

Core Working Group Meeting Notes (21-Mar-2019)

Attendees:

The screenshot shows a window titled "Participants (18)" with a search bar at the top containing the text "Type to filter...". Below the search bar is a list of 18 participants, each with a circular profile picture, their name, and their affiliation. To the right of each name are icons for video, microphone, and chat, some of which are crossed out with a red slash, indicating they are disabled. The participants listed are:

Name	Company	Video	Microphone	Chat
Trevor Conn	(Host, me)	On	On	On
Akram Ahmad	(Dell)	Off	Off	On
André Srinivasan		Off	Off	On
dianaa		Off	Off	On
Emad Attia	(Intel)	Off	Off	On
Eno		Off	Off	On
Eric Cotter		Off	Off	On
Jim Wang		Off	Off	On
Jim White		Off	Off	On
Joan Duran		Off	Off	On
Malini Bhandaru		Off	Off	On
Michael Hall	(LF)	Off	Off	On
Mike		Off	Off	On
Rodney Hess		Off	Off	On
Steve Osselton		Off	Off	On
Tobias Mosby	(Intel)	Off	Off	On
tonyespy		Off	Off	On
Walt Morton	- Intel	Off	Off	On

Old Business

- Intel -- Meeting scheduled w/Microsoft RE: Go plugins support for Windows
 - *It was rescheduled. Still pending.*

- Operating State Unmarshalling Follow-up
 - *Try OperatingState as pointer on ProvisionWatcher, does that trigger Unmarshal when not specified in the request?*
 - *Answer, no it does not.*
 - *Investigate tags which could be opposite of omitempty and require the field on the request*
 - *Didn't find any such thing. Tags related to marshaling are described starting at line 63 below*
<https://golang.org/src/encoding/json/encode.go>
 - *We have two options currently. They are gross, but they are options.*
 - *Remarshal in the controller method to get our current validation to fire*
 - *Add UnmarshalJSON methods to model types and put the knowledge of valid child type states into the parents.*
 - *E.g. ProvisionWatcher instance checks to see if the OperatingState is empty after unmarshalling.*

New Business

- Documentation needs
 - GoDoc versus published docs
 - *Malini to ask her team for how /whether GoDocs can be used to document not only master but previous feature branches as well.*
 - *Hopeful for response later today*
- Discuss denormalized versus normalized (pseudo) data representation
 - Example: Events/Readings relationship – Discuss denormalization
 - Edge cases around reading updates and backporting to events
- Discuss differences in solutions within Redis provider – Delhi versus Scheduler
 - What should be emphasized – functional completeness, implementation consistency, both?
 - Possibly functional completeness for Edinburgh, implementation consistency via dot releases and Fuji, et al.
- Support-logging for Edinburgh will NOT have Redis support. We have made the choice to use the File-based logging.

Denormalized (redis)

Event1

Reading1

Reading2

Reading3

Readings by ValueDescriptor (1,2,3)

Readings by Device (1,2,3)

Readings by XYZ (1,2,3)

Normalized (mongo)

Events collection

Event1

Reading1 ID

Reading2 ID

Reading3 ID

Readings collection

Reading 1

Reading 2

Reading 3

For next week

- *Device Profile and Device Service simplification issues (Tony)*