

## API Reference

# ETSI GS QKD 004 push mode specification, QKD server

API Version: 1.0.1

Server description of the QKD endpoint for processing ETSI GS QKD 004 push mode requests

### CONTACT

**URL:** <https://qkd-kms.ait.ac.at>

# INDEX

<b>1. ETSI004</b>	<b>3</b>
1.1 POST /open-connect	3
1.2 GET /get-key	5
1.3 PUT /close	7

# API

## 1. ETSI004

### 1.1 POST /open-connect

#### open key stream

The open-connect message connects a KMS to the QKD Device. As per sequence definition, the QKD device will assign the key\_stream\_id at the source node, therefore null has to be given as key\_stream\_id input parameter. At the destination, the key\_stream\_id has to be specified to the same value given at the source. The QKD can suggest alternative QoS due to its capabilities.

### REQUEST

#### REQUEST BODY - application/json

NAME	TYPE	DESCRIPTION
source*	string	the source URI of the request.
destination*	string	the destination URI of the request.
key_stream_id*	string	ID of the key stream. If `null` specified it is up to the QKD device to define the UUID. At the destination it should match the source `key_stream_id`.
<b>qos*</b>	object	
key_chunk_size	integer	length of key chunks in Byte.
max_bps	integer	in bit per second.
min_bps	integer	in bit per second.
jitter	integer	in bit per second.
priority	integer	Meaning up to implementation.
timeout	integer	Timeout in msec after which the call will be aborted, returning an error.
ttl	integer	Time in seconds after which the keys for this key stream ID shall be erased from the kms key store.
metadata_mimetype	string	MIME type of the metadata of the `get` request.

### RESPONSE

**STATUS CODE - 201:** Key stream successfully created.

#### RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
<b>OBJECT WITH BELOW STRUCTURE</b>		
key_stream_id	string	key stream ID. If not specified it is up to the QKD device to define the UUID.
<b>qos*</b>	object	
key_chunk_size	integer	length of key chunks in Byte.
max_bps	integer	in bit per second.
min_bps	integer	in bit per second.
jitter	integer	in bit per second.
priority	integer	Meaning up to implementation.

NAME	TYPE	DESCRIPTION
timeout	integer	Timeout in msec after which the call will be aborted, returning an error.
ttl	integer	Time in seconds after which the keys for this key stream ID shall be erased from the kms key store.
metadata_mimetype	string	MIME type of the metadata of the `get` request.
status	enum	<b>ALLOWED:0, 1, 2, 3, 4, 5, 6, 7, 8</b> Status code:  0 = Successful; HTTP code 200.  1 = Successful connection, but peer not connected; HTTP code 200.  2 = GET_KEY failed because insufficient key available; HTTP code 500.  3 = GET_KEY failed because peer application is not yet connected; HTTP code 400.  4 = No QKD connection available; HTTP code 500.  5 = OPEN_CONNECT failed because the KSID is already in use; HTTP code 400.  6 = TIMEOUT_ERROR: The call failed because the specified TIMEOUT; HTTP code 500.  7 = OPEN failed because requested QoS settings could not be met, counter proposal included in return has occurred; HTTP code 500.  8 = GET_KEY failed because metadata field size insufficient. Returned Metadata_size value holds minimum needed size of metadata; HTTP code 400.

**STATUS CODE - 400:** Client sent a malformed input.

#### RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
<b>OBJECT WITH BELOW STRUCTURE</b>		
status*	enum	<b>ALLOWED:0, 1, 2, 3, 4, 5, 6, 7, 8</b> Status code:  0 = Successful; HTTP code 200.  1 = Successful connection, but peer not connected; HTTP code 200.  2 = GET_KEY failed because insufficient key available; HTTP code 500.  3 = GET_KEY failed because peer application is not yet connected; HTTP code 400.  4 = No QKD connection available; HTTP code 500.  5 = OPEN_CONNECT failed because the KSID is already in use; HTTP code 400.  6 = TIMEOUT_ERROR: The call failed because the specified TIMEOUT; HTTP code 500.  7 = OPEN failed because requested QoS settings could not be met, counter proposal included in return has occurred; HTTP code 500.  8 = GET_KEY failed because metadata field size insufficient. Returned Metadata_size value holds minimum needed size of metadata; HTTP code 400.

**STATUS CODE - 500:** The server experienced an internal error.

#### RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
<b>OBJECT WITH BELOW STRUCTURE</b>		

NAME	TYPE	DESCRIPTION
status*	enum	<b>ALLOWED:0, 1, 2, 3, 4, 5, 6, 7, 8</b> Status code: 0 = Successful; HTTP code 200. 1 = Successful connection, but peer not connected; HTTP code 200. 2 = GET_KEY failed because insufficient key available; HTTP code 500. 3 = GET_KEY failed because peer application is not yet connected; HTTP code 400. 4 = No QKD connection available; HTTP code 500. 5 = OPEN_CONNECT failed because the KSID is already in use; HTTP code 400. 6 = TIMEOUT_ERROR: The call failed because the specified TIMEOUT; HTTP code 500. 7 = OPEN failed because requested QoS settings could not be met, counter proposal included in return has occurred; HTTP code 500. 8 = GET_KEY failed because metadata field size insufficient. Returned Metadata_size value holds minimum needed size of metadata; HTTP code 400.

## 1.2 GET /get-key

Start the pushed key retrieval process.

The get-key message informs the QKD device that it can start pushing keys.

### REQUEST

#### REQUEST BODY - application/json

NAME	TYPE	DESCRIPTION
key_stream_id*	string	corresponding to the one given in `open-connect`.
index	integer	increasing index of the key chunk in the key stream. Wrapping at UINT_MAX.
<b>metadata</b>	object	
size	integer	size of buffer in characters.
buffer	string	metadata in json format.

### RESPONSE

**STATUS CODE - 205:** Successful registration.

#### RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
<b>OBJECT WITH BELOW STRUCTURE</b>		

NAME	TYPE	DESCRIPTION
status*	enum	<b>ALLOWED:</b> 0, 1, 2, 3, 4, 5, 6, 7, 8 Status code:  0 = Successful; HTTP code 200.  1 = Successful connection, but peer not connected; HTTP code 200.  2 = GET_KEY failed because insufficient key available; HTTP code 500.  3 = GET_KEY failed because peer application is not yet connected; HTTP code 400.  4 = No QKD connection available; HTTP code 500.  5 = OPEN_CONNECT failed because the KSID is already in use; HTTP code 400.  6 = TIMEOUT_ERROR: The call failed because the specified TIMEOUT; HTTP code 500.  7 = OPEN failed because requested QoS settings could not be met, counter proposal included in return has occurred; HTTP code 500.  8 = GET_KEY failed because metadata field size insufficient. Returned Metadata_size value holds minimum needed size of metadata; HTTP code 400.
<b>metadata</b>	object	
size	integer	size of buffer in characters.
buffer	string	metadata in json format.

**STATUS CODE - 400:** Client sent a malformed input.

#### RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
<b>OBJECT WITH BELOW STRUCTURE</b>		
status*	enum	<b>ALLOWED:</b> 0, 1, 2, 3, 4, 5, 6, 7, 8 Status code:  0 = Successful; HTTP code 200.  1 = Successful connection, but peer not connected; HTTP code 200.  2 = GET_KEY failed because insufficient key available; HTTP code 500.  3 = GET_KEY failed because peer application is not yet connected; HTTP code 400.  4 = No QKD connection available; HTTP code 500.  5 = OPEN_CONNECT failed because the KSID is already in use; HTTP code 400.  6 = TIMEOUT_ERROR: The call failed because the specified TIMEOUT; HTTP code 500.  7 = OPEN failed because requested QoS settings could not be met, counter proposal included in return has occurred; HTTP code 500.  8 = GET_KEY failed because metadata field size insufficient. Returned Metadata_size value holds minimum needed size of metadata; HTTP code 400.
<b>metadata</b>	object	
size	integer	size of buffer in characters.
buffer	string	metadata in json format.

**STATUS CODE - 500:** The server experienced an internal error.

#### RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
<b>OBJECT WITH BELOW STRUCTURE</b>		

NAME	TYPE	DESCRIPTION
status*	enum	<b>ALLOWED:0, 1, 2, 3, 4, 5, 6, 7, 8</b> Status code:  0 = Successful; HTTP code 200.  1 = Successful connection, but peer not connected; HTTP code 200.  2 = GET_KEY failed because insufficient key available; HTTP code 500.  3 = GET_KEY failed because peer application is not yet connected; HTTP code 400.  4 = No QKD connection available; HTTP code 500.  5 = OPEN_CONNECT failed because the KSID is already in use; HTTP code 400.  6 = TIMEOUT_ERROR: The call failed because the specified TIMEOUT; HTTP code 500.  7 = OPEN failed because requested QoS settings could not be met, counter proposal included in return has occurred; HTTP code 500.  8 = GET_KEY failed because metadata field size insufficient. Returned Metadata_size value holds minimum needed size of metadata; HTTP code 400.
<b>metadata</b>	object	
size	integer	size of buffer in characters.
buffer	string	metadata in json format.

## 1.3 PUT /close

### close the key stream

The close message disconnects the KMS from the QKD device.

## REQUEST

### REQUEST BODY - application/json

NAME	TYPE	DESCRIPTION
key_stream_id*	string	corresponding to the one given in `open-connect`.

## RESPONSE

**STATUS CODE - 204:** Key stream successfully closed.

### RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
<b>OBJECT WITH BELOW STRUCTURE</b>		

---

NAME	TYPE	DESCRIPTION
------	------	-------------

---

status*	enum	<b>ALLOWED:</b> 0, 1, 2, 3, 4, 5, 6, 7, 8 Status code:  0 = Successful; HTTP code 200.  1 = Successful connection, but peer not connected; HTTP code 200.  2 = GET_KEY failed because insufficient key available; HTTP code 500.  3 = GET_KEY failed because peer application is not yet connected; HTTP code 400.  4 = No QKD connection available; HTTP code 500.  5 = OPEN_CONNECT failed because the KSID is already in use; HTTP code 400.  6 = TIMEOUT_ERROR: The call failed because the specified TIMEOUT; HTTP code 500.  7 = OPEN failed because requested QoS settings could not be met, counter proposal included in return has occurred; HTTP code 500.  8 = GET_KEY failed because metadata field size insufficient. Returned Metadata_size value holds minimum needed size of metadata; HTTP code 400.
---------	------	--

---

**STATUS CODE - 400:** Client sent a malformed input.

**RESPONSE MODEL - application/json**

---

NAME	TYPE	DESCRIPTION
------	------	-------------

---

**OBJECT WITH BELOW STRUCTURE**

status*	enum	<b>ALLOWED:</b> 0, 1, 2, 3, 4, 5, 6, 7, 8 Status code:  0 = Successful; HTTP code 200.  1 = Successful connection, but peer not connected; HTTP code 200.  2 = GET_KEY failed because insufficient key available; HTTP code 500.  3 = GET_KEY failed because peer application is not yet connected; HTTP code 400.  4 = No QKD connection available; HTTP code 500.  5 = OPEN_CONNECT failed because the KSID is already in use; HTTP code 400.  6 = TIMEOUT_ERROR: The call failed because the specified TIMEOUT; HTTP code 500.  7 = OPEN failed because requested QoS settings could not be met, counter proposal included in return has occurred; HTTP code 500.  8 = GET_KEY failed because metadata field size insufficient. Returned Metadata_size value holds minimum needed size of metadata; HTTP code 400.
---------	------	--

---

**STATUS CODE - 500:** The server experienced an internal error.

**RESPONSE MODEL - application/json**

---

NAME	TYPE	DESCRIPTION
------	------	-------------

---

**OBJECT WITH BELOW STRUCTURE**

---

<b>NAME</b>	<b>TYPE</b>	<b>DESCRIPTION</b>
-------------	-------------	--------------------

---

status*	enum	<b>ALLOWED:0, 1, 2, 3, 4, 5, 6, 7, 8</b>
---------	------	--

Status code:

0 = Successful; HTTP code 200.

1 = Successful connection, but peer not connected; HTTP code 200.

2 = GET\_KEY failed because insufficient key available; HTTP code 500.

3 = GET\_KEY failed because peer application is not yet connected; HTTP code 400.

4 = No QKD connection available; HTTP code 500.

5 = OPEN\_CONNECT failed because the KSID is already in use; HTTP code 400.

6 = TIMEOUT\_ERROR: The call failed because the specified TIMEOUT; HTTP code 500.

7 = OPEN failed because requested QoS settings could not be met, counter proposal included in return has occurred; HTTP code 500.

8 = GET\_KEY failed because metadata field size insufficient. Returned Metadata\_size value holds minimum needed size of metadata; HTTP code 400.

---