

Example – V(I)SS to REST/HTTP conversion

(1)



- For getRequest, use HTTP GET on signal path URL (see below)
- For setRequest, use HTTP PUT on signal path URL (or, arguably POST provides the more correct semantics)
- Convert signal path, from "**Signal.Body.Trunk.IsLocked**" to the following resource locator:
"https://*server*/api/**signal/body/trunk/islocked**"
- Presumably, use JSON format for data exchange
- Use same JSON schema as defined in VISS v1

Example – V(I)SS to REST/HTTP conversion



URLs used in REST lend themselves well to hierarchy:
Requesting subtree gets everything below that level:

- GET `https://server/api/signal/body/`
Will return **all** possible **body** related signals in one go
- GET `https://server/api/signal/body/door`
Will return **all door** related information
- GET `https://server/api/signal/body/door/frontleft/islocked`
Will return only the value of `isLocked` for the left door.

Example – V(I)SS to REST/HTTP conversion



VISS (v1) however allows for flexible wildcards

Signal.Body.Door.*.isLocked → tell me lock status of all doors.

Wildcard mapping in REST/HTTP should be possible.

Proposals: *(Based on input from W3C F2F Munich April 20):*

1) Same as VISS (v1) using some valid wildcard character in the URL:

Example: Let's say “_”:

→ **https://server/api/signal/body/door/_/islocked**

or:

2) Use Xpath (but likely limit the features since Xpath is very flexible & large)

→ Would need more detailed proposal

Example – V(I)SS to REST/HTTP conversion



HTTP Return codes:

This is well defined in ViWi – strive to reuse it.