ii. Fees Collection

Event : Fees Collection

Enrolment Fees data items are submitted by the Partners through their Training Management Systems. The source system will submit the requests to their respective DL members. This leads to events being registered across the SSG Training Ecosystem DL, thereby notifying SSG to continue processing the application.

Each Fees Collection event tracks the combination of a Trainee, Company (Employer, where applicable), Course Run and Partner.

The data submitted for Fees Collection should include all the following attributes:

Payload Attributes	Event Fields	Mandatory	Input /Output	Data Type	Description
(enrolment) action	Action	Y	Input	String	Defines the action to be taken by this request For Fees Collection the action should be "update"
(enrolment) (trainingPartner) code	Training Partner Code	Y	Input	String	Training Partner's Code
(enrolment) (trainingPartner) uen	Training Partner UEN	Y	Input	String	Training Partner's organisation UEN
(enrolment) (trainee) (fees) collectionStatus	Fees Collection Status	Y	Input	String	Fees Collection Status E.g. Pending Payment / Partial Payment / Full Payment / Cancelled
(enrolment) referenceNumber	Enrolment Reference Number	Y	Input /Output	String	Enrolment Reference Number Generated by SSG after successful validation
(enrolment) status	Enrolment Status	Y	Output	String	Status of enrolment generated by SSG. E.g. Confirmed/Cancelled ***SSG will not cancel an enrolment unless Training Providers submit a cancel enrolment request***

Input Request Parameters :

The input to the DL Interface comprises of the following parts:

- **Header** : consists of fields which are required by the DL Interface to process the payload

Attributes	Data Type	Description
eventType	String	Type of the event (FeesCollection)
primaryKey	String	Primary Lookup Key CONCAT(Course Reference Number,Trainee ID)
secondaryKey	String	Secondary Lookup Key (Course Run ID)
tertiaryKey	String	Tertiary Lookup Key For Fees Collection Request: "Enrolment Reference Number"
trainingPartnerUen	String	Training Partner's UEN
trainingPartnerCode	String	Training Partner's Branch Code
schemaLocation	String	Location of the JSON schema file stored locally in the system
schemaVersion	String	Version of the schema

The schemaLocation field should point to the location of the file in SSG's Repository. For data validation before writing to the DLT, this JSON schema file could be retrieved from SSG's Repository and stored in the local file system.

- **Payload**: consists of the actual Fees Collection data; all data within the payload is to be encrypted before writing to the DLT

Attributes	Data Type	Description
payload	Serialized JSON Object	This is the set of nested JSON objects of Fees Collection data.

- **Public Payload**: additional fields which are **not encrypted** before writing to the DLT

Attributes	Data Type	Description
tags	Array of String	This is an Array of String tags for future use.

(source) dateTime	String	Source System time stamp in seconds ISO 8601 Datetime format
(source) timeStampInMilliSeconds	String	Source System time stamp in milli seconds Unix Datetime format
(ack) dateTime	String	Acknowledgement time stamp in seconds. To be provided as "-1" by Source System ISO 8601 Datetime format
(ack) timeStampInMilliSeconds	String	Acknowledgement time stamp in milli seconds. To be provided as "-1" by Source System Unix Datetime format

In order to align with SSG's time, the source system should be in sync with the Amazon Time Sync Service via Network Time Protocol (NTP). If the source system time is not aligned with SSG's time, this could lead to improper sequencing of data updates, which in turn would lead to erroneous data submission via the DLT.

- **DLT Data**: fields updated by the Distributed Ledger. Any values supplied by the source system will not be used.

Attributes	Data Type	Description
eventSource	String	Source system initiating the request
timeStamp	String	Time stamp updated by the DL
validationResult	String	Result of the validation performed by SSG

The values for the fields in the DLT schema should not be blank; if any field is left blank it will lead to errors in data submission to SSG

This is a sample of the serialized JSON object to be supplied as input to the DL Interface by Source System:

FessColl-Req

```
{
        "header": {
                "eventType": "FeesCollection",
                "primaryKey": "TGS-0026008-ESS0118316H",
                "secondaryKey": "10026",
                "tertiaryKey": "ENR-1912-000123",
                "trainingPartnerUen": "T08GB0032G",
                "trainingPartnerCode":"T08GB0032G-01",
                "schemaLocation": "<TBC>",
                "schemaVersion": "TGS_v1.0"
        },
        "payload": {
            "enrolment": {
                "action" : "update",
                "trainingPartner": {
                        "code": "T08GB0032G-01",
                        "uen": "T08GB0032G"
                },
                "trainee": {
                    "fees": {
                            "collectionStatus": "pending payment"
                    }
                },
                "referenceNumber":"ENR-1912-000123"
         }
        },
        "publicPayload": {
                "tags": ["TBC"],
                "source": {
                        "dateTime": "2020-05-04 20:58:17",
                        "timeStampInMilliSeconds": "1588597097880"
                },
                "ack": {
                        "dateTime": "-1",
                        "timeStampInMilliSeconds": "-1"
                }
        },
        "dltData": {
                "eventSource": "",
                "timeStamp": "",
                "validationResult": ""
        }
}
```

The above data is written to the DLT, after the primary key is hashed and payload is encrypted. The DLT-specific fields are populated accordingly and the "validationResult" is set to "TGS-300" which means "PENDING_VALIDATION" until SSG validates the data items.

Response Parameters :

Upon receiving this information, SSG performs validation and returns a response; the response depends on the result of the validation.

The following fields are updated as explained below:

- **Header** : all other fields and values from the request header are retained; only the Tertiary key is updated

Attributes	Data Type	Description
tertiaryKey	String	If validation is successful, **Enrolment Reference Number** If validation failed, **-1**
ALL OTHER FIELDS & VALUES FROM THE REQUEST HEADER ARE RETAINED		

- **Payload**: consists of the actual Fees Collection data; all data within the payload is to be encrypted before writing to the DLT

Attributes	Data Type	Description
payload	Serialized JSON Object	This is the JSON object of Fees Collection data.

- **Public Payload**: additional fields which are **not encrypted** before writing to the DLT

Attributes	Data Type	Description
tags	Array of String	This is an Array of String tags for future use.
(ack) dateTime	String	Acknowledgement time stamp with seconds. To be updated by SSG. ISO 8601 Datetime format
(ack) timeStampInMilliSeconds	String	Acknowledgement time stamp with milli seconds. To be updated by SSG. Unix Datetime format

- **DLT Data**: all fields are updated by SSG to indicate the result of validation

Attributes	Data Type	Description
eventSource	String	SSG
timeStamp	String	Time stamp updated by the DL
validationResult	String	Status of validation updated by SSG If validation is successful, the code is TGS-200 If validation failed, TGS_4XX

NOTE : The description for the validation error codes should be fetched from the API, details of the API will be provided in future.

A sample successful response of validation from SSG is shown below. The primary keys as read from the DLT are shown in hashed form and the payload is depicted after decryption.

FeesColl-Resp-Valid

```
{
        "header": {
                "eventType": "FeesCollection",
                "primaryKey":
"80e00f124e8bd67257fd0291a8491c3b2ce3ee838ff9dafd91a841da0f7c329174eaaa0006e289e5536f46d0529be058",
                "secondaryKey": "10026",
                "tertiaryKey": "ENR-1912-000123",
                "trainingPartnerUen": "T08GB0032G",
                "trainingPartnerCode":"T08GB0032G-01",
                "schemaLocation": "<TBC>",
                "schemaVersion": "TGS_v1.0"
        },
        "payload": {
            "enrolment": {
                "action" : "update",
                "trainingPartner": {
                        "code": "T08GB0032G-01",
                        "uen": "T08GB0032G"
                },
                "trainee": {
                    "fees": {
                            "collectionStatus": "pending payment"
                    }
                },
                "referenceNumber": "ENR-1912-000123",
                "status": "Confirmed"
                }
        },
        "publicPayload": {
                "tags": ["TBC"],
                "source": {
                        "dateTime": "2020-05-04 20:58:17",
                        "timeStampInMilliSeconds": "1588597097880"
                },
                "ack": {
                        "dateTime": "2020-05-05 20:58:17",
                        "timeStampInMilliSeconds": "1588683497880"
                }
        },
        "dltData": {
                "eventSource": "SSG",
                "timeStamp": "2020-05-04T20:58:38.251Z",
                "validationResult": "TGS-200"
        }
}
```

A sample failure response of validation from SSG is shown below. In this scenario, the enrolment is considered to have failed and a reference number is not assigned by SSG. The primary keys as read from the DLT are shown in hashed form and the payload is depicted after decryption.

FeesColl-Resp-Inv

```
{
        "header": {
                "eventType": "FeesCollection",
                "primaryKey":
"80e00f124e8bd67257fd0291a8491c3b2ce3ee838ff9dafd91a841da0f7c329174eaaa0006e289e5536f46d0529be058",
                "secondaryKey": "10026",
                "tertiaryKey": "-1",
                "trainingPartnerUen": "T08GB0032G",
                "trainingPartnerCode":"T08GB0032G-01",
                "schemaLocation": "<TBC>",
                "schemaVersion": "TGS_v1.0"
        },
        "payload": {
            "enrolment": {
                "action" : "update",
                "trainingPartner": {
                        "code": "T08GB0032G-01",
                        "uen": "T08GB0032G"
                },
                "trainee": {
                    "fees": {
                            "collectionStatus": "pending payment"
                    }
                },
                        "referenceNumber": "-1"
                }
        },
        "publicPayload": {
                "tags": ["TBC"],
                "source": {
                        "dateTime": "2020-05-04 20:58:17",
                        "timeStampInMilliSeconds": "1588597097880"
                },
                "ack": {
                        "dateTime": "2020-05-05 20:58:17",
                        "timeStampInMilliSeconds": "1588683497880"
                }
        },
        "dltData": {
                "eventSource": "SSG",
                "timeStamp": "2020-05-04T20:58:38.251Z",
                "validationResult": "TGS-4XX"
        }
}
```

Submission of data via the DLT

Submission of Fees Collection records to the SSG Training Ecosystem DL Network will only be for the following case:

- **Record Update**

For the Fees Collection event, the update action requires some pre-requisite conditions to be met. In order to update a Fees Collection event, a pre-existing valid Enrolment record (with SSG-issued Enrolment reference number) should be present. Also, it is not valid to update the Fees Collection event for a record whose Grants status is Approved. If these pre-requisites are satisfied, partners may update the Fees Collection event.

Record Update: This indicates the case when records are updated:

1. For an update, partners are to generate a Fees Collection event for a given primary key and secondary key; the tertiary key should be set to the Enrolment reference number. The action field should be set to update. Once this is submitted, the partner should wait for the response from SSG for this event. No new events pertaining to that record are to be submitted in the meantime.
2. SSG validates the submission and if the data is valid, returns the corresponding status in the validationResult field.
3. Partners listening for events from the DL Network can read this response and extract the result of validation.