



Payment API

Version 2.2.4

Confidential

TABLE OF CONTENTS

CHANGE REQUEST REFERENCE	4
DOCUMENT SUMMARY	4
DOCUMENT CHANGE HISTORY.....	4
OVERVIEW	6
DOCUMENT USAGE	6
GENERAL INFORMATION	6
<i>Amounts.....</i>	<i>6</i>
<i>Country Codes</i>	<i>6</i>
<i>Currency Codes.....</i>	<i>6</i>
<i>Distances</i>	<i>6</i>
<i>Dates/Times</i>	<i>6</i>
<i>Duplicate transaction checking</i>	<i>6</i>
API SUMMARY	7
HTTP RESPONSE CODES	7
AUTHENTICATION	8
INTRODUCTION	8
AUTHENTICATION	8
<i>Sample Header</i>	<i>8</i>
PAIRING TOKEN	9
CARD ON FILE TOKEN	9
SAFEGUARDING INFORMATION	9
PAYMENT API RESOURCES	10
I. AUTHORIZE RESOURCE	10
II. CAPTURE RESOURCE	10
III. TOKENIZE RESOURCE.....	10
I. AUTHORIZATION RESOURCE.....	11
SUMMARY	11
<i>Transaction Types</i>	<i>11</i>
<i>Additional/Optional Data.....</i>	<i>11</i>
<i>AVS/CVV2 Verification.....</i>	<i>11</i>
A. POST /PAIRING/:PAIRINGTOKEN/AUTHORIZE.....	12
<i>Pairing Token Authorize Resource URLs</i>	<i>12</i>
<i>Pairing Token Authorize Parameters.....</i>	<i>12</i>
<i>Pairing Token Authorize Response Codes.....</i>	<i>13</i>
<i>Sample Pairing Token Authorize Request (POST):.....</i>	<i>13</i>
B. POST /PAIRING/:PAIRINGTOKEN/PREAUTHORIZE	15
<i>Pairing Token Preauthorize Resource URLs.....</i>	<i>15</i>
<i>Pairing Token Preauthorize Parameters</i>	<i>15</i>
<i>Pairing Token Preauthorize Response Codes.....</i>	<i>16</i>
<i>Sample Pairing Token Preauthorize Request (POST):</i>	<i>16</i>
II. CAPTURE RESOURCE	18

SUMMARY	18
<i>Additional/Optional Data</i>	18
A. POST /PAIRING/:PAIRINGTOKEN/CAPTURE.....	19
<i>Pairing Token Capture Resource URLs</i>	19
<i>Pairing Token Capture Parameters</i>	19
<i>Pairing Token Capture Response Codes</i>	20
<i>Sample Pairing Token Capture Request (POST):</i>	20
III. TOKENIZE RESOURCE	21
SUMMARY	21
A. POST /TOKENIZE.....	22
<i>Create Token Resource URLs</i>	22
<i>Create Token Parameters</i>	22
<i>Create Token Response Codes</i>	22
<i>Sample Create Token Request (POST):</i>	23
B. DELETE TOKENIZE/:CARDTOKEN	24
<i>Delete Token Resource URLs</i>	24
<i>Delete Token Parameters</i>	24
<i>Delete Token Response Codes</i>	24
<i>Sample Delete Token Request (DELETE):</i>	24

Change Request Reference

Change Request (CR) Number(s):	
CR Submitter(s):	
Date of CR Submission to:	

Document Summary

Document Title:	CMT – Payment API
Owner:	Creative Mobile Technologies
Status:	(check one box) <input type="checkbox"/> DRAFT <input checked="" type="checkbox"/> Approved
Template Version:	1.1

Document Change History

Date of Change	Version	Reason for Change	Summary of Change	Build #	Author
Jun, 10, 2011	1.0		Payment API created	1.0	J. Backof
June 20, 2011	1.1		Added additional Level 2/3 Fields		J. Backof
August 2, 2011	1.2		Updated encryption algorithms, enc. Token and format for expiration date		J. Backof
August 5, 2011	1.3		Removed Error Code 9 From all transaction types		J. Backof
September 5, 2011	1.4		Added Transaction Type “M” – Offline Sale to authorizations. Also added Authorization Code for transaction type “M” Updated user_id to employee_id and made it a String value to allow for more flexible employee numbers.		J. Backof
November 20, 2011	1.5		Added decline scenarios.		J. Backof
April 3, 2012	2.0.0		Updated services schema to accommodate for level 2 and 3 data capture. Renamed methods to reduce confusion. Added optional service data for linkage to CMT's FleetNet Portal.		J. Golden/ J. Backof

September 17, 2012	2.0.1		Added more descriptive text around REST Calls		J. Backof
October 19, 2012	2.0.2		Included additional fields for CabConnect card processing. Included Test card data.		J. Golden
December 5, 2012	2.0.3		Correct URLs and updated OAuth section		J. Golden
February 14, 2013	2.0.4		Included truncated account number and expiration date for auth and capture responses		J. Golden
April 4, 2013	2.0.5		Included CVV and Zip Code for Token request as well as additional response codes		J. Golden
August 13, 2013	2.1.3		Streamline error codes and format, remove COF token update, remove Device Capture, added section headers		S. Selikoff
August 28, 2013	2.1.5		Added Pairing Preauthorization and Pairing Capture		S. Selikoff
September 11, 2013	2.1.6		Updated examples from XML to JSON		S. Selikoff
November 11, 2013	2.2.4		Correct JSON sample messages		S. Selikoff

Overview

Creative Mobile Technologies has developed a RESTful API to allow developers to programmatically interface with our payment processing infrastructure. All resources use JSON payloads over secure http (port 443) and are currently available to authorized CMT customers with valid merchant accounts.

Document Usage

This document was designed to be used by developers to wish to integrate with CMT's payment processing infrastructure. A basic understanding of RESTful services, JSON and communications over HTTP are required.

General Information

Amounts

Amounts do not include cents nor thousands delimiters. For a \$1,017.65 amount, 101765 is sent and the last 2 digits are assumed to be cents. All amounts are assumed to be in the currency code specified in the request. If a currency code is not given, USD is assumed to be the currency.

Country Codes

The country codes used are from the standard [ISO 3166-1 alpha-2 specification](#). Please see [Appendix B](#) for supported countries or contact CMT to ensure the country is supported.

Currency Codes

The currency codes used are from the standard [ISO 4217 Currency Codes](#). Please see [Appendix B](#) for supported currencies or contact CMT to ensure the currency is supported.

Distances

All distances are in kilometers unless otherwise specified.

Dates/Times

All optional dates should be passed in as the [ISO 8601](#) date format, YYYY-MM-DDTHH:mm:ss+ UTC Offset. (eg. 2012-07-16T13:24:00+0000 is 1:24 pm on July 16, 2012 UTC).

Duplicate transaction checking

CMT supports checking for duplicate transactions. When configured, CMT will validate that the combination of transaction type, customer reference number, transaction amount and last 4 of the credit card number are not repeated within the last 5 minutes. If repeated transaction(s) are sent, the transactions that are deemed duplicate will not be sent onto the bank. NOTE: The new transactions will only be checked against transactions that were approved or partially approved.

API Summary

CMT's Payment API is a RESTful service providing developers access to process credit, debit and private label cards through CMT's payment processing infrastructure.

CMT supports the following card types:

- Major Credit Cards (Visa, MasterCard, Discover, JCB, Diners, American Express)
- Pin-less Debit
- Certain Private Label Cards (contact CMT)
- Certain University Cards (contact CMT)

This API provides methods to authorize cards by passing track II swipe data (for programmatic interfaces with card readers) or account numbers and expiry dates as in the case of manual or card not present environments.

HTTP Response Codes

The following table lists the possible http response codes returned by the CMT Platform API and their corresponding description.

Code	Description
200	Request processed successfully. Check response codes for additional status indicators.
400	Bad request. Includes validation errors, bad JSON.
401	Unauthorized. The response code is returned if: <ul style="list-style-type: none"> - OAuth authentication failure - Session timeout (if session is supported)
403	Forbidden. Customers will receive this response code if they try to access a method without proper authorization.
500	Server Error. Retry request or contact CMT.

Authentication

Introduction

Each request to the CMT Payment API requires authentication. CMT employs a signature authentication strategy based on oauth 1.0a.

Authentication

To authenticate to the CMT Payment API, implement or download a client based on oauth 1.0a. More information on oauth can be found at <http://oauth.net> and for information or third party resources and libraries, visit:

<http://oauth.net>
<http://oauth.net/code/>
<http://hueniverse.com/oauth/guide/authentication/>

Once your oauth client is in place, you will be provided an oauth consumer key and oauth consumer secret key. Please do not share your secret key with anyone and obfuscate any reference to this key in your libraries.

Field	Description
OAuth Consumer Key	Unique key which grants developers access to specific resources and fleets.
OAuth Consumer Secret Key	Private key issued to developers which is used in the hashing algorithm. See below.

Sample Header

Below is a sample header request for a Payment API oauth request.

```
request: merchants/97b8ef25bc064bbb9eaf0e75858f5c73/authorize
Accept:[application/json]
Authorization:[OAuth oauth_consumer_key="api-consumer-1", oauth_nonce="-
2277426177509978136", oauth_signature="5PICyq0XvXbwS2Fv1AvMetaTxM0%3D",
oauth_signature_method="HMAC-SHA1", oauth_timestamp="1352825875",
oauth_version="1.0"]
Content-Type:[application/json]
```


Pairing Token

The previously established pairing token referencing the unique trip and device created for a single pairing session.

Card on File Token

The previously establishing pairing token for an existing card on file.

Safeguarding Information

In order to ensure the security and safety of cardholder data, CMT strongly suggests storing API keys, secret keys and merchant tokens encrypted. Never share these keys/tokens with anyone.

Payment API Resources

I. Authorize Resource

The **Authorize** resource provides an interface to authorize credit card sales, preauthorizations and credits. The **Authorize** resource takes as input either track II data or an account number/expiration date. Please see the field list below for a detailed description of the input parameters including types, length and if the field is optional.

II. Capture Resource

For preauthorized transactions, it is necessary to perform a delayed capture. The **Capture** resource provides this interface. Authorized transactions will return a transaction ID which must be used when calling the **Capture** resource. If a capture is not sent, the preauthorization will expire after 3 days and the transaction will have to be re-authorized.

III. Tokenize Resource

The **Tokenize** resource provides developers the ability to create a token in place of cardholder data, update tokenized data and delete a token. The token does not use the cardholder data to create the token so there is no way to get the cardholder information with just the token alone.

I. Authorization Resource

Summary

Several transaction types are available for authorizations including sales, credits, preauthorizations, and voice authorizations depending on the business requirements.

Transaction Types

Sale

A Sale should be used if there is only one final transaction and the money should be collected immediately. Once CMT receives the transaction, it is immediately marked and will be settled on the next settlement cycle. No further action is necessary.

Preauthorization

Preauthorizations should be used to put a hold on money until services are rendered. Once rendered, a [Capture command](#) (see below) should be called to release the hold and settle the funds. Please note that the time between authorization and capture can impact interchange rates. Please contact CMT for more details and best practices.

Additional/Optional Data

CMT supports optional data fields to support Purchase Card Level 2 and Level 3 data information as well as service data for other CMT services (trip sheet collection and reporting).

Level 2/3 Data

CMT supports passing Purchasing Card Level 2 information (such as purchase order number, tax amount, and postal codes, etc...). CMT Also supports Level 3 data and line item records which include line item details for purchases. See the description below as not card types support all fields.

AVS/CVV2 Verification

CMT supports address and card verification services. When configured, CMT will validate AVS and/or CVV2 data if passed. Please contact CMT to activate this service.

Resource	Description
A. POST /pairing/:pairingToken/authorize	Provides access to create authorizations using a previously established pairing token.
B. POST /pairing/:pairingToken/preauthorize	Provides access to create preauthorizations using a previously established pairing token.

A. POST /pairing/:pairingToken/authorize

This resource is designed to authorize credit cards using a matching mobile pairing token. This resource does not take a transaction type. The resource supports transactions of type: authorization.

Pairing Token Authorize Resource URLs

Production: <https://payment.cmtapi.com/v2/pairing/:pairingToken/authorize>

Sandbox: <https://payment-sandbox.cmtapi.com/v2/pairing/:pairingToken/authorize>

Pairing Token Authorize Parameters

pairingToken (required) - See [Pairing Token](#)

If the event that the response sub-code is HTTP 400 indicating an error, then only two fields, responseCode and responseMessage, will be returned.

URL Parameters

Field	DataType	Size	Required	Notes
:pairingToken	String	100	x	Unique pairing token received during the pairing process.

Request Field Description

Field	DataType	Size	Required	Notes
totalInCents	Integer		x	Total amount to be authorized. This amount will be paid to the operator/driver/fleet. Must match total amount reported in trip, including tip.
encryptionKeyVersion	Integer			Version of the key used in the encryption
encryptionToken	String	50		Send "CMT_PAYNET" encrypted (if encryption is not 0)
encryptionAlgorithm	Integer			One of the following 2 integer values: 0 : None 1 : 3DES
customerReferenceNumber	String	50		
ehailServiceFeeInCents	Integer		x	The service fee charged by the e-hail provider. Set to 0 if you are not providing it to us.
currencyCode	String	5		Currency code for the transaction. Options are USD (default), GBP, EUR, and CAD.
accountNumber	String	200	x (if cardOnFileToken is not set)	The primary account number for authorization
expiryDate	String	4	x (if cardOnFileToken is not set)	The expiry date in YYMM format
cvv2	String	5		Card verification value
zipCode	String	9		Either the 5 or 9 digit representation
cardOnFileToken	String		x (if accountNumber is not set)	Card on file token

Response Field Description

Field	DataType	Size	Required	Notes
tripId	Integer		x	The trip associated with the authorization
totalInCents	Integer		x	Total amount of the authorization
authorizationCode	String	8	x	The authorization code
lastFour	String	4	x	The last 4 digits of the credit card
cardType	String	10	x	The card type used in the transaction
transactionId	Long			Unique identifier used to reference the authorization
responseCode	Integer		x	See response codes in next tables.
responseMessage	String	100	x	Additional message on response

Pairing Token Authorize Response Codes

Success Sub-codes (HTTP 200)

Code	Description
1	Approved
2	Partial Approval

Error Sub-codes (HTTP 400)

Code	Description
601	Invalid fields: [See message for more details]
602	Invalid pairing token
603	Total amount does not match recorded amount for trip
604	Invalid CVV2
605	Invalid AVS
606	Unable to decrypt data
607	Declined
608	Error processing request

Sample Pairing Token Authorize Request (POST):

URL: <https://payment.cmtapi.com/v2/pairing/dg760hgsF3gfe67873DfgghJ/authorize>

```
{
  "totalInCents": 1084,
  "ehailServiceFeeInCents": 0,
  "accountNumber": "8888888888888888",
  "expiryDate": "1113",
  "cvv2": "123"
}
```

Sample Successful Pairing Token Authorize Response (HTTP 200):

```
{
  "totalInCents": 1084,
  "responseCode": 1,
  "responseMessage": "Approved",
  "tripId": 123,
  "authorizationCode": "AR32564",
  "lastFour": "8888",
  "cardType": "VISA",
  "transactionId": 12345678901234
}
```

Unsuccessful Pairing Token Authorize Response (HTTP 400):

```
{  
  "responseCode": 602,  
  "responseMessage": "Invalid pairing token"  
}
```

B. POST /pairing/:pairingToken/preauthorize

This resource is designed to preauthorize credit cards using a matching mobile pairing token. In the case of successfully processing, a preauthorization should be followed by a pairing capture covered later in this document.

The amount for the preauthorization does not need to match the total for the trip, as this service is commonly called early in the trip when the total is not known. The amount preauthorized should be greater than or equal to the *expected* amount of the trip.

If the final capture amount is less than or equal to the preauthorization for the trip, then the capture should be unconditionally approved. Alternatively, if the final capture amount is higher than the preauthorization for the trip, then the entire transaction may be declined at the time of capture.

Pairing Token Preauthorize Resource URLs

Production: <https://payment.cmtapi.com/v2/pairing/:pairingToken/preauthorize>

Sandbox: <https://payment-sandbox.cmtapi.com/v2/pairing/:pairingToken/preauthorize>

Pairing Token Preauthorize Parameters

pairingToken (required) - See [Pairing Token](#)

If the event that the response sub-code is HTTP 400 indicating an error, then only two fields, responseCode and responseMessage, will be returned.

URL Parameters

Field	DataType	Size	Required	Notes
:pairingToken	String	100	x	Unique pairing token received during the pairing process.

Request Field Description

Field	DataType	Size	Required	Notes
totalInCents	Integer		X	Total amount to be preauthorized. The should be greater than or equal to the expected final capture amount, although this is not required.
encryptionKeyVersion	Integer			Version of the key used in the encryption
encryptionToken	String	50		Send "CMT_PAYNET" encrypted (if encryption is not 0)
encryptionAlgorithm	Integer			One of the following 2 integer values: 0 : None 1 : 3DES
customerReferenceNumber	String	50		
ehailServiceFeeInCents	Integer		x	The service fee charged by the e-hail provider. Set to 0 if you are not providing it to us.
currencyCode	String	5		Currency code for the transaction. Options are USD (default), GBP, EUR, and CAD.
accountNumber	String	200	x (if cardOnFileToken is not set)	The primary account number for preauthorization

expiryDate	String	4	x (if cardOnFileToken is not set)	The expiry date in YYMM format
cvv2	String	5		Card verification value
zipCode	String	9		Either the 5 or 9 digit representation
cardOnFileToken	String		x (if accountNumber is not set)	Card on file token

Response Field Description

Field	DataType	Size	Required	Notes
tripId	Integer		x	The trip associated with the preauthorization
totalInCents	Integer		x	Total amount of the preauthorization
lastFour	String	4	x	The last 4 digits of the credit card
cardType	String	10	x	The card type used in the transaction
transactionId	Long			Unique identifier used to reference the authorization that should be used to later capture the authorization.
responseCode	Integer		x	See response codes in next tables.
responseMessage	String	100	x	Additional message on response

Pairing Token Preauthorize Response Codes

Success Sub-codes (HTTP 200)

Code	Description
1	Approved
2	Partial Approval

Error Sub-codes (HTTP 400)

Code	Description
601	Invalid fields: [See message for more details]
602	Invalid pairing token
604	Invalid CVV2
605	Invalid AVS
606	Unable to decrypt data
607	Declined
608	Error processing request

Sample Pairing Token Preauthorize Request (POST):

URL:

https://payment.cmtapi.com/v2/pairing/dg760hgsF3gfe67873DfgghJ/preauthorize

```
{
  "totalInCents": 10000,
  "ehailServiceFeeInCents": 0,
  "accountNumber": "8888888888888888",
  "expiryDate": "1010",
  "cvv2": "123"
}
```

Sample Successful Pairing Token Preauthorize Response (HTTP 200):

```
{
```



```
"totalInCents": 10000,  
"responseCode": 1,  
"responseMessage": "Approved",  
"tripId": 123,  
"lastFour": "8888",  
"cardType": "VISA",  
"transactionId": 12345678901234  
}
```

Unsuccessful Pairing Token Preauthorize Response (HTTP 400):

```
{  
  "responseCode": 602,  
  "responseMessage": "Invalid pairing token"  
}
```

II. Capture Resource

Summary

The ***capture*** method provides a programmatic interface to capture (settle) credit card preauthorizations. The ***capture*** method takes as input a **Capture Request** and returns a **Capture Response**. Please see the field list below for a detailed description of the input parameters including types, length and if the field is optional. Also note that when capturing a transaction, Purchase Card Level 2 and Level 3 data may optionally be passed.

Additional/Optional Data

CMT supports optional data fields to support Purchase Card Level 2 and Level 3 data information.

Level 2/3 Data

CMT supports passing Purchasing Card Level 2 information (such as purchase order number, tax amount, and postal codes, etc...). CMT Also supports Level 3 data and line item records which include line item details for purchases. See the description below as not card types support all fields.

Resource	Description
A. POST pairing/:pairingToken/capture	Provides access to capture preauthorizations using a previously established pairing token

A. POST /pairing/:pairingToken/capture

This resource is designed to capture credit cards using a matching mobile pairing token. The Pairing Token and Transaction Id must match the values used for the preauthorization. If the pairing token has changed, a new preauthorization must be established, or alternatively, a direct authorization must be used.

The amount for the capture must match the recorded amount for the trip using the pairing token information. If the final capture amount is less than or equal to the preauthorization for the trip, then the capture should be unconditionally approved. Alternatively, if the final capture amount is higher than the preauthorization for the trip, then the entire transaction may be declined at the time of capture.

Pairing Token Capture Resource URLs

Production: <https://payment.cmtapi.com/v2/pairing/:pairingToken/capture>

Sandbox: <https://payment-sandbox.cmtapi.com/v2/pairing/:pairingToken/capture>

Pairing Token Capture Parameters

pairingToken (required) - See [Pairing Token](#)

If the event that the response sub-code is HTTP 400 indicating an error, then only two fields, responseCode and responseMessage, will be returned.

URL Parameters

Field	DataType	Size	Required	Notes
:pairingToken	String	100	x	Unique pairing token received during the pairing process.

Request Field Description

Field	DataType	Size	Required	Notes
totalInCents	Integer		x	Total amount to be captured. This amount will be paid to the operator/driver/fleet. Must match total amount reported in trip, including tip.
transactionId	Long		x	Unique identifier used to reference the previously established preauthorization

Response Field Description

Field	DataType	Size	Required	Notes
tripId	Integer		x	The trip associated with the preauthorization
totalInCents	Integer		x	Total amount of the preauthorization
lastFour	String	4	x	The last 4 digits of the credit card
cardType	String	10	x	The card type used in the transaction
transactionId	Long			Unique identifier used to reference the authorization that should be used to later capture the authorization.
authorizationCode	String	8	x	The authorization code
responseCode	Integer		x	See response codes in next tables.
responseMessage	String	100	x	Additional message on response

Pairing Token Capture Response Codes

Success Sub-codes (HTTP 200)

Code	Description
1	Approved
2	Partial Approval

Error Sub-codes (HTTP 400)

Code	Description
601	Invalid fields: [See message for more details]
602	Invalid pairing token
603	Total amount does not match recorded amount for trip
604	Invalid CVV2
605	Invalid AVS
606	Unable to decrypt data
607	Declined
608	Error processing request

Sample Pairing Token Capture Request (POST):

URL: <https://payment.cmtapi.com/v2/pairing/dg760hgsF3gfe67873DfgghJ/capture>

```
{
  "totalInCents": 1084,
  "transactionId": 13212132312
}
```

Sample Successful Pairing Token Capture Response (HTTP 200):

```
{
  "totalInCents": 1084,
  "responseCode": 1,
  "responseMessage": "Approved",
  "tripId": 123,
  "authorizationCode": "AR32564",
  "lastFour": "8888",
  "cardType": "VISA",
  "transactionId": 12345678901234
}
```

Unsuccessful Pairing Token Capture Response (HTTP 400):

```
{
  "responseCode": 602,
  "responseMessage": "Invalid pairing token"
}
```

III. Tokenize Resource

Summary

The Tokenization Resources provide programmatic access to create and delete credit card tokens. Please note that this functionality is not a profile storage for customer data.

Supported cards are American Express, Mastercard, Visa, Discover, Diner's Club, and JCB.

Resource	Description
A. POST /tokenize	Provides access to tokenize credit card data.
B. DELETE tokenize/:cardToken	Provides access to delete tokenized credit card data.

A. POST /tokenize

This method is designed to tokenize credit card data.

Create Token Resource URLs

Production: <https://payment.cmtapi.com/v2/tokenize>

Sandbox: <https://payment-sandbox.cmtapi.com/v2/tokenize>

Create Token Parameters

None

If the event that the response sub-code is HTTP 400 indicating an error, then only two fields, responseCode and responseMessage, will be returned.

Request Field Description

Field	DataType	Size	Required	Notes
accountNumber	String	200	x	Card account number
expiryDate	String	4	x	Card expiration date in YYMM format
validateAccountInformation	Boolean			If true, a preauthorization is sent to verify the cardholder data to protect against fraud. The default is false.
cvv	String	8		Card verification value
zipCode	String	9		Either the 5 or 9 digit representation

Response Field Description

Field	DataType	Size	Required	Notes
responseCode	Integer		x	See response codes in next tables.
responseMessage	String	100	x	Message description
cardType	String	16	x	One of the following six values: AMERICAN_EXPRESS, VISA, JCB, DISCOVER, MASTERCARD, DINERS_CLUB
lastFour	String	4	x	Last four digits of the tokenized credit card number
cardOnFileToken	String	100	x	Token to be used to call the service to use the saved card information

Create Token Response Codes

Success Sub-codes (HTTP 200)

Code	Description
1	Success

Error Sub-codes (HTTP 400)

Code	Description
601	Invalid fields: [See message for more details]
602	Invalid card type or card type not supported
603	Invalid data
604	Invalid CVV2
605	Invalid AVS
606	Invalid CVV2 and AVS

CMT – Payment API	
-------------------	--

607	Declined if the validateAccountInformation flag was set to true and card was not able to be authorized.
608	Error processing request

Sample Create Token Request (POST):

URL: *https://payment.cmtapi.com/v2/tokenize*

```
{
  "accountNumber": "8888888888888888",
  "expiryDate": "1504",
  "validateAccountInformation": false,
  "cvv2": "123",
  "zipCode": "11111"
}
```

Sample Successful Create Token Response (HTTP 200):

```
{
  "responseCode": 1,
  "responseMessage": "Success",
  "cardType": "VISA",
  "lastFour": "8888",
  "cardOnFileToken": "da43fdfsdfsdfsdfs"
}
```

Unsuccessful Create Token Response (HTTP 400):

```
{
  "responseCode": 601,
  "responseMessage": "Invalid card type or card type not supported"
}
```

B. DELETE tokenize/:cardToken

This method is designed to delete tokenized credit card data.

Delete Token Resource URLs

Production: <https://payment.cmtapi.com/v2/tokenize/:cardToken>

Sandbox: <https://payment-sandbox.cmtapi.com/v2/tokenize/:cardToken>

Delete Token Parameters

cardToken (required) - See [Card on File Token](#)

If the event that the response sub-code is HTTP 400 indicating an error, then only two fields, responseCode and responseMessage, will be returned.

Response Field Description

Field	DataType	Size	Required	Notes
responseCode	Integer		x	See response codes in next tables.
responseMessage	String	100	x	Message description

Delete Token Response Codes

Success Sub-codes (HTTP 200)

Code	Description
1	Success

Error Sub-codes (HTTP 400)

Code	Description
601	Invalid fields: [See message for more details]
603	Invalid data
608	Error processing request

Sample Delete Token Request (DELETE):

URL: <https://payment.cmtapi.com/v2/tokenize/da43fdfsdfsdfs/>

Sample Successful Delete Token Response (HTTP 200):

```
{
  "responseCode": 1,
  "responseMessage": "Success"
}
```

Unsuccessful Delete Token Response (HTTP 400):

```
{
  "responseCode": 608,
  "responseMessage": "Error processing request"
}
```


}