



Edinburgh 1.0 Features

July 2019

Introduction



Trevor Conn (@tsconn23)

- Senior Principal Software Developer
- Dell Technologies, Inc
- IoT Division
 - Chair of EdgeX Foundry Core Working Group



<https://github.com/edgexfoundry>

High level feature list (Core Services)

- Targets Go 1.11.x
 - You should still be able to use Go 1.12.x with no adverse effect
 - Modules for dependency mgmt. No more glide.
- Modules
 - Go-mod-core-contracts (request-based contracts for service interaction)
 - Used to be part of the mono-repo: <https://github.com/edgexfoundry/edgex-go/tree/delhi/pkg>
 - Go-mod-registry (configuration and service discovery)
 - Go-mod-messaging (pub/sub via ZeroMQ. Want to add RabbitMQ? This is where you'd do it.)
- CBOR Support for binary data ingestion
- Redis Support
 - Core-Metadata, Core-Data, Support-Notifications, Support-Scheduler, Export-Client
 - If you're using Redis, set support-logging to use "File" logging.
 - Delhi – BSONID as a type within the contract models (ID property)
 - Edinburgh – ID is now a string. Use a UUID.
- Correlation ID usage for service interaction tracing
 - Output level set to "TRACE" by default
- Scheduler service rewrite
 - Delhi implementation was incomplete
- Export Service Deprecation
 - Long term, use the App-Functions-SDK

High level feature list (Device Service SDK)

- DeviceProfile changes
 - When defining a device-profile.yaml document, properties have changed
 - Resources => deviceCommands
 - Commands => coreCommands
 - Float values are now encoded base64 by default. However this can be toggled via the “floatEncoding” property (deviceResources → properties → value → floatEncoding)
- Device model changes
 - Addressable was replaced with Protocols
 - Map[string]ProtocolProperties
 - AutoEvents property is new
 - Used by a device service internally for interacting with a device.
 - Eliminates REST calls to/from the Scheduler
- Reading model changes
 - BinaryValue property added to support CBOR
- AutoEvents
 - Within the SDK itself, there is now an autonomous scheduler that can utilize AutoEvents definitions on a per-device basis for actuation and readings (pull)

Application Functions SDK (brand new – v0.1.0)

- This will replace the Export services
- New SDK for building flexible, function-based processing pipelines
 - Function pipeline is declarative currently, not configuration based.
 - See <https://github.com/edgexfoundry/app-functions-sdk-go/tree/master/examples/advanced-filter-convert-publish>
- Events are received via message bus in the same manner as export-distro
 - Note that events are now wrapped in a “MessageEnvelope” type with some metadata properties