Implementation Guide

Electronic Payment Server

March 6, 2023

API Version 1.0

Document Summary

This document provides guidance for building payment solutions within the petroleum convenience industry for processing payments using the Electronic Payment Server (EPS) Specification. This document focuses on the transaction flows and message contents for processing financial transactions using the EPS RESTful API. This Implementation Guide is part of a suite of documents, including the Business Requirements, Process Document, and Use Cases.

Copyright © IFSF, CONEXXUS, INC., 2023, All Rights Reserved

Contributors

Brian Russell, Verifone
Casey Brant, Conexxus
Clerley Silveira, PDI
Dan Harrell, Invenco
Darryl Miller, Verifone
Donna Perkins, Conexxus
Ian A. Brown, IFSF
Jack Dickinson, Dover Fueling Solutions
Kevin Eckelkamp, Comdata
Khaled El Manawhly, Bulloch Technologies
Kim Seufer, Conexxus
Linda Toth, Conexxus
Sue Chan, W. Capra

Revision History

Revision Date	Revision Number	Revision Editor(s)	Revision Changes
March 6, 2023	Version 1.0	Kim Seufer, Conexxus	Final Release Version 1.0
November 1, 2022	Draft 1.0	Casey Brant, Conexxus	Replaced broken links
October 24, 2022	Draft 0.92	Kim Seufer, Conexxus	Resolved comments after legal review
September 12, 2022	Draft 0.91	Casey Brant, Conexxus	Updated message names to match the redocs and accepted changes in preparation for second legal review.
August 31, 2022	Draft 0.9	Linda Toth, Conexxus	Updates after technical legal review
August 10, 2022	Draft 0.8	Casey Brant, Conexxus	Accepted changes in preparation for legal review
July 22, 2022	Draft 0.7	Sue Chan, W. Capra	Additional formatting
July 22, 2022	Draft 0.6	Sue Chan, W. Capra	Clarifications to Section 8.1 and formatting

Copyright © IFSF, CONEXXUS, INC., 2023, All Rights Reserved

July 12, 2022	Draft 0.5	Sue Chan, W. Capra	Updates: watermark, copyright, formatting, clarification, discussion during WG meeting
June 24, 2022	Draft 0.4	Sue Chan, W. Capra	Updates – added Establish Connection and POS Connection Process sections
May 18, 2022	Draft 0.3	Sue Chan, W. Capra	Updates
April 8, 2022	Draft 0.2	Darryl Miller, Verifone	Updates to InitialDraft
February 11, 2022	Draft 0.1	Kim Seufer, Conexxus	Initial Draft

Copyright © IFSF, CONEXXUS, INC., 2023, All Rights Reserved

Copyright Statement

Copyright © IFSF, CONEXXUS, INC., 2023, All Rights Reserved

The content (content being images, text or any other medium contained within this document which is eligible of copyright protection) are jointly copyrighted by Conexxus and IFSF. All rights are expressly reserved.

IF YOU ACQUIRE THIS DOCUMENT FROM IFSF. THE FOLLOWING STATEMENT ON THE USE OF COPYRIGHTED MATERIAL APPLIES:

You may print or download to a local hard disk extracts for your own business use. Any other redistribution or reproduction of part or all of the contents in any form is prohibited.

You may not, except with our express written permission, distribute to any third party. Where permission to distribute is granted by IFSF, the material must be acknowledged as IFSF copyright and the document title specified. Where third party material has been identified, permission from the respective copyright holder must be sought.

You agree to abide by all copyright notices and restrictions attached to the content and not to remove or alter any such notice or restriction.

Subject to the following paragraph, you may design, develop and offer for sale products which embody the functionality described in this document.

No part of the content of this document may be claimed as the Intellectual property of any organisation other than IFSF Ltd and Conexxus, Inc, and you specifically agree not to claim patent rights or other IPR protection that relates to:

a) the content of this document; or
b) any design or part thereof that embodies the content of this document whether in whole or part.

For further copies and amendments to this document please contact: IFSF Technical Services via the IFSF Web Site (www.ifsf.org).

IF YOU ACQUIRE THIS DOCUMENT FROM CONEXXUS, THE FOLLOWING STATEMENT ON THE USE OF COPYRIGHTED MATERIAL APPLIES:

Conexxus members may use this document for purposes consistent with the adoption of the Conexxus Standard (and/or the related documentation), as detailed in the Implementation Guide; however, Conexxus must pre-approve any inconsistent uses in writing.

Except in the limited case set forth explicitly in this Copyright Statement, the Member shall not modify, adapt, merge, transform, copy, or create derivative works of the Conexxus Standard, including the documentation suite and the application programing interface ("API"). Conexxus recognizes that the API may include multiple Definition Files, and accordingly recognizes and agrees that the Member may implement one, some, or all Definition Files within the API, unless otherwise specified in the Implementation Guide, provided that each Definition File implemented is implemented in full. Here implementing a Definition File in full means that all functionality defined by the Conexxus Standard for the Definition File is implemented. Regardless of whether the Member implements one, some, or all Definition Files, the Member agrees to abide by all requirements under this Copyright Statement for each of the Definition Files implemented.

Note that some functionality within a Definition File is specified for predefined error or non-implementation codes to be returned. For functionality where such predefined codes are specified, returning such a predefined code constitutes an implementation. However, in such cases, a Member may not return codes or values different from the predefined codes, nor may the Member simply not implement the functionality, as this would create a Definition File that was not fully implemented as required under this Copyright Statement.

The Member hereby waives and agrees not to assert or take advantage of any defense based on copyright fair use. The Member, as well as any and all of the Member's development partners who are responsible for implementing the Conexxus Standard for the Member or may have access to the Conexxus Standard, must be made aware of, and agree to comply with, all requirements under this Copyright Statement prior to accessing any documentation or API.

Conexxus recognizes the limited case where a Member wishes to create a derivative work that comments on, or otherwise explains or assists in its own implementation, including citing or referring to the standard, specification, code, protocol, schema, or guideline, in whole or in part. The Member may do so **ONLY** for the purpose of explaining or assisting in its implementation of the Conexxus Standard and the Member shall acquire no right to ownership of such derivative work. Furthermore, the Member may share such derivative work **ONLY** with another Conexxus Member who possesses appropriate document rights or with an entity that is a direct contractor of the Conexxus Member who is responsible for implementing the standard for the Member. In so doing, a Conexxus Member shall require its development partners to download Conexxus documents, API, and schemas directly from the Conexxus website. A Conexxus Member may not furnish this document in any form, along with any derivative works, to non-members of Conexxus or to Conexxus Members who do not possess document rights or who are not direct contractors of the Member, including to any direct contractor of the Member who does not agree in writing to comply with the terms of this Copyright Statement. A Member may demonstrate its Conexxus membership at a level that includes document rights by presenting an unexpired digitally signed Conexxus membership certificate.

Copyright © IFSF, CONEXXUS, INC., 2023, All Rights Reserved

This document may not be modified in any way, including removal of the copyright notice or references to Conexxus. However, a Member has the right to make draft changes to schema or API code for trial use, which must then be submitted to Conexxus for consideration to be included in the existing standard. Translations of this document into languages other than English shall continue to reflect the Conexxus copyright notice.

The limited permissions granted above are perpetual and will not be revoked by Conexxus, Inc. or its successors or assigns, except in the circumstance where an entity, who is no longer a member in good standing but who rightfully obtained Conexxus Standards as a former member, is acquired by a non-member entity. In such circumstances, Conexxus may revoke the grant of limited permissions or require the acquiring entity to establish rightful access to Conexxus Standards through membership.

Disclaimers

IF YOU ACQUIRE THIS DOCUMENT FROM CONEXXUS, THE FOLLOWING DISCALIMER STATEMENT APPLIES:

Conexxus makes no warranty, express or implied, about, nor does it assume any legal liability or responsibility for, the accuracy, completeness, or usefulness of any information, product, or process described in these materials, even if such liability was disclosed to Conexxus or was foreseeable. Although Conexxus uses commercially reasonable best efforts to ensure this work product is free of any encumbrances from third-party intellectual property rights (IPR), it cannot guarantee that such IPR does not exist now or in the future. Conexxus further notifies each user of this standard that its individual method of implementation may result in infringement of the IPR of others. Accordingly, each user is encouraged to seek legal advice from competent counsel to carefully review its implementation of this standard and obtain appropriate licenses where needed.

Copyright © IFSF, CONEXXUS, INC., 2023, All Rights Reserved

Table of Contents

Copyright © IFSF, CONEXXUS, INC., 2023, All Rights Reserved

Project

Electronic Payment Server

1 Introduction and Overview

The Electronic Payment Server (EPS) Specification is a joint standard shared by Conexxus and IFSF and supports global EPS implementations. The intent of this API group is to provide functionality necessary for the global market to manage credit, debit, fleet, and other proprietary card transaction processing from the Point of Sale (POS) to the EPS.

The Specification supports payment inside at a POS device and payment from an outdoor payment terminal (OPT), including unattended dispensers and car washes. In the EPS environment, the PIN pad is no longer a device of the POS and the EPS can function as a standalone processor. The EPS's connection to one or more hosts or front-end processors (FEPs) segregates payment functionality within the EPS. In addition, prompts presented to the customer are the same regardless of the POS. The Specification also supports standardized messaging between a Point of Purchase/Point of Interaction (POP/POI) terminal and the EPS. The EPS Specification separates payment from the other components of a fuel retailing site, such as a point of sale (POS) system or outdoor sales processor (OSP). Separating payment functionality eliminates the burden of maintaining network specific software on the POS, may reduce PCI scope, and may improve interoperability. In addition, by using an EPS, making changes to POS software applications to add or modify features and functionality no longer require payment recertification. Software in the EPS can be updated and downloaded independently from the POS and may not require a site visit, thereby potentially reducing cost and providing greater POS independence. Card processing within the EPS is table driven and unique by merchant, which provides greater flexibility and card acceptance support. The card table supports known debit, credit, prepaid, and fleet card processing using multiple entry methods, including, but not limited to magnetic stripe, ICC, RFID, and keyboard entry.

Copyright © IFSF, CONEXXUS, INC., 2023, All Rights Reserved

2 Architecture

2.1 API Architecture

This API group follows the normal structure as described in “Open Retailing Design Rules for APIs OAS3.0”.

This API uses RESTful Web Services, associating required functionality with resources and operations on those resources. For handling unsolicited events from the service provider to the client, it uses HTML5 constructs such as "Server Sent Events" and "Web Sockets." The interfaces are "highly cohesive" and "loosely coupled" in order to provide maximum flexibility to the implementer, and to allow implementation of an individual API definition file (ADF) if that construction is useful to the implementer.

2.2 Logical Entities

This section outlines the logical entities for the EPS API Specification. The term “entity” is used in this document to differentiate logical processing functionality without regard to its physical location in an implementation.

EPS – Electronic Payment Server
POS – Point of Sale
OSP – Outside Sales Processor
POI – Point of Interaction (also known as Point of Purchase (POP), can include PinPads, Receipt Printers, Scanners)
OPT – Outside Payment Terminal – can contain OSP and POI functions
FDC - Forecourt Device Controller

3 Security Considerations

For security considerations, please refer to the Threat Model document for this API. Also, Conexxus provides an overall “Technical Security Considerations” document that should be the basis of the security implementation of this API. This document outlines best practices for implementing technology at retail locations. In addition, there is an “Open Retailing API Implementation Guide: Security” document that addresses the security aspects of API transport technologies.

4 Protocol

This API group follows the standard recommendations for protocol described in “Open Retailing API Implementation Guide - Transport Alternatives”.

Copyright © IFSF, CONEXXUS, INC., 2023, All Rights Reserved

5 Data Model

N/A

6 Data Specification

The details of the data specification can be found in the “docs/Schema Documentation” directory as “Redoc” generated HTML files.

7 Internationalization

The EPS Specification is a joint specification developed by Conexxus and IFSF. It supports international implementations and data elements (e.g., currency, units of measure, language, country codes). Translations, currency exchange rates, and multi-language support are implementation specific, which makes them the responsibility of the EPS provider.

8 Implementation Details

8.1 API Overview

The API Group is divided into several API Definition Files.

The API Definition File (ADF) details are documented separately as listed below.

Three of the ADF files (connection, eps, and dca) are required when implementing this standard. POP/POI interfaces are also required but may be a legacy interface, and/or the poi API documented in this standard. This accommodates different implementation schedules among POP/POI vendors, as well as when POP/POI devices are upgraded/replaced at the site.

When implementing any single API Definition File, the implementer must implement the entire file protocol as provided in the API. If the functionality is not supported in an implementation, it should return an appropriate error code.

Note: Each of the definitions below can be found in the “../Schema Documentation” directory relative to this current document, named as shown below, i.e., “<definition-name>-redoc.html” would be “connection-redoc.html” for the first definition below.

> connection: Contains the resource used to establish a connection to the EPS or as part of the heartbeat process to verify the EPS is online.

Copyright © IFSF, CONEXXUS, INC., 2023, All Rights Reserved

eps: Contains the resources used by a POS, Site System or OSP to interact with the EPS.

dca: (Data Configuration APIs): Contains the resources used by a POS, Site System, OSP or POI for administrative functions.

poi: Contains the resources used by a POI to interact with the EPS.

8.1.1 Events

POS, POIs, and other workstations should establish an event stream (Server Sent Event (SSE)) with the EPS, subscribing to specific events of interest. The EPS will then be able to send event messages to the appropriate workstation(s). Each message contains an "event:" field followed by a "data:" description.

Note: The redoc for the file below can be found in the "../Schema Documentation" directory relative to this current document.

sse-events-definitions-only: This file is not to be used as an actual API resource, but rather as an example that describes the events that the EPS can send along with information regarding the action that would be performed by the workstation that received the event.

8.2 Operation Details

8.2.1 Establish Connection

- The `/connection` message is sent by each workstation to establish the connection with the EPS
- Each workstation provides its `workstationID`, which will be unique per the implementation.
- An EPS may want to validate the workstations that are requesting a connection – if so, the EPS would need to know the possible `workstationID` that should connect. This validation is outside the scope of the implementation specifics of the EPS API standard.
- An EPS may want to know the location from which the workstation is requesting a connection, especially for an EPS that manages/processes several sites. An implementation may desire that each workstation also provide a `locationID` in addition to the `workstationID` in the `/connection` message. This would be implementation specific.

8.2.2 POI Registration Process

This subsection provides the details associated with the POI registration to process.

Register the POI with the EPS

Copyright © IFSF, CONEXXUS, INC., 2023, All Rights Reserved

- POI sends to the EPS the POST `/POIs/registration`
    - The `poiID` needs to be unique per implementation. It is recommended that the serial number is used as the `poiID`.
    - The POI has been previously configured with a unique logical ID per site.
    - This request would be sent upon `poiID` start-up.

Request a connection with the EPS

- POI sends to the EPS the POST`/connection`
    - The ID would be the `poiID` and the `IDType` would be `workstationID`

Send the workstation settings

- POI sends to the EPS the POST `/workstationSettings`
    - The `workstationID` would be the same as the `poiID`
    - The `workstationType` would be `pointOfInteraction`

Request the event stream endpoint

- POI sends to the EPS the GET `/sse-events`
    - Register for all applicable events

Establish the event stream (Server Stream Event)

- POI to establish an HTTP connection to the endpoint and listen for the events.

8.2.3 POS Connection Process

This subsection provides the details associated with the POS connection process.

Request a connection with the EPS

- POS sends to the EPS the POST `/connection`
    - The `ID` should be unique for the site.
    - The `IDType` would be `workstationID`

Send the workstation settings

- POS sends to the EPS the POST `/workstationSettings`
    - The `workstationID` would be the same as the `ID` sent in the `/connection`.
    - The `workstationType` would be `pointOfService`

Request the event stream endpoint

- POS sends to the EPS the GET `/sse-events`
    - Register for all applicable events

Establish the event stream (Server Stream Event)

- POS to establish an HTTP connection to the endpoint and listen for the events.

Copyright © IFSF, CONEXXUS, INC., 2023, All Rights Reserved

Request the list of POIs registered with the EPS.

- POS to send a GET `/POIs`.
- The EPS responds with the list of POIs, including the `logicalID` and `poiID`.
- The POS has been previously configured to use a specific `logicalID`.
- The POS will use the `poiID` associated with that `logicalID` in all EPS API calls.

8.2.4 Heartbeat Process

The EPS will send the `heartBeatEvent` to each workstation (e.g., POI, POS) at a set interval, during idle times, previously agreed upon in the implementation (e.g., 45 seconds). If the workstation has not received a `heartBeatEvent` per the interval, the workstation should begin its reconnect logic.

8.2.5 Reconnect Logic

The workstation should perform reconnect logic as follows:

- Send to the EPS a POST `/connection`.
- Upon successful response, set up the event stream as follows:
  - Send a GET `/sse-events`, registering for all applicable events.
  - Establish an HTTP connection to the endpoint and listen for the events.
- If an unsuccessful response is received or if the POST `/connection` times-out, wait a random amount of time and resend the POST `/connection`. The random retry will help to reduce an overload of network connection traffic.

8.2.6 Reconciliation Process

The POS will send a message to the EPS to start the reconciliation process (POST `/reconciliation`). This normally will occur during the POS end-of-day process.

The POS will send the message to the EPS to perform the reconciliation process in either a synchronous or asynchronous mode with the POS, using the `synchronous` field.

Synchronous:

- `synchronous` is set to “yes” or TRUE
- The EPS will only respond with a 200 success when the EPS has finished its reconciliation process.
- When the POS receives the 200 success, it can perform any of the GET `/reconciliation/{reconciliationID}` calls that it desires.

Asynchronous:

- `synchronous` is set to “no” or FALSE
- The EPS will respond immediately with a 200 success.
- The EPS will send a `reconciliationCompletionEvent` when it has finished its reconciliation process.

Copyright © IFSF, CONEXXUS, INC., 2023, All Rights Reserved

- When the POS receives the `reconciliationCompletionEvent`, it can perform any of the GET `/reconciliation/{reconciliationID}` calls that it desires.

A.References

A.1 Normative References

From "OpenRetailing: API Design Guidelines":

- Open Retailing API Design Rules for JSON
- Open Retailing API Implementation Guide – Security
- Open Retailing API Implementation Guide - Transport Alternatives
- Open Retailing Design Rules for APIs OAS3.0

Conexxus Standards:

- Technical Security Considerations: This document provides high-level technical security guidance for Conexxus standards. Please note you must be logged into the Conexxus website to access this document.

External Standards:

- Hypertext Transfer Protocol (HTTP/1.1) RFC 7231
- RESTFul Web Services
- Open API Specification Version 3.0.3
- HTML5

A.2 Non-Normative References

N/A

Copyright © IFSF, CONEXXUS, INC., 2023, All Rights Reserved

B.Glossary

Term	Definition
EPS	Electronic Payment Server
POI	Point of Interaction
FEPs	Front-End Processors
POP	Point of Purchase
OSP	Outdoor Sales Processor
POS	Point of Sale
OAS	OpenAPI Specification
IFSF	International Forecourt Standards Forum
OPT	Outdoor Payment Terminal

Copyright © IFSF, CONEXXUS, INC., 2023, All Rights Reserved