



Evaluation and validation of connected
mobility in real open systems beyond
5GS

Quality on Demand API

HORIZON JU Innovation Actions | 101139048 |
ENVELOPE - HORIZON-JU-SNS-2023



Co-funded by
the European Union

6GSNS

Table of contents

TABLE OF CONTENTS	I
1 INTRODUCTION	2
2 ENVELOPE API OVERVIEW	2
3 SEQUENCE DIAGRAMS	3
4 ENVELOPE API DEFINITION	5
4.1 <i>Resource structure</i>	5
4.1.1 Resource URIs	5
4.2 <i>Data model</i>	6
4.2.1 CreateSession data type	6
4.2.2 SessionInfo data type	7
4.2.3 ExtendSessionDuration data type	9
4.2.4 RetrieveSessionsInput data type	9
4.3 <i>API operations</i>	9
4.3.1 Resource: QoD sessions	9
4.3.2 Resource: Individual QoD sessions	10
4.3.3 Resource: Extension of a QoD session	12
4.3.4 Resource: Retrieve all the sessions	13

1 Introduction

The Quality-On-Demand (QoD) API provides a programmable interface for developers and API consumers to request a certain quality of service (like prioritized throughput or stable latency) for application data flows. The API abstracts the complexity of underlying technologies, such as 4G and 5G mobile network, leveraging the developers of having in-depth knowledge of network configurations.

QoD APIs are particularly relevant in ENVELOPE, where the reference vertical is the Automotive sector, to offer to application functions the possibility to achieve low latency in vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communications. This is crucial for applications like autonomous driving and collision avoidance, where even millisecond-level delays can impact safety.

2 API Overview

API name	Description	Modules Involved	Use Cases
Name of the API	Short summary of the API	List of 5GS or other modules interacting with this API	List of Use Cases using the API
Quality on Demand	The QoD API lets developers request network performance levels like low latency or high throughput, without needing deep telecom knowledge. It's key for Automotive use cases in ENVELOPE, enabling reliable V2V and V2I communications for safety-critical applications.	Network APIs	Italian and Dutch site UCs

3 Sequence Diagrams

3.1.1 Creation of a QoS session

Initiates a new Quality of Service session. An application requests specific network quality, and the API confirms session establishment with relevant details.

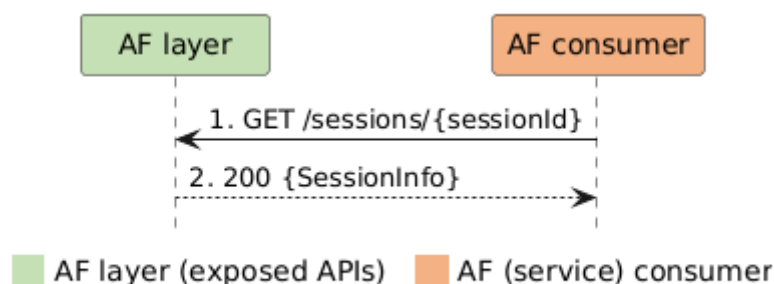


AF Consumer requests the creation of a new prioritized data flow (QoS session) to the AF Layer.

AF Layer confirms the establishment of the prioritized data flow with session details, enabling enhanced communication.

3.1.2 Get Individual QoS session

Retrieves details of a specific, existing QoS session. Allows applications to check the status and parameters of an active session.

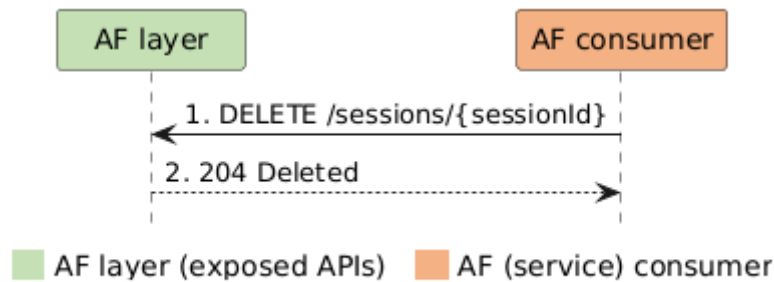


AF Consumer queries the AF Layer to retrieve details of a specific, active prioritized data flow.

AF Layer provides current information about the requested prioritized data flow's parameters and status for monitoring.

3.1.3 Delete Individual QoS session

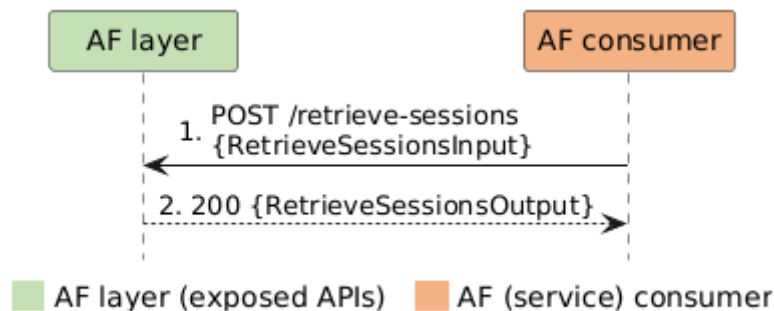
Terminates an existing QoS session. This releases reserved network resources when the specific quality is no longer needed.



AF Consumer sends a request to the AF Layer to terminate an existing prioritized data flow. AF Layer confirms the release of network resources previously allocated for that specific prioritized data flow.

3.1.4 Retrieve all QoS sessions

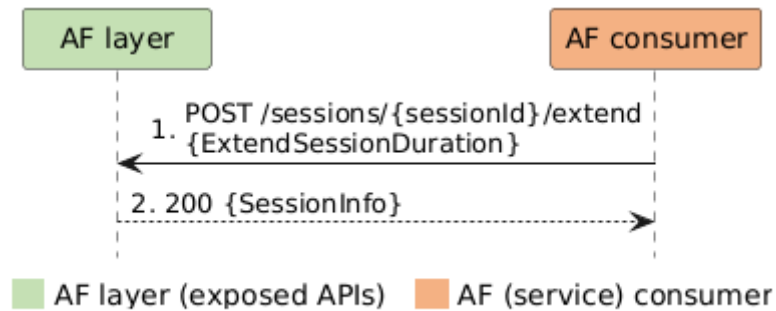
Fetches information on all active QoS sessions for a given consumer. Provides an overview for managing multiple active sessions.



AF Consumer requests from the AF Layer a list of all currently active prioritized data flows for its services. AF Layer provides an overview of all established prioritized data flows to the AF Consumer.

3.1.5 Extension of a QoS session

Extends the duration of an active QoS session. Allows applications to prolong guaranteed quality of service without creating a new session.



AF Consumer sends a request to the AF Layer to extend the duration of an existing prioritized data flow.

AF Layer confirms the extended validity of the prioritized data flow, ensuring continuous low-latency communication.

4 API definition

4.1 Resource structure

4.1.1 Resource URIs

The root of the resource URIs of this API shall be:

{apiRoot}/{apiName}/{apiVersion}

where apiName is set to quality-on-demand and apiVersion is set to v1

The resource URI structure of this API is introduced in Figure 1.

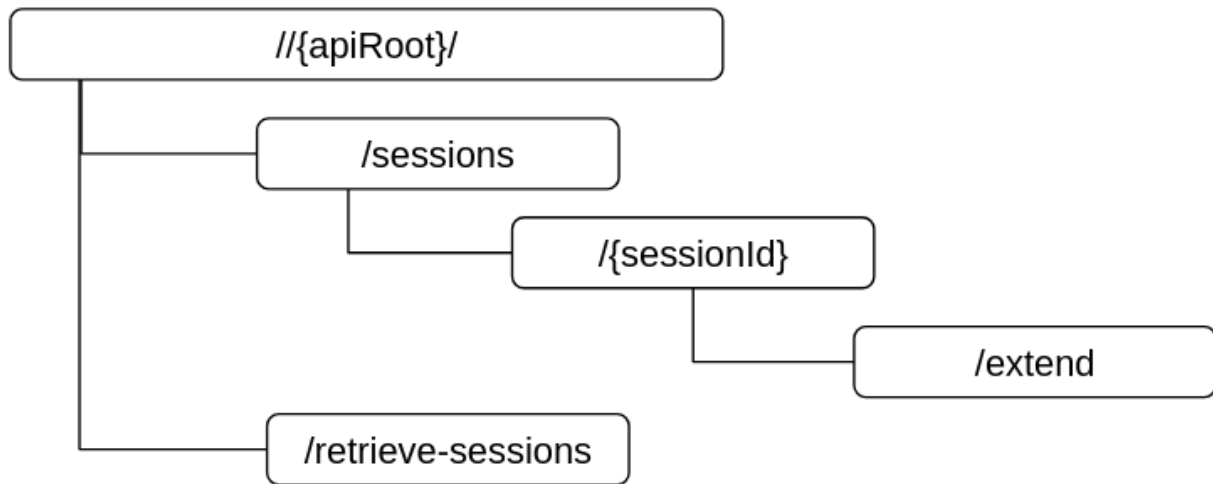


Figure 1 Resource URI structure for the QoS CAMARA API.

Table 1 introduces an overview of the resources and of the methods.

Table 1. Resources and methods overview.

Resource name	Resource URI	HTTP method	Description
Creation of a QoS session	/sessions	POST	Create a new QoS session
Individual QoS sessions	/sessions/{sessionId}	GET	Get QoS session information
		DELETE	Delete a QoS session
Extension of a QoS session	/sessions/{sessionId}/extend	POST	Extend the duration of an active session
QoS information per device	/retrieve-sessions	POST	Get all QoS profiles

4.2 Data model

4.2.1 CreateSession data type

Table 2. CreateSession data type

Field	Data Type	Cardinality	Description
device	Structure (inlined)	0..1	End-user equipment.
→ phoneNumber	string	1	A public identifier.
→ networkAccessIdentifier	string	1	A public identifier.

→ ipv4Address	Structure (inlined)	1	The device should be a single IPv4 address.
→ → publicAddress	ipv4	1	A single IPv4 address.
→ → privateAddress	ipv4	1	A single IPv4 address.
→ → publicPort	integer	1	TCP or UDP port.
→ ipv6Address	ipv6	1	The device should be an IPv6 address.
applicationServer	Structure (inlined)	1	A server hosting service.
→ ipv4Address	string	1	IPv4 address may be provided.
→ ipv6Address	string	1	IPv6 address may be provided.
devicePorts	Structure (inlined)	0..1	The ports used locally.
→ ranges	array of object	1	TCP or UDP port ranges.
→ → from	integer	1	Start of the port range.
→ → to	integer	1	End of the port range.
→ ports	array of integers	1	A list of single ports or ranges.
applicationServerPorts	Structure (inlined)	1	Ports used by the application server.
→ ranges	array of object	1..N	TCP or UDP port ranges.
→ → from	integer	1	Start of the port range.
→ → to	integer	1	End of the port range.
→ ports	array of integers	1..N	A list of single ports or ranges.
qosProfile	string	1	A unique name for QoS.
sink	string (uri)	0..1	The address to which events about all status changes of the session
sinkCredential	Structure (inlined)	0..1	A sink credential provides authentication or authorization information necessary to enable delivery of events to a target.
→ credentialType	String (enum)	1	The type of the credential. Allowed values are
duration	Integer	1	Requested session duration in seconds

4.2.2 SessionInfo data type

Table 3. SessionInfo data type

Field	Data Type	Cardinality	Description
device	Structure (inlined)	0..1	End-user equipment.
→ phoneNumber	string	1	A public identifier.
→ networkAccessIdentifier	string	1	A public identifier.

→ ipv4Address	Structure (inlined)	1	The device should be a single IPv4 address.
→ → publicAddress	ipv4	1	A single IPv4 address.
→ → privateAddress	ipv4	1	A single IPv4 address.
→ → publicPort	integer	1	TCP or UDP port.
→ ipv6Address	ipv6	1	The device should be an IPv6 address.
applicationServer	Structure (inlined)	1	A server hosting service.
→ ipv4Address	string	1	IPv4 address may be provided.
→ ipv6Address	string	1	IPv6 address may be provided.
devicePorts	Structure (inlined)	0..1	The ports used locally.
→ ranges	array of object	1	TCP or UDP port ranges.
→ → from	integer	1	Start of the port range.
→ → to	integer	1	End of the port range.
→ ports	array of integers	1	A list of single ports or ranges.
applicationServerPorts	Structure (inlined)	1	Ports used by the application server.
→ ranges	array of object	1..N	TCP or UDP port ranges.
→ → from	integer	1	Start of the port range.
→ → to	integer	1	End of the port range.
→ ports	array of integers	1..N	A list of single ports or ranges.
qosProfile	string	1	A unique name for QoS.
sink	string (uri)	0..1	The address to which events about all status changes of the session
sinkCredential	Structure (inlined)	0..1	A sink credential provides authentication or authorization information necessary to enable delivery of events to a target.
→ credentialType	String (enum)	1	The type of the credential. Allowed values are
duration	Integer	1	Requested session duration in seconds
sessionId	String	1	Session ID in UUID format
startedAt	String	0..1	Date and time when the QoS status became "AVAILABLE"
expiresAt	String	0..1	
qosStatus	String (enum)	1	Date and time of the QoS session expiration
statusInfo	String (enum)	0..1	Reason for the new qosStatus. Allowed values are [DURATION_EXPIRED, NETWORK_TERMINATED, DELETE_REQUEST]

4.2.3 ExtendSessionDuration data type

Table 4. ExtendSessionDuration data type

Field	Data Type	Cardinality	Description
requestedAdditionalDuration	Integer	1	Additional duration in seconds to be added to the current session duration

4.2.4 RetrieveSessionsInput data type

Table 5. RetrieveSessionsInput data type

Field	Data Type	Cardinality	Description
device	Structure (inlined)	1	End-user equipment.
→ phoneNumber	string	1	A public identifier.
→ networkAccessIdentifier	string	1	A public identifier.
→ ipv4Address	Structure (inlined)	1	The device should be a single IPv4 address.
→ → publicAddress	ipv4	1	A single IPv4 address.
→ → privateAddress	ipv4	1	A single IPv4 address.
→ → publicPort	integer	1	TCP or UDP port.
→ ipv6Address	ipv6	1	The device should be an IPv6 address.

4.3 API operations

4.3.1 Resource: QoS sessions

The resource URI is: {apiRoot}/{apiName}/{apiVersion}/sessions

The content of the methods requests and response is JSON and it is indicated by setting the content type "application/json".

4.3.1.1 Resource methods

4.3.1.1.1 POST

Create QoS Session to manage latency/throughput priorities

If the qosStatus in the API response is "AVAILABLE" and a notification callback is provided the API consumer will receive in addition to the response a QOS_STATUS_CHANGED event notification with qosStatus as AVAILABLE.

If the qosStatus in the API response is REQUESTED, the client will receive either

- a QOS_STATUS_CHANGED event notification with qosStatus as AVAILABLE after the network notifies that it has created the requested session, or
- a QOS_STATUS_CHANGED event notification with qosStatus as UNAVAILABLE and statusInfo as NETWORK_TERMINATED after the network notifies that it has failed to provide the requested session.

A QOS_STATUS_CHANGED event notification with qosStatus as UNAVAILABLE will also be send if the network terminates the session before the requested duration expired

Parameters: no parameters

Table 6. Data structures supported by the POST request on this resource

Request body	Data type	Cardinality	Notes
	CreateSession	1	Parameters to create a new session

Table 7. Data structures supported by the POST response on this resource

Response body	Data type	Cardinality	Response Codes	Notes
	SessionInfo	1	201	Session created
	ErrorResponse	0..1	400	Invalid input for createSession operation
	ErrorResponse	0..1	401	Unauthorized
	ErrorResponse	0..1	403	Forbidden
	ErrorResponse	0..1	409	Conflict
	ErrorResponse	0..1	500	Internal server error
	ErrorResponse	0..1	501	Not Implemented
	ErrorResponse	0..1	503	Service unavailable

4.3.2 Resource: Individual QoS sessions

The resource URI is: {apiRoot}/{apiName}/{apiVersion}/sessions/{sessionId}

4.3.2.1 Resource methods

4.3.2.1.1 GET

Querying for QoS session resource information details

Parameters support by this method are listed in Table 8.

Table 8. URI query parameters supported by the GET method on this resource

Name	Data type	Cardinality	Notes
sessionId	string	1	Session ID that was obtained from the createSession operation

Table 9. Data structures supported by the GET response on this resource

	Data type	Cardinality	Response Codes	Notes
Response body	SessionInfo	1	201	Session retrieved
	ErrorResponse	0..1	400	Invalid input for DeleteSession operation
	ErrorResponse	0..1	401	Unauthorized
	ErrorResponse	0..1	403	Forbidden
	ErrorResponse	0..1	409	Conflict
	ErrorResponse	0..1	500	Internal server error
	ErrorResponse	0..1	501	Not Implemented
	ErrorResponse	0..1	503	Service unavailable

4.3.2.1.2 DELETE

Release resources related to QoS session

If the notification callback is provided and the qosStatus of the session was AVAILABLE the client will receive in addition to the response a QOS_STATUS_CHANGED event with

qosStatus as UNAVAILABLE and
statusInfo as DELETE_REQUESTED There will be no notification event if the qosStatus was already UNAVAILABLE.

Parameters support by this method are listed in Table 8.

Table 10. URI query parameters supported by the GET method on this resource

Name	Data type	Cardinality	Notes
sessionId	string	1	Session ID that was obtained from the createSession operation

Table 11. Data structures supported by the GET response on this resource

Response body	Data type	Cardinality	Response Codes	Notes
	<empty>	1	204	Session deleted
	ErrorResponse	0..1	400	Invalid input for createSession operation
	ErrorResponse	0..1	401	Unauthorized
	ErrorResponse	0..1	403	Forbidden
	ErrorResponse	0..1	409	Conflict
	ErrorResponse	0..1	500	Internal server error
	ErrorResponse	0..1	501	Not Implemented
	ErrorResponse	0..1	503	Service unavailable

4.3.3 Resource: Extension of a QoS session

The resource URI is: {apiRoot}/{apiName}/{apiVersion}/sessions/{sessionId}

4.3.3.1 Resource methods

4.3.3.1.1 POST

Extend the overall session duration of an active QoS session (with qosStatus = AVAILABLE). The overall duration of the QoS session, including the additional extended duration, shall not exceed the maximum duration limit fixed for the QoS Profile. If the current duration plus the value of requestedAdditionalDuration exceeds the maximum limit, the new overall duration shall be capped to the maximum value allowed. An example: For a QoS profile limited to a maxDuration of 50,000 seconds, a QoS session was originally created with duration 30,000 seconds. Before the session expires, the developer requests to extend the session by another 30,000 seconds:

Previous duration: 30,000 seconds

Requested additional duration: 30,000 seconds

New overall session duration: 50,000 seconds (the maximum allowed)

Table 12. URI query parameters supported by the GET method on this resource

Name	Data type	Cardinality	Notes
sessionId	string	1	Session ID that was obtained from the createSession operation

Table 13. Data structures supported by the POST request on this resource

Request body	Data type	Cardinality	Notes
	ExtendSessionDuration	1	Attributes required to extend the duration of an active session

Table 14. Data structures supported by the POST response on this resource

Response body	Data type	Cardinality	Response Codes	Notes
	SessionInfo	1	201	Session extended
	ErrorResponse	0..1	400	Invalid input for ExtendSession operation
	ErrorResponse	0..1	401	Unauthorized
	ErrorResponse	0..1	403	Forbidden
	ErrorResponse	0..1	409	Conflict
	ErrorResponse	0..1	500	Internal server error
	ErrorResponse	0..1	501	Not Implemented
	ErrorResponse	0..1	503	Service unavailable

4.3.4 Resource: Retrieve all the sessions

The resource URI is: {apiRoot}/{apiName}/{apiVersion}/retrieve-sessions

4.3.4.1 Resource methods

4.3.4.1.1 POST

Querying for QoS session resource information details for a device

Parameters: none

Table 15. Data structures supported by the POST request on this resource

Request body	Data type	Cardinality	Notes
	RetrieveSessionsInput	1	Parameters to get QoS session information by device

Table 16. Data structures supported by the POST response on this resource

Response body	Data type	Cardinality	Response Codes	Notes
	RetrieveSessionsOutput	1	200	QoS session information for a given device
	ErrorResponse	0..1	400	Invalid input for Retrieve Sessions

	ErrorResponse	0..1	401	Unauthorized
	ErrorResponse	0..1	403	Forbidden
	ErrorResponse	0..1	409	Conflict
	ErrorResponse	0..1	500	Internal server error
	ErrorResponse	0..1	501	Not Implemented
	ErrorResponse	0..1	503	Service unavailable