INTERNATIONAL STANDARD

ISO 20078-2

Second edition
2021-11

Road vehicles — Extended vehicle (ExVe) web services —

Part 2: Access

Véhicules routiers — Web services du véhicule étendu (ExVe) —

Partie 2: Accès

Reference number
ISO 20078-2:2021(E)

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Contents

Page

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of ISO documents should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT), see www.iso.org/iso/foreword.html.

This document was prepared by Technical Committee ISO/TC 22, *Road vehicles,* Subcommittee SC 31, *Data communication*.

This second edition cancels and replaces the first edition (ISO 20078-2:2019), which has been technically revised.

The main changes are as follows:

— added definition of a push method, allowing the offering party to push resources to the accessing party according to subscription;

— added definition of reusable subscription profiles, used to store URI locations and authorization information from the accessing party and used by offering party to push subscribed resources;

— defined requirements for container management API;

— added Annex A and digital Annex B describing container management API;

— revised error format;

— redefined resource versioning.

A list of all parts in the ISO 20078 series can be found on the ISO website.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html.

Road vehicles — Extended vehicle (ExVe) web services —

Part 2:
Access

1 Scope

This document defines how to access resources on a web-services interface of an offering party using the Hypertext Transfer Protocol Secure (HTTPS). Resources can be accessed through request/reply and/or requested to be pushed.

The Representational State Transfer (REST) architectural pattern is chosen as a common way to format resource paths both for request/reply and push. Some specific extensions to this pattern are defined to allow for asynchronous resource requests, such as, for example, forcing readouts of data from a connected vehicle.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO 20078-1, *Road vehicles — Extended vehicle (ExVe) web services — Part 1: Content and definitions*

ISO 8601 (all parts), *Date and time — Representations for information interchange*

3 Terms, definitions and abbreviated terms

3.1 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO 20078-1 apply.

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

— ISO Online browsing platform: available at https://www.iso.org/obp

— IEC Electropedia: available at https://www.electropedia.org/

3.2 Abbreviated terms

For the purposes of this document, the abbreviated terms given in ISO 20078-1 apply.

4 Representational state transfer-based interface

4.1 General

4.1.1 Introduction

There are three different ways to access resources for the accessing party, request/reply, push, and push with a subsequent request/reply. Request/reply is the recommended method, but push can be used when needed. In most cases only one method is used for a particular resource, but sometimes both

request/reply and push are needed. Latency, message size and frequency are examples of requirements to consider when selecting between request/reply, push, and push with subsequent request/reply.

4.1.2 Request/reply

Figure 1 shows an example of request/reply sequence, where the accessing party requests a resource using HTTP GET and the offering party returns it in the payload.

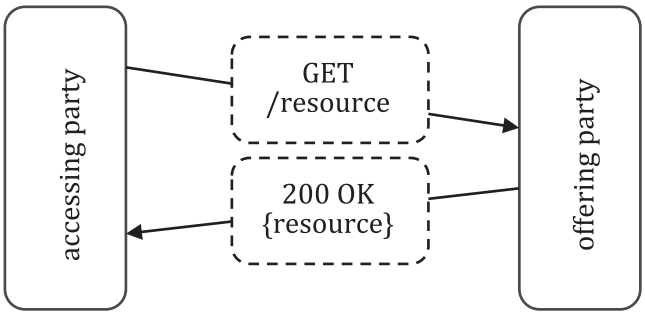

Figure 1 — Request/reply example

4.1.3 Push

Figure 2 shows an example of push sequence, where the accessing party initiates a push of a resource using HTTP POST. The offering party confirms the subscription and starts to push the resource using HTTP POST. Figure 2 shows an example where the push is done two times.

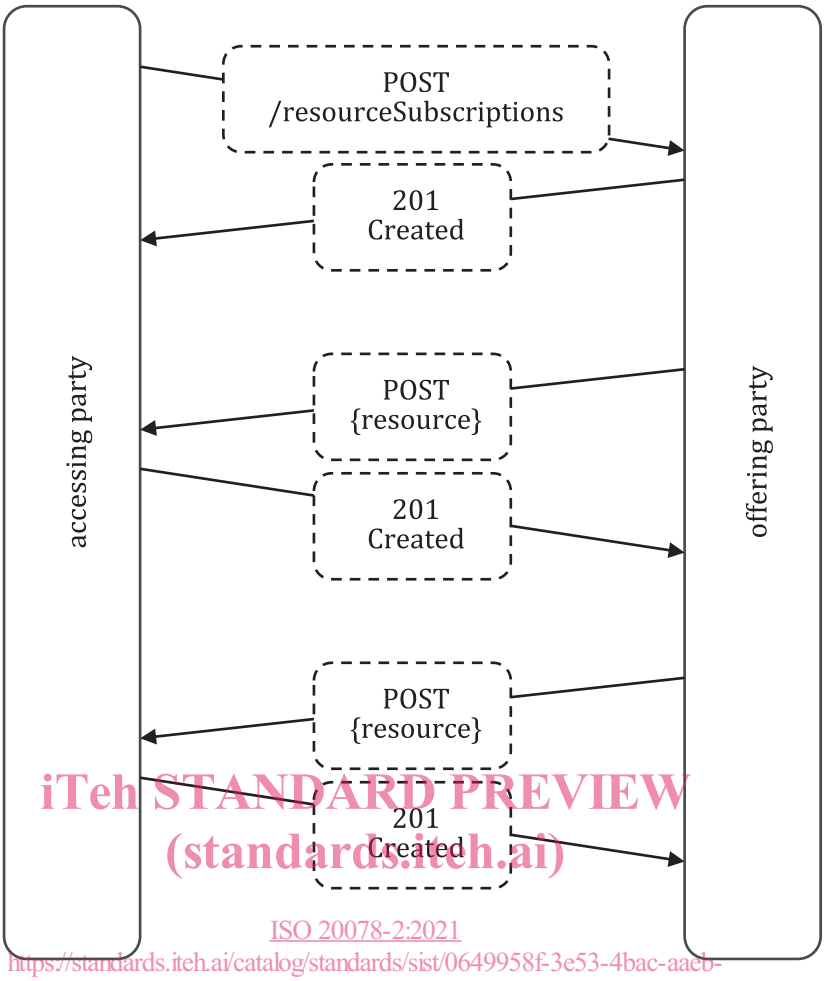

Figure 2 — Push example

Figure 3 shows an example of push sequence, where there is no explicit initiation of the push from the accessing party. It is instead initiated by for instance rules or configuration at the offering party. It is outside the scope of this document to describe how this is agreed between the accessing party and the offering party.

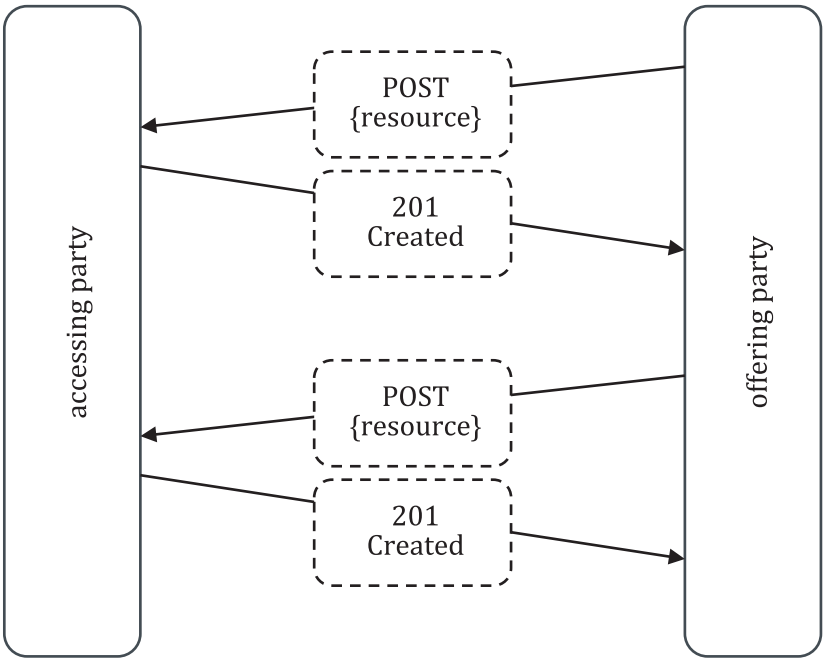

Figure 3 — Push example without explicit initiation

4.1.4 Push with subsequent request/reply

Figure 4 shows an example of push with subsequent request/reply, where the accessing party initiates a push of a resource using HTTP POST. The offering party confirms the subscription and starts to push the resource reference using HTTP POST. The accessing party uses the resource reference to request the resource from the offering party.

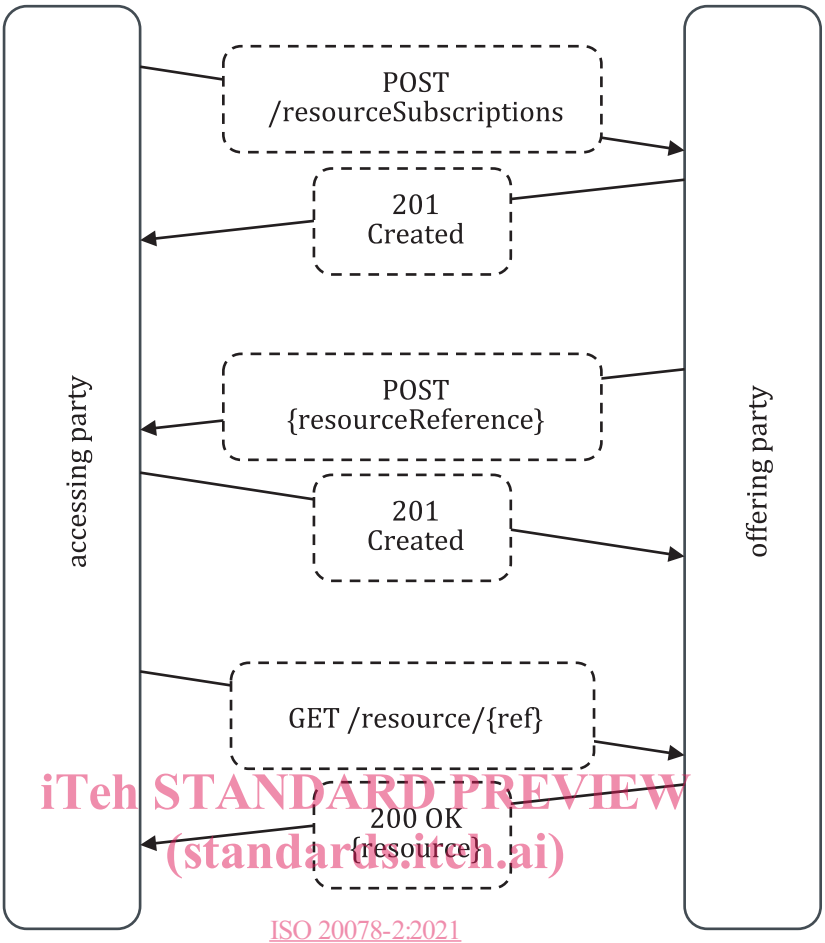

Figure 4 — Push with subsequent request/reply example

Figure 5 shows an example of push with subsequent request/reply, where there is no explicit initiation of the push from the accessing party. It is instead initiated by, for instance, rules or configuration at the offering party. It is outside the scope of this document to describe how this is agreed between the accessing and the offering party.

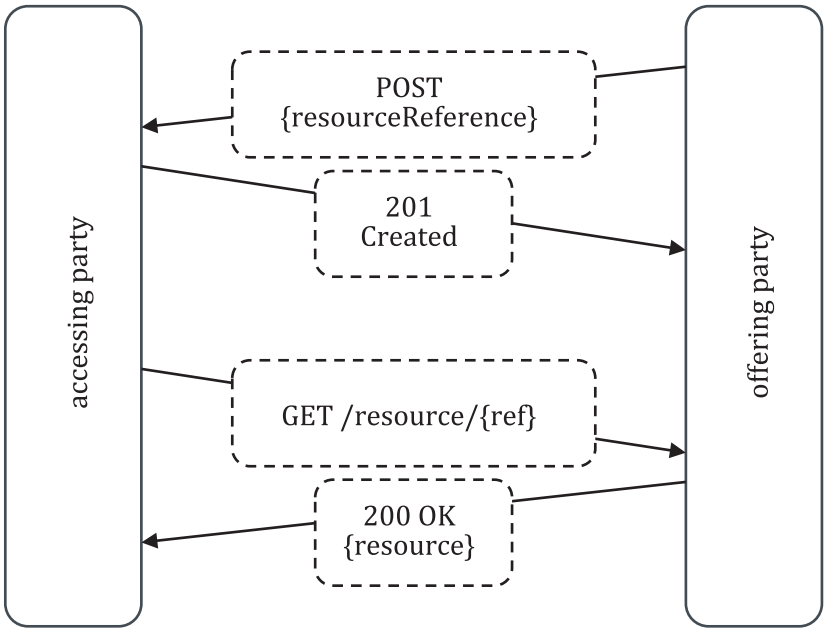

Figure 5 — Push with subsequent request/reply example without explicit initiation

4.1.5 Requirements

The following defines the requirements on a REST[9]-based web service interface using HTTPS[6][7] based on transport layer security (TLS) to give the accessing party secure access to resources provided by the offering party. The requirements in this document are valid both for request/reply and push unless otherwise stated.

REQ_04_01_01	The REST-based web services interface implementation shall use HTTPS as the transport protocol with TLS.

REQ_04_01_02	HTTP shall only be used with version 1.1 or higher compatible versions.

REQ_04_01_03	TLS shall only be used with version 1.2 or higher versions.

REQ_04_01_04	The request/reply REST web service shall be a strict client-server interaction, where the accessing party (client) sends a request and the offering party (server) sends a response.

NOTE Resources can be transferred both in the request and the response.

REQ_04_01_05	The request/reply REST implementation shall be stateless; i.e. the offering party server shall not maintain any accessing party client context or session information.

Due to REQ_04_01_05, each request-response pair is handled independently from one another. Each client request by the accessing party contains all information required by the server of the offering party to successfully respond to the request, including a representation of the client state when necessary.

REQ_04_01_06	A push initiation REST web service shall be a strict client-server interaction, where the accessing party (client) sends a request and the offering party (server) sends a response.

REQ_04_01_07	The push REST web service shall be a strict client-server interaction, where the offering party (client) sends a request and the accessing party (server) sends a response.

4.2 Resources

REQ_04_02_01	Information on the server shall be exposed as resources expressed as plural nouns.

NOTE 1 This holds true even when the resource is only available one time on a connected vehicle (e.g. "odometers").

REQ_04_02_02	The exposed resources shall be uniquely identified in the form of Uniform Resource Identifiers (URIs).

The resources, the resource groups or the containers, and how to apply those on a specific presentation or application layer of the accessing party are described in ISO 20078-1. How an accessing party authenticates and how it is authorized for access to resources is described in ISO 20078-3.

REQ_04_02_03	The offering party shall define the base URIs of the web services offered by the offering party.

Table 1 — Examples of possible ExVe-based URIs

Resource	Description
https://{example.com}/exve/	URI based on sub directory
https://exve.{example.com}/	URI based on sub domain

REQ_04_02_04	The offering party shall comply with the URI resource paths defined by specific ExVe standard applications.

Table 2 — Examples of possible ExVe resource URIs

Resource	Description
{base_URI}/{resourcePath}	Resources based on path
{base URI}/vehicles	A list of the available vehicles as defined by authorization context
{base URI}/vehicles/{vehicleId}/dtcReadouts/	Read all Diagnostic Trouble Codes for a specific vehicle
{base URI}/vehicles/{vehicleId}/ecus/{ecuId}/dtcReadouts/	Read all Diagnostic Trouble Codes for a specific ECU of a specific vehicle

There are two primary elements defining an URI: entities and resources (Table 2). Entities are the fundamental objects representing, e.g. vehicles, ECUs, drivers and fleets. Resources are the actual data, aggregated information or functionalities associated with an entity and a specific use case.

REQ_04_02_05	Resources shall be named and described.

EXAMPLE Fuel level can be an example of a single data item resource, vehicle position can be an aggregated resource consisting of several data items (e.g. latitude, longitude, sample time) and lock and unlock the vehicle can be a functionality resource.

REQ_04_02_06	The offering party should have the possibility to extend resources, but shall not be able to reduce resources.

Thus, by REQ_04_02_06 it is not possible to remove data items from a resource, other than through an update of the underlying use case specification. It is however possible to add data items to a resource (i.e. versioning).

REQ_04_02_07	Aggregated resources shall only include or cross-reference necessary data items for the complete and correct operation of the related use case.

NOTE 2 REQ_04_02_07 ensures an accessing party only receives data items necessary for fulfilling the intended need and nothing else when accessing the resource (i.e. data economy).

See also ISO 20078-1:2021, 8.3, for further details.

REQ_04_02_08	A resource within an already existing use case should not be redefined.

An application may extend the recommended patterns below if none of them meets the needs of the use case.

REQ_04_02_09	Each use case shall define how the HTTP operations GET, POST, PUT, PATCH, and DELETE are supported for each defined resource and what the response for each operation is.

REQ_04_02_10	All URI elements shall be written in lower camel case notation.

NOTE 3 The {baseURI} in following patterns refers to the offering party root URI (see Table 3).

Table 3 — Examples of base URI expresses REQ_04_02_09 and REQ_04_02_01

Normative generic format	{baseURI}/{entities}
Example	{baseURI}/vehicles Request a list of all vehicles available to the accessing party.

REQ_04_02_11	Relations of entities shall be expressed using subresources.

NOTE 4 See Table 4.

Table 4 — Examples of sub-URI's expresses REQ_04_02_11 and REQ_04_02_01

Normative generic format	{baseURI}/{entities}/{ID}/{entities2}/{ID2}
Example	{baseURI}/fleets/12/vehicles/456 Request information on vehicle with id 456 of fleet with id 12.
	{baseURI}/vehicles/456/ecus/789 Request information on ECU with id 789 of vehicle with id 456.

REQ_04_02_12	A resource shall be placed after the entity to which it belongs in the URI.

NOTE 5 See Table 5.

Table 5 — Examples of descriptive URI's expresses REQ_04_02_12 and REQ_04_02_01

Normative generic format	{baseURI}/{entities}/{ID}/{resource}
Example	{baseURI}/vehicles/123/positions Request all positions for vehicle with id 123.
	{baseURI}/vehicles/456/odometers Request the odometer value for vehicle with id 456.
	{baseURI}/vehicles/456/tirePressures Request tire pressures of all wheels on the vehicle with id 456.

REQ_04_02_13	If filtering of the response is needed, query parameters shall be used.

NOTE 6 Several query parameters can be added to a request.

NOTE 7 See Table 6.

Table 6 — Examples of filtering responses expresses REQ_04_02_13 and REQ_04_02_01

Normative generic format	{baseURI}/{entities}?{filter}={filterValue}
	{baseURI}/{entities}?{filter}={filterValue}&{filter2}={filterValue2}
Example	{baseURI}/vehicles?ignitionState=on Request all vehicles with ignition on.
	{baseURI}/vehicles/123/positions?startDate=...&endDate=... Request the positions for vehicle with id 123 registered in given date span in the ISO 8601 series date-time format.

REQ_04_02_14	If sorting is needed, query parameters shall be used.

NOTE 8 See Table 7.

Table 7 — Examples of query parameters expresses REQ_04_02_14 and REQ_04_02_01

Normative generic format	{baseURI}/{entities}?{sorting}={sortingValue}
	{baseURI}/{entities}?{sorting}={sortingValue}&{sorting2}={sortingValue2}
Example	{baseURI}/vehicles?sortField=id&sortOrder=asc

REQ_04_02_15	If selection of subsets of resources is needed, query parameters shall be used.

NOTE 9 See Table 8.

Table 8 — Examples of query parameters for subsets expresses REQ_04_02_15 and REQ_04_02_01

Normative generic format	{baseURI}/{entities}?{id}={ID}
	{baseURI}/{entities}?{id}={ID}&{id2}={ID2}
Example	{baseURI}/vehicles?id=123&id=124
	{baseURI}/vehicles?id=YS2RX20001754836

Identifiers may come in multiple formats including, but not limited to, VIN or pseudonymized IDs.

REQ_04_02_16	Pseudonymized IDs may be simple numerical IDs, UUIDs or any other alphanumerical scheme.

NOTE 10 See Table 9.

Table 9 — Examples of pseudonymized IDs represented by numerical IDs expresses REQ_04_02_16 and REQ_04_02_01

Normative generic format	{baseURI}/{entities}?{id}={ID}
	{baseURI}/{entities}?{id}={ID}&{id2}={ID2}
Example	{baseURI}/vehicles?id=ce5d5e3d-28bc-475f-8ef7-b5cb9c8039d4&id=f95ce756-42fc-48b2-8873-86553f6df5cc
	{baseURI}/vehicles?id=456

REQ_04_02_17	For large resource responses pagination may be used.

NOTE 11 See Table 10.

Table 10 — Examples of pagination expresses REQ_04_02_17 and REQ_04_02_01

Normative generic format	{baseURI}/{entities}?start={value}&limit={count}
Example	{baseURI}/vehicles?start=20&limit=10

REQ_04_02_18	A GET request on the returned location may return the total amount of results. If used, it shall be part of the message body using the keyword “exveTotal”.

EXAMPLE The following example expresses REQ_04_02_18:

```
{
 "results": {
 "exveTotal": "150",
... }
}
```

REQ_04_02_19	If wildcards are used, a wildcard (*) shall access all subentities or resources of all parent entities.

NOTE 12 See Table 11.

Table 11 — Examples of wildcards in URIs expresses REQ_04_02_19 and REQ_04_02_01

Normative generic format	{baseURI}/{entities}/*/{entities2}
Example	{baseURI}/vehicles/*/positions Request all positions for all vehicles.
	{baseURI}/vehicles/*/ecus Request all ECU’s of all vehicles.

REQ_04_02_20	If wildcards are used, a wildcard (*) may be combined with other filtering, including IDs, to access a smaller number of entities.

NOTE 13 See Table 12.

Table 12 — Examples of wildcards in URIs while filtering by IDs expresses REQ_04_02_20 and REQ_04_02_01

Normative generic format	{baseURI}/{entities}/*/{resource}?{id}={ID}&{id2}={ID2}&{filter}={filterValue}&{filter2}={filterValue2}
Example	{baseURI}/vehicles/*/odometers?id=456&id=789& startDate=...&endDate=... Request all odometer values from vehicles with id 456 and 789 registered within the given time span in the ISO 8601 series date-time format.

REQ_04_02_21	For fully anonymized access, no entity IDs shall be used in the URIs.

NOTE 14 See Table 13.

Table 13 — Examples of an anonymized access expresses REQ_04_02_21 and REQ_04_02_01

Normative generic format	{baseURI}/{entities}/{entities2}
Example	{baseURI}/vehicles/hazardWarnings?isActive=true

REQ_04_02_22	A push resource shall be suffixed with subscriptions.

NOTE 15 See Table 14.

Table 14 — Examples of push resources regarding REQ_04_02_22

Normative generic format	{baseURI}/{resource}Subscriptions
Example	POST {baseURI}/positionSubscriptions?vehicleId=123 Subscribe to positions for vehicle with id 123.
	GET {baseURI}/positionSubscriptions Request a list of all position subscriptions.
	POST {baseURI}/vehicles/456/tirePressureSubscriptions?pressure=low Subscribe to low tire pressures of all wheels on the vehicle with id 456.

REQ_04_02_23	The accessing party shall define the base URIs of the web services offered by the accessing party. (Compare with REQ_04_02_03).

4.3 Subscription profiles

Push of subscribed resources requires URI locations and authorization information from the accessing party. This information is stored in subscription profiles. Subscription profiles can be created in a separate step before creating subscriptions or directly when creating a subscription. In both cases the subscription profile will be assigned an id and can be reused by any number of subscriptions.