CHANGELOG

v14.2.0

- Added package comment to make `github.com/Azure/go-autorest` importable.

v14.1.1

Bug Fixes

- Change `x-ms-authorization-auxiliary` header value separator to comma.

v14.1.0

New Features

- Added `azure.SetEnvironment()` that will update the global environments map with the specified values.

v14.0.1

Bug Fixes

- Fix race condition when refreshing token.
- Fixed some tests to work with Go 1.14.

v14.0.0

Breaking Changes

- By default, the `DoRetryForStatusCodes` functions will no longer infinitely retry a request when the response returns an HTTP status code of 429 (StatusTooManyRequests). To opt in to the old behavior set `autorest.Count429AsRetry` to `false`.

New Features

- Variable `autorest.Max429Delay` can be used to control the maximum delay between retries when a 429 is received with no `Retry-After` header. The default is zero which means there is no cap.

v13.4.0

New Features

- Added field `SendDecorators` to the `Client` type. This can be used to specify a custom chain of SendDecorators per client.
- Added method `Client.Send()` which includes logic for selecting the preferred chain of SendDecorators.

v13.3.3

Bug Fixes

- Fixed connection leak when retrying requests.

- Enabled exponential back-off with a 2-minute cap when retrying on 429.
- Fixed some cases where errors were inadvertently dropped.

v13.3.2

Bug Fixes

- Updated `autorest.AsStringSlice()` to convert slice elements to their string representation.

v13.3.1

- Updated external dependencies.

Bug Fixes

v13.3.0

New Features

- Added support for shared key and shared access signature token authorization.
  - `autorest.NewSharedKeyAuthorizer()` and dependent types.
  - `autorest.NewSASTokenAuthorizer()` and dependent types.
- Added `ServicePrincipalToken.SetCustomRefresh()` so a custom refresh function can be invoked when a token has expired.

Bug Fixes

- Fixed `cli.AccessTokensPath()` to respect `AZURE_CONFIG_DIR` when set.
- Support parsing error messages in XML responses.

v13.2.0

New Features

- Added the following functions to replace their versions that don't take a context.
  - `adal.InitiateDeviceAuthWithContext()`
  - `adal.CheckForUserCompletionWithContext()`
  - `adal.WaitForUserCompletionWithContext()`

v13.1.0

New Features

- Added support for MSI authentication on Azure App Service and Azure Functions.

v13.0.2

Bug Fixes

- Always retry a request even if the sender returns a non-nil error.

v13.0.1

Bug Fixes

- Fixed `autorest.WithQueryParameters()` so that it properly encodes multi-value query parameters.

v13.0.0

Breaking Changes

The `tracing` package has been rewritten to provide a common interface for consumers to wire in the tracing package of their choice. What this means is that by default no tracing provider will be compiled into your program and setting the `AZURE_SDK_TRACING_ENABLED` environment variable will have no effect. To enable this previous behavior you must now add the following import to your source file.

```
import _ "github.com/Azure/go-autorest/tracing/opencensus"
```

The APIs required by autorest-generated code have remained but some APIs have been removed and new ones added. The following APIs and variables have been removed (the majority of them were moved to the `opencensus` package).

- tracing.Transport
- tracing.Enable()
- tracing.EnableWithAIForwarding()
- tracing.Disable()

The following APIs and types have been added

- tracing.Tracer
- tracing.Register()

To hook up a tracer simply call `tracing.Register()` passing in a type that satisfies the `tracing.Tracer` interface.

v12.4.3

Bug Fixes

- `autorest.MultiTenantServicePrincipalTokenAuthorizer` will now properly add its auxiliary bearer tokens.

v12.4.2

Bug Fixes

- Improvements to the fixes made in v12.4.1.
  - Remove `override` stanza from Gopkg.toml and `replace` directive from go.mod as they don't apply when being consumed as a dependency.
  - Switched to latest version of `ocagent` that still depends on protobuf v1.2.
  - Add indirect dependencies to the `required` clause with matching `constraint` stanzas so that `dep` dependencies match go.sum.

v12.4.1

Bug Fixes

- Updated OpenCensus and OCAgent versions to versions that don't depend on v1.3+ of protobuf as it was breaking kubernetes.
- Pinned opencensus-proto to a version that's compatible with our versions of OpenCensus and OCAgent.

v12.4.0

New Features

- Added `autorest.WithPrepareDecorators` and `autorest.GetPrepareDecorators` for adding and retrieving a custom chain of PrepareDecorators to the provided context.

v12.3.0

New Features

- Support for multi-tenant via x-ms-authorization-auxiliary header has been added for client credentials with secret scenario; this basically bundles multiple OAuthConfig and ServicePrincipalToken types into corresponding MultiTenant* types along with a new authorizer that adds the primary and auxiliary token headers to the reqest. The authenticaion helpers have been updated to support this scenario; if environment var AZURE_AUXILIARY_TENANT_IDS is set with a semicolon delimited list of tenants the multi-tenant codepath will kick in to create the appropriate authorizer. See `adal.NewMultiTenantOAuthConfig` , `adal.NewMultiTenantServicePrincipalToken` and `autorest.NewMultiTenantServicePrincipalTokenAuthorizer` along with their supporting types and methods.
- Added `autorest.WithSendDecorators` and `autorest.GetSendDecorators` for adding and retrieving a custom chain of SendDecorators to the provided context.
- Added `autorest.DoRetryForStatusCodesWithCap` and `autorest.DelayForBackoffWithCap` to enforce an upper bound on the duration between retries.

v12.2.0

New Features

- Added `autorest.WithXML` , `autorest.AsMerge` , `autorest.WithBytes` preparer decorators.
- Added `autorest.ByUnmarshallingBytes` response decorator.
- Added `Response.IsHTTPStatus` and `Response.HasHTTPStatus` helper methods for inspecting HTTP status code in `autorest.Response` types.

Bug Fixes

- `autorest.DelayWithRetryAfter` now supports HTTP-Dates in the `Retry-After` header and is not limited to just 429 status codes.

v12.1.0

New Features

- Added `to.ByteSlicePtr()` .
- Added blob/queue storage resource ID to `azure.ResourceIdentifier` .

v12.0.0

Breaking Changes

In preparation for modules the following deprecated content has been removed.

- async.NewFuture()
- async.Future.Done()
- async.Future.WaitForCompletion()
- async.DoPollForAsynchronous()
- The `utils` package
- validation.NewErrorWithValidationError()
- The `version` package

v11.9.0

New Features

- Add `ResourceIdentifiers` field to `azure.Environment` containing resource IDs for public and sovereign clouds.

v11.8.0

New Features

- Added `autorest.NewClientWithOptions()` to support endpoints that require free renegotiation.

v11.7.1

Bug Fixes

- Fix missing support for http(s) proxy when using the default sender.

v11.7.0

New Features

- Added methods to obtain a ServicePrincipalToken on the various credential configuration types in the `auth` package.

v11.6.1

Bug Fixes

- Fix ACR DNS endpoint for government clouds.
- Add Cosmos DB DNS endpoints.
- Update dependencies to resolve build breaks in OpenCensus.

v11.6.0

New Features

- Added type `autorest.BasicAuthorizer` to support Basic authentication.

v11.5.2

Bug Fixes

- Fixed `GetTokenFromCLI` did not work with zsh.

v11.5.1

Bug Fixes

- In `Client.sender()` set the minimum TLS version on HTTP clients to 1.2.

v11.5.0

New Features

- The `auth` package has been refactored so that the environment and file settings are now available.
- The methods used in `auth.NewAuthorizerFromEnvironment()` are now exported so that custom authorization chains can be created.
- Added support for certificate authorization for file-based config.

v11.4.0

New Features

- Added `adal.AddToUserAgent()` so callers can append custom data to the user-agent header used for ADAL requests.
- Exported `adal.UserAgent()` for parity with `autorest.Client`.

v11.3.2

Bug Fixes

- In `Future.WaitForCompletionRef()` if the provided context has a deadline don't add the default deadline.

v11.3.1

Bug Fixes

- For an LRO PUT operation the final GET URL was incorrectly set to the Location polling header in some cases.

v11.3.0

New Features

- Added method `ServicePrincipalToken()` to `DeviceFlowConfig` type.

v11.2.8

Bug Fixes

- Deprecate content in the `version` package. The functionality has been superseded by content in the `autorest` package.

v11.2.7

Bug Fixes

- Fix environment variable name for enabling tracing from `AZURE_SDK_TRACING_ENABELD` to `AZURE_SDK_TRACING_ENABLED`. Note that for backward compatibility reasons, both will work until the next major version release of the package.

v11.2.6

Bug Fixes

- If zero bytes are read from a polling response body don't attempt to unmarshal them.

v11.2.5

Bug Fixes

- Removed race condition in `autorest.DoRetryForStatusCodes`.

v11.2.4

Bug Fixes

- Function `cli.ProfilePath` now respects environment `AZURE_CONFIG_DIR` if available.

v11.2.1

NOTE: Versions of Go prior to 1.10 have been removed from CI as they no longer work with golint.

Bug Fixes

- Method `MSIConfig.Authorizer` now supports user-assigned identities.
- The adal package now reports its own user-agent string.

v11.2.0

New Features

- Added `tracing` package that enables instrumentation of HTTP and API calls. Setting the env variable `AZURE_SDK_TRACING_ENABLED` or calling `tracing.Enable` will start instrumenting the code for metrics and traces. Additionally, setting the env variable `OCAGENT_TRACE_EXPORTER_ENDPOINT` or calling `tracing.EnableWithAIForwarding` will start the instrumentation and connect to an App Insights Local Forwarder that is needs to be running. Note that if the AI Local Forwarder is not running tracking will still be enabled. By default, instrumentation is disabled. Once enabled, instrumentation can also be programatically disabled by calling `Disable`.
- Added `DoneWithContext` call for checking LRO status. `Done` has been deprecated.

Bug Fixes

- Don't use the initial request's context for LRO polling.
- Don't override the `refreshLock` and the `http.Client` when unmarshalling `ServicePrincipalToken` if it is already set.

v11.1.1

Bug Fixes

- When creating a future always include the polling tracker even if there's a failure; this allows the underlying response to be obtained by the caller.

v11.1.0

New Features

- Added `auth.NewAuthorizerFromCLI` to create an authorizer configured from the Azure 2.0 CLI.
- Added `adal.NewOAuthConfigWithAPIVersion` to create an OAuthConfig with the specified API version.

v11.0.1

New Features

- Added `x5c` header to client assertion for certificate Issuer+Subject Name authentication.

v11.0.0

Breaking Changes

- To handle differences between ADFS and AAD the following fields have had their types changed from `string` to `json.Number`
    - ExpiresIn
    - ExpiresOn
    - NotBefore

New Features

- Added `auth.NewAuthorizerFromFileWithResource` to create an authorizer from the config file with the specified resource.
- Setting a client's `PollingDuration` to zero will use the provided context to control a LRO's polling duration.

v10.15.5

Bug Fixes

- In `DoRetryForStatusCodes`, if a request's context is cancelled return the last response.

v10.15.4

Bug Fixes

- If a polling operation returns a failure status code return the associated error.

v10.15.3

Bug Fixes

- Initialize the polling URL and method for an LRO tracker on each iteration, favoring the Azure-AsyncOperation header.

v10.15.2

Bug Fixes

- Use fmt.Fprint when printing request/response so that any escape sequences aren't treated as format specifiers.

v10.15.1

Bug Fixes

- If an LRO API returns a `Failed` provisioning state in the initial response return an error at that point so the caller doesn't have to poll.
- For failed LROs without an OData v4 error include the response body in the error's `AdditionalInfo` field to aid in diagnosing the failure.

v10.15.0

New Features

- Add initial support for request/response logging via setting environment variables. Setting `AZURE_GO_SDK_LOG_LEVEL` to `LogInfo` will log request/response without their bodies. To include the bodies set the log level to `LogDebug`. By default the logger writes to strerr, however it can also write to stdout or a file if specified in `AZURE_GO_SDK_LOG_FILE`. Note that if the specified file already exists it will be truncated. IMPORTANT: by default the logger will redact the Authorization and Ocp-Apim-Subscription-Key headers. Any other secrets will *not* be redacted.

v10.14.0

New Features

- Added package version that contains version constants and user-agent data.

Bug Fixes

- Add the user-agent to token requests.

v10.13.0

- Added support for additionalInfo in ServiceError type.

v10.12.0

New Features

- Added field ServicePrincipalToken.MaxMSIRefreshAttempts to configure the maximun number of attempts to refresh an MSI token.

v10.11.4

Bug Fixes

- If an LRO returns http.StatusOK on the initial response with no async headers return the response body from Future.GetResult().
- If there is no "final GET URL" return an error from Future.GetResult().

v10.11.3

Bug Fixes

- In IMDS retry logic, if we don't receive a response don't retry.
  - Renamed the retry function so it's clear it's meant for IMDS only.
- For error response bodies that aren't OData-v4 compliant stick the raw JSON in the ServiceError.Details field so the information isn't lost.
  - Also add the raw HTTP response to the DetailedResponse.
- Removed superfluous wrapping of response error in azure.DoRetryWithRegistration().

v10.11.2

Bug Fixes

- Validation for integers handles int and int64 types.

v10.11.1

Bug Fixes

- Adding User information to authorization config as parsed from CLI cache.

v10.11.0

New Features

- Added NewServicePrincipalTokenFromManualTokenSecret for creating a new SPT using a manual token and secret
- Added method ServicePrincipalToken.MarshalTokenJSON() to marshall the inner Token

v10.10.0

New Features

- Most ServicePrincipalTokens can now be marshalled/unmarshall to/from JSON (ServicePrincipalCertificateSecret and ServicePrincipalMSISecret are not supported).
- Added method ServicePrincipalToken.SetRefreshCallbacks().

v10.9.2

Bug Fixes

- Refreshing a refresh token obtained from a web app authorization code now works.

v10.9.1

Bug Fixes

- The retry logic for MSI token requests now uses exponential backoff per the guidelines.
- IsTemporaryNetworkError() will return true for errors that don't implement the net.Error interface.

v10.9.0

Deprecated Methods

Old Method	New Method

azure.NewFuture()	azure.NewFutureFromResponse()
Future.WaitForCompletion()	Future.WaitForCompletionRef()

New Features

- Added azure.NewFutureFromResponse() for creating a Future from the initial response from an async operation.
- Added Future.GetResult() for making the final GET call to retrieve the result from an async operation.

Bug Fixes

- Some futures failed to return their results, this should now be fixed.

v10.8.2

Bug Fixes

- Add nil-gaurd to token retry logic.

v10.8.1

Bug Fixes

- Return a TokenRefreshError if the sender fails on the initial request.
- Don't retry on non-temporary network errors.

v10.8.0

- Added NewAuthorizerFromEnvironmentWithResource() helper function.

v10.7.0

New Features

- Added *WithContext() methods to ADAL token refresh operations.

v10.6.2

- Fixed a bug on device authentication.

v10.6.1

- Added retries to MSI token get request.

v10.6.0

- Changed MSI token implementation. Now, the token endpoint is the IMDS endpoint.

v10.5.1

Bug Fixes

- `DeviceFlowConfig.Authorizer()` now prints the device code message when running `go test`. `-v` flag is required.

v10.5.0

New Features

- Added NewPollingRequestWithContext() for use with polling asynchronous operations.

Bug Fixes

- Make retry logic use the request's context instead of the deprecated Cancel object.

v10.4.0

New Features

- Added helper for parsing Azure Resource ID's.
- Added deprecation message to utils.GetEnvVarOrExit()

v10.3.0

New Features

- Added EnvironmentFromURL method to load an Environment from a given URL. This function is particularly useful in the private and hybrid Cloud model, where one may define their own endpoints
- Added TokenAudience endpoint to Environment structure. This is useful in private and hybrid cloud models where TokenAudience endpoint can be different from ResourceManagerEndpoint

v10.2.0

New Features

- Added endpoints for batch management.

v10.1.3

Bug Fixes

- In Client.Do() invoke WithInspection() last so that it will inspect WithAuthorization().
- Fixed authorization methods to invoke p.Prepare() first, aligning them with the other preparers.

v10.1.2

- Corrected comment for auth.NewAuthorizerFromFile() function.

v10.1.1

- Updated version number to match current release.

v10.1.0

New Features

- Expose the polling URL for futures.

Bug Fixes

- Add validation.NewErrorWithValidationError back to prevent breaking changes (it is deprecated).

v10.0.0

New Features

- Added target and innererror fields to ServiceError to comply with OData v4 spec.
- The Done() method on futures will now return a ServiceError object when available (it used to return a partial value of such errors).
- Added helper methods for obtaining authorizers.
- Expose the polling URL for futures.

Bug Fixes

- Switched from glide to dep for dependency management.
- Fixed unmarshaling of ServiceError for JSON bodies that don't conform to the OData spec.
- Fixed a race condition in token refresh.

Breaking Changes

- The ServiceError.Details field type has been changed to match the OData v4 spec.
- Go v1.7 has been dropped from CI.
- API parameter validation failures will now return a unique error type validation.Error.
- The adal.Token type has been decomposed from adal.ServicePrincipalToken (this was necessary in order to fix the token refresh race).

v9.10.0

- Fix the Service Bus suffix in Azure public env
- Add Service Bus Endpoint (AAD ResourceURI) for use in Azure Service Bus RBAC Preview

v9.9.0

New Features

- Added EventGridKeyAuthorizer for key authorization with event grid topics.

Bug Fixes

- Fixed race condition when auto-refreshing service principal tokens.

v9.8.1

Bug Fixes

- Added http.StatusNoContent (204) to the list of expected status codes for long-running operations.
- Updated runtime version info so it's current.

v9.8.0

New Features

- Added type azure.AsyncOpIncompleteError to be returned from a future's Result() method when the operation has not completed.

v9.7.1

Bug Fixes

- Use correct AAD and Graph endpoints for US Gov environment.

v9.7.0

New Features

- Added support for application/octet-stream MIME types.

v9.6.1

Bug Fixes

- Ensure Authorization header is added to request when polling for registration status.

v9.6.0

New Features

- Added support for acquiring tokens via MSI with a user assigned identity.

v9.5.3

Bug Fixes

- Don't remove encoding of existing URL Query parameters when calling autorest.WithQueryParameters.
- Set correct Content Type when using autorest.WithFormData.

v9.5.2

Bug Fixes

- Check for nil *http.Response before dereferencing it.

v9.5.1

Bug Fixes

- Don't count http.StatusTooManyRequests (429) against the retry cap.
- Use retry logic when SkipResourceProviderRegistration is set to true.

v9.5.0

New Features

- Added support for username + password, API key, authoriazation code and cognitive services authentication.
- Added field SkipResourceProviderRegistration to clients to provide a way to skip auto-registration of RPs.
- Added utility function AsStringSlice() to convert its parameters to a string slice.

Bug Fixes

- When checking for authentication failures look at the error type not the status code as it could vary.

v9.4.2

Bug Fixes

- Validate parameters when creating credentials.
- Don't retry requests if the returned status is a 401 (http.StatusUnauthorized) as it will never succeed.

v9.4.1

Bug Fixes

- Update the AccessTokensPath() to read access tokens path through AZURE_ACCESS_TOKEN_FILE. If this environment variable is not set, it will fall back to use default path set by Azure CLI.
- Use case-insensitive string comparison for polling states.

v9.4.0

New Features

- Added WaitForCompletion() to Future as a default polling implementation.

Bug Fixes

- Method Future.Done() shouldn't update polling status for unexpected HTTP status codes.

v9.3.1

Bug Fixes

- DoRetryForStatusCodes will retry if sender.Do returns a non-nil error.

v9.3.0

New Features

- Added PollingMethod() to Future so callers know what kind of polling mechanism is used.
- Added azure.ChangeToGet() which transforms an http.Request into a GET (to be used with LROs).

v9.2.0

New Features

- Added support for custom Azure Stack endpoints.
- Added type azure.Future used to track the status of long-running operations.

Bug Fixes

- Preserve the original error in DoRetryWithRegistration when registration fails.

v9.1.1

- Fixes a bug regarding the cookie jar on `autorest.Client.Sender`.

v9.1.0

New Features

- In cases where there is a non-empty error from the service, attempt to unmarshal it instead of uniformly calling it an "Unknown" error.
- Support for loading Azure CLI Authentication files.
- Automatically register your subscription with the Azure Resource Provider if it hadn't been previously.

Bug Fixes

- RetriableRequest can now tolerate a ReadSeekable body being read but not reset.

- Adding missing Apache Headers

v9.0.0

> ***IMPORTANT:*** *This release was intially labeled incorrectly as* `v8.4.0`*. From the time it was released, it should have been marked* `v9.0.0` *because it contains breaking changes to the MSI packages. We appologize for any inconvenience this causes.*

Adding MSI Endpoint Support and CLI token rehydration.

v8.3.1

Pick up bug fix in adal for MSI support.

v8.3.0

Updates to Error string formats for clarity. Also, adding a copy of the http.Response to errors for an improved debugging experience.

v8.2.0

New Features

- Add support for bearer authentication callbacks
- Support 429 response codes that include "Retry-After" header
- Support validation constraint "Pattern" for map keys

Bug Fixes

- Make RetriableRequest work with multiple versions of Go

v8.1.1

Updates the RetriableRequest to take advantage of GetBody() added in Go 1.8.

v8.1.0

Adds RetriableRequest type for more efficient handling of retrying HTTP requests.

v8.0.0

ADAL refactored into its own package. Support for UNIX time.

v7.3.1

- Version Testing now removed from production bits that are shipped with the library.

v7.3.0

- Exposing new `RespondDecorator`, `ByDiscardingBody`. This allows operations to acknowledge that they do not need either the entire or a trailing portion of accepts response body. In doing so, Go's http library can reuse HTTP connections more readily.
- Adding `PrepareDecorator` to target custom BaseURLs.

- Adding ACR suffix to public cloud environment.
- Updating Glide dependencies.

v7.2.5

- Fixed the Active Directory endpoint for the China cloud.
- Removes UTF-8 BOM if present in response payload.
- Added telemetry.

v7.2.3

- Fixing bug in calls to `DelayForBackoff` that caused doubling of delay duration.

v7.2.2

- autorest/azure: added ASM and ARM VM DNS suffixes.

v7.2.1

- fixed parsing of UTC times that are not RFC3339 conformant.

v7.2.0

- autorest/validation: Reformat validation error for better error message.

v7.1.0

- preparer: Added support for multipart formdata - WithMultiPartFormdata()
- preparer: Added support for sending file in request body - WithFile
- client: Added RetryDuration parameter.
- autorest/validation: new package for validation code for Azure Go SDK.

v7.0.7

- Add trailing / to endpoint
- azure: add EnvironmentFromName

v7.0.6

- Add retry logic for 408, 500, 502, 503 and 504 status codes.
- Change url path and query encoding logic.
- Fix DelayForBackoff for proper exponential delay.
- Add CookieJar in Client.

v7.0.5

- Add check to start polling only when status is in [200,201,202].
- Refactoring for unchecked errors.
- azure/persist changes.
- Fix 'file in use' issue in renewing token in deviceflow.
- Store header RetryAfter for subsequent requests in polling.
- Add attribute details in service error.

v7.0.4

- Better error messages for long running operation failures

v7.0.3

- Corrected DoPollForAsynchronous to properly handle the initial response

v7.0.2

- Corrected DoPollForAsynchronous to continue using the polling method first discovered

v7.0.1

- Fixed empty JSON input error in ByUnmarshallingJSON
- Fixed polling support for GET calls
- Changed format name from TimeRfc1123 to TimeRFC1123

v7.0.0

- Added ByCopying responder with supporting TeeReadCloser
- Rewrote Azure asynchronous handling
- Reverted to only unmarshalling JSON
- Corrected handling of RFC3339 time strings and added support for Rfc1123 time format

The `json.Decoder` does not catch bad data as thoroughly as `json.Unmarshal`. Since `encoding/json` successfully deserializes all core types, and extended types normally provide their custom JSON serialization handlers, the code has been reverted back to using `json.Unmarshal`. The original change to use `json.Decode` was made to reduce duplicate code; there is no loss of function, and there is a gain in accuracy, by reverting.

Additionally, Azure services indicate requests to be polled by multiple means. The existing code only checked for one of those (that is, the presence of the `Azure-AsyncOperation` header). The new code correctly covers all cases and aligns with the other Azure SDKs.

v6.1.0

- Introduced `date.ByUnmarshallingJSONDate` and `date.ByUnmarshallingJSONTime` to enable JSON encoded values.

v6.0.0

- Completely reworked the handling of polled and asynchronous requests
- Removed unnecessary routines
- Reworked `mocks.Sender` to replay a series of `http.Response` objects
- Added `PrepareDecorators` for primitive types (e.g., bool, int32)

Handling polled and asynchronous requests is no longer part of `Client#Send`. Instead new `SendDecorators` implement different styles of polled behavior. See `autorest.DoPollForStatusCodes` and `azure.DoPollForAsynchronous` for examples.

v5.0.0

- Added new RespondDecorators unmarshalling primitive types
- Corrected application of inspection and authorization PrependDecorators

v4.0.0

- Added support for Azure long-running operations.
- Added cancelation support to all decorators and functions that may delay.
- Breaking: `DelayForBackoff` now accepts a channel, which may be nil.

v3.1.0

- Add support for OAuth Device Flow authorization.
- Add support for ServicePrincipalTokens that are backed by an existing token, rather than other secret material.
- Add helpers for persisting and restoring Tokens.
- Increased code coverage in the github.com/Azure/autorest/azure package

v3.0.0

- Breaking: `NewErrorWithError` no longer takes `statusCode int` .
- Breaking: `NewErrorWithStatusCode` is replaced with `NewErrorWithResponse` .
- Breaking: `Client#Send()` no longer takes `codes ...int` argument.
- Add: XML unmarshaling support with `ByUnmarshallingXML()`
- Stopped vending dependencies locally and switched to Glide. Applications using this library should either use Glide or vendor dependencies locally some other way.
- Add: `azure.WithErrorUnlessStatusCode()` decorator to handle Azure errors.
- Fix: use `net/http.DefaultClient` as base client.
- Fix: Missing inspection for polling responses added.
- Add: CopyAndDecode helpers.
- Improved `./autorest/to` with `[]string` helpers.
- Removed golint suppressions in .travis.yml.

v2.1.0

- Added `StatusCode` to `Error` for more easily obtaining the HTTP Reponse StatusCode (if any)

v2.0.0

- Changed `to.StringMapPtr` method signature to return a pointer
- Changed `ServicePrincipalCertificateSecret` and `NewServicePrincipalTokenFromCertificate` to support generic certificate and private keys

v1.0.0

- Added Logging inspectors to trace http.Request / Response
- Added support for User-Agent header
- Changed WithHeader PrepareDecorator to use set vs. add
- Added JSON to error when unmarshalling fails
- Added Client#Send method
- Corrected case of "Azure" in package paths
- Added "to" helpers, Azure helpers, and improved ease-of-use
- Corrected golint issues

v1.0.1

- Added CHANGELOG.md

v1.1.0

- Added mechanism to retrieve a ServicePrincipalToken using a certificate-signed JWT
- Added an example of creating a certificate-based ServicePrincipal and retrieving an OAuth token using the certificate

v1.1.1

- Introduce godeps and vendor dependencies introduced in v1.1.1