COMDATA®
CORPORATE PAYMENTS

IMPLEMENTING COMDATA WEBHOOKS USER GUIDE

Disclaimer

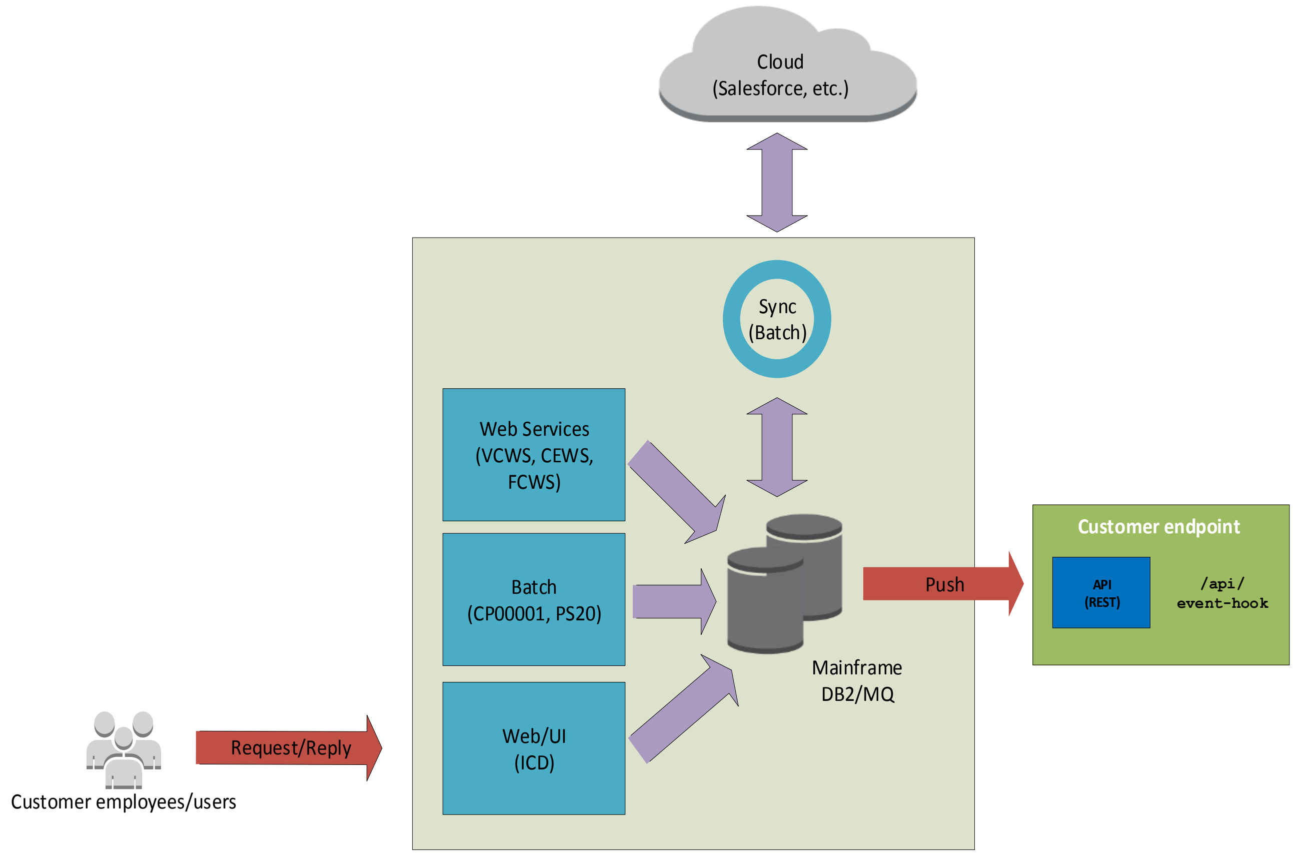
The information furnished herein by Comdata is proprietary and confidential and is intended for Comdata customers and/or Comdata internal use. It should not be duplicated, published, or disclosed in whole or in part without the prior written permission of Comdata.

Table of Contents

Overview

A Comdata partner typically interacts with the Comdata Card platform via web services, batch file processing, or web applications hosted on Iconnectdata.com (ICD). These interactions require a partner to initiate a request (submit a web service request, send a batch file, or submit a web application request) and wait for a response from platform.

To complement the UI, batch, and web services integration options, Comdata developed a publishing platform to ***notify partners in near real time of card and payment-related events***.



This document includes information on how to subscribe to and consume events delivered by the Comdata Card Delivery platform.

Event Payload Reference

The Comdata Card Platform delivers a JSON payload to a subscriber's pre-defined HTTP endpoint for various payment-related events. Details of each event payload are listed in the sections below.

Comdata Card Platform Supported Events:

Event	Notification Trigger
Card status changes	When a card status changes or when a card is blocked
Card transactions	When a card transaction is occurs (AUTH, POST, CREDIT, etc.)
Customer vendor changes	When a customer vendor is added or altered

Obtaining sample payloads:

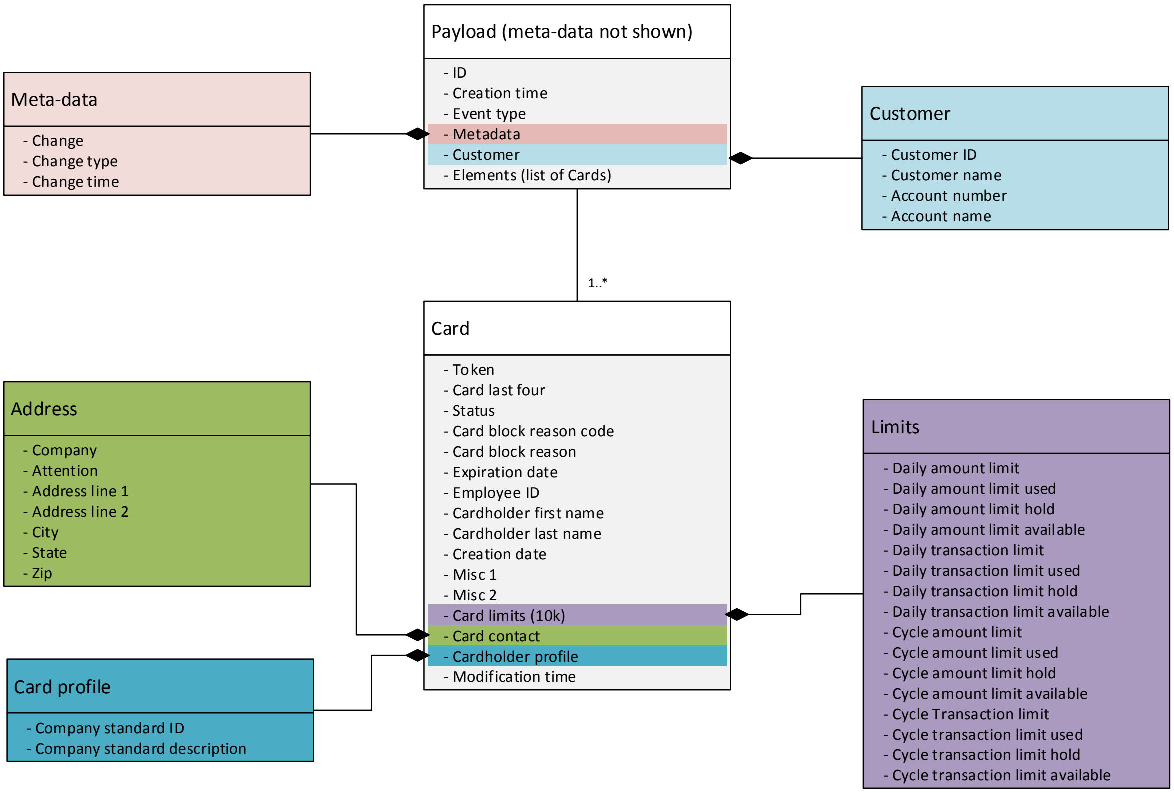
Your TRR can provide you sample JSON payloads and a Swagger definitions for each event type.

Card Status Change Event

Notification Triggers

- Status of card has changed
- Block reason on a card had been added or changed

Payload Model



Card Status Definitions

Status Code	Description
A	ACTIVE - card is active
B	BLOCK - card is blocked
C	CREATED – card has been created
D	DELETED - card has been deleted
E	EXPIRED – card is expired
F	FRAUD - card is labeled for fraud
G	PARTIAL BLOCK -- card is for immediate purchases without having a physical card present
K	CARD CHARGE BACK – bank initiated credit
L	LOST CARD – card is labeled as lost
M	MOVE – card has moved to a different CUSTID
S	STOLEN – card labeled as stolen
T	CARD TRANSFER - from one cardholder to another
V	VRU PIN BLK – Blocked due to VRU Pin
X	PERMANENT (SECURITY) BLOCK – These do have reason codes associated with them (see Block reason codes)

Block Reason Codes

Card Block Statuses

Reason code	Reason Description
1	Comdata Automated Block -unknown reason
2	Comdata internal - payee opt out
3	Comdata Automated block - net credit remaining rule
4	Comdata automated block - force post - under card rule
5	Payment limit cap
6	Overpayment
7	Partial payment
8	Online only
9	Security
10	Advisory details not matching
11	Threshold
20	Replaced
56	VP Prefund Block
57	VP Auto Block
58	Fraud override mode
59	Cardholder Confirmed Fraud
60	Manual Permanent Block
999	This is a test
9000	Expired Card Block

Reason code	Reason Description
100	CR - Payee opt out
101	CR - incorrect payee
102	CR - void Card
103	CR - Duplicate Payment
104	CR - Payment Reissued
200	Vehicle out of Service
201	New Vehicle
203	Vehicle Sold
204	Vehicle Transferred
205	Temporary Vehicle
300	Card Damaged
400	New Cardholder
401	Cardholder Transfer
402	Cardholder Leave
403	Cardholder Vacation
404	Cardholder Termed
405	Cardholder out of service
406	Comcheck mobile blocked Card

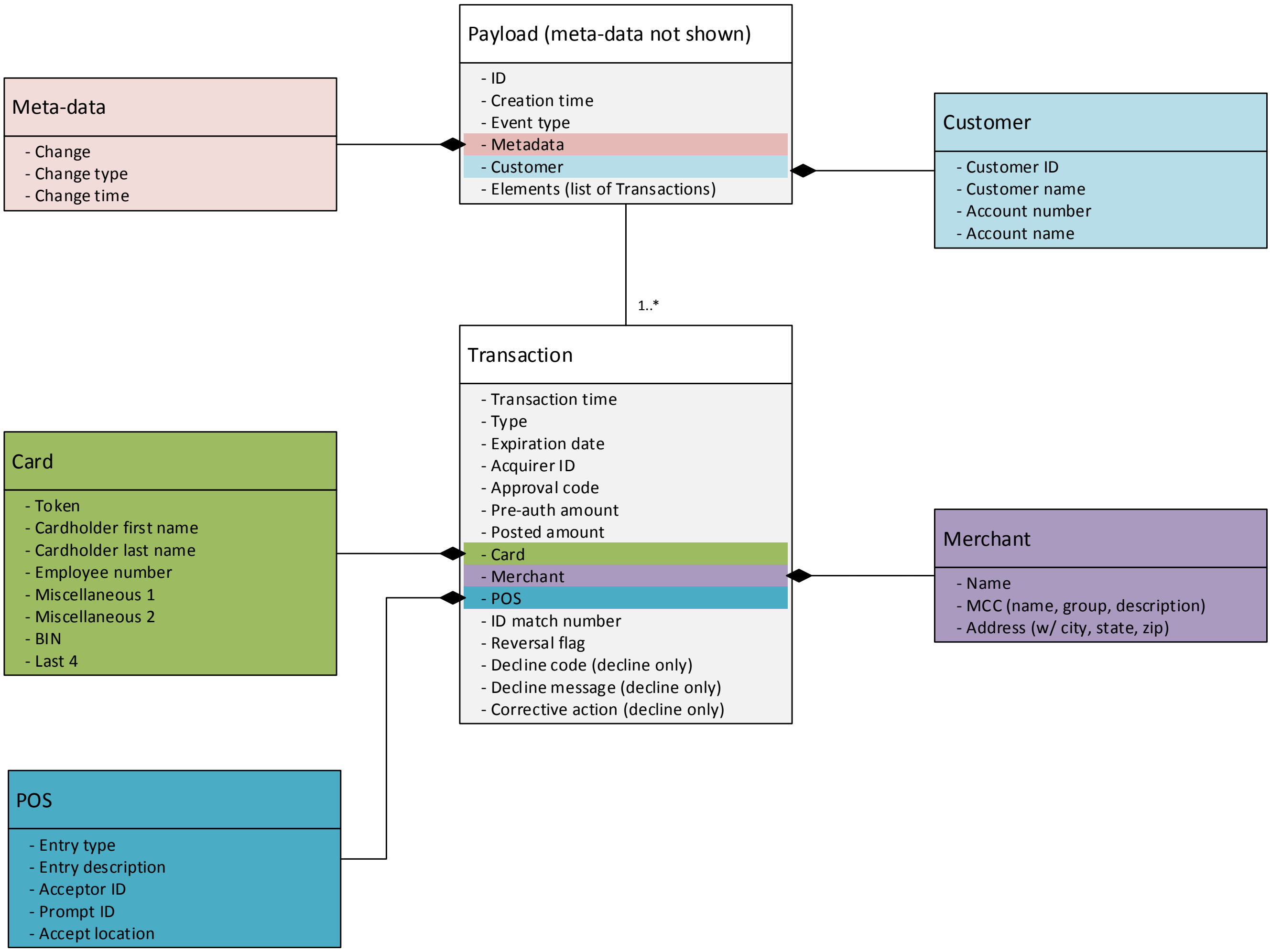
Card Transaction Event

Notification Triggers

The Comdata Card Platform publishes events for the following card transaction:

Transaction type	Definition	Comments
Pre-authorization	Uses preset amount higher or equal to the actual spend to ensure that the card is going to be good before knowing final amount. Primarily used at gas station pumps and hotels.	Typically, a "Pay at Pump" (MCC 5542) or a transaction with ISO-8583 DE61.7 equal to '4' ("Pre-Authorization")
Authorization	Similar to a PRE AUTHORIZATION but for a known amount. When a merchant sends an authorization request, Comdata returns either a decline or an approval code.	
Decline	Any reason for transaction not to fall into any other bucket.	
Post	The actual movement of funds, usually occurring within 24-48 hours after the AUTHORIZATION. This completes the transaction.	Multiple possible for an authorization
Force Post	When a merchant by-passes the authorization process, and goes directly to posting. Client will have chargeback rights if the authorization is by-passed. If a merchant does it too much, they can lose their permission to accept MCs.	No authorization associated with this transaction
Credit	When the merchant sends funds back to the card, behaving similar to a FORCE POST.	An authorization is not required
Reversal	A new transaction that replicates the original transaction but with debit amounts shown as credit amounts and vice versa	Multiple possible for an authorization
Authorization expiration		

Payload Model

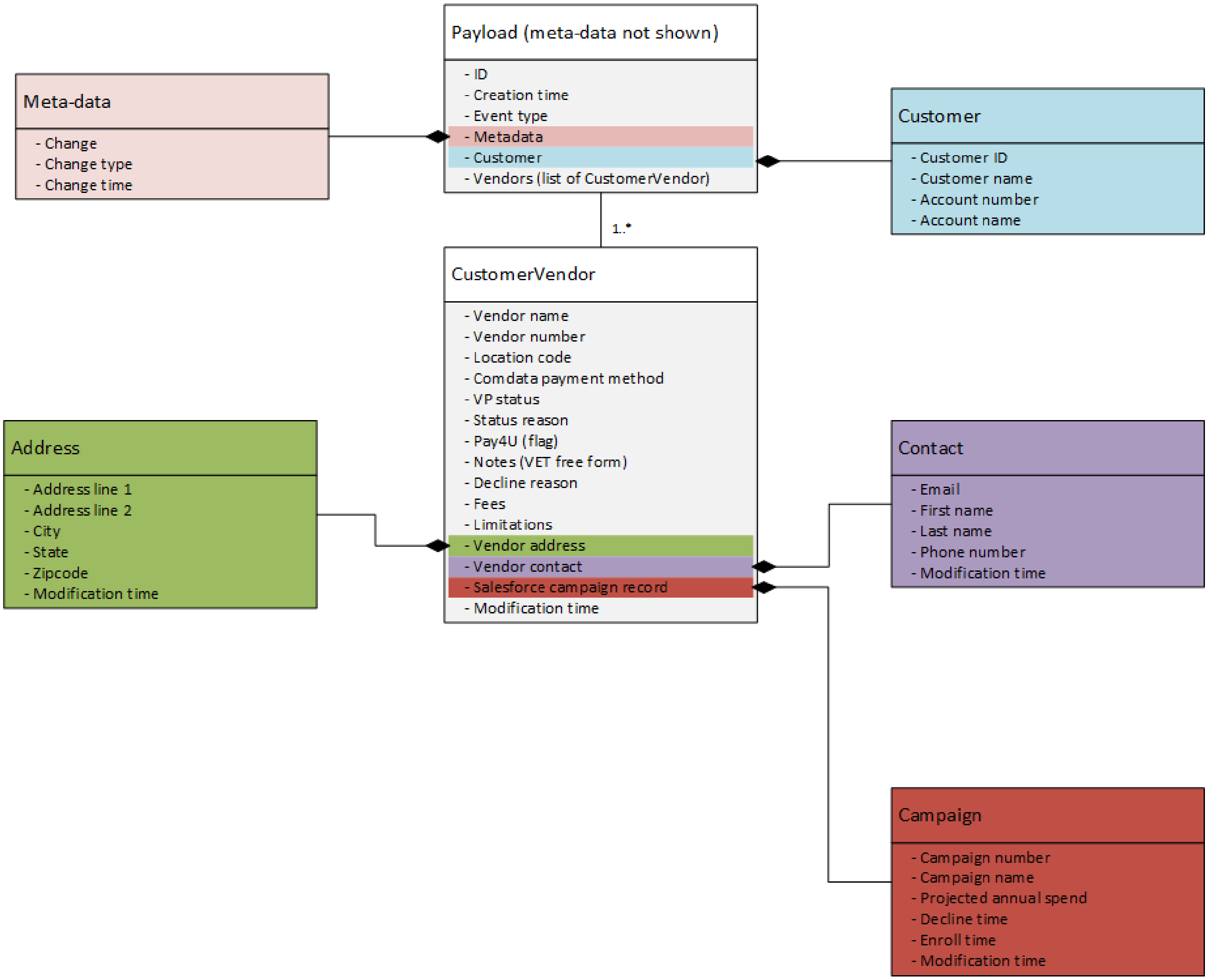


Vendor Update Event

Notification Triggers

- Customer vendor added
- Customer vendor record updated (contact information, campaign information, etc.)

Payload Model



Subscription API

Partners manage their subscriptions to Comdata Card webhook events through a secure JSON/REST Subscription API. With the API, they may subscribe to, unsubscribe from, alter, and view their subscriptions. The operations listed below provide an overview of each operation. Details of these operations can be found in the corresponding *FleetCor Subscription API* Swagger document (http://swagger.io) provided by your TRR.

Interactions with the Subscription API are protected and require an access. You must use the Cogito API to exchange your user credentials for an ID token. Details of this process are outlined in the *Obtaining an ID Token* section of this guide.

The following rules are enforced by the API:

- Multiple subscriptions to the same event are allowed
- Each URL subscribed to the same event must be unique
- The same URL may be subscribed to multiple events

Subscribe to an Event

This operation assigns a partner's HTTP endpoint to accept delivery of JSON payloads associated to a Comdata Card webhook event.

Input:

HTTP Verb / Path	`POST /webhooks`

HTTP headers (required)	`Content-Type: application/json`
	`Authorization: {{ID TOKEN}}`

Path parameters	(None)
Request body	JSON (Subscription)

Output:

Response body	JSON (Subscription) Returns the created subscription

Unsubscribe from an Event

This operation removes a partner's HTTP endpoint from receiving Comdata Card webhook events.

Input:

HTTP Verb / Path	`DELETE /webhooks/{id}`

HTTP headers (required)	`Content-Type: application/json`
	`Authorization: {{ID TOKEN}}`

Path parameters	id – The id of the subscription to remove (unsubscribe)
Request body	(None)

Output:

Response body	(None)

Alter Subscription

This operation modifies the configurations related to an existing subscription, giving a partner the ability to make changes to the receiving endpoint, to security settings, and to delivery policy rules.

Input:

HTTP Verb / Path	`PUT /webhooks/{id}`

HTTP headers (required)	`Content-Type: application/json`
	`Authorization: {{ID TOKEN}}`

Path parameters	id – The id of the subscription to alter
Request body	JSON (Subscription)

Output:

Response body	JSON (Subscription) Returns the altered subscription

List Subscriptions

This operation retrieves all subscriptions linked to a partner.

Input:

HTTP Verb / Path	`GET /webhooks`
	`GET /webhooks/{id}`

HTTP headers (required)	`Content-Type: application/json`
	`Authorization: {{ID TOKEN}}`

Path parameters	id – The id of the subscription to fetch
Request body	JSON (Subscription)

Output:

Response body	JSON (Array of Subscription) for GET /webhooks - or - JSON (Subscription) for GET/webhooks/{id}

Subscription API Properties

Subscription Model Properties

Element Name	Req	Type	Max Length	Comments/Example Value
Event	Y	String	N/A	Event to subscribe **Valid values:** - `card-status-events` - `card-transaction-events` - `vendor-events`
Endpoint	Y	String	100	URL of your REST API **Required:** HTTPS endpoint.
securityPolicy	N	Object	-	Subscription security policy object
securityPolicy.signatureSecret	N	String	50	A secret that you provide to be used to compute the SHA-256 checksum of the event payload provided as an HTTP header value. **Signature HTTP Header:** `X-FC-SIGNATURE` *Optional but strongly recommended.*

Element Name	Req	Type	Max Length	Comments/Example Value
securityPolicy.apiKey	N	String	50	Value sent along with an event payload in the HTTP header which can be used by the consuming web service for authentication purposes. The API key is sent as the HTTP header `X-FC-API-KEY` by default, although this can be modified with the subscription property `apiKeyHeader`. *Optional but strongly recommended.*
securityPolicy.apiKeyHeader	N	String	50	Value of the HTTP header of your API key sent with the event payload. **Default:** `X-FC-API-KEY`
deliveryPolicy	Y	Object	-	Subscription delivery policy object
deliveryPolicy.retries	Y	Number	1 to 100	Maximum number of re-delivery attempts
deliveryPolicy.delay	Y	Number	1 to 3600	Number of *seconds* between delivery re-attempts
deliveryPolicy.maxTPS	Y	Number	1 to 100	Maximum number of deliveries per second

See the *Fleetcor Subscription API* swagger specification for more information about the subscription model.

HTTP Response Codes

Below are a list of response codes that may be returned by one or more of the Subscription API operations. Consult the Swagger specification for a complete list of error codes returned by each operation.

Response	HTTP response code	Response body
Subscription successfully added	201 – CREATED	JSON (Subscription)
Subscription successfully altered	200 – OK	JSON (Subscription)
Subscription successfully removed	204 – NO CONTENT	(None)
Subscription list successfully returned	200 – OK	JSON (*Array* of Subscription)
Input validation error	400 – BAD REQUEST	JSON (Error)
Subscription not found	404 – NOT FOUND	(None)
Authentication failure	403 – FORBIDDEN	JSON (Error)

Notes on Checksum Calculation (X-FC-SIGNATURE)

The Comdata Card Delivery system publishes a SHA-256 checksum of each event payload body to a subscribed endpoint IF the subscription defines a signature secret (see Subscription API section for details). This allows your endpoint to validate the integrity of the payload by computing the checksum of the received message body and comparing it to the `X-FC-SIGNATURE` HTTP header value.

Computing the Checksum:

The checksum is the SHA-256 hash of the event payload body and the signature secret value registered by the customer:

```
checksum = SHA-256(payloadBody + signatureSecret)
```

Example:

Consider an event payload to a subscribed endpoint with a signature secret “SECRET123” and a delivery payload with the following data:

HTTP Headers:

```
Content-Type: application/json
X-FC-API-KEY: secret-key-abc
X-FC-SIGNATURE: adedf939321…
```

Body:

```
{"event": "This is an event payload message"}
```

To compute the checksum, append the signature secret (“SECRET123”) to the *entire* payload body and apply SHA-256.

```
// Compute the checksum using signature secret and payload
body = {"event": "This is an event payload message"}
secret = “SECRET123”
localChecksum = SHA-256(body + secret)

// Compare computed checksum to checksum provided with payload
deliveryChecksum = request.getHttpHeader(“X-FC-SIGNATURE”)
isSafePayload = (localChecksum == deliveryChecksum)
```

Obtaining an ID Token (Cognito API)

The Fleetcor Subscription API requires a valid, non-expired ID token sent along with each request. AWS Cognito provides an API to exchange your credentials for ID tokens. You must change your password when logging into Cognito for the first time. After that you must obtain ID tokens to interact with the subscription API.

Cognito API operations used:

- `AWSCognitoIdentityProviderService.RespondToAuthChallenge` (first time login)
- `AWSCognitoIdentityProviderService.InitiateAuth`

Your TRR will provide you with a POSTMAN collection that includes the requests required to initiate the above operations. It also includes examples for changing/resetting passwords. Contact your TRR for more information.

For more information about the Cognito API:
https://docs.aws.amazon.com/cognito-user-identity-pools/latest/APIReference

Your Credentials

During the on-boarding process, your TRR will send credentials and configuration information required to authenticate and subscribe to Fleetcor events via the Subscription API. This information includes:

- Temporary credentials (username & temporary password)
- Authentication CLIENT ID (string)
- AWS Cognito authentication URL: https://cognito-idp.us-east-1.amazonaws.com
- Fleetcor Subscription API URL: https://subscribe.fleetcorpayments.com

Your credentials along with the CLIENT ID will be required for all interactions with Cognito. The sections below summarize how to interact with the `InitiateAuth` and `RespondToAuthChallenge` Cognito operations. For more detailed information about the Cognito API reference documentation.

RespondToAuthChallenge Operation (First-Time Activation)

You will be prompted to change your password when you first attempt to exchange credentials using the Cognito `InitiateAuth` operation. To invoke this operation, you will need to provide the session token returned with the `InitateAuth` response along with your username, the CLIENT ID, and your new password.

Input:

HTTP Verb / Path	POST https://cognito-idp.us-east-1.amazonaws.com
HTTP headers (required)	X-Amz-Target: AWSCognitoIdentityProviderService.RespondToAuthChallenge Content-Type: application/x-amz-json-1.1

Request body	{
	"ChallengeName": "NEW_PASSWORD_REQUIRED",
	"ChallengeResponses": {
	"NEW_PASSWORD" : **"{{cognito-new-password}}",**
	"USERNAME" : **"{{cognito-username}}"**
	},
	"ClientId": **"{{cognito-client-id}}"**,
	"Session": "**{{session-token}}"**
	}

Password policy:

* Minimum eight characters

* 1+ special characters

* 1+ uppercase characters

* 1+ lowercase characters

Session token notes:

You obtain this value when logging in to Cognito for the first time via `InitiateAuth`.

InitiateAuth Operation (Credential Exchange)

Use this operation to exchange credentials for ID tokens after your initial user activation. To invoke this operation, you will need to provide your username and password along with CLIENT ID.

Input:

HTTP Verb / Path	POST https://cognito-idp.us-east-1.amazonaws.com
HTTP headers (required)	X-Amz-Target: AWSCognitoIdentityProviderService.InitiateAuth Content-Type: application/x-amz-json-1.1

Request body	{
	"AuthParameters" : {
	"USERNAME" : **"{{cognito-username}}"**,
	"PASSWORD" : **"{{cognito-password}}"**
	},
	"AuthFlow" : "USER_PASSWORD_AUTH",
	"ClientId" : **"{{cognito-client-id}}"**
	}

First-time Use (Password Change and User Activation)

You will be prompted to change your password after the first login. Follow these steps to successfully set up your new password:

1) Invoke the "initiate auth" Cognito operation, providing your username, password, and client ID. The service response will contain a "session" attribute which you must use in the next step. Copy this value and continue.

2) Invoke the "response to auth challenge" operation, providing your username, client id, and the new password you wish to use. The service will respond with an access token, id token, and a refresh token. You can use the id token to interact with system for up to 60 minutes.

Exchanging Credentials for an ID Token

You will use the Cognito `InitiateAuth` operation to exchange your credentials for ID tokens. See the `IntiateAuth` section above for more details.