfig() and setConfig() from clients to avoid confusion around whether things like base\_url, message\_factory, etc. should be able to be retrieved or modified.
* Added getDefaultOption() and setDefaultOption() to ClientInterface
* functions.php functions were renamed using snake\_case to match PHP idioms
* Added support for HTTP\_PROXY, HTTPS\_PROXY, and GUZZLE\_CURL\_SELECT\_TIMEOUT environment variables
* Added the ability to specify custom sendAll() event priorities
* Added the ability to specify custom stream context options to the stream adapter.
* Added a functions.php function for get\_path() and set\_path()
* CurlAdapter and MultiAdapter now use a callable to generate curl resources
* MockAdapter now properly reads a body and emits a headers event
* Updated Url class to check if a scheme and host are set before adding ":" and "//". This allows empty Url (e.g., "") to be serialized as "".
* Parsing invalid XML no longer emits warnings
* Curl classes now properly throw AdapterExceptions
* Various performance optimizations
* Streams are created with the faster Stream\create() function
* Marked deprecation\_proxy() as internal
* Test server is now a collection of static methods on a class

## 4.0.0-rc.1 - 2014-03-15

* See <https://github.com/guzzle/guzzle/blob/master/UPGRADING.md#3x-to-40>

## 3.8.1 - 2014-01-28

* Bug: Always using GET requests when redirecting from a 303 response
* Bug: CURLOPT\_SSL\_VERIFYHOST is now correctly set to false when setting $certificateAuthority to false in Guzzle\Http\ClientInterface::setSslVerification()
* Bug: RedirectPlugin now uses strict RFC 3986 compliance when combining a base URL with a relative URL
* Bug: The body of a request can now be set to "0"
* Sending PHP stream requests no longer forces HTTP/1.0
* Adding more information to ExceptionCollection exceptions so that users have more context, including a stack trace of each sub-exception
* Updated the $ref attribute in service descriptions to merge over any existing parameters of a schema (rather than clobbering everything).
* Merging URLs will now use the query string object from the relative URL (thus allowing custom query aggregators)
* Query strings are now parsed in a way that they do no convert empty keys with no value to have a dangling =. For example foo&bar=baz is now correctly parsed and recognized as foo&bar=baz rather than foo=&bar=baz.
* Now properly escaping the regular expression delimiter when matching Cookie domains.
* Network access is now disabled when loading XML documents

## 3.8.0 - 2013-12-05

* Added the ability to define a POST name for a file
* JSON response parsing now properly walks additionalProperties
* cURL error code 18 is now retried automatically in the BackoffPlugin
* Fixed a cURL error when URLs contain fragments
* Fixed an issue in the BackoffPlugin retry event where it was trying to access all exceptions as if they were CurlExceptions
* CURLOPT\_PROGRESS function fix for PHP 5.5 (69fcc1e)
* Added the ability for Guzzle to work with older versions of cURL that do not support CURLOPT\_TIMEOUT\_MS
* Fixed a bug that was encountered when parsing empty header parameters
* UriTemplate now has a setRegex() method to match the docs
* The debug request parameter now checks if it is truthy rather than if it exists
* Setting the debug request parameter to true shows verbose cURL output instead of using the LogPlugin
* Added the ability to combine URLs using strict RFC 3986 compliance
* Command objects can now return the validation errors encountered by the command
* Various fixes to cache revalidation (#437 and 29797e5)
* Various fixes to the AsyncPlugin
* Cleaned up build scripts

## 3.7.4 - 2013-10-02

* Bug fix: 0 is now an allowed value in a description parameter that has a default value (#430)
* Bug fix: SchemaFormatter now returns an integer when formatting to a Unix timestamp (see <https://github.com/aws/aws-sdk-php/issues/147>)
* Bug fix: Cleaned up and fixed URL dot segment removal to properly resolve internal dots
* Minimum PHP version is now properly specified as 5.3.3 (up from 5.3.2) (#420)
* Updated the bundled cacert.pem (#419)
* OauthPlugin now supports adding authentication to headers or query string (#425)

## 3.7.3 - 2013-09-08

* Added the ability to get the exception associated with a request/command when using MultiTransferException and CommandTransferException.
* Setting additionalParameters of a response to false is now honored when parsing responses with a service description
* Schemas are only injected into response models when explicitly configured.
* No longer guessing Content-Type based on the path of a request. Content-Type is now only guessed based on the path of an EntityBody.
* Bug fix: ChunkedIterator can now properly chunk a \Traversable as well as an \Iterator.
* Bug fix: FilterIterator now relies on \Iterator instead of \Traversable.
* Bug fix: Gracefully handling malformed responses in RequestMediator::writeResponseBody()
* Bug fix: Replaced call to canCache with canCacheRequest in the CallbackCanCacheStrategy of the CachePlugin
* Bug fix: Visiting XML attributes first before visiting XML children when serializing requests
* Bug fix: Properly parsing headers that contain commas contained in quotes
* Bug fix: mimetype guessing based on a filename is now case-insensitive

## 3.7.2 - 2013-08-02

* Bug fix: Properly URL encoding paths when using the PHP-only version of the UriTemplate expander See <https://github.com/guzzle/guzzle/issues/371>
* Bug fix: Cookie domains are now matched correctly according to RFC 6265 See <https://github.com/guzzle/guzzle/issues/377>
* Bug fix: GET parameters are now used when calculating an OAuth signature
* Bug fix: Fixed an issue with cache revalidation where the If-None-Match header was being double quoted
* Guzzle\Common\AbstractHasDispatcher::dispatch() now returns the event that was dispatched
* Guzzle\Http\QueryString::factory() now guesses the most appropriate query aggregator to used based on the input. See <https://github.com/guzzle/guzzle/issues/379>
* Added a way to add custom domain objects to service description parsing using the operation.parse\_class event. See <https://github.com/guzzle/guzzle/pull/380>
* cURL multi cleanup and optimizations

## 3.7.1 - 2013-07-05

* Bug fix: Setting default options on a client now works
* Bug fix: Setting options on HEAD requests now works. See #352
* Bug fix: Moving stream factory before send event to before building the stream. See #353
* Bug fix: Cookies no longer match on IP addresses per RFC 6265
* Bug fix: Correctly parsing header parameters that are in <> and quotes
* Added cert and ssl\_key as request options
* Host header can now diverge from the host part of a URL if the header is set manually
* Guzzle\Service\Command\LocationVisitor\Request\XmlVisitor was rewritten to change from using SimpleXML to XMLWriter
* OAuth parameters are only added via the plugin if they aren't already set
* Exceptions are now thrown when a URL cannot be parsed
* Returning false if Guzzle\Http\EntityBody::getContentMd5() fails
* Not setting a Content-MD5 on a command if calculating the Content-MD5 fails via the CommandContentMd5Plugin

## 3.7.0 - 2013-06-10

* See UPGRADING.md for more information on how to upgrade.
* Requests now support the ability to specify an array of $options when creating a request to more easily modify a request. You can pass a 'request.options' configuration setting to a client to apply default request options to every request created by a client (e.g. default query string variables, headers, curl options, etc.).
* Added a static facade class that allows you to use Guzzle with static methods and mount the class to \Guzzle. See Guzzle\Http\StaticClient::mount.
* Added command.request\_options to Guzzle\Service\Command\AbstractCommand to pass request options to requests created by a command (e.g. custom headers, query string variables, timeout settings, etc.).
* Stream size in Guzzle\Stream\PhpStreamRequestFactory will now be set if Content-Length is returned in the headers of a response
* Added Guzzle\Common\Collection::setPath($path, $value) to set a value into an array using a nested key (e.g. $collection->setPath('foo/baz/bar', 'test'); echo $collection['foo']['bar']['bar'];)
* ServiceBuilders now support storing and retrieving arbitrary data
* CachePlugin can now purge all resources for a given URI
* CachePlugin can automatically purge matching cached items when a non-idempotent request is sent to a resource
* CachePlugin now uses the Vary header to determine if a resource is a cache hit
* Guzzle\Http\Message\Response now implements \Serializable
* Added Guzzle\Cache\CacheAdapterFactory::fromCache() to more easily create cache adapters
* Guzzle\Service\ClientInterface::execute() now accepts an array, single command, or Traversable
* Fixed a bug in Guzzle\Http\Message\Header\Link::addLink()
* Better handling of calculating the size of a stream in Guzzle\Stream\Stream using fstat() and caching the size
* Guzzle\Common\Exception\ExceptionCollection now creates a more readable exception message
* Fixing BC break: Added back the MonologLogAdapter implementation rather than extending from PsrLog so that older Symfony users can still use the old version of Monolog.
* Fixing BC break: Added the implementation back in for Guzzle\Http\Message\AbstractMessage::getTokenizedHeader(). Now triggering an E\_USER\_DEPRECATED warning when used. Use $message->getHeader()->parseParams().
* Several performance improvements to Guzzle\Common\Collection
* Added an $options argument to the end of the following methods of Guzzle\Http\ClientInterface: createRequest, head, delete, put, patch, post, options, prepareRequest
* Added an $options argument to the end of Guzzle\Http\Message\Request\RequestFactoryInterface::createRequest()
* Added an applyOptions() method to Guzzle\Http\Message\Request\RequestFactoryInterface
* Changed Guzzle\Http\ClientInterface::get($uri = null, $headers = null, $body = null) to Guzzle\Http\ClientInterface::get($uri = null, $headers = null, $options = array()). You can still pass in a resource, string, or EntityBody into the $options parameter to specify the download location of the response.
* Changed Guzzle\Common\Collection::\_\_construct($data) to no longer accepts a null value for $data but a default array()
* Added Guzzle\Stream\StreamInterface::isRepeatable
* Removed Guzzle\Http\ClientInterface::setDefaultHeaders(). Use $client->getConfig()->setPath('request.options/headers/{header\_name}', 'value'). or $client->getConfig()->setPath('request.options/headers', array('header\_name' => 'value'))`.
* Removed Guzzle\Http\ClientInterface::getDefaultHeaders(). Use $client->getConfig()->getPath('request.options/headers').
* Removed Guzzle\Http\ClientInterface::expandTemplate()
* Removed Guzzle\Http\ClientInterface::setRequestFactory()
* Removed Guzzle\Http\ClientInterface::getCurlMulti()
* Removed Guzzle\Http\Message\RequestInterface::canCache
* Removed Guzzle\Http\Message\RequestInterface::setIsRedirect
* Removed Guzzle\Http\Message\RequestInterface::isRedirect
* Made Guzzle\Http\Client::expandTemplate and getUriTemplate protected methods.
* You can now enable E\_USER\_DEPRECATED warnings to see if you are using a deprecated method by setting Guzzle\Common\Version::$emitWarnings to true.
* Marked Guzzle\Http\Message\Request::isResponseBodyRepeatable() as deprecated. Use $request->getResponseBody()->isRepeatable() instead.
* Marked Guzzle\Http\Message\Request::canCache() as deprecated. Use Guzzle\Plugin\Cache\DefaultCanCacheStrategy->canCacheRequest() instead.
* Marked Guzzle\Http\Message\Request::canCache() as deprecated. Use Guzzle\Plugin\Cache\DefaultCanCacheStrategy->canCacheRequest() instead.
* Marked Guzzle\Http\Message\Request::setIsRedirect() as deprecated. Use the HistoryPlugin instead.
* Marked Guzzle\Http\Message\Request::isRedirect() as deprecated. Use the HistoryPlugin instead.
* Marked Guzzle\Cache\CacheAdapterFactory::factory() as deprecated
* Marked 'command.headers', 'command.response\_body' and 'command.on\_complete' as deprecated for AbstractCommand. These will work through Guzzle 4.0
* Marked 'request.params' for Guzzle\Http\Client as deprecated. Use [request.options][params].
* Marked Guzzle\Service\Client::enableMagicMethods() as deprecated. Magic methods can no longer be disabled on a Guzzle\Service\Client.
* Marked Guzzle\Service\Client::getDefaultHeaders() as deprecated. Use $client->getConfig()->getPath('request.options/headers')`.
* Marked Guzzle\Service\Client::setDefaultHeaders() as deprecated. Use $client->getConfig()->setPath('request.options/headers/{header\_name}', 'value')`.
* Marked Guzzle\Parser\Url\UrlParser as deprecated. Just use PHP's parse\_url() and percent encode your UTF-8.
* Marked Guzzle\Common\Collection::inject() as deprecated.
* Marked Guzzle\Plugin\CurlAuth\CurlAuthPlugin as deprecated. Use $client->getConfig()->setPath('request.options/auth', array('user', 'pass', 'Basic|Digest');
* CacheKeyProviderInterface and DefaultCacheKeyProvider are no longer used. All of this logic is handled in a CacheStorageInterface. These two objects and interface will be removed in a future version.
* Always setting X-cache headers on cached responses
* Default cache TTLs are now handled by the CacheStorageInterface of a CachePlugin
* CacheStorageInterface::cache($key, Response $response, $ttl = null) has changed to cache(RequestInterface $request, Response $response);
* CacheStorageInterface::fetch($key) has changed to fetch(RequestInterface $request);
* CacheStorageInterface::delete($key) has changed to delete(RequestInterface $request);
* Added CacheStorageInterface::purge($url)
* DefaultRevalidation::\_\_construct(CacheKeyProviderInterface $cacheKey, CacheStorageInterface $cache, CachePlugin $plugin) has changed to DefaultRevalidation::\_\_construct(CacheStorageInterface $cache, CanCacheStrategyInterface $canCache = null)
* Added RevalidationInterface::shouldRevalidate(RequestInterface $request, Response $response)

## 3.6.0 - 2013-05-29

* ServiceDescription now implements ToArrayInterface
* Added command.hidden\_params to blacklist certain headers from being treated as additionalParameters
* Guzzle can now correctly parse incomplete URLs
* Mixed casing of headers are now forced to be a single consistent casing across all values for that header.
* Messages internally use a HeaderCollection object to delegate handling case-insensitive header resolution
* Removed the whole changedHeader() function system of messages because all header changes now go through addHeader().
* Specific header implementations can be created for complex headers. When a message creates a header, it uses a HeaderFactory which can map specific headers to specific header classes. There is now a Link header and CacheControl header implementation.
* Removed from interface: Guzzle\Http\ClientInterface::setUriTemplate
* Removed from interface: Guzzle\Http\ClientInterface::setCurlMulti()
* Removed Guzzle\Http\Message\Request::receivedRequestHeader() and implemented this functionality in Guzzle\Http\Curl\RequestMediator
* Removed the optional $asString parameter from MessageInterface::getHeader(). Just cast the header to a string.
* Removed the optional $tryChunkedTransfer option from Guzzle\Http\Message\EntityEnclosingRequestInterface
* Removed the $asObjects argument from Guzzle\Http\Message\MessageInterface::getHeaders()
* Removed Guzzle\Parser\ParserRegister::get(). Use getParser()
* Removed Guzzle\Parser\ParserRegister::set(). Use registerParser().
* All response header helper functions return a string rather than mixing Header objects and strings inconsistently
* Removed cURL blacklist support. This is no longer necessary now that Expect, Accept, etc. are managed by Guzzle directly via interfaces
* Removed the injecting of a request object onto a response object. The methods to get and set a request still exist but are a no-op until removed.
* Most classes that used to require a Guzzle\Service\Command\CommandInterface typehint now request a Guzzle\Service\Command\ArrayCommandInterface.
* Added Guzzle\Http\Message\RequestInterface::startResponse() to the RequestInterface to handle injecting a response on a request while the request is still being transferred
* The ability to case-insensitively search for header values
* Guzzle\Http\Message\Header::hasExactHeader
* Guzzle\Http\Message\Header::raw. Use getAll()
* Deprecated cache control specific methods on Guzzle\Http\Message\AbstractMessage. Use the CacheControl header object instead.
* Guzzle\Service\Command\CommandInterface now extends from ToArrayInterface and ArrayAccess
* Added the ability to cast Model objects to a string to view debug information.

## 3.5.0 - 2013-05-13

* Bug: Fixed a regression so that request responses are parsed only once per oncomplete event rather than multiple times
* Bug: Better cleanup of one-time events across the board (when an event is meant to fire once, it will now remove itself from the EventDispatcher)
* Bug: Guzzle\Log\MessageFormatter now properly writes "total\_time" and "connect\_time" values
* Bug: Cloning an EntityEnclosingRequest now clones the EntityBody too
* Bug: Fixed an undefined index error when parsing nested JSON responses with a sentAs parameter that reference a non-existent key
* Bug: All \_\_call() method arguments are now required (helps with mocking frameworks)
* Deprecating Response::getRequest() and now using a shallow clone of a request object to remove a circular reference to help with refcount based garbage collection of resources created by sending a request
* Deprecating ZF1 cache and log adapters. These will be removed in the next major version.
* Deprecating Response::getPreviousResponse() (method signature still exists, but it's deprecated). Use the HistoryPlugin for a history.
* Added a responseBody alias for the response\_body location
* Refactored internals to no longer rely on Response::getRequest()
* HistoryPlugin can now be cast to a string
* HistoryPlugin now logs transactions rather than requests and responses to more accurately keep track of the requests and responses that are sent over the wire
* Added getEffectiveUrl() and getRedirectCount() to Response objects

## 3.4.3 - 2013-04-30

* Bug fix: Fixing bug introduced in 3.4.2 where redirect responses are duplicated on the final redirected response
* Added a check to re-extract the temp cacert bundle from the phar before sending each request

## 3.4.2 - 2013-04-29

* Bug fix: Stream objects now work correctly with "a" and "a+" modes
* Bug fix: Removing Transfer-Encoding: chunked header when a Content-Length is present
* Bug fix: AsyncPlugin no longer forces HEAD requests
* Bug fix: DateTime timezones are now properly handled when using the service description schema formatter
* Bug fix: CachePlugin now properly handles stale-if-error directives when a request to the origin server fails
* Setting a response on a request will write to the custom request body from the response body if one is specified
* LogPlugin now writes to php://output when STDERR is undefined
* Added the ability to set multiple POST files for the same key in a single call
* application/x-www-form-urlencoded POSTs now use the utf-8 charset by default
* Added the ability to queue CurlExceptions to the MockPlugin
* Cleaned up how manual responses are queued on requests (removed "queued\_response" and now using request.before\_send)
* Configuration loading now allows remote files

## 3.4.1 - 2013-04-16

* Large refactoring to how CurlMulti handles work. There is now a proxy that sits in front of a pool of CurlMulti handles. This greatly simplifies the implementation, fixes a couple bugs, and provides a small performance boost.
* Exceptions are now properly grouped when sending requests in parallel
* Redirects are now properly aggregated when a multi transaction fails
* Redirects now set the response on the original object even in the event of a failure
* Bug fix: Model names are now properly set even when using $refs
* Added support for PHP 5.5's CurlFile to prevent warnings with the deprecated @ syntax
* Added support for oauth\_callback in OAuth signatures
* Added support for oauth\_verifier in OAuth signatures
* Added support to attempt to retrieve a command first literally, then ucfirst, the with inflection

## 3.4.0 - 2013-04-11

* Bug fix: URLs are now resolved correctly based on <https://datatracker.ietf.org/doc/html/rfc3986#section-5.2>. #289
* Bug fix: Absolute URLs with a path in a service description will now properly override the base URL. #289
* Bug fix: Parsing a query string with a single PHP array value will now result in an array. #263
* Bug fix: Better normalization of the User-Agent header to prevent duplicate headers. #264.
* Bug fix: Added number type to service descriptions.
* Bug fix: empty parameters are removed from an OAuth signature
* Bug fix: Revalidating a cache entry prefers the Last-Modified over the Date header
* Bug fix: Fixed "array to string" error when validating a union of types in a service description
* Bug fix: Removed code that attempted to determine the size of a stream when data is written to the stream
* Bug fix: Not including an oauth\_token if the value is null in the OauthPlugin.
* Bug fix: Now correctly aggregating successful requests and failed requests in CurlMulti when a redirect occurs.
* The new default CURLOPT\_TIMEOUT setting has been increased to 150 seconds so that Guzzle works on poor connections.
* Added a feature to EntityEnclosingRequest::setBody() that will automatically set the Content-Type of the request if the Content-Type can be determined based on the entity body or the path of the request.
* Added the ability to overwrite configuration settings in a client when grabbing a throwaway client from a builder.
* Added support for a PSR-3 LogAdapter.
* Added a command.after\_prepare event
* Added oauth\_callback parameter to the OauthPlugin
* Added the ability to create a custom stream class when using a stream factory
* Added a CachingEntityBody decorator
* Added support for additionalParameters in service descriptions to define how custom parameters are serialized.
* The bundled SSL certificate is now provided in the phar file and extracted when running Guzzle from a phar.
* You can now send any EntityEnclosingRequest with POST fields or POST files and cURL will handle creating bodies
* POST requests using a custom entity body are now treated exactly like PUT requests but with a custom cURL method. This means that the redirect behavior of POST requests with custom bodies will not be the same as POST requests that use POST fields or files (the latter is only used when emulating a form POST in the browser).
* Lots of cleanup to CurlHandle::factory and RequestFactory::createRequest

## 3.3.1 - 2013-03-10

* Added the ability to create PHP streaming responses from HTTP requests
* Bug fix: Running any filters when parsing response headers with service descriptions
* Bug fix: OauthPlugin fixes to allow for multi-dimensional array signing, and sorting parameters before signing
* Bug fix: Removed the adding of default empty arrays and false Booleans to responses in order to be consistent across response location visitors.
* Bug fix: Removed the possibility of creating configuration files with circular dependencies
* RequestFactory::create() now uses the key of a POST file when setting the POST file name
* Added xmlAllowEmpty to serialize an XML body even if no XML specific parameters are set

## 3.3.0 - 2013-03-03

* A large number of performance optimizations have been made
* Bug fix: Added 'wb' as a valid write mode for streams
* Bug fix: Guzzle\Http\Message\Response::json() now allows scalar values to be returned
* Bug fix: Fixed bug in Guzzle\Http\Message\Response where wrapping quotes were stripped from getEtag()
* BC: Removed Guzzle\Http\Utils class
* BC: Setting a service description on a client will no longer modify the client's command factories.
* BC: Emitting IO events from a RequestMediator is now a parameter that must be set in a request's curl options using the 'emit\_io' key. This was previously set under a request's parameters using 'curl.emit\_io'
* BC: Guzzle\Stream\Stream::getWrapper() and Guzzle\Stream\Stream::getSteamType() are no longer converted to lowercase
* Operation parameter objects are now lazy loaded internally
* Added ErrorResponsePlugin that can throw errors for responses defined in service description operations' errorResponses
* Added support for instantiating responseType=class responseClass classes. Classes must implement Guzzle\Service\Command\ResponseClassInterface
* Added support for additionalProperties for top-level parameters in responseType=model responseClasses. These additional properties also support locations and can be used to parse JSON responses where the outermost part of the JSON is an array
* Added support for nested renaming of JSON models (rename sentAs to name)
* CachePlugin
  + Added support for stale-if-error so that the CachePlugin can now serve stale content from the cache on error
  + Debug headers can now added to cached response in the CachePlugin

## 3.2.0 - 2013-02-14

* CurlMulti is no longer reused globally. A new multi object is created per-client. This helps to isolate clients.
* URLs with no path no longer contain a "/" by default
* Guzzle\Http\QueryString does no longer manages the leading "?". This is now handled in Guzzle\Http\Url.
* BadResponseException no longer includes the full request and response message
* Adding setData() to Guzzle\Service\Description\ServiceDescriptionInterface
* Adding getResponseBody() to Guzzle\Http\Message\RequestInterface
* Various updates to classes to use ServiceDescriptionInterface type hints rather than ServiceDescription
* Header values can now be normalized into distinct values when multiple headers are combined with a comma separated list
* xmlEncoding can now be customized for the XML declaration of a XML service description operation
* Guzzle\Http\QueryString now uses Guzzle\Http\QueryAggregator\QueryAggregatorInterface objects to add custom value aggregation and no longer uses callbacks
* The URL encoding implementation of Guzzle\Http\QueryString can now be customized
* Bug fix: Filters were not always invoked for array service description parameters
* Bug fix: Redirects now use a target response body rather than a temporary response body
* Bug fix: The default exponential backoff BackoffPlugin was not giving when the request threshold was exceeded
* Bug fix: Guzzle now takes the first found value when grabbing Cache-Control directives

## 3.1.2 - 2013-01-27

* Refactored how operation responses are parsed. Visitors now include a before() method responsible for parsing the response body. For example, the XmlVisitor now parses the XML response into an array in the before() method.
* Fixed an issue where cURL would not automatically decompress responses when the Accept-Encoding header was sent
* CURLOPT\_SSL\_VERIFYHOST is never set to 1 because it is deprecated (see 5e0ff2ef20f839e19d1eeb298f90ba3598784444)
* Fixed a bug where redirect responses were not chained correctly using getPreviousResponse()
* Setting default headers on a client after setting the user-agent will not erase the user-agent setting

## 3.1.1 - 2013-01-20

* Adding wildcard support to Guzzle\Common\Collection::getPath()
* Adding alias support to ServiceBuilder configs
* Adding Guzzle\Service\Resource\CompositeResourceIteratorFactory and cleaning up factory interface

## 3.1.0 - 2013-01-12

* BC: CurlException now extends from RequestException rather than BadResponseException
* BC: Renamed Guzzle\Plugin\Cache\CanCacheStrategyInterface::canCache() to canCacheRequest() and added CanCacheResponse()
* Added getData to ServiceDescriptionInterface
* Added context array to RequestInterface::setState()
* Bug: Removing hard dependency on the BackoffPlugin from Guzzle\Http
* Bug: Adding required content-type when JSON request visitor adds JSON to a command
* Bug: Fixing the serialization of a service description with custom data
* Made it easier to deal with exceptions thrown when transferring commands or requests in parallel by providing an array of successful and failed responses
* Moved getPath from Guzzle\Service\Resource\Model to Guzzle\Common\Collection
* Added Guzzle\Http\IoEmittingEntityBody
* Moved command filtration from validators to location visitors
* Added extends attributes to service description parameters
* Added getModels to ServiceDescriptionInterface

## 3.0.7 - 2012-12-19

* Fixing phar detection when forcing a cacert to system if null or true
* Allowing filename to be passed to Guzzle\Http\Message\Request::setResponseBody()
* Cleaning up Guzzle\Common\Collection::inject method
* Adding a response\_body location to service descriptions

## 3.0.6 - 2012-12-09

* CurlMulti performance improvements
* Adding setErrorResponses() to Operation
* composer.json tweaks

## 3.0.5 - 2012-11-18

* Bug: Fixing an infinite recursion bug caused from revalidating with the CachePlugin
* Bug: Response body can now be a string containing "0"
* Bug: Using Guzzle inside of a phar uses system by default but now allows for a custom cacert
* Bug: QueryString::fromString now properly parses query string parameters that contain equal signs
* Added support for XML attributes in service description responses
* DefaultRequestSerializer now supports array URI parameter values for URI template expansion
* Added better mimetype guessing to requests and post files

## 3.0.4 - 2012-11-11

* Bug: Fixed a bug when adding multiple cookies to a request to use the correct glue value
* Bug: Cookies can now be added that have a name, domain, or value set to "0"
* Bug: Using the system cacert bundle when using the Phar
* Added json and xml methods to Response to make it easier to parse JSON and XML response data into data structures
* Enhanced cookie jar de-duplication
* Added the ability to enable strict cookie jars that throw exceptions when invalid cookies are added
* Added setStream to StreamInterface to actually make it possible to implement custom rewind behavior for entity bodies
* Added the ability to create any sort of hash for a stream rather than just an MD5 hash

## 3.0.3 - 2012-11-04

* Implementing redirects in PHP rather than cURL
* Added PECL URI template extension and using as default parser if available
* Bug: Fixed Content-Length parsing of Response factory
* Adding rewind() method to entity bodies and streams. Allows for custom rewinding of non-repeatable streams.
* Adding ToArrayInterface throughout library
* Fixing OauthPlugin to create unique nonce values per request

## 3.0.2 - 2012-10-25

* Magic methods are enabled by default on clients
* Magic methods return the result of a command
* Service clients no longer require a base\_url option in the factory
* Bug: Fixed an issue with URI templates where null template variables were being expanded

## 3.0.1 - 2012-10-22

* Models can now be used like regular collection objects by calling filter, map, etc.
* Models no longer require a Parameter structure or initial data in the constructor
* Added a custom AppendIterator to get around a PHP bug with the \AppendIterator

## 3.0.0 - 2012-10-15

* Rewrote service description format to be based on Swagger
  + Now based on JSON schema
  + Added nested input structures and nested response models
  + Support for JSON and XML input and output models
  + Renamed commands to operations
  + Removed dot class notation
  + Removed custom types
* Broke the project into smaller top-level namespaces to be more component friendly
* Removed support for XML configs and descriptions. Use arrays or JSON files.
* Removed the Validation component and Inspector
* Moved all cookie code to Guzzle\Plugin\Cookie
* Magic methods on a Guzzle\Service\Client now return the command un-executed.
* Calling getResult() or getResponse() on a command will lazily execute the command if needed.
* Now shipping with cURL's CA certs and using it by default
* Added previousResponse() method to response objects
* No longer sending Accept and Accept-Encoding headers on every request
* Only sending an Expect header by default when a payload is greater than 1MB
* Added/moved client options:
  + curl.blacklist to curl.option.blacklist
  + Added ssl.certificate\_authority
* Added a Guzzle\Iterator component
* Moved plugins from Guzzle\Http\Plugin to Guzzle\Plugin
* Added a more robust backoff retry strategy (replaced the ExponentialBackoffPlugin)
* Added a more robust caching plugin
* Added setBody to response objects
* Updating LogPlugin to use a more flexible MessageFormatter
* Added a completely revamped build process
* Cleaning up Collection class and removing default values from the get method
* Fixed ZF2 cache adapters

## 2.8.8 - 2012-10-15

* Bug: Fixed a cookie issue that caused dot prefixed domains to not match where popular browsers did

## 2.8.7 - 2012-09-30

* Bug: Fixed config file aliases for JSON includes
* Bug: Fixed cookie bug on a request object by using CookieParser to parse cookies on requests
* Bug: Removing the path to a file when sending a Content-Disposition header on a POST upload
* Bug: Hardening request and response parsing to account for missing parts
* Bug: Fixed PEAR packaging
* Bug: Fixed Request::getInfo
* Bug: Fixed cases where CURLM\_CALL\_MULTI\_PERFORM return codes were causing curl transactions to fail
* Adding the ability for the namespace Iterator factory to look in multiple directories
* Added more getters/setters/removers from service descriptions
* Added the ability to remove POST fields from OAuth signatures
* OAuth plugin now supports 2-legged OAuth

## 2.8.6 - 2012-09-05

* Added the ability to modify and build service descriptions
* Added the use of visitors to apply parameters to locations in service descriptions using the dynamic command
* Added a json parameter location
* Now allowing dot notation for classes in the CacheAdapterFactory
* Using the union of two arrays rather than an array\_merge when extending service builder services and service params
* Ensuring that a service is a string before doing strpos() checks on it when substituting services for references in service builder config files.
* Services defined in two different config files that include one another will by default replace the previously defined service, but you can now create services that extend themselves and merge their settings over the previous
* The JsonLoader now supports aliasing filenames with different filenames. This allows you to alias something like '\_default' with a default JSON configuration file.

## 2.8.5 - 2012-08-29

* Bug: Suppressed empty arrays from URI templates
* Bug: Added the missing $options argument from ServiceDescription::factory to enable caching
* Added support for HTTP responses that do not contain a reason phrase in the start-line
* AbstractCommand commands are now invokable
* Added a way to get the data used when signing an Oauth request before a request is sent

## 2.8.4 - 2012-08-15

* Bug: Custom delay time calculations are no longer ignored in the ExponentialBackoffPlugin
* Added the ability to transfer entity bodies as a string rather than streamed. This gets around curl error 65. Set body\_as\_string in a request's curl options to enable.
* Added a StreamInterface, EntityBodyInterface, and added ftell() to Guzzle\Common\Stream
* Added an AbstractEntityBodyDecorator and a ReadLimitEntityBody decorator to transfer only a subset of a decorated stream
* Stream and EntityBody objects will now return the file position to the previous position after a read required operation (e.g. getContentMd5())
* Added additional response status codes
* Removed SSL information from the default User-Agent header
* DELETE requests can now send an entity body
* Added an EventDispatcher to the ExponentialBackoffPlugin and added an ExponentialBackoffLogger to log backoff retries
* Added the ability of the MockPlugin to consume mocked request bodies
* LogPlugin now exposes request and response objects in the extras array

## 2.8.3 - 2012-07-30

* Bug: Fixed a case where empty POST requests were sent as GET requests
* Bug: Fixed a bug in ExponentialBackoffPlugin that caused fatal errors when retrying an EntityEnclosingRequest that does not have a body
* Bug: Setting the response body of a request to null after completing a request, not when setting the state of a request to new
* Added multiple inheritance to service description commands
* Added an ApiCommandInterface and added getParamNames() and hasParam()
* Removed the default 2mb size cutoff from the Md5ValidatorPlugin so that it now defaults to validating everything
* Changed CurlMulti::perform to pass a smaller timeout to CurlMulti::executeHandles

## 2.8.2 - 2012-07-24

* Bug: Query string values set to 0 are no longer dropped from the query string
* Bug: A Collection object is no longer created each time a call is made to Guzzle\Service\Command\AbstractCommand::getRequestHeaders()
* Bug: + is now treated as an encoded space when parsing query strings
* QueryString and Collection performance improvements
* Allowing dot notation for class paths in filters attribute of a service descriptions

## 2.8.1 - 2012-07-16

* Loosening Event Dispatcher dependency
* POST redirects can now be customized using CURLOPT\_POSTREDIR

## 2.8.0 - 2012-07-15

* BC: Guzzle\Http\Query
  + Query strings with empty variables will always show an equal sign unless the variable is set to QueryString::BLANK (e.g. ?acl= vs ?acl)
  + Changed isEncodingValues() and isEncodingFields() to isUrlEncoding()
  + Changed setEncodeValues(bool) and setEncodeFields(bool) to useUrlEncoding(bool)
  + Changed the aggregation functions of QueryString to be static methods
  + Can now use fromString() with querystrings that have a leading ?
* cURL configuration values can be specified in service descriptions using curl. prefixed parameters
* Content-Length is set to 0 before emitting the request.before\_send event when sending an empty request body
* Cookies are no longer URL decoded by default
* Bug: URI template variables set to null are no longer expanded

## 2.7.2 - 2012-07-02

* BC: Moving things to get ready for subtree splits. Moving Inflection into Common. Moving Guzzle\Http\Parser to Guzzle\Parser.
* BC: Removing Guzzle\Common\Batch\Batch::count() and replacing it with isEmpty()
* CachePlugin now allows for a custom request parameter function to check if a request can be cached
* Bug fix: CachePlugin now only caches GET and HEAD requests by default
* Bug fix: Using header glue when transferring headers over the wire
* Allowing deeply nested arrays for composite variables in URI templates
* Batch divisors can now return iterators or arrays

## 2.7.1 - 2012-06-26

* Minor patch to update version number in UA string
* Updating build process

## 2.7.0 - 2012-06-25

* BC: Inflection classes moved to Guzzle\Inflection. No longer static methods. Can now inject custom inflectors into classes.
* BC: Removed magic setX methods from commands
* BC: Magic methods mapped to service description commands are now inflected in the command factory rather than the client \_\_call() method
* Verbose cURL options are no longer enabled by default. Set curl.debug to true on a client to enable.
* Bug: Now allowing colons in a response start-line (e.g. HTTP/1.1 503 Service Unavailable: Back-end server is at capacity)
* Guzzle\Service\Resource\ResourceIteratorApplyBatched now internally uses the Guzzle\Common\Batch namespace
* Added Guzzle\Service\Plugin namespace and a PluginCollectionPlugin
* Added the ability to set POST fields and files in a service description
* Guzzle\Http\EntityBody::factory() now accepts objects with a \_\_toString() method
* Adding a command.before\_prepare event to clients
* Added BatchClosureTransfer and BatchClosureDivisor
* BatchTransferException now includes references to the batch divisor and transfer strategies
* Fixed some tests so that they pass more reliably
* Added Guzzle\Common\Log\ArrayLogAdapter

## 2.6.6 - 2012-06-10

* BC: Removing Guzzle\Http\Plugin\BatchQueuePlugin
* BC: Removing Guzzle\Service\Command\CommandSet
* Adding generic batching system (replaces the batch queue plugin and command set)
* Updating ZF cache and log adapters and now using ZF's composer repository
* Bug: Setting the name of each ApiParam when creating through an ApiCommand
* Adding result\_type, result\_doc, deprecated, and doc\_url to service descriptions
* Bug: Changed the default cookie header casing back to 'Cookie'

## 2.6.5 - 2012-06-03

* BC: Renaming Guzzle\Http\Message\RequestInterface::getResourceUri() to getResource()
* BC: Removing unused AUTH\_BASIC and AUTH\_DIGEST constants from
* BC: Guzzle\Http\Cookie is now used to manage Set-Cookie data, not Cookie data
* BC: Renaming methods in the CookieJarInterface
* Moving almost all cookie logic out of the CookiePlugin and into the Cookie or CookieJar implementations
* Making the default glue for HTTP headers ';' instead of ','
* Adding a removeValue to Guzzle\Http\Message\Header
* Adding getCookies() to request interface.
* Making it easier to add event subscribers to HasDispatcherInterface classes. Can now directly call addSubscriber()

## 2.6.4 - 2012-05-30

* BC: Cleaning up how POST files are stored in EntityEnclosingRequest objects. Adding PostFile class.
* BC: Moving ApiCommand specific functionality from the Inspector and on to the ApiCommand
* Bug: Fixing magic method command calls on clients
* Bug: Email constraint only validates strings
* Bug: Aggregate POST fields when POST files are present in curl handle
* Bug: Fixing default User-Agent header
* Bug: Only appending or prepending parameters in commands if they are specified
* Bug: Not requiring response reason phrases or status codes to match a predefined list of codes
* Allowing the use of dot notation for class namespaces when using instance\_of constraint
* Added any\_match validation constraint
* Added an AsyncPlugin
* Passing request object to the calculateWait method of the ExponentialBackoffPlugin
* Allowing the result of a command object to be changed
* Parsing location and type sub values when instantiating a service description rather than over and over at runtime

## 2.6.3 - 2012-05-23

* [BC] Guzzle\Common\FromConfigInterface no longer requires any config options.
* [BC] Refactoring how POST files are stored on an EntityEnclosingRequest. They are now separate from POST fields.
* You can now use an array of data when creating PUT request bodies in the request factory.
* Removing the requirement that HTTPS requests needed a Cache-Control: public directive to be cacheable.
* [Http] Adding support for Content-Type in multipart POST uploads per upload
* [Http] Added support for uploading multiple files using the same name (foo[0], foo[1])
* Adding more POST data operations for easier manipulation of POST data.
* You can now set empty POST fields.
* The body of a request is only shown on EntityEnclosingRequest objects that do not use POST files.
* Split the Guzzle\Service\Inspector::validateConfig method into two methods. One to initialize when a command is created, and one to validate.
* CS updates

## 2.6.2 - 2012-05-19

* [Http] Better handling of nested scope requests in CurlMulti. Requests are now always prepares in the send() method rather than the addRequest() method.

## 2.6.1 - 2012-05-19

* [BC] Removing 'path' support in service descriptions. Use 'uri'.
* [BC] Guzzle\Service\Inspector::parseDocBlock is now protected. Adding getApiParamsForClass() with cache.
* [BC] Removing Guzzle\Common\NullObject. Use <https://github.com/mtdowling/NullObject> if you need it.
* [BC] Removing Guzzle\Common\XmlElement.
* All commands, both dynamic and concrete, have ApiCommand objects.
* Adding a fix for CurlMulti so that if all of the connections encounter some sort of curl error, then the loop exits.
* Adding checks to EntityEnclosingRequest so that empty POST files and fields are ignored.
* Making the method signature of Guzzle\Service\Builder\ServiceBuilder::factory more flexible.

## 2.6.0 - 2012-05-15

* [BC] Moving Guzzle\Service\Builder to Guzzle\Service\Builder\ServiceBuilder
* [BC] Executing a Command returns the result of the command rather than the command
* [BC] Moving all HTTP parsing logic to Guzzle\Http\Parsers. Allows for faster C implementations if needed.
* [BC] Changing the Guzzle\Http\Message\Response::setProtocol() method to accept a protocol and version in separate args.
* [BC] Moving ResourceIterator\* to Guzzle\Service\Resource
* [BC] Completely refactored ResourceIterators to iterate over a cloned command object
* [BC] Moved Guzzle\Http\UriTemplate to Guzzle\Http\Parser\UriTemplate\UriTemplate
* [BC] Guzzle\Guzzle is now deprecated
* Moving Guzzle\Common\Guzzle::inject to Guzzle\Common\Collection::inject
* Adding Guzzle\Version class to give version information about Guzzle
* Adding Guzzle\Http\Utils class to provide getDefaultUserAgent() and getHttpDate()
* Adding Guzzle\Curl\CurlVersion to manage caching curl\_version() data
* ServiceDescription and ServiceBuilder are now cacheable using similar configs
* Changing the format of XML and JSON service builder configs. Backwards compatible.
* Cleaned up Cookie parsing
* Trimming the default Guzzle User-Agent header
* Adding a setOnComplete() method to Commands that is called when a command completes
* Keeping track of requests that were mocked in the MockPlugin
* Fixed a caching bug in the CacheAdapterFactory
* Inspector objects can be injected into a Command object
* Refactoring a lot of code and tests to be case insensitive when dealing with headers
* Adding Guzzle\Http\Message\HeaderComparison for easy comparison of HTTP headers using a DSL
* Adding the ability to set global option overrides to service builder configs
* Adding the ability to include other service builder config files from within XML and JSON files
* Moving the parseQuery method out of Url and on to QueryString::fromString() as a static factory method.

## 2.5.0 - 2012-05-08

* Major performance improvements
* [BC] Simplifying Guzzle\Common\Collection. Please check to see if you are using features that are now deprecated.
* [BC] Using a custom validation system that allows a flyweight implementation for much faster validation. No longer using Symfony2 Validation component.
* [BC] No longer supporting "{{ }}" for injecting into command or UriTemplates. Use "{}"
* Added the ability to passed parameters to all requests created by a client
* Added callback functionality to the ExponentialBackoffPlugin
* Using microtime in ExponentialBackoffPlugin to allow more granular backoff strategies.
* Rewinding request stream bodies when retrying requests
* Exception is thrown when JSON response body cannot be decoded
* Added configurable magic method calls to clients and commands. This is off by default.
* Fixed a defect that added a hash to every parsed URL part
* Fixed duplicate none generation for OauthPlugin.
* Emitting an event each time a client is generated by a ServiceBuilder
* Using an ApiParams object instead of a Collection for parameters of an ApiCommand
* cache.\* request parameters should be renamed to params.cache.\*
* Added the ability to set arbitrary curl options on requests (disable\_wire, progress, etc.). See CurlHandle.
* Added the ability to disable type validation of service descriptions
* ServiceDescriptions and ServiceBuilders are now Serializable