

Pull Channels

- Pull channels, on page 1

Pull channels

A pull channel is a data delivery mechanism that

- enables clients to retrieve (pull) data from a source instead of passively receiving it,
- supports multiple modes for event handling (such as HTTP, event replay, and de-duplication), and
- allows configuration for scalability and event retention based on system requirements

HTTP streaming

Your application can use the HTTP pull channel to retrieve events over HTTP/2 or HTTP/1.x. Send a `GET` request to the Firehose HTTP endpoint. Cisco Spaces continuously sends newline-delimited JSON events while the connection remains active.

The OpenAPI specification for this endpoint includes the complete `EventRecord` schema and the supported event payloads. For more information, refer to HTTP Streaming in the Cisco Spaces Firehose documentation.

Code snippet:

```
GET /api/partners/v1/firehose/events
```

Ensure your application meets these integration specifications:

- Your application must support secure (HTTPS) connections to the endpoint.
- Events are encoded as JSON and separated by a newline character. For sample JSON format events, see Sample Events JSON format.
- For on-premise applications, you can request a replica of the Firehose stream for standby or secondary instances by using the replicaId query parameter (integer; default 1).

Note **replicaId** is supported only for on-premise applications.

You can request to replay events from a specific timestamp by using the from Timestamp parameter:

- Provide the timestamp as the number of milliseconds since epoch.
- If you do not specify fromTimestamp, the HTTP Pull endpoint sends only events received after the HTTP connection is established.

Event replay and de-duplication

To manage event de-duplication and ensure data continuity, follow these best practices:

- Use the unique identifier (record UID) present in all events to de-duplicate and avoid processing duplicate events.
- During production deployments, if a restart occurs, use the EventsStreamRequest with the from Timestamp value in combination with de-duplication methods to prevent missing events or data loss.

Event retention window

Scaling with multiple receivers

Use these guidelines to scale your application and optimize throughput for the Firehose API:

- To distribute load across multiple receivers in your application, or to increase throughput for high-latency connections, you can open multiple connections to the Firehose API HTTP endpoint.
- Your application can have up to 12 receivers.
- When you use more than one connection, you must specify the partitions each connection receives by setting the **minPartition** (default one) and **maxPartition** (default 12) parameters in the GET request.