Event Publishing Specification

This document provides the build specification to clients who wish to receive learning event notifications programmatically.

Changelog

2023-03-15	Added several refX fields to event_specific_detail
2023-01-20	Clarified meaning of UUID in eventContext Corrected typo in learning_path_completed sample payload
2022-12-20	Added LEARNING_PATH_COMPLETED event type
2022-12-13	Corrected payload property names from camel case to snake case
2022-01-12	Added first_name and last_name fields to COURSE_COMPLETED payload in event_specific_detail to aid with the resolution of matching errors

Overview

In general, a client will reach out to Praesidium to identify relevant events and an endpoint to which to publish them in near real-time. Authentication credentials (to be used with HTTP Basic Authentication) may be provided securely to Praesidium as well. Praesidium will then set up the event subscription for the client in question. From the time of setup onward, any relevant events that happen will be communicated in near real-time to the endpoint specified by the client.

Event Payload

The event payload will take the form of a JSON Object as follows:

```
{
  "version": <STRING>,
    "event_type": <STRING>,
  "event_timestamp": <STRING>,
  "event_context": <JSON>,
  "event_specific_detail": <JSON>
```

```
}
```

where
version - the version string (value: "1.0")
event_type - what kind of event is being reported. Event type will determine the structure of event_context (see below)
event_timestamp - a UTC timestamp in ISO format (i.e. YYYY-MM-DD HH24:MM:SS)
event_context - a JSON object with a specific structure depending upon event_type
event_specific_detail - a JSON object with additional contextual information that may differ depending on event_type

Accepted Return Values

In keeping with general RESTful principles, HTTP Status codes should be used. A return value of 400 should be used if the event payload violates the specification above or is otherwise malformed. A return value of 200, 201 or 202 should be returned to indicate that the event record was successfully received.

Event Types

COURSE_COMPLETED

The event context structure will consist of 2 fields inside the JSON object

1. uuid - This field will contain the UUID of the user as returned by user creation via the REST API
2. user – This field will contain the email address needed to identify the user who completed the course
3. course – This field will contain a JSON object with 2 fields:
    1. id: the course sku for the course completed
    2. name: the course name for the course completed

The event_specific_detail field now contains an object user_detail with the following fields:

- first_name
- last_name
- clientExternalId - this field will hold the value from the Academy user record that the client has provided uniquely identifying this user in their system. (Note: this field may be null if the client is not using this feature.
- ref3 - this corresponds to the ref3 element in the Academy user record
- ref4 - this corresponds to the ref4 element in the Academy user record
- ref5 - this corresponds to the ref5 element in the Academy user record
- ref7 - this corresponds to the ref7 element in the Academy user record
- ref8 - this corresponds to the ref8 element in the Academy user record
- ref9 - this corresponds to the ref9 element in the Academy user record

Example Payload


```
{
    "version": "1.0",
    "event_type": "COURSE_COMPLETED",
    "event_timestamp": "2018-03-01 17:45:37",
    "event_context": {
        "uuid": "aaaaaaaa-bbbb-cccc-dddd-ffffffffffff",
        "user": "email@gmail.com",
        "course": {
            "id": "CON20938ES",
            "name": "Duty to Report: Mandated Reporter"
        }
    },
    "event_specific_detail": {
        "user_detail": {
            "first_name": "Tester",
            "last_name": "Testerman",
            "clientExternalId": "1234569",
            "ref3": "arbitrary text",
            "ref4": "arbitrary text2",
            "ref5": "arbitrary text3",
            "ref7": "arbitrary text4",
            "ref8": "arbitrary text5",
            "ref9": "arbitrary text6",
        }
    }
}
```


LEARNING_PATH_COMPLETED

The event context structure will consist of 2 fields inside the JSON object

1. uuid - This field will contain the UUID of the user as returned by user creation via the REST API
2. user – This field will contain the email address needed to identify the user who completed the course
3. learning_path – This field will contain a JSON object with 2 fields:
    1. id: the learning path sku for the learning path completed
    2. name: the learning path name for the learning path completed

The event_specific_detail field now contains an object user_detail with the following fields:

- first_name
- last_name
- clientExternalId - this field will hold the value from the Academy user record that the client has provided uniquely identifying this user in their system. (Note: this field may be null if the client is not using this feature.
- ref3 - this corresponds to the ref3 element in the Academy user record
- ref4 - this corresponds to the ref4 element in the Academy user record
- ref5 - this corresponds to the ref5 element in the Academy user record
- ref7 - this corresponds to the ref7 element in the Academy user record
- ref8 - this corresponds to the ref8 element in the Academy user record
- ref9 - this corresponds to the ref9 element in the Academy user record

Example Payload

```
{
"version": "1.0",
    "event_type": "LEARNING_PATH_COMPLETED",
"event_timestamp": "2018-03-01 17:45:37",
"event_context": {
        "uuid": "aaaaaaaa-bbbb-cccc-dddd-ffffffffffff",
        "user": "email@gmail.com",
        "learning_path": {
            "id": "CONLP10023EN",
            "name": "Duty to Report: Mandated Reporter"
        }
    },
"event_specific_detail": {
        "user_detail": {
            "first_name": "Tester",
            "last_name": "Testerman",
            "clientExternalId": "1234569",
            "ref3": "arbitrary text",
            "ref4": "arbitrary text2",
            "ref5": "arbitrary text3",
            "ref7": "arbitrary text4",
            "ref8": "arbitrary text5",
            "ref9": "arbitrary text6",

        }
    }
```

```
}
```

Testing

COURSE_COMPLETED Test Endpoint

This endpoint will allow a client to manually trigger the mechanism which sends course completion events back to client systems. The client can specify an endpoint to which to send the test event payload. The client will send a payload to the Praesidium endpoint and a success response will be sent to the endpoint that the client specified.

Endpoint: https://test.praesidiumacademy.com/portal/event_pub_webhooks/client_course_action

Method: POST

Sample Payload (note the "T" embedded in the "date" field format)

```
{
"client_id": "35F5C5A985D111EB857A0A3ECA36592D",
    "user_guid": "584adf35-85d1-11eb-857a-0a3eca36592d",
    "email": "lcarl@notreallythere.com",
    "location_guid": "5f383262-85d1-11eb-857a-0a3eca36592d",
    "courseSku": "TCCE1001",
    "date": "2021-03-11T12:01:03",
    "url": "https://test.client.eventpublication/endpoint"
}
```

> Note that the url parameter is optional. When present, the current client event publishing endpoint will be changed to the url value in the payload and will be the endpoint used until it is changed again.

Sample Response (HTTP 200)

```
{
    "message": "success"
}
```

Revision #14
Created 14 December 2022 19:47:04 by Scott Fulbright
Updated 9 May 2023 12:29:09 by Scott Fulbright