CMT TPEP 2.0 Mobile API

v1.0.13

TABLE OF CONTENTS

Change Request Reference

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

Document Summary

Document Title:	CMT TPEP 2.0 Mobile API
Owner:	Creative Mobile Technologies
Status:	(*check one box*) ■ DRAFT ☐Approved
Template Version:	1.0

Document Change History

Date of Change	Version	Reason for Change	Summary of Change	Author
January 15, 2013	1.0.1	Initial Draft		J. Backof/ M.Dinowitz
February 15, 2013	1.0.2	TLC Feedback	Added additional pairing method. Updates to trip end fields.	J. Backof/ M.Dinowitz
April 24, 2013	1.0.3	Additional payment fields	Added additional fields for authorization request.	J. Backof
May 30, 2013	1.0.5	Pairing Token Updates and streamlined payment workflow	Added PUT call to update pairing token Updated Payment Workflow so payments can be called asynchronously.	J. Backof
June 14, 2013	1.0.6	Summary Update	RideLinQ Images, pairing update, auto-pay	J. Backof / M. Dinowitz
July 1, 2013	1.0.7	Updated error handling	Updated error codes for all methods; Added driver/medallion confirmation method.	M. Dalen
July 16, 2013	1.0.8	Modified pairing authorization resource	Removed fields from pairing authorization resource; updated trip callback message	M. Dalen
July 24, 2013	1.0.9	Updated error fields and removed tokenization update	Clarified error response fields; added total to trip callback message; removed tokenization update method; updated tokenization and payment authorization error codes	M. Dalen
Aug. 8, 2013	1.0.10	Invalid driver, Trip resource	Added error code 107 for an invalid driver to external pairing resource; Made tripdata call its own resource.	M. Dalen
Aug. 26, 2013	1.0.11	Pairing failure callback	Added callback to notify of a pairing failure. Added error code to trip resource to notify of a pairing failure.	M. Dalen
Oct. 7, 2013	1.0.12	Authorization disabled for non-autopay	Disabled authorization when autoCompletePayment flag is set to false.	M. Dalen
Dec. 6, 2013	1.0.13	Pairing/Unpair/ Update acknowledgement	Added the ability for the vehicle to acknowledge a pairing, unpair, and update request.	M. Dalen

General Information

Overview
CMT's TPEP 2.0 Mobile API provides (T2MAPI) programmatic access to CMT's Mobile Middleware services and offers an integration point with in-vehicle equipment.

Audience
The audience for this document is typically an application developer familiar with RESTful services and JSON. Developers should also be familiar with OAuth and/or general request signing techniques.

Getting Started with the API
To get started with the API, a developer must obtain credentials from CMT for authentication. Once authenticated, developers can access resources using standard RESTFul calls.

The T2MAPI was designed to integrate either directly with mobile devices, or server to server in the case where developers wish to proxy requests. CMT requires all communication between app and host be encrypted using standard protocol encryption techniques.

Dates/Times
All dates passed to the API must be formatted according to the ISO 8601 date format standard, using the mask YYYY-MM-DDTHH:mm:ss+ UTC Offset. (eg. 2012-07-16T13:24:00+0000 is 1:24 pm on July 16, 2012 UTC).

Currency Codes
The currency codes used are from the standard ISO 4217 Currency Codes.

Amounts
All currency amounts are represented in cents. For example, $1,017.65 is represented as 101765. 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.

Integers
Integers are assumed to be a non-negative number unless otherwise stated.

Transport

All requests and responses are formatted using JSON. All communications are over SSL (HTTPS).

HTTP Response Codes

The following table lists the possible HTTP response codes returned by the CMT TPEP 2.0 Mobile API and their corresponding description.

Code	Description
200	Request processed successfully.
201	Creation request processed successfully. 201 is typically returned on a POST request to create a resource.
400	Message format or validation exception. This is typically returned when the message format is incorrect.
401	Unauthorized. The response code is returned if there is an authentication failure.
403	Forbidden. The response code returned when trying to access a resource without proper authorization.
500	Server Error. Retry request.

Additionally, many resources may return sub-codes describing additional status information. In these cases, the response will follow this format:

```
{
	"responseCode":101,
	"message":"The medallion is invalid"
}
```

Please see individual resource descriptions for more detail.

Authentication and Authorization

Introduction

Each request to the T2MAPI requires authentication. CMT employs a signature authentication strategy based on OAuth 1.0a.

Authentication

To authenticate to T2MAPI, implement or download a client based on OAuth 1.0a. More information on OAuth can be found at http://oauth.net. 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.

Sample Header

Below is a sample header request for a T2MAPI OAuth request.

```
request: /pairing
Accept:[application/json]
Authorization:[OAuth oauth_consumer_key="api-consumer-1", oauth_nonce="-
2277426177509978136", oauth_signature="5PICyq0XvXbwS2FvlAvMetaTxM0%3D",
oauth_signature_method="HMAC-SHA1", oauth_timestamp="1352825875",
oauth_version="1.0"]
Content-Type:[application/json]
```

Authorization

Once a developer is properly authenticated to the CMT T2MAPI, their credentials will be authorized for each resource request. If a developer is unauthorized to access a particular resource, an HTTP status of 403 (Forbidden) will be returned.

API Overview

The following section describes the overall process to integrate with in-vehicle equipment, receive end of trip data and provide payment information to complete the trip.

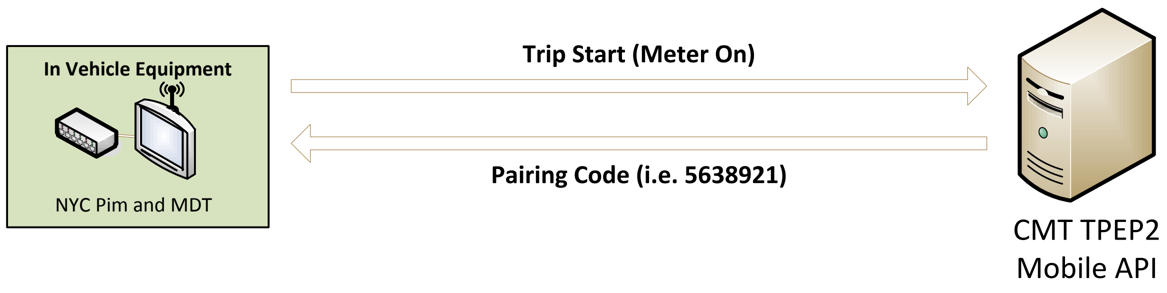
Pairing Summary

Pairing allows the e-hail application provider to associate their customer to a specific vehicle, for a specific trip. Once paired, the application provider can leverage CMT's payment processing and trip data services. Pairing is a **required step** for application provider in-vehicle payment processing.

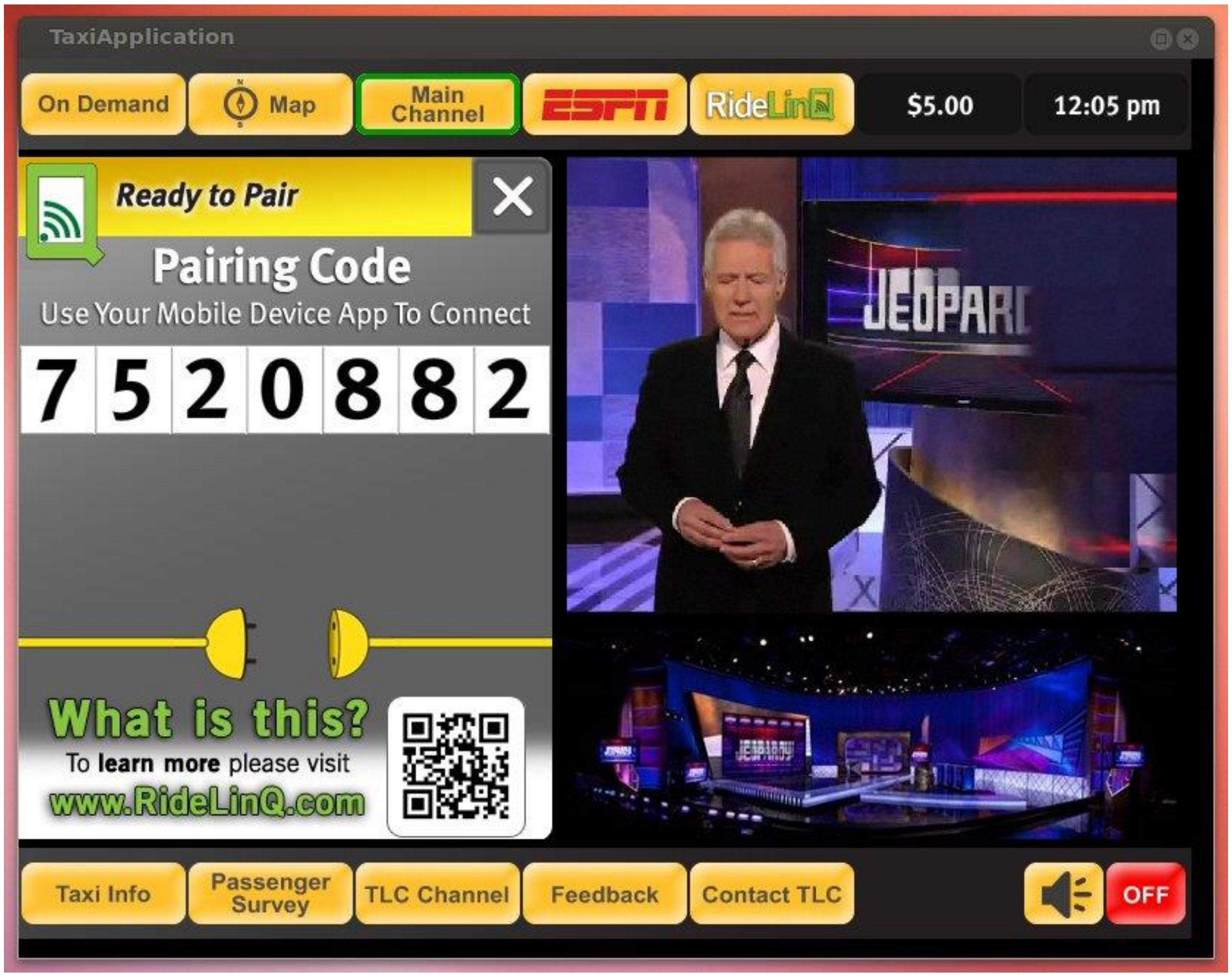
The Pairing Process:

1) Pairing Code sent to vehicle at trip-start

Shortly after the meter is hired, the vehicle sends a message to CMT indicating that the trip has started. In response, CMT sends a short numeric **'pairing code'** to the vehicle. This message is displayed to the passenger after the prologue has been completed. The per-trip pairing code is preferable to simply entering the vehicle's medallion number, as it will help reduce vehicle 'pair-spoofing'.



The pairing code is displayed to the passenger via the rear-seat display system as shown in the example on the next page:



2) E-Hail application submits pairing request to CMT

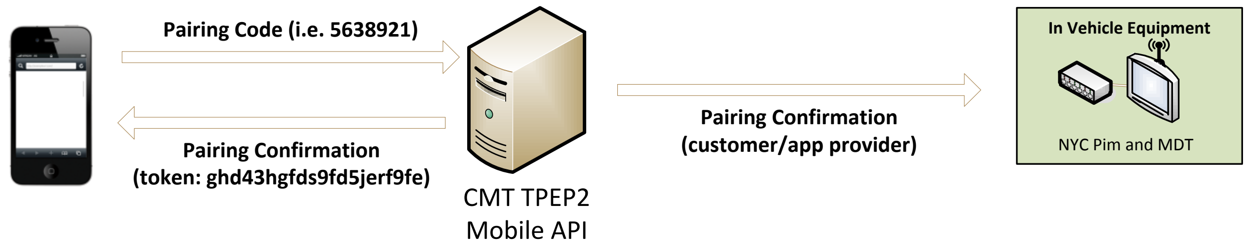
The passenger enters the pairing code into their E-Hail application and along with additional data from the E-Hail provider, the pairing request is sent to CMT. The pairing request contains customer defined tip data, and also includes an ‘Auto Complete Payment’ flag which is discussed in more detail later in this document.

3) CMT validates request and sends responses

When CMT receives the '**Pairing Request**' from the E-Hail application, the vendor credentials and pairing code are validated, and the appropriate success or error responses are sent to both the E-Hail application, and the vehicle associated with the pairing code. The response from a successful pairing request includes the medallion, trip identifier, driver id, and a globally unique pairing token.

NOTE: The unique 'Pairing Token' returned in the pairing request response is REQUIRED for all subsequent requests to the CMT TPEP2 MAPI. The Pairing Token is used in lieu of the original Pairing Code, as it ensures that the E-Hail application received the pairing request response successfully. Requests without valid Pairing Tokens will be rejected.

When the E-Hail application receives the response, the provided information may be used to indicate successful or unsuccessful pairing. CMT will also send the status of the pairing request to the vehicle.

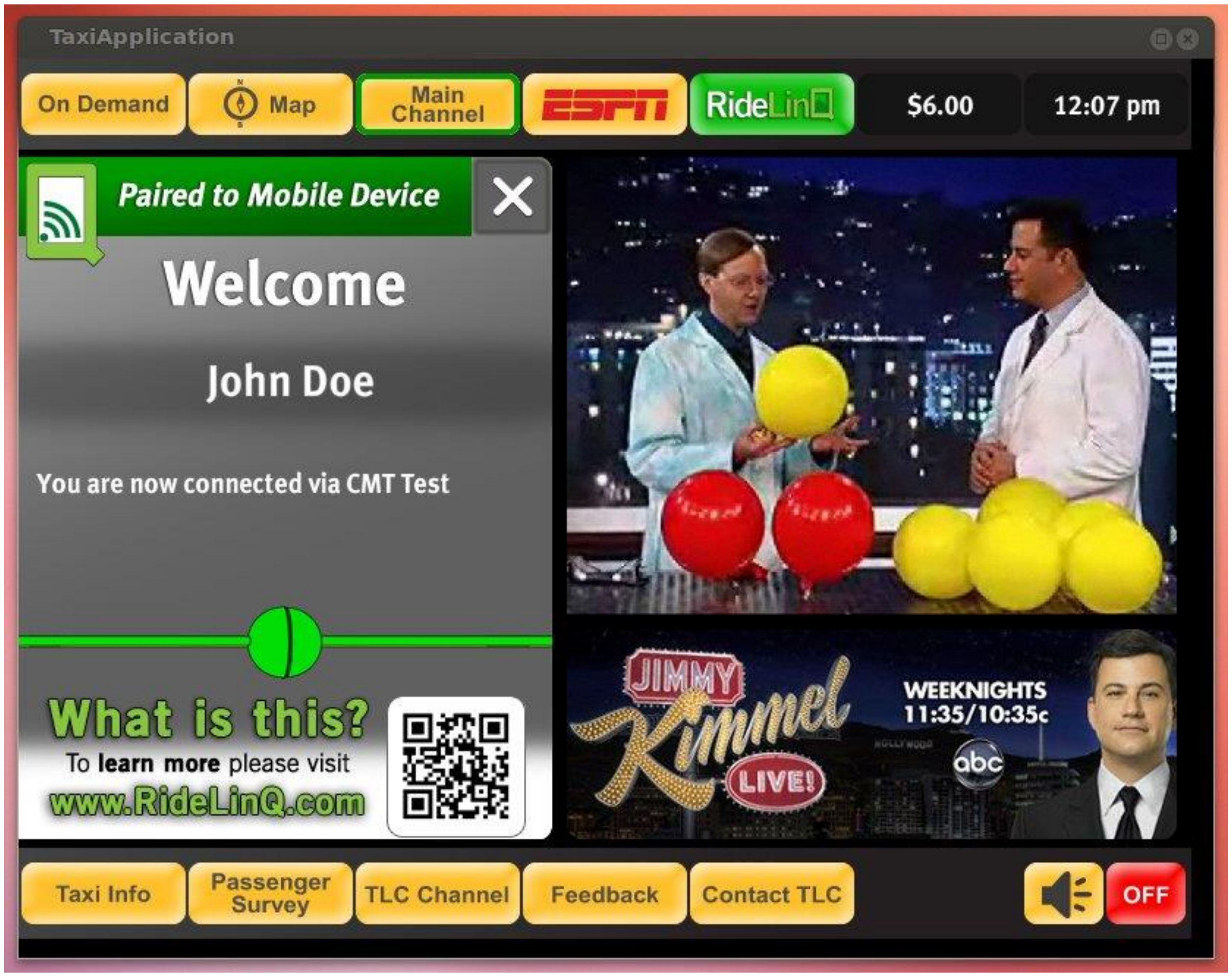


3A) Vehicle sends acknowledgement (Optional)

If the pairing initialization resource was used, the vehicle will send an acknowledgement message to the Mobile API server. This acknowledgement will be passed on to the E-Hail application through a callback or the trip resource.

4) Pairing request disposition displayed in vehicle and E-Hail application

In the vehicle, the rear-seat GUI will clearly indicate the final disposition of the pairing attempt. An example of a successful pairing is shown on the next page:



NOTE: CMT recommends that the E-hail application display a dialog to the effect of: "You are in vehicle <medallion>, driver id is <driverId>" which will be returned in the pairing response.

This will confirm the pairing for the customer, and provide the opportunity to unpair (see below) in the unlikely event of an incorrect pairing.

In the event of an error, the E-Hail customer may attempt to start the pairing process again by initiating the same request as described in Step 2. The original pairing code is valid for the entire duration of the trip. The customer must pair before the payment process in the vehicle begins.

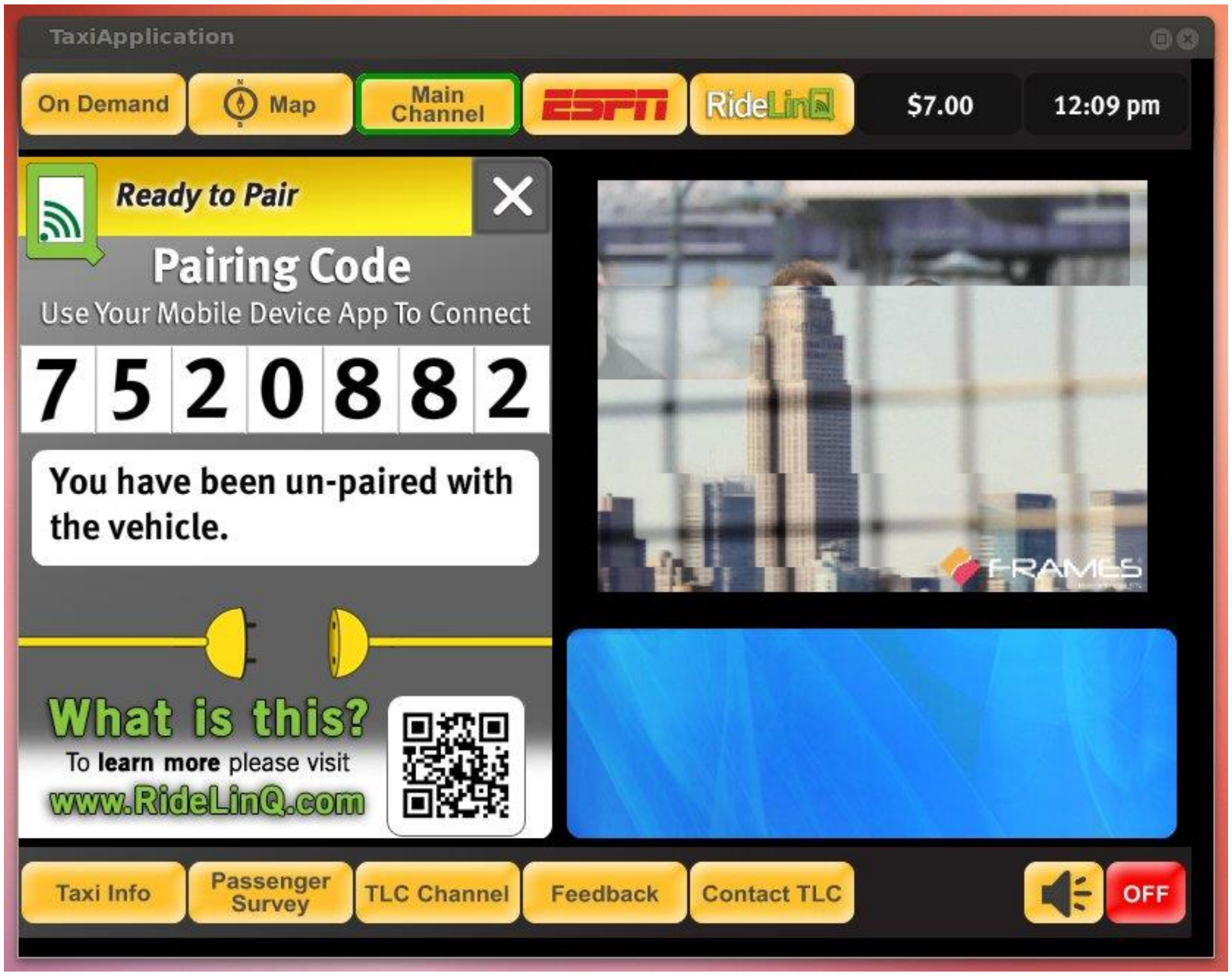
Alternative Pairing Method
In addition to in-vehicle pairing, which requires passenger action to complete the pairing process, CMT offers a method for E-Hail providers to manage the pairing through their own logic. Many E-Hail providers supply driver-side applications which allow for the confirmation of customer/vehicle pairing independent of CMT. In these cases, the E-Hail provider can send a 'host pairing' request that bypasses the use of the in-vehicle generated pairing code. This works similarly to in-vehicle pairing, except the vehicle medallion number is sent in lieu of the pairing code.

If the medallion identified in the host-pairing request is not already participating in a paired trip, CMT will 'trust' the E-Hail providers host-pairing request, and return a pairing token back to the E-Hail provider. As previously indicated, the pairing token is required for all subsequent requests made to the CMT TPEP2 MAPI. Additionally, CMT will send pairing disposition messages to the vehicle identified by the medallion in the request, as if the passenger had used the in-vehicle pairing code.

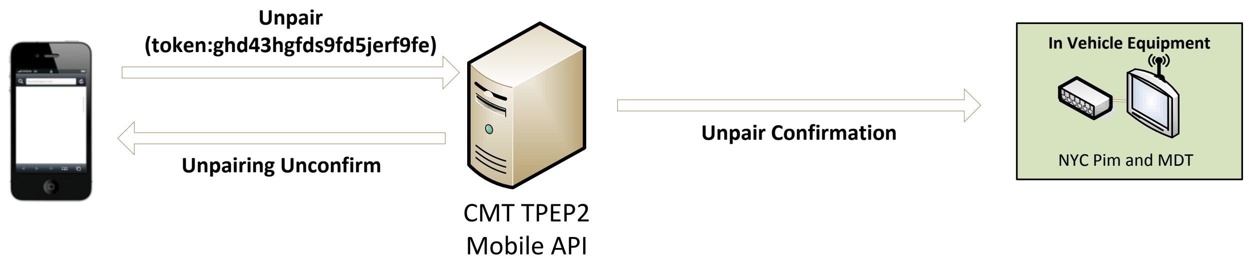
Un-pairing
During the course of a trip, a customer may decide to break their linkage with the CMT vehicle. This may occur due to an invalid pairing, or simply a customer initiated 'un-pairing'. In these cases, the E- Hail application may send an '**Un-pair Request**' to CMT. The CMT service will respond to the request with either a success or fail response.

If the unpairing initialization resource was used, the request will be acknowledged through either a callback or the trip resource.

Additionally, CMT will send the '**Un-pair Response**' to the vehicle to display to the customer on the rear-seat GUI as shown below.



You have been un-paired with the vehicle.

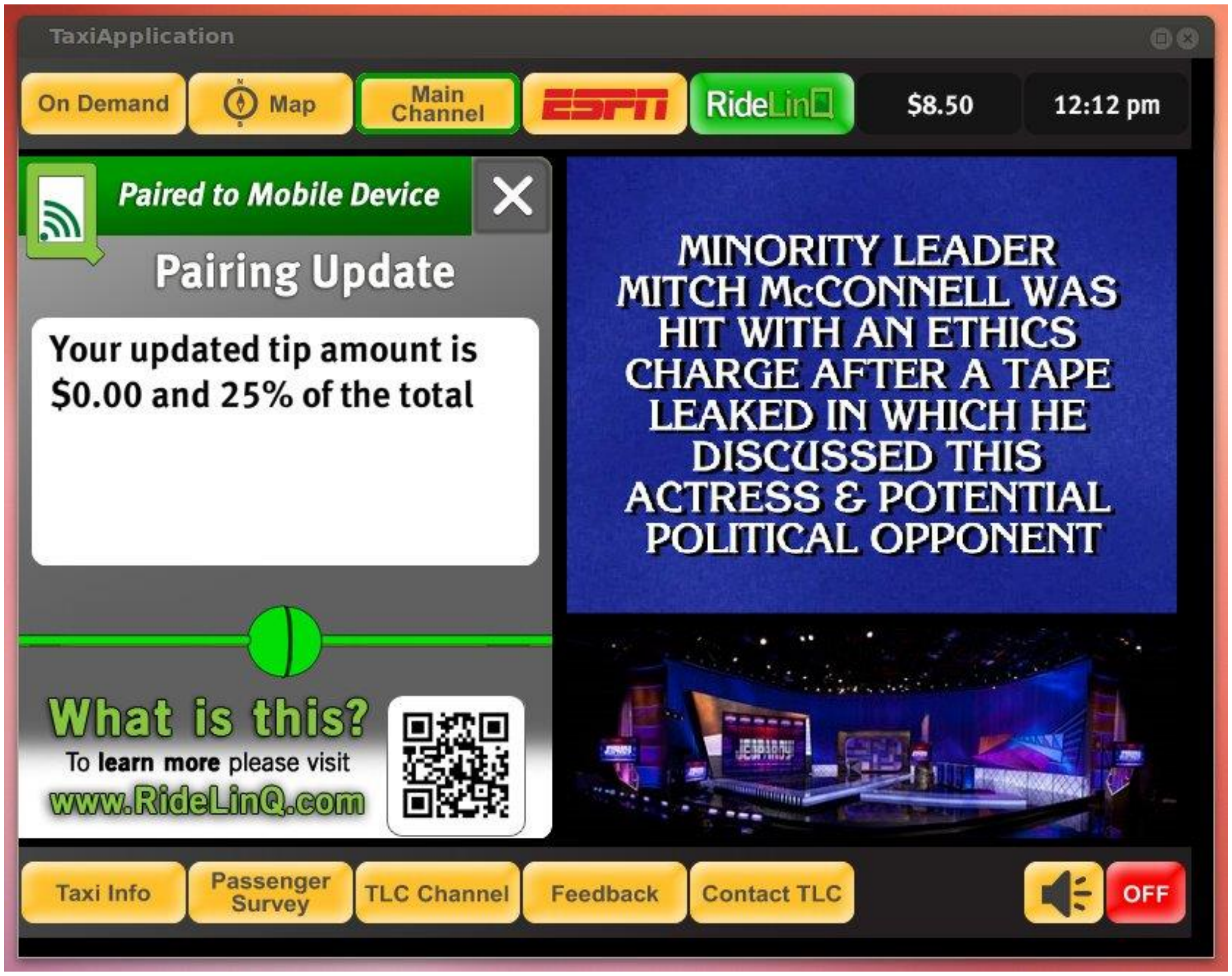


Even after 'un-pairing', the customer may initiate pairing again at any point before the payment process using the original pairing code for that trip.

NOTE: While the pairing code may be used multiple times during the same trip if the customer chooses to 'unpair', the resulting 'Pairing Token' returned by CMT from a 'Pairing Request' will always be unique.

Pairing Update

In the course of the trip, it is possible to change some of the values sent in the initial pairing request. If the E-Hail customer wishes to change default tip amounts specified, or change how the payment process will proceed at trip end, the 'Pairing Update' is employed. As with any changes to pairing state, this will be indicated in the vehicle as shown below.



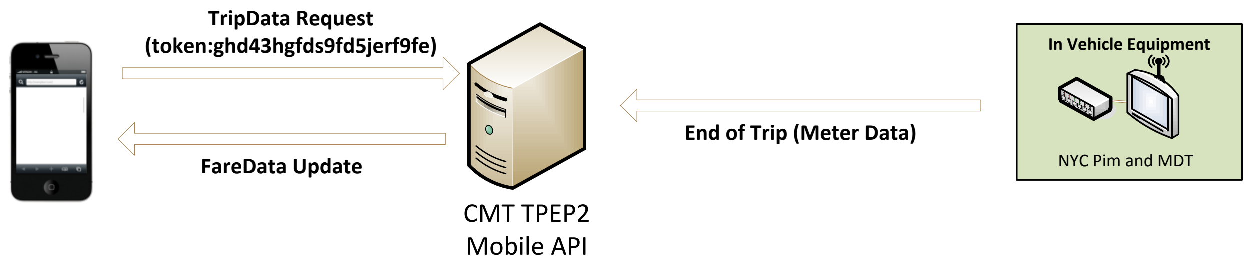
Your updated tip amount is $0.00 and 25% of the total
What is this?
To learn more please visit
www.RideLinQ.com
MINORITY LEADER MITCH McCONNELL WAS HIT WITH AN ETHICS CHARGE AFTER A TAPE LEAKED IN WHICH HE DISCUSSED THIS ACTRESS & POTENTIAL POLITICAL OPPONENT
Taxi Info
Passenger Survey
TLC Channel
Feedback
Contact TLC
OFF

TripData Overview

The trip resource is used to obtain trip related information regarding paired trips. The E-Hail application may call this resource at any time, but paired trip data will only be available when the payment process has begun or, in the case of a pairing initialization request, when the pairing is acknowledged. Generally, the payment process begins when the driver's meter makes the state transition from 'time-off' to 'meter-off'. Due to network latency, there may be some delay between the meter state transition and the availability of trip data when calling the trip resource.

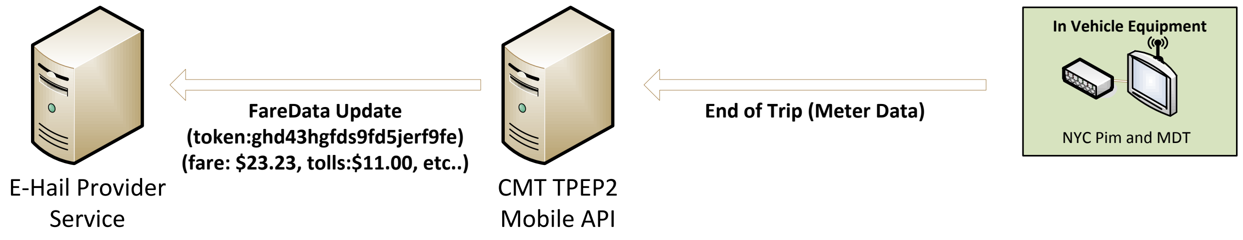
There are two mechanisms by which the E-Hail provider may receive trip data. The first method is a traditional polling mechanism where the E-Hail Provider makes periodic requests for the trip data. The second involves the E-Hail provider including a callback URL in the initial pairing request.

Mechanism 1. Polling Resource
When using the polling method, CMT recommends that the E-Hail application poll for trip data at least once every 5-10 seconds. In the event of an un-pairing during the trip, the E-Hail application trip polling thread or process should be terminated.



Mechanism 2. Callback Process
If the E-Hail Provider chooses to implement the callback feature, CMT will post a JSON message to the specified callback resource in the pairing request. CMT will retry the delivery of this message and the E-Hail provider must return an HTTP response code in order to ensure CMT executes its retry logic properly.

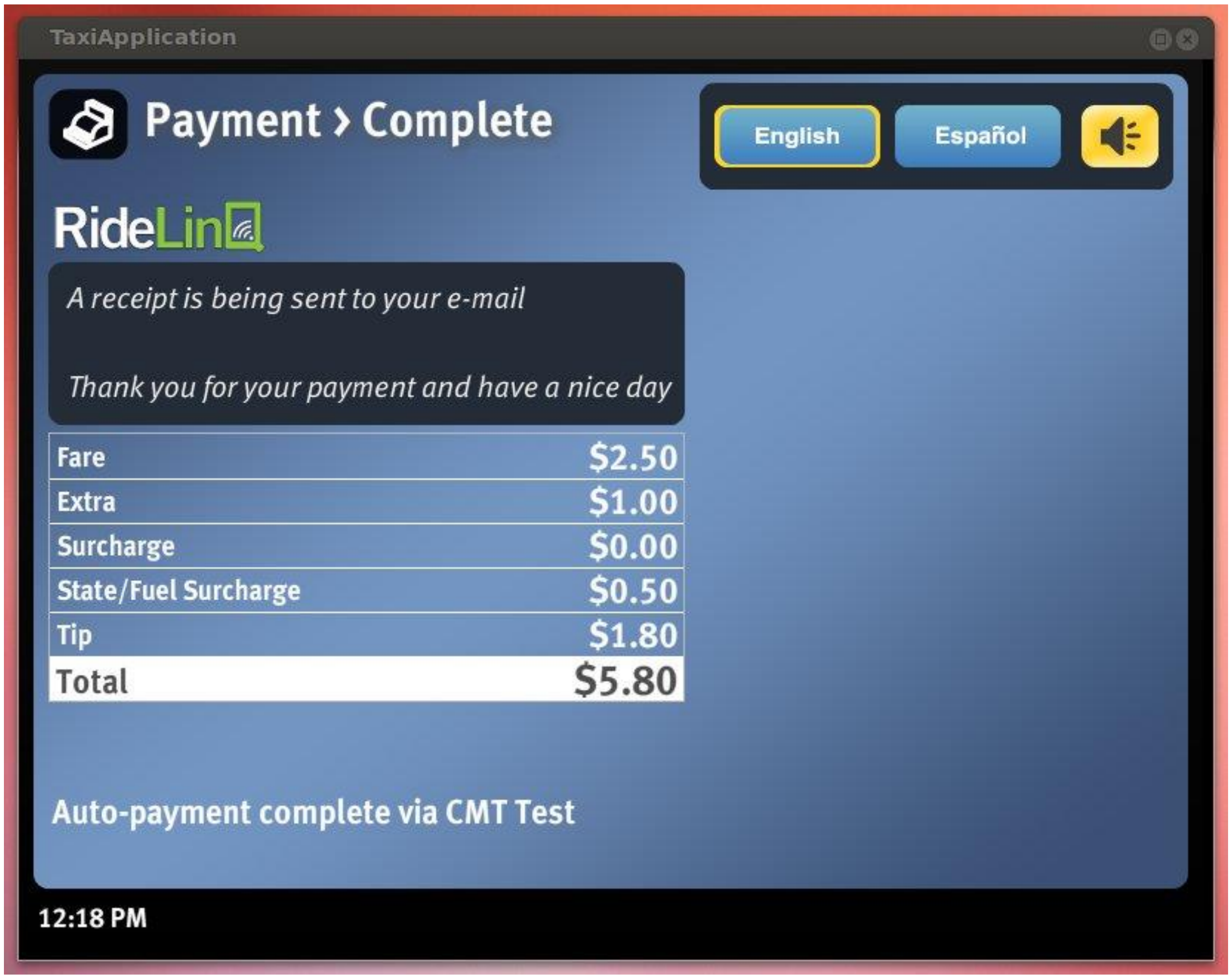
NOTE: E-Hail providers choosing to use the callback method must work with CMT's integration team before implementation.



Auto-Complete Payment

Previously, it was mentioned that when sending the 'Pairing Request' there is a flag that may be sent indicating whether or not use auto complete payment. When the meter goes off, there are two possible ways to complete the payment from an in-vehicle perspective.

(1) Asynchronous Authorization - If 'Auto-Complete Payment' is enabled in the 'Pairing Request' or 'Pairing Update', the customer is directed immediately to a final disposition screen, and the driver sees the final payment screen on the driver's mobile data terminal (MDT). From a ride perspective, the trip is now complete, and the customer may exit the vehicle. Behind the scenes, CMT will guarantee the delivery of the trip information to the E-Hail provider via the established callback URL or polling previously described. The E-Hail provider uses this information to create the authorization request to CMT. A sample of the final disposition screen when 'Auto-Complete Payment' is enabled is shown below:



A receipt is being sent to your e-mail

Thank you for your payment and have a nice day

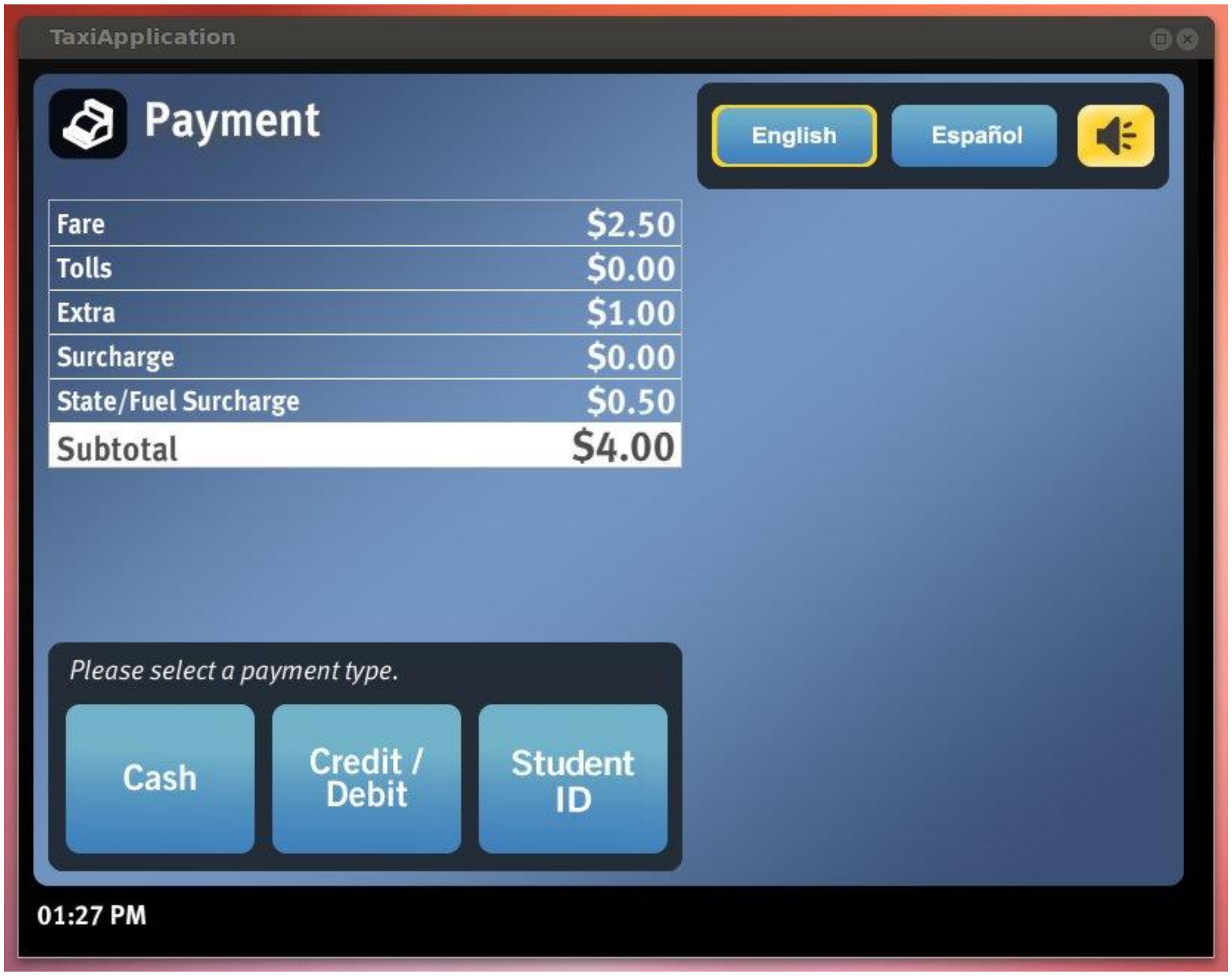
Fare	$2.50
Extra	$1.00
Surcharge	$0.00
State/Fuel Surcharge	$0.50
Tip	$1.80
Total	$5.80

Auto-payment complete via CMT Test

12:18 PM

NOTE: When 'Auto-Complete Payment' is enabled, the E-Hail provider is responsible to ensure payment to the driver.

(2) In-Vehicle Authorization – If 'Auto-Complete Payment' is not enabled, payment proceeds as usual (synchronously) in the vehicle. A sample of the initial payment screen is shown below:



Fare	$2.50
Tolls	$0.00
Extra	$1.00
Surcharge	$0.00
State/Fuel Surcharge	$0.50
Subtotal	$4.00

Please select a payment type.

Payment Overview

The payment resource allows certified TPEP 2.0 E-Hail applications to leverage CMT's PCI-DSS certified payment infrastructure to process payment card authorization requests. The CMT payment infrastructure has been used since the implementation of the original TPEP, and provides robust payment gateway services for thousands of New York City taxis. It conforms to the TLC's service level requirements for in-vehicle payment authorizations.

The E-Hail application constructs a payment request using the data obtained in the trip request.

The CMT payment processing gateway supports VISA, MasterCard, American Express, Discover, Diners, JCB, and certain private label cards. PIN/DEBIT transactions are not supported at this time.

NOTE: When constructing the request, total field must EXACTLY match the total amount sent from the trip response, or the request will be rejected. This is done to validate the amount being authorized, and ensure the integrity of the transaction.

Pairing Resources

The Pairing Resources provide developers access to pairing and un-pairing functionality as well as trip data integration and payment. Pairing is required in order to integrate with CMT's TPEP data collection and reporting backend.

Resource	Description
POST /pairing	Pairs the e-hail app to a CMT vehicle
POST /pairing/external	Pairs the e-hail app to a CMT vehicle via E-Hail Provider logic
DELETE /pairing/:pairingToken	Un-pairs the e-hail app from a CMT vehicle
PUT /pairing/:pairingToken	Updates current pairing data.
GET /pairing/:pairingToken/tripdata	Retrieves trip data for the paired vehicle/trip.
POST /pairing/:pairingToken/authorize	Processes payment for the paired vehicle/trip.
GET /confirm	Confirms validity of driver and medallion.

POST /pairing

Pairs an e-hail application to a vehicle for payment and trip collection and gets a pairing token (used in subsequent requests).

Resource URL

Production: https://mobile.cmtapi.com/v1/pairing

Sandbox: https://mobile-sandbox.cmtapi.com/v1/pairing

Request Field Description

Field	DataType	Size	Required	Notes
pairingCode	String	20	Y	Code to link with CMT
customerId	String	50	Y	Unique identifier for the mobile user (customer)
customerName	String	75	Y	Customer name for display
latitude	Double		Y	Latitude in degrees of device requesting the pairing token
longitude	Double		Y	Longitude in degrees of the device requesting the pairing token
callbackUrl	String	500	N	url used for callback features.
autoCompletePayment	Boolean		Y	Indicates if payment will occur automatically by the ehail provider (true) or if payment will be completed using the in-vehicle equipment (false).
autoTipPercentage	Integer		N*	Indicates the percentage tip automatically used. Valid ranges are 0 to 200. *autoTipPercentage and/or autoTipAmount is required if autoCompletePayment is enabled.
autoTipAmount	Double		N*	Indicates the amount tip automatically be used. Valid ranges are 0 to 200. *autoTipPercentage and/or autoTipAmount is required if autoCompletePayment is enabled.

Successful Response Field Description (HTTP 200)

Field	DataType	Size	Required	Notes
pairingToken	String	30	Y	Unique pairing token used in all subsequent requests.
pairingCode	String	10	Y	Returned pairing code.
medallion	String	20	Y	Medallion
tripId	Integer		Y	Trip Identifier
driverId	Integer		Y	Driver Identifier

Error Sub-codes (HTTP 400)

Code	Description
101	The pairing code is invalid. See message for specifics.
102	Missing or invalid customer ID. See message for specifics.
103	Pairing code already in use.
104	Invalid Customer Name. See message for specifics.
105	Invalid Latitude/Longitude. See message for specifics.
106	Invalid callback URL. See message for specifics.
444	Validation exception. See message for specifics.
446	Deserialization error.
443	Persistence error

Example Request (POST)

POST URL: *https://mobile.cmtapi.com/v1/pairing*

Request (JSON)
```
{
       "pairingCode":"1234567",
       "customerId":"3231432232",
       "customerName":"Test Customer",
       "latitude":78.342342,
       "longitude":-43.893454,
       "callbackUrl":"https://host.appcompany.com/faredata/",
       "autoCompletePayment":true,
       "autoTipPercentage":25,
       "autoTipAmount":0.0

}
```

Successful Response (JSON)
HTTP Status: **201 (Created)**
```
{
       "pairingToken":"dg76Jkil90hgsF3gfe67873DfgghJ",
       "pairingCode":"1234567",
       "medallion":"1N42",
       "tripId":34,
       "driverId":2342344
}
```

Unsuccessful Response (JSON)
HTTP Status: **400**
```
{
       “responseCode”:101,
       “message”:"The medallion is invalid"
}
```

POST /pairing/external

Pairs an e-hail application to a vehicle for payment and trip collection and gets a pairing token (used in subsequent requests). This method is only used when an E-Hail provider is performing the pairing function.

Resource URL

Production: https://mobile.cmtapi.com/v1/pairing/external

Sandbox: https://mobile-sandbox.cmtapi.com/v1/pairing/external

Request Field Description

Field	DataType	Size	Required	Notes
medallion	String	20	Y	Code to link with CMT
driverId	String	20	Y	Hack license or driver id used in the pairing
customerId	String	50	Y	Unique identifier for the mobile user (customer)
customerName	String	75	Y	Customer name for display
latitude	Double		Y	Latitude in degrees of device requesting the pairing token
longitude	Double		Y	Longitude in degrees of the device requesting the pairing token
callbackUrl	String	500	N	url used for callback features.
autoCompletePayment	Boolean		Y	Indicates if payment will occur automatically by the ehail provider (true) or if payment will be completed using the in-vehicle equipment (false).
autoTipPercentage	Integer		N*	Indicates the percentage tip automatically used. Valid ranges are 0 to 200. *autoTipPercentage and/or autoTipAmount is required if autoCompletePayment is enabled.
autoTipAmount	Double		N*	Indicates the amount tip automatically be used. Valid ranges are 0 to 200. *autoTipPercentage and/or autoTipAmount is required if autoCompletePayment is enabled.

Successful Response Field Description (HTTP 200)

Field	DataType	Size	Required	Notes
pairingToken	String	30	Y	Unique pairing token used in all subsequent requests.
pairingCode	String	10	Y	Returned pairing code.
medallion	String	20	Y	Medallion
tripId	Integer		Y	Trip Identifier
driverId	Integer		Y	Driver Identifier

Error Sub-codes (HTTP 400)

Code	Description
102	Missing or invalid customer ID. See message for specifics.
103	Unable to pair with this driver/vehicle
104	Invalid Customer Name. See message for specifics.
105	Invalid Latitude/Longitude. See message for specifics.
106	Invalid callback URL. See message for specifics.
107	Invalid driver. Driver specified is not in given vehicle.
444	Validation exception. See message for specifics.
446	Deserialization error.
443	Persistence error

Example Request (POST)

POST URL: *https://mobile.cmtapi.com/v1/pairing/external*

Request (JSON)
```
{
	"medallion":"1n45",
	"driverId":"1232323",
	"customerId":"3231432232",
	"customerName":"Test Customer",
	"latitude":78.342342,
	"longitude":-43.893454,
	"callbackUrl":"https://host.appcompany.com/faredata/",
	"autoCompletePayment":true,
	"autoTipPercentage":15,
	"autoTipAmount":3.75
}
```

Successful Response (JSON)
HTTP Status: **201 (Created)**
```
{
	"pairingToken":"dg76Jkil90hgsF3gfe67873DfgghJ",
	"pairingCode":"1234567",
	"medallion":"1N42",
	"tripId":34,
	"driverId":2342344
}
```

Unsuccessful Response (JSON)
HTTP Status: **400**
```
{
	“responseCode”:108,
	“message”:"The pairing code is invalid"
}
```

DELETE /pairing/:pairingToken

Un-pairs an e-hail application from a CMT vehicle.

Resource URL

Production: https://mobile.cmtapi.com/v1/pairing/:pairingToken

Sandbox: https://mobile-sandbox.cmtapi.com/v1/paring/:pairingToken

URL Parameters

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

Error Sub-codes (HTTP 400)

Code	Description
101	The pairing token is invalid
102	Missing or invalid customer id
207	Already un-paired
208	Invalid vendor for un-pairing
443	Persistence error

Example Request (DELETE)

DELETE URL: *https://mobile.cmtapi.com/v1/pairing/dg76Jkil90hgsF3gfe67873DfgghJ*

Successful Response (JSON)

```
HTTP Status: 200 (Success)
```

Unsuccessful Response (JSON)

```
HTTP Status: 400
{
	“responseCode”:101,
	“message”:"pairing token is invalid"
}
```

PUT /pairing/:pairingToken

Augments the current pairing with updated customer, callback and auto-pay fields.

Resource URL

Production: https://mobile.cmtapi.com/v1/pairing/dg76Jkil90hgsF3gfe67873DfgghJ

Sandbox: https://mobile-sandbox.cmtapi.com/v1/pairing/dg76Jkil90hgsF3gfe67873DfgghJ

Request Field Description

Field	DataType	Size	Required	Notes
customerId	String	50	Y	Unique identifier for the mobile user (customer)
customerName	String	75	Y	Customer name for display
latitude	Double		Y	Latitude in degrees of device requesting the pairing token
longitude	Double		Y	Longitude in degrees of the device requesting the pairing token
callbackUrl	String	500	N	url used for callback features.
autoCompletePayment	Boolean		Y	Indicates if payment will occur automatically by the ehail provider (true) or if payment will be completed using the in-vehicle equipment (false).
autoTipPercentage	Integer		N*	Indicates the percentage tip automatically used. Valid ranges are 0 to 200. *autoTipPercentage and/or autoTipAmount is required if autoCompletePayment is enabled.
autoTipAmount	Double		N*	Indicates the amount tip automatically be used. Valid ranges are 0 to 200. *autoTipPercentage and/or autoTipAmount is required if autoCompletePayment is enabled.

Error Sub-codes (HTTP 400)

Code	Description
102	Missing or invalid customer ID. See message for specifics.
103	Unable to locate pairing information
104	Invalid Customer Name. See message for specifics.
105	Invalid Latitude/Longitude. See message for specifics.
106	Invalid callback URL. See message for specifics.
444	Validation exception. See message for specifics.
446	Deserialization error.
443	Persistence error.

Example Request (POST)

PUT URL: *https://mobile.cmtapi.com/v1/pairing/dg76Jkil90hgsF3gfe67873DfgghJ*

Request (JSON)

```
{
       "customerId":"321232322",
       "customerName":"Updated Customer",
       "latitude":78.342342,
       "longitude":-43.893454,
       "callbackUrl":"https://host.appcompany.com/faredata/",
       "autoCompletePayment":true,
       "autoTipPercentage":0,
       "autoTipAmount":10.35

}
```

Successful Response (JSON)

HTTP Status: **200 (Success)**

Unsuccessful Response (JSON)

HTTP Status: **400**

```
{
	“responseCode”:101,
	“message”:"pairing token is invalid"
}
```

GET /confirm

Confirms existence of driver and medallion in CMT’s system.

Resource URL

Production: https://mobile.cmtapi.com/v1/confirm

Sandbox: https://mobile-sandbox.cmtapi.com/v1/confirm

URL Parameters

Field	DataType	Size	Required	Notes
driverId	String	20	Y	Driver hack license to be confirmed
medallion	String	20	Y	Vehicle medallion to be confirmed

Error Sub-codes (HTTP 400)

Code	Description
450	Invalid driver
451	Invalid medallion

Example Request (GET)

GET URL: *https://mobile.cmtapi.com/v1/confirm?driverId=8888&medallion=JDC1001*

Successful Response (JSON)

HTTP Status: **200 (Success)**

Unsuccessful Response (JSON)

HTTP Status: **400**

```
{
	“responseCode”:450,
	“message”:"Driver not found"
}
```

Pairing Initialization Resources

The Pairing Initialization Resources duplicate the functionality of the Pairing Resources, but with the additional feature of asynchronous confirmation of pairing, unpairing, and updating messages. All requests made through these resources should be considered in a pending state until a confirmation is received.

Use of the Pairing Initialization Resources requires the implementation of a callback server. See the Callback section for more details.

Resource	Description
POST /init/pairing	Pairs the e-hail app to a CMT vehicle.
DELETE /init/pairing/:pairingToken	Un-pairs the e-hail app from a CMT vehicle
PUT /init/pairing/:pairingToken	Updates current pairing data.

POST /init/pairing

Pairs an e-hail application to a vehicle for payment and trip collection and gets a pairing token (used in subsequent requests).

Resource URL

Production: https://mobile.cmtapi.com/v1/init/pairing

Sandbox: https://mobile-sandbox.cmtapi.com/v1/init/pairing

Request Field Description

Field	DataType	Size	Required	Notes
pairingCode	String	20	Y	Code to link with CMT
customerId	String	50	Y	Unique identifier for the mobile user (customer)
customerName	String	50	Y	Customer name for display
latitude	Double		Y	Latitude of device requesting the pairing token
longitude	Double		Y	Longitude of the device requesting the pairing token
callbackUrl	String	500	N	url used for callback features.
autoCompletePayment	Boolean		Y	Indicates if payment will occur automatically by the ehail provider (true) or if payment will be completed using the in-vehicle equipment (false).
autoTipPercentage	Integer		N*	Indicates the percentage tip automatically used. Valid ranges are 0 to 200. *autoTipPercentage and/or autoTipAmount is required if autoCompletePayment is enabled.
autoTipAmount	Double		N*	Indicates the amount tip automatically be used. Valid ranges are 0 to 200. *autoTipPercentage and/or autoTipAmount is required if autoCompletePayment is enabled.

Response Field Description

Field	DataType	Size	Required	Notes
pairingToken	String	30	Y	Unique pairing token used in all subsequent requests.
pairingCode	String	10	Y	Returned pairing code.
medallion	String	20	Y	Medallion
tripId	Integer		Y	Trip Identifier
driverId	Integer		Y	Driver Identifier
timeoutSeconds	Long		Y	The time in seconds after which the request will be considered to have failed.

Error Sub-codes (HTTP 400)

Code	Description
101	The pairing code is invalid
102	Missing or invalid customer ID
103	Pairing code already in use
104	Invalid Customer Name
105	Invalid Lat/Lon
106	Invalid Callback URL

Example Request (POST)

POST URL: *https://mobile.cmtapi.com/v1/init/pairing*

Request (JSON)

```
{
	"pairingCode":"1234567",
        "customerId":"3231432232",
        "customerName":"Test Customer",
        "latitude":78.342342,
        "longitude":-43.893454,
        "callbackUrl":"https://host.appcompany.com/faredata/",
        "autoCompletePayment":true,
        "autoTipPercentage":25,
        "autoTipAmount":0.0
}
```

Successful Response (JSON)

HTTP Status: **201 (Created)**

```
{
        "pairingToken":"dg76Jkil90hgsF3gfe67873DfgghJ",
        "pairingCode":"1234567",
        "medallion":"1N42",
        "tripId":34,
        "driverId":2342344,
        "timeoutSeconds":240
}
```

Unsuccessful Response (JSON)

HTTP Status: **400**

```
{
        "code":101,
        "description":"missing or invalid customer id"
}
```

POST /init/pairing/external

Pairs an e-hail application to a vehicle for payment and trip collection and gets a pairing token (used in subsequent requests). This method is only used when an E-Hail provider is performing the pairing function.

Resource URL

Production: https://mobile.cmtapi.com/v1/init/pairing/external

Sandbox: https://mobile-sandbox.cmtapi.com/v1/init/pairing/external

Request Field Description

Field	DataType	Size	Required	Notes
medallion	String	20	Y	Unique identifier for the vehicle
driverId	String	20	Y	Hack license or driver id used in the pairing
customerId	String	50	Y	Unique identifier for the mobile user (customer)
customerName	String	50	Y	Customer name for display
latitude	Double		Y	Latitude of device requesting the pairing token
longitude	Double		Y	Longitude of the device requesting the pairing token
callbackUrl	String	500	N	url used for callback features.
autoCompletePayment	Boolean		Y	Indicates if payment will occur automatically by the ehail provider (true) or if payment will be completed using the in-vehicle equipment (false).
autoTipPercentage	Integer		N*	Indicates the percentage tip automatically used. Valid ranges are 0 to 200. *autoTipPercentage and/or autoTipAmount is required if autoCompletePayment is enabled.
autoTipAmount	Double		N*	Indicates the amount tip automatically be used. Valid ranges are 0 to 200. *autoTipPercentage and/or autoTipAmount is required if autoCompletePayment is enabled.

Response Field Description

Field	DataType	Size	Required	Notes
pairingToken	String	30	Y	Unique pairing token used in all subsequent requests.
pairingCode	String	10	Y	Returned pairing code.
medallion	String	20	Y	Medallion
tripId	Integer		Y	Trip Identifier
driverId	Integer		Y	Driver Identifier
timeoutSeconds	Long		Y	The time in seconds after which the request will be considered to have failed.

Error Sub-codes (HTTP 400)

Code	Description
101	The pairing code is invalid
102	Missing or invalid customer ID
103	Unable to pair with this driver/vehicle. Usually returned if the vehicle is not in a trip currently
104	Invalid Customer Name
105	Invalid Lat/Lon
106	Invalid Callback URL
107	Invalid driver. Driver specified is not in given vehicle.

Example Request (POST)

POST URL: *https://mobile.cmtapi.com/v1/init/pairing/external*

Request (JSON)

```
{
	"medallion":"EH9009",
	"medallion":"9999",
	"customerId":"3231432232",
	"customerName":"Test Customer",
	"latitude":78.342342,
	"longitude":-43.893454,
	"callbackUrl":"https://host.appcompany.com/faredata/",
	"autoCompletePayment":true,
	"autoTipPercentage":25,
	"autoTipAmount":0.0
}
```

Successful Response (JSON)

HTTP Status: **201 (Created)**

```
{
	"pairingToken":"dg76Jkil90hgsF3gfe67873DfgghJ",
	"pairingCode":"1234567",
	"medallion":"1N42",
	"tripId":34,
	"driverId":2342344,
	"timeoutSeconds":240
}
```

Unsuccessful Response (JSON)

HTTP Status: **400**

```
{
	"code":101,
	"description":"missing or invalid customer id"
}
```

DELETE /init/pairing/:pairingToken

Un-pairs an e-hail application from a CMT vehicle.

Resource URL

Production: https://mobile.cmtapi.com/v1/init/pairing/:pairingToken

Sandbox: https://mobile-sandbox.cmtapi.com/v1/init/paring/:pairingToken

URL Parameters

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

Response Field Description

Field	DataType	Size	Required	Notes
timeoutSeconds	Long		Y	The time in seconds after which the request will be considered to have failed.

Error Sub-codes (HTTP 400)

Code	Description
101	The pairing token is invalid
102	Missing or invalid customer id

Example Request (DELETE)

DELETE URL: *https://mobile.cmtapi.com/v1/init/pairing/dg76JkiI90hgsF3gfe67873DfgghJ*

Successful Response (JSON)

```
HTTP Status: 200 (Success)
{
        "timeoutSeconds":240
}
```

Unsuccessful Response (JSON)

```
HTTP Status: 400
{
        "code":101,
        "description":"pairing token is invalid"
}
```

PUT /init/pairing/:pairingToken

Augments the current pairing with updated auto-pay fields.

Resource URL

Production: https://mobile.cmtapi.com/v1/init/pairing/dg76Jkil90hgsF3gfe67873DfgghJ
Sandbox: https://mobile-sandbox.cmtapi.com/v1/init/pairing/dg76Jkil90hgsF3gfe67873DfgghJ

Request Field Description

Field	DataType	Size	Required	Notes
customerId	String	50	Y	Unique identifier for the mobile user (customer)
customerName	String	75	Y	Customer name for display
latitude	Double		Y	Latitude in degrees of device requesting the pairing token
longitude	Double		Y	Longitude in degrees of the device requesting the pairing token
callbackUrl	String	500	N	url used for callback features.
autoCompletePayment	Boolean		Y	Indicates if payment will occur automatically by the ehail provider (true) or if payment will be completed using the in-vehicle equipment (false).
autoTipPercentage	Integer		N*	Indicates the percentage tip automatically used. Valid ranges are 0 to 200. *autoTipPercentage and/or autoTipAmount is required if autoCompletePayment is enabled.
autoTipAmount	Double		N*	Indicates the amount tip automatically be used. Valid ranges are 0 to 200. *autoTipPercentage and/or autoTipAmount is required if autoCompletePayment is enabled.

Response Field Description

Field	DataType	Size	Required	Notes
timeoutSeconds	Long		Y	The time in seconds after which the request will be considered to have failed.

Error Sub-codes (HTTP 400)

Code	Description
102	Missing or invalid customer ID
104	Invalid Customer Name
105	Invalid Lat/Lon
106	Invalid callback URL
103	No data found for pairing token

Example Request (POST)

PUT URL: *https://mobile.cmtapi.com/v1/init/pairing/dg76Jkil90hgsF3gfe67873DfgghJ*

Request (JSON)

```
{
	"customerId":"321232322",
	"customerName":"Updated Customer",
	"latitude":78.342342,
	"longitude":-43.893454,
	"callbackUrl":"https://host.appcompany.com/faredata/",
	"autoCompletePayment":true,
	"autoTipPercentage":0,
	"autoTipAmount":10.35
}
```

Successful Response (JSON)

```
HTTP Status: 200 (Success)
{
	"timeoutSeconds":240
}
```

Unsuccessful Response (JSON)

```
HTTP Status: 400
{
	"code":101,
	"description":"No data found for pairing token"
}
```

Callbacks

Callbacks are sent from the Mobile API server to the Ehail provider’s callback URL in response to specific events. Callbacks are posted in JSON format to a REST resource

Callback Type	Description
TripData	Sent when a trip ends and includes all fare information.
Pairing failure	Sent if the vehicle receives a pairing request but the payment process has already begun.
Pairing acknowledgement	Sent when a vehicle acknowledges a pairing or pairing update initialization request.
Unpair acknowledgement	Sent when a vehicle acknowledges an unpair initialization request.

TripData Callback

If a callback URL is specified in the pairing resource, CMT will forward events to a RESTFul callback service when events occur in the vehicle. The JSON below defines the message format used in the tripData Callback.

Callback Field Description

Field	DataType	Size	Required	Notes
type	String	4	Y	Set to "TRIP" for tripData callbacks.
tripId	Integer		Y	The id of the trip.
driverId	Integer		Y	The driver id for the trip.
medallion	String		Y	The medallion of the vehicle.
pairingToken	String		Y	The pairing token for the trip.
startTime	String		Y	The start time of the trip, in UTC ISO 8601 format (See example request below)
endTime	String		Y	The end time of the trip, in UTC ISO 8601 format
total	Integer		Y	The total cost of the trip, including fare, fareAtAlternateRate, extra, surcharge, tax, and tip.
fare	Integer		Y	The fare amount in cents at the initial rate class.
fareAtAlternateRate	Integer		Y	The fare amount in cents after a rate change (generally only has a value when going from rate 1 to 4).
extra	Integer		Y	Extra costs for the trip.
tip	Integer		Y	The tip amount for the trip.
surcharge	Integer		Y	The surcharge amount for the trip.
tax	Integer		Y	The tax amount for the trip.
rateAtTripStart	Integer		Y	The rate class at trip start.
rateAtTripEnd	Integer		Y	The rate class at trip end.
rateChangeTime	String		Y	The time, in UTC ISO 8601 format, at which the rate class changed. Empty if no rate change.
distance	Double		Y	The distance of the trip.
autoTipPercentage	Integer		Y	The automatically-set percentage used to calculate the tip.
autoTipAmount	Integer		Y	The automatically-set tip amount.
tollHistory	Array		Y	An array of any tolls included in the trip. Can be empty.
tollName	String			The name of the toll.
tollAmount	Integer			The amount of the toll.

Example Request (POST)

```
{
    "type": "TRIP",
    "tripId": 12,
    "driverId": 435343,
    "startTime": "2012-12-16T13:24:00-500Z",
    "endTime": "2012-12-16T13:44:00-500Z",
    "fare": 421,
    "fareAtAlternateRate": 3600,
    "extra": 123,
    "tip": 123,
    "surcharge": 32,
    "tax": 50,
    "rateAtTripStart": 1,
    "rateAtTripEnd": 4,
    "rateChangeTime": "2012-12-16T13:24:45-500Z",
    "distance": 12332,
    "tollHistory": [
        {
            "tollName": "Toll 1",
            "tollAmount": 1200
        }
    ]
}
```

Successful Callback Response (JSON)
HTTP Status: **200 (Success)**

Unsuccessful Callback Response (JSON)
HTTP Status: **400**

NOTE: E-Hail providers choosing to use the callback method must work with CMT's integration team before implementation.

Pairing Acknowledgement Callback

When a vehicle acknowledges a pairing initialization request, a callback notification will be sent to the Callback URL. The message will be JSON-formatted and include the below fields.

Callback Field Description

Field	DataType	Size	Required	Notes
type	String		Y	Set to "PAIRING" for pairing acknowledgement callbacks.
tripId	Integer		Y	The id of the trip.
driverId	Integer		Y	The driver id for the trip.
medallion	String		Y	The medallion of the vehicle.
pairingToken	String		Y	The pairing token for the trip.
customerId	String		Y	The id of the customer that paired.
customerName	String		Y	The name of the customer that paired.
autoCompletePayment	Boolean		Y	Indicates if payment will occur automatically by the ehail provider (true) or if payment will be completed using the in-vehicle equipment (false).
autoTipPercentage	Integer		N*	The automatically-set percentage used to calculate the tip.
autoTipAmount	Integer		N*	The automatically-set tip amount.

*Only one of autoTipPercentage and autoTipAmount will be included in the callback, depending on which was set in the pairing request.

Example Request (POST)

```
{
    "type": "PAIRING",
    "driverId": 435343,
    "medallion": "EH9009",
    "pairingToken": "dg76Jkil90hgsF3gfe67873DfgghJ",
    "tripId": 1234,
    "customerId": "123456A",
    "customerName": "Charles Xavier",
    "autoCompletePayment": true,
    "autoTipPercentage": 20
}
```

Successful Callback Response (JSON)

HTTP Status: **200 (Success)**

Unsuccessful Callback Response (JSON)

HTTP Status: **400**

NOTE: E-Hail providers choosing to use the callback method must work with CMT's integration team before implementation.

Pairing Failure Callback

If a non-initialization pairing attempt failed because it did not reach the vehicle until after the driver has already started payment, a callback notification will be sent to the Callback URL. The message will be JSON-formatted and include the below fields.

Callback Field Description

Field	DataType	Size	Required	Notes
type	String		Y	Set to “FAILURE” for pairing failure callbacks.
driverId	Integer		Y	The driver id for the trip.
medallion	String		Y	The medallion of the vehicle.
pairingToken	String		Y	The pairing token for the trip.

Example Request (POST)

```
{
    "type": “FAILURE”,
    "driverId": 435343,
    "medallion": "EH9009",
    "pairingToken": "dg76Jkil90hgsF3gfe67873DfgghJ
}
```

Successful Callback Response (JSON)

HTTP Status: **200 (Success)**

Unsuccessful Callback Response (JSON)

HTTP Status: **400**

NOTE: E-Hail providers choosing to use the callback method must work with CMT’s integration team before implementation.

Unpair Acknowledgement Callback

When a vehicle acknowledges an unpair initialization request, a callback notification will be sent to the Callback URL. The message will be JSON-formatted and include the below fields.

Callback Field Description

Field	DataType	Size	Required	Notes
type	String		Y	Set to "UNPAIR" for pairing failure callbacks.
driverId	Integer		Y	The driver id for the trip.
medallion	String		Y	The medallion of the vehicle.
tripId	Integer		Y	The trip id for the trip.
pairingToken	String		Y	The pairing token for the trip.

Example Request (POST)

```
{
    "type": “UNPAIR”,
    "driverId": 435343,
    "medallion": "EH9009",
    "pairingToken": "dg76Jkil90hgsF3gfe67873DfgghJ",
    "tripId": 1234
}
```

Successful Callback Response (JSON)

HTTP Status: **200 (Success)**

Unsuccessful Callback Response (JSON)

HTTP Status: **400**

NOTE: E-Hail providers choosing to use the callback method must work with CMT's integration team before implementation.

Payment Resources

The Payment Resources provide developers access to payment functionality.

Resource	Description
POST /pairing/:pairingToken/authorize	Processes payment for the paired vehicle/trip.

POST /pairing/:pairingToken/authorize

Resource URL

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

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

URL Parameters

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

Request Field Description

Field	DataType	Size	Required	Notes
totalInCents	Integer		Y	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		N	Version of the key used in the encryption
encryptionToken	String	50	N	Send “CMT_PAYNET” encrypted (if encryption is not 0)
encryptionAlgorithm	Integer		N	0=None, 1=3DES
customerReferenceNumber	String	50	N	
ehailServiceFeeInCents	Integer		Y	The service fee charged by the e-hail provider. Set to 0 if you are not providing it to us.
currencyCode	String	5	N	Currency code for the transaction. Options are USD (default), GBP, EUR, and CAD.
accountNumber	String	200	Y (if cardOnFileToken is not set)	The primary account number for authorization
expiryDate	String	4	Y (if cardOnFileToken is not set)	The expiry date in YYMM format
cvv2	String	5	N	Card verification value
zipCode	String	9	N	Either the 5 or 9 digit representation
cardOnFileToken	String		Y (if accountNumber is not set)	Card on file token

Response Field Description

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

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
619	Payment not enabled for this trip

Example Request (POST)

POST URL: *https://payment.cmtapi.com/v2/pairing/dg76Jkil90hgsF3gfe67873DfgghJ/authorize*

Request (JSON)

```
{
      "totalInCents":5382,
      "ehailServiceFeeInCents":0,
      "accountNumber":"4111111111111111",
      "expiryDate":"1605",
      "cvv2":"142"
}
```

Successful Response (JSON)

HTTP Status: **200 (Success)**

```
{
      "responseCode":1,
      "responseMessage":"Approved",
      "tripId":1223,
      "authorizationCode":"AR32564",
      "lastFour":"1111",
      "cardType":"VISA",
      "transactionId":12345678901234

}
```

Unsuccessful Response (JSON)

HTTP Status: **400**

```
{
      “responseCode”:602,
      “message”:"Invalid pairing token"
}
```

Trip Resource

The Trip resource provides access to trip data after a trip has been completed. **Please note that all trip resources require a valid oauth token to be passed.**

Resource	Description
GET /trip/{pairingToken}	Gets the trip details associated with the provided pairing token.

GET /trip/:pairingToken

This method is designed to retrieve trip data.

Resource URL

Production: https://mobile.cmtapi.com/v1/trip/:pairingToken

Sandbox: https://mobile-sandbox.cmtapi.com/v1/trip/:pairingToken

URL Parameters

Field	DataType	Size	Required	Notes
:pairingToken	String	250	Y	The token associated with the trip to be returned.

Response Field Description

Field	DataType	Size	Required	Notes
type	String	4	Y	Set to "TRIP" for tripData callbacks.
tripId	Integer		Y	The id of the trip.
driverId	Integer		Y	The driver id for the trip.
pairingToken	String		Y	The pairing token for the trip.
startTime	String		Y	The start time of the trip, in UTC ISO 8601 format (See example request below)
endTime	String		Y	The end time of the trip, in UTC ISO 8601 format
total	Integer		Y	The total cost of the trip, including fare, fareAtAlternateRate, extra, surcharge, tax, and tip.
fare	Integer		Y	The fare amount in cents at the initial rate class.
fareAtAlternateRate	Integer		Y	The fare amount in cents after a rate change (generally only has a value when going from rate 1 to 4).
extra	Integer		Y	Extra costs for the trip.
tip	Integer		Y	The tip amount for the trip.
surcharge	Integer		Y	The surcharge amount for the trip.
tax	Integer		Y	The tax amount for the trip.
rateAtTripStart	Integer		Y	The rate class at trip start.
rateAtTripEnd	Integer		Y	The rate class at trip end.
rateChangeTime	String		Y	The time, in UTC ISO 8601 format, at which the rate class changed. Empty if no rate change.
distance	Double		Y	The distance of the trip.
autoTipPercentage	Integer		Y	The automatically-set percentage used to calculate the tip.
autoTipAmount	Integer		Y	The automatically-set tip amount.
tollHistory	Array		Y	An array of any tolls included in the trip. Can be empty.
tollName	String			The name of the toll.
tollAmount	Integer			The amount of the toll.

Error Sub-codes (HTTP 400)

Code	Description
441	Trip not found or has not ended yet
103	Pairing failed due to timing of request

Example Request (GET)

GET URL: `http://localhost:8080/v1/trip/ MTM2NDFiYWItOGU2Yy00N2IxLTkzZDgtNzEyNGM1Y2YwMzM3`

Successful Response (JSON)

HTTP Status: **200 (Success)**

```
{
      "startTime":"2013-07-22T05:20:50.712Z",
      "driverId":90009,
      "autoTipAmount":5.0,
      "autoTipPercentage":10,
      "tripId":6,
      "endTime":"2013-07-22T05:42:49.952Z",
      "pairingToken":"MTM2NDFiYWItOGU2Yy00N2IxLTkzZDgtNzEyNGM1Y2YwMzM3",
      "fare":400,
      "fareAtAlternateRate":0,
      "tip":180,
      "surcharge":0,
      "tax":0,
      "rateChangeTime":"",
      "distance":0.0,
      "rateAtTripStart":1,
      "rateAtTripEnd":1,
      "tollHistory":[],
      "type":"TRIP",
      "extra":0
}
```

Unsuccessful Response (JSON)

HTTP Status: **400**

```
{
      "code":441,
      "description":"Trip data not found"
}
```

Tokenization Resources

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
POST /tokenize	Provides access to tokenize credit card data.
DELETE /tokenize/:cardToken	Provides access to delete tokenized credit card data.

POST /tokenize

This method is designed to tokenize credit card data.

Resource URL

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

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

Request Field Description

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

* Only used if validateAccountInformation is true.

Response Field Description

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

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

Example Request (POST)

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

Request (JSON)

```
{
    "accountNumber":"8888888888",
    "expiryDate":"1504"
}
```

Successful Response (JSON)

```
HTTP Status: 200 (Success)
{
        "responseCode":1,
        "responseMessage":"Success",
        "cardType":"VISA",
        "lastFour":"1234",
        "cardOnFileToken":"abcdefghijklmnop123456"
}
```

Unsuccessful Response (JSON)

```
HTTP Status: 400
{
      “responseCode”:601,
      “message”:"Invalid card type or card type not supported"
}
```

DELETE tokenize/:cardToken

This method is designed to delete tokenized credit card data.

Resource URL

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

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

Response Field Description

Field	DataType	Size	Required	Notes
responseCode	Integer		Y	See sub-codes below
responseMessage	String	100	Y	Message description

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

Example Request (DELETE)

DELETE URL: *https://payment.cmtapi.com/v2/tokenize/da43fdsfdsfdsfds/*

PUT Data Response (JSON)

```
HTTP Status: 200 (Success)
{
        "responseCode":1,
        "responseMessage":"Success"
}
```

Unsuccessful Response (JSON)

```
HTTP Status: 400
{
      “responseCode”:608,
      “message”:" Error processing request"
}
```

Appendix A

Authorization Response Codes

Code	***Message***
0	An error occurred
1	Approved
2	Declined
10	Partial approval
20	Invalid CVV2
21	No AVS match
99	Processor error