Sending Data to Streams

Module Objectives

In this module, you will look at:

- Sending events to an existing Stream
- Sending data to an existing Stream using the `curl` command

Sending Events to a Stream

An event can be sent to a Stream by an HTTP POST method to the URL of the Stream:

```
POST <base-url>/streams/<stream-id>
```

Parameter

`<stream-id>`

Description

Name of an existing Stream

HTTP Responses

Status Code

`200 OK`

Description

The event was successfully received

Status Code

`404 Not Found`

Description

The Stream does not exist

Note: The response will always have an empty body

Sending Events to a Stream: Example

```
POST <base-url>/streams/mystream
```

Sending Events to a Stream: Passing Headers

- The body of the request must contain the event in binary form
- Pass headers for the event as HTTP headers, prefixing them with the *stream-id*: `<stream-id>.<property>:<string value>`
- After receiving the request, the HTTP handler transforms it into a Stream event:

1. The body of the event is an identical copy of the bytes found in the body of the HTTP post request
2. If the request contains any headers prefixed with the *stream-id*, the *stream-id* prefix is stripped from the header name and the header is added to the event

Sending Events to a Stream: Example using curl

```
curl -X POST -d "$line" <base-url>/streams/mystream
```

where

`$line` is a line from a log file, such as

```
165.225.156.91 - - [09/Jan/2014:21:28:53 -0400] "GET /index.html HTTP/1.1" 200 225
"http://continuuity.com" "Mozilla/4.08 [en] (Win98; I ;Nav)"
```

Module Summary

You should now be able to:

- Send events to an existing Stream
- Sending data to an existing Stream using `curl`

Module Completed

Chapter Index