

What is the correct format for the "delivery" section when updating a stream's configuration?

After looking closely at the spec, my question is which is correct: the example in section 7 (https://openid.net/specs/openid-sse-framework-1_0.html#updating-a-streams-configuration) or the definition in section 11

(https://openid.net/specs/openid-sse-framework-1_0.html#delivery-meta)

Specifically, should it look like

```
"delivery": {  
  "delivery_method":  
    "https://schemas.openid.net/secevent/risc/delivery-method/push",  
  "url": "https://receiver.example.com/events"  
}
```

OR

```
"delivery": {  
  "method":  
    "https://schemas.openid.net/secevent/risc/delivery-method/push",  
  "endpoint_url": "https://receiver.example.com/events"  
}
```

Google has interpreted the latter to be correct:

https://developers.google.com/identity/protocols/risc#config_stream

When exactly are Event Streams created or deleted?

There seems to be two options for how Event Stream entities would actually be created / destroyed:

Option 1: Out of Band

Our assumption is that Event Stream creation and deletion will happen out-of-band. That is, the stream *configuration* will happen via calls to the Transmitter endpoints, but the stream itself will have already been created via manual configuration done at the time that an Admin is configuring the Transmitter and Receiver to know about one another.

The main reason for our assumption is that in the [Stream Configuration](#), the “aud” property is **Read-Only**. This implies that it's being set out of band, as the API does not expose a way to set that.

Additionally, the fact that [Updating a Stream Config](#) can result in a 404 indicates that outside of the *configuration*, the existence of a stream entity is being checked.

If our assumption is correct, there are a few other questions:

- Why isn't there a 404 status code for calls to [Delete](#) a stream config?
- Related: Does deleting a stream's config wipe out the subjects / queued SETs, or just reset the configuration to some transmitter-defined default but preserve the rest of the state of the stream?

Option 2: Corresponding to POST and DELETE requests to the "stream" endpoint.

The other option is that the streams themselves will be created (or deleted) just-in-time by calls to the [Update](#) and [Delete](#) config endpoints.

If this is the case, the evidence suggested in option 1 would need to be corrected in the spec. Specifically:

- Why would [Updating](#) a config ever lead to a 404?
- When would the "aud" of the stream config be specified?

Some in-the-weeds questions about HTTP status codes

Why does [Add Subject](#) return a 200, but [Remove Subject](#) return a 204?

Should [Add Subject](#) and [Remove Subject](#) also return 404 if the stream doesn't exist?

Why is there no 404 for [Triggering a Verification Event](#)?

How should we specify the user format for Complex or Aliases subject formats when describing a stream's configuration?

In the Stream Configuration object:

format

Read-Write, The Subject Identifier Format that the Receiver wants for the events. If not set then the Transmitter might decide to use a type that discloses more information than necessary.

Complex Subject does not have a "format" field.

For Aliases, how would we define multiple formats?

There appears to be a typo in the section on updating a stream's status.

The POST body contains a [JSON](#) object with the following fields: 200 OK response containing a [JSON](#) representation of the updated stream status in the body.