

Rental Car Manager™
SERIOUSLY SIMPLE SOFTWARE

API v3.1 Developers Guide

September 2021

Contents

Introduction.....	4
Getting Started	4
API Credentials	4
Request Signing	4
Demo & Sample Code	6
Checking for Errors.....	6
Datasets returned – specs and format	7
The API booking process	7
Agent API	10
Server call syntax	10
Booking Amount Calculations.....	10
Making a booking	12
Step 1 – Get locations and vehicle categories.....	12
Step 2 – Get available vehicles and rates	13
Step 3 – Select a vehicle and get fees & insurances.....	15
Step 4 – Make a Booking or Quotation	17
Step 5 – Submit payment information (Not applicable for Agent bookings)	22
Using the Credit Card Vault	24
Method 1: Implementing Vault by directly displaying the Vault Entry Screen.	24
Method 2: Implementing Vault using RCM’s Secure Vault Screen	25
Step 1: Loading RCM’s Secure Vault screen	25
Step 2: Sending the Confirmation Email	26
Sending Email Confirmations.....	26
Booking Information	27
Editing a booking or Quotation	29
Adding or editing Additional Drivers for an existing booking	31
Cancelling a booking (Agent API only)	34
Get Agent Bookings (Agent API only)	35
Other Useful API Methods.....	36
Get Location Details (locationdetails)	36
Dataset Reference	37
Troubleshooting	51
Google Analytics Ecommerce Tracking.....	52
Google Tag Manager	52
Google Analytics.....	53
API Error Codes.....	53

Introduction

Getting Started

API Credentials

Contact support@rentalcamanager.com to request access to our development sandbox.

If you provide us with the details of the rental car company that you are integrating with, we can create an API for the live data as well.

Request Signing

If you have been issued with a hash secret then you will need to sign each request using the HMACSHA256 protocol.

To create the signature:

1. Get the part of the request URL following the domain information (including the entire query string).
2. Use the hash secret to sign the part of the URL using HMACSHA256.
3. Add the hex encoded result to the header of the request as the signature.

Sample VB Code

Step 1 API call - <https://apis.rentalcamanager.com/booking/v3.1/<key>/step1>

UrlPart = /booking/v3.1/<key>/step1

```
Private Function CreateHMACSHA256Signature(sharedSecret As String, UrlPart As String) As String
    Dim textToHash As String = UrlPart
    Dim encoding = New System.Text.UTF8Encoding
    Dim myHMAC As New HMACSHA256(encoding.GetBytes(sharedSecret))
    Dim hashedBytes As Byte() = myHMAC.ComputeHash(encoding.GetBytes(textToHash))
    Dim strTemp As New StringBuilder(hashedBytes.Length * 2)
    Dim hex As String
    For Each b As Byte In hashedBytes
        hex = Conversion.Hex(b)
        If hex.Length = 1 Then hex = "0" & hex
        strTemp.Append(hex)
    Next
    Dim hash As String = strTemp.ToString

    Return hash
End Function
```

Sample C# code

Step 1 API call - <https://apis.rentalcarmanager.com/booking/v3.1/<key>/step1>
UrlPart = /booking/v3.1/<key>/step1

```
private string CreateHMACSHA256Signature(string sharedSecret, string UrlPart)
{
    string textToHash = UrlPart;
    dynamic encoding = new System.Text.UTF8Encoding();
    HMACSHA256 myHMAC = new HMACSHA256(encoding.GetBytes(sharedSecret));

    byte[] hashedBytes = myHMAC.ComputeHash(encoding.GetBytes(textToHash));

    StringBuilder strTemp = new StringBuilder(hashedBytes.Length * 2);
    string hex = null;
    foreach (byte b in hashedBytes) {
        hex = Conversion.Hex(b);
        if (hex.Length == 1)
            hex = "0" + hex;
        strTemp.Append(hex);
    }
    string hash = strTemp.ToString();

    return hash;
}
```

Add the hash result to the request header as:
signature: <hash>

PHP Example

To create the hash:

```
$hash = hash_hmac('sha256', UrlPart, SHARED_SECRET);
```

Example:

```
$hash = hash_hmac('sha256', '/booking/v3.1/<apikey>/step1', SHARED_SECRET);
```

Curl request:

```
curl --request GET \
  --url https://apis.rentalcarmanager.com/booking/v3.1/<ApiKey>/step1 \
  --header 'signature: $hash'
```

Demo & Sample Code

Demo link

For a working demonstration of our API go to:

<https://web.rentacarmanager.com/API32/WebAPIDemo/>

Sample Code & Testing

Sample code for .NET and PHP can be downloaded from our sandbox area at https://sandbox.rentacarmanager.com/developer/sample_code_3_1

You can make test API calls for each step of the booking page at https://sandbox.rentacarmanager.com/developer/livetest_3_1

You will need a username and password to be able to log in to the sandbox area and download the sample code. Please contact support@rentacarmanager.com if you do not have sandbox credentials.

Checking for Errors

After making an API call, always check the **rcmErrors** dataset first. If this is empty, it is safe to proceed. See the [Troubleshooting](#) section for more information.

Datasets returned – specs and format

The results of API calls are returned as a JavaScript string with variables initialised with JSON data in this format:

```
Dataset1Name = [{"field1name":"field1value","field2name":"field2value"...etc}];  
Dataset2Name = [{"field1name":"field1value","field2name":"field2value"...etc}];  
...
```

In some cases the results returned are a variable(s) in this format:

```
Variable1Name='value';Variable2Name='value';
```

The API booking process

The RCM API and Agent API allow you to make unallocated bookings for a specified vehicle category, for a specified pickup and dropoff date. An unallocated booking is a booking that has been made for a particular category (e.g. small car), but has not been allocated to a particular vehicle. Allocation of a booking to a specific vehicle is done by the rental car company.

In order to make a booking, you first need to know what locations and vehicle categories have been made available by the rental car company for selection on their website or for selection by agents.

When the end user or agent has selected the booking dates, driver age and category type, then a call is made to the server to get the availability of the selected category.

The user or agent then selects a particular category, the insurance and optional extras as setup by the rental car company. The selected category and optional extras are then sent to the server via an API call to make a booking or quote.

Via the API, you can create either:

- A quote,
- An “on request” booking, or
- An unallocated reservation.

The next stage of the process is the payment. If you are collecting payment for the booking, then you can save the payment information through the API so that the payment will be recorded against the reservation in Rental Car Manager.

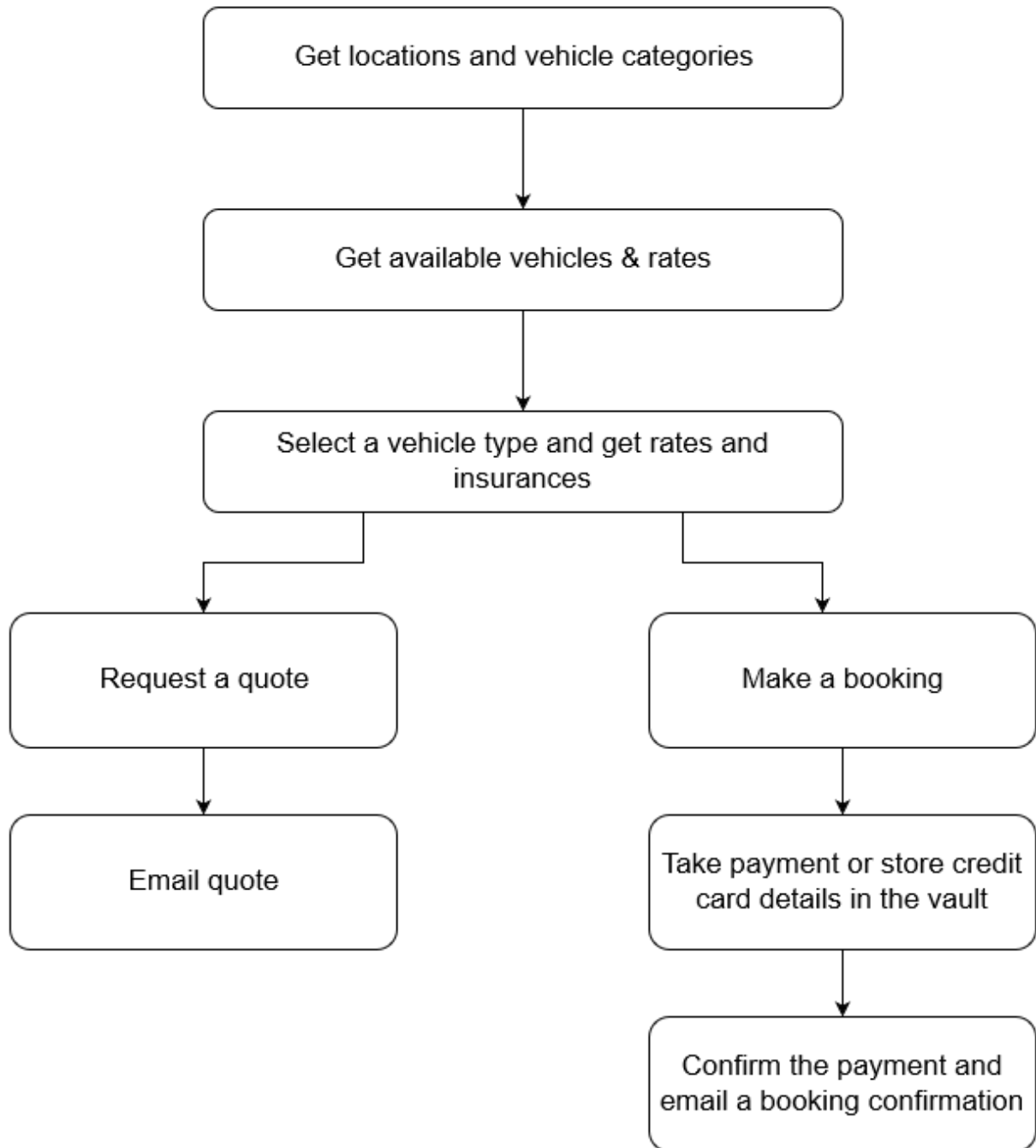
To make a booking through the API, you need use the following process:

1. Get the locations and vehicle categories
2. Get the available vehicles and rates for the selected date range, driver age and vehicle category
3. Select a vehicle and get fees & insurances
4. Submit a booking or quote
5. Submit payment information (not applicable for agent bookings)

It is important that you follow this process in the order mentioned above. You can't just assume that all locations or categories have been made available by the rental car company for online bookings.

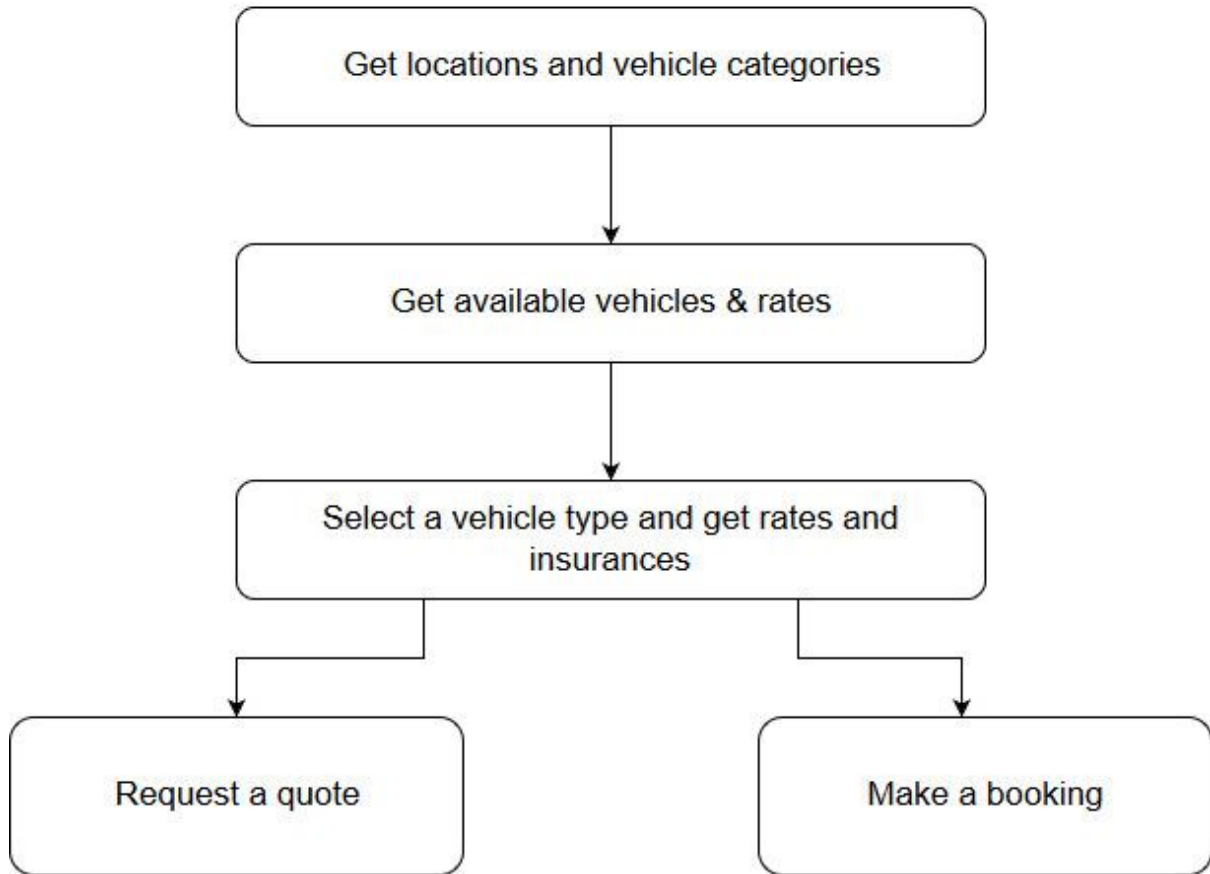
Rental Car Manager API v3

Booking Process



Rental Car Manager API v3

Agent Booking Process



Agent API

The Agent API is the same as the RCM API except that agents have access to extra methods that allow developers to:

- Cancel bookings, and
- Get a list of bookings for an agent

These extra methods are only available if you have been issued an Agent API key.

Server call syntax

The main syntax for calling the API for all methods is as follows:

```
https://apis.rentalcarmanager.com/booking/v3.1/<key>/<method>{/parameters/}
```

key: Secure key issued to each rental car company or agent

method: The name of the method call as outlined in the next sections

Booking Amount Calculations

There are two datasets that you should be looking at, the first is rcmAvailableCars and the second one is rcmMandatoryFees, please see the image below for a definition of both of these.

rcmAvailableCars contains the information about the cars and what the rental rate is for them. You should be using the field totalrateafterdiscount to get the rental rate for the booking.

rcmMandatoryFees contains a listing of the mandatory fees that need to be applied to the booking. This is where the holiday and after hours fees will be.

If you haven't been using the data in the rcmMandatoryFees dataset, this would explain why the fees haven't been included in the booking by the agent.

If you look at the image below you can see there are 4 fields in the dataset rcmMandatoryFees which you use to determine the cost of the fee:

- Numofdays – this is the number of days for the booking.
- Fees – this is the value of the mandatory fee.
- Percentagetotalcost – this is either true or false and is used when calculating the amount for a Percentage type fee.
- Type – this can be one of three values
 1. Daily – if it is this value, you calculate the cost of the mandatory fee by performing the following calculation on the fields within the dataset, numofdays * fees.
 2. Fixed – if this is the value, the cost of the mandatory fee is the value in the fees field in the dataset
 3. Percentage – if this is the value, you calculate the cost of the fee in two different ways
 - If percentageTotalCost is set to True, then you calculate the cost of the mandatory fee by performing the following calculation on the fields within the dataset:
*fees * total cost of booking* (the total cost of the booking is the rental rates plus any optional fees the customer has chosen).

- If `percentageTotalCost` is set to `False`, then you calculate the cost of the mandatory fee by performing the following calculation on the fields within the dataset:
*fees * totalrateafterdiscount* (the percentage is only applied to the rental part of the booking).

Making a booking

Step 1 – Get locations and vehicle categories

Step1 returns important information in order to start the booking process, including renting locations, office times, information about vehicle categories, holidays etc.

Note: You will need this step to be able to pass in the correct identifiers for future steps. It is possible to skip this step and go straight to Step 2 which also provides the same information.

Method name:

step1

Method call:

```
https://apis.rentalcarmanager.com/booking/v3.1/<key>/step1
```

Parameters:

None

Sample call:

```
https://apis.rentalcarmanager.com/booking/v3.1/<key>/step1
```

Returned datasets:

- rcmLocationInfo
- rcmOfficeTimes
- rcmCategoryTypeInfo
- rcmDriverAgesInfo
- rcmAgentInfo (agent API only)
- rcmErrors

Step 2 – Get available vehicles and rates

Method name:

step2

Method call:

```
https://apis.rentalcarmanager.com/booking/v3.1/<key>/step2
/<category_type_id>
/<pickup_location_id>
/<pickup_location_date>
/<pickup_location_time>
/<dropoff_location_id>
/<dropoff_location_date>
/<dropoff_location_time>
/<age_id>
/<details_flag>
/{campaign_code}
```

Parameter List:

Name	Type	Size (max)	Mandatory	Description
category_type_id	Int		Yes	Category Type ID matching the ID in RCM system Retrieved from dataset rcmCategoryTypeInfo
pickup_location_id	Int		Yes	Location ID matches the ID of that location within RCM system Retrieved from dataset rcmLocationInfo
pickup_location_date	String	10	Yes	Date string using next format: dd_MM_yyyy It is important that the European date format is used and not American so first day and then month
pickup_location_time	String	5	Yes	Time string using the next format: HH_mm It is important you use 24 hour time and not 12 hour time with am/pm!
dropoff_location_id	Int		Yes	Location ID matches the ID of that location within RCM system Retrieved from dataset rcmLocationInfo
dropoff_location_date	String	10	Yes	Date string using next format: dd_MM_yyyy It is important that the European date format is used and not American so first day and then month
dropoff_location_time	String	5	Yes	Time string using the next format: HH_mm It is important you use 24 hour time and

				not 12 hour time with am/pm!
age_id	Int		Yes	This is the ID for age used within RCM system not the actual age! Retrieved from dataset rcmDriverAgesInfo
details_flag	bit		Yes	Indicates if you wish to download the payment detail breakdown for each day. 0=no details (default) 1=details
campaign_code	String	10	No	Optional code that has to match campaign code setup within the RCM system, default is '-'

Sample call:

```
https://apis.rentalcarmanager.com/booking/v3.1/<key>/step2/1/1/08_08_2016/09_00/1/15_08_2016/09_00/15/1/-
```

Returned datasets:

- rcmLocationInfo
- rcmOfficeTimes
- rcmCategoryTypeInfo
- rcmDriverAgesInfo
- rcmLocationFees
- rcmAvailableCarDetail (may be empty depending on details flag being set to 1)
- rcmAvailableCars
- rcmMandatoryFees
- rcmAgentInfo (agent API only)
- rcmErrors

Step 3 – Select a vehicle and get fees & insurances

Method name:

Step3

Method call:

```

https://apis.rentalcarmanager.com/booking/v3.1/<key>/step3
/<category_type_id>
/<pickup_location_id>
/<pickup_location_date>
/<pickup_location_time>
/<dropoff_location_id>
/<dropoff_location_date>
/<dropoff_location_time>
/<age_id>
/<car_category_id>
/<details_flag>
/{campaign_code}
    
```

Parameters:

Name	Type	Size (max)	Mandatory	Description
category_type_id	Int		Yes	Category Type ID matching the ID in RCM system Retrieved from rcmCategoryTypeInfo dataset
pickup_location_id	Int		Yes	Location ID matches the ID of that location within RCM system Retrieved from rcmLocationInfo dataset
pickup_location_date	String	10	Yes	Date string using next format: dd_MM_yyyy It is important that the European date format is used and not American so first day and then month
pickup_location_time	String	5	Yes	Time string using the next format: HH_mm It is important you use 24 hour time and not 12 hour time with am/pm!
dropoff_location_id	Int		Yes	Location ID matches the ID of that location within RCM system. Retrieved from RCMLocationInfo dataset
dropoff_location_date	String	10	Yes	Date string using next format: dd_MM_yyyy It is important that the European date format is used and not American so first day and then month
Dropoff_location_time	String	5	Yes	Time string using the next format: HH_mm It is important you use 24 hour time and

				not 12 hour time with am/pm!
age_id	Int		Yes	This is the ID for age used within RCM system not the actual age! Retrieved from rcmDriverAgesInfo dataset
car_category_id	Int		Yes	The selected Category ID from step2. Retrieved from rcmAvailableCars dataset (field = carsizeid)
details_flag	bit		Yes	Indicates if you wish to download the payment detail breakdown for each day. 0=no details (default) 1=details
campaign_code	String	10	No	Optional code that has to match campaign code setup within the RCM system, leave as '-' if not required.

Returned datasets:

- rcmDriverAgesInfo
- rcmLocationFees
- rcmAvailableCarDetails Optional depending on details flag being set to 1
- rcmAvailableCars
- rcmMandatoryFees
- rcmOptionalFees
- rcmInsuranceOptions
- rcmKmCharges
- rcmRentalSource
- rcmCountries
- rcmAreaOfUse
- rcmTaxInclusive Boolean variable (true/false)
- rcmTaxRate Numeric variable (0.10 -> 10%)
- rcmStateTax Numeric variable (0.10 -> 10%)

Example:

```
https://apis.rentalcarmanager.com/booking/v3.1/<key>/step3/1/4/31_08_2015/10_00/4/07_09_2015/10_00/15/2/0
```


Step 4 – Make a Booking or Quotation

In step 4 of the booking process, information selected in step 3 is saved in Rental Car Manager as either a quote or an unallocated booking.

Please note:

Standard bookings:

When a booking is made using the API, it is saved as a quote by default until payment has been confirmed by calling the *confirmpayment* method. The rental car company can override this default behaviour by changing a system setting that forces all bookings made through the API to be saved as an unallocated booking even if payment has not been submitted.

A booking can be changed from quote to unallocated booking and vice versa using the *booking_type* parameter of the *editbooking* method.

Agent Bookings:

Agent bookings don't use the *confirmpayment* method. These bookings will be saved as the booking type that you specify in the API call. For example, if you set the *booking_type* parameter to 2 then it will be saved as a booking and does not change unless you change it using the *editbooking* method.

Method name:

booking

Method call:

```
https://apis.rentalcarmanager.com/booking/v3.1/<key>/booking
  /<category_type_id>
  /<pickup_location_id>
  /<pickup_location_date>
  /<pickup_location_time>
  /<dropoff_location_id>
  /<dropoff_location_date>
  /<dropoff_location_time>
  /<age_id>
  /<car_category_id>
  /<booking_type>
  /<insurance_id>
  /<extra_kms_id>
  /<transmission>
  /<sendemail>
  /?<data>
```

Parameters:

Name	Type	Size (max)	Mandatory	Description
category_type_id	Int		Yes	Category Type ID matching the ID in RCM system

pickup_location_id	Int		Yes	Location ID matches the ID of that location within RCM system
pickup_location_date	String	10	Yes	Date string using next format: dd_mm_yyyy It is important that the European date format is used and not American so first day and then month
pickup_location_time	String	5	Yes	Time string using the next format: HH_mm It is important you use 24 hour time and not 12 hour time with am/pm! (example: 14_30)
dropoff_location_id	Int		Yes	Location ID matches the ID of that location within RCM system
dropoff_location_date	String	10	Yes	Date string using next format: dd_mm_yyyy It is important that the European date format is used and not American so first day and then month
dropoff_location_time	String	5	Yes	Time string using the next format: HH_mm It is important you use 24 hour time and not 12 hour time with am/pm!
age_id	Int		Yes	This is the ID for age used within RCM system not the actual age!
Car_category_id	Int		Yes	The selected Category ID from step 2
booking_type	Int		Yes	Indicates the type of booking Quote or actual booking 1=quote 2=booking
insurance_id	Int		Yes	Insurance ID corresponding to the ID used within RCM Part of optional Extras
extra_kms_id	Int		Yes	Extra Kms ID corresponding with the ID in RCM system Part of optional Extras
transmission	Int		Yes	Indicates what the preference is for Car

				transmission 0 = No Preference 1 = Auto 2 = Manual
sendemail	Int	No		Determines the functionality when it comes to sending email confirmations 0 - No Email 1 - Default Behaviour * 2 - Always send email * Default Behaviour for Sending email confirmations Quotes - The confirmation email gets sent when either the method Booking is successful or, in the case of an Agent booking, if the agent has been setup to receive emails. Bookings - The confirmation email gets sent in the following circumstances, the payment/card collection has been successful, the Rental Car Company has been setup not to save cards, or in the case of an Agent booking, the agent has been setup to receive emails.
data	String	2048	Yes	Base64 data string build up of the next data groups: CustomerData OptionalData ReferralID (optional) CampaignCode (optional) AgentCode (optional – Agent API only) AgentName (optional – Agent API only) rcmNewsletter (optional) RefNo (optional) AgentEmail (optional – Agent API only) Random value to Prevent Caching

The "data" parameter is a base64 encoded string made up of the following:

CustomerData + "|" + OptionalData + "|" + ReferralID + "|" + CampaignCode + "|" + AgentCode + "|" + AgentName + "|" + rcmNewsLetter + "|" + RefNo + "|" + AgentEmail + "|" + <Random value to Prevent Caching>

CustomerData:

```

fmn:<firstname: alphanumeric(30)>,
lnm:<lastname: alphanumeric(40)>,
eml:<email: email_string(50)>,
phn:<phone number: alphanumeric(20)>,
mob:<mobile phone number: alphanumeric(25)>,
dob:<date of birth: date (dd/mm/yyyy)>,
lcn:<license number: number(70)>,

```

```
lci:<country id where license was issues: number>,
lce:<date license expires: date (dd/mm/yyyy)>,
adr:<address: text(80)>,
cty:<city: alphanumeric(50)>,
sta:<state: alphanumeric(30)>,
pcd:<postal code: alphanumeric(10)>,
cnt:<country id>,
fax:<fax number(15)>,
fus:<found us id: number>,
rmk:<remarks: alphanumeric(250)>,
not:<number of people travelling: number>,
fln:<flight no arrival (50)> ,
flo:<flight no departure (50)>,
flc:<collection point: text(80)>,
flr:<return point: text(80)>,
aru:<area of use id: number>
```

!!!! ONLY Normal text characters are allowed, any strange character will cause a validation error.

A-Z a-z 0-9 . - # @ /

Plus Unicode: u0080-u0250 to allow for European characters

Example:

```
fnm:test,lnm:test,eml:email@email.com,phn:555555555,mob:,dob:01/01/1900,lcn:,lci:7,
lce:1/1/1900,adr:test,cty:test,sta:QLD,pcd:4301,cnt:7,fax:,fus:,rmk:,not:2,fln:TG123,flo:,
flc:Airport,flr:,aru:
```

Optional Data:

```
<Optional Extra ID>:<Quantity>,...
eq: 9:2,11:1,15:3
```

ReferralID:

ReferralID is a numeric value that is used to identify affiliate or referrer to the website to credit that affiliate with the booking e.g.: 123

CampaignCode:

A valid Rental Car Manager campaign code.

AgentCode:

A valid agent code from Rental Car Manager (rcmAgentInfo dataset). Only used for agent bookings.

AgentName:

Can be any name e.g. you may want to use the name of the agent's employee that made the booking. Only used for agent bookings.

rcmNewsletter:

Either "True" or "False"

RefNo:

Use this field to save a custom reference number against the booking

AgentEmail:

Can be any email address e.g. you may want to use the email address of the agent's employee that made the booking. Only used for agent bookings.

Example:

```
https://apis.rentalcarmanager.com/booking/v3.1/<key>/booking/1/4/31_08_2015/10_00/4/07_09_2015/10_00/15/2/1/2/1/0/1/?<data>
```

Returned variables:

- rcmReservationRef – String variable (Random Unique String)
- rcmReservationNo – Numeric variable (sequential booking number)

Step 5 – Submit payment information (Not applicable for Agent bookings)

If you are processing a payment at the time of booking, then you can save the payment information against the booking by calling the *confirmpayment* method.

Method name:

confirmpayment

Method call:

```
https://apis.rentalcarmanager.com/booking/v3.1/<key>/confirmpayment/<refno>/<data>
```

Parameters:

Name	Type	Size (max)	Mandatory	Description
<i>refno</i>	String		Yes	The value <i>rcmReservationRef</i> returned from method Booking
<i>data</i>	String		Yes	Base64 data string build up of the next data groups: <ul style="list-style-type: none">• PaymentData• PaymentScenario (optional)• Random value to Stop Caching

The “data” parameter is a base64 encoded string made up of the following:

PaymentData + "|" PaymentScenario + "|" + <Random value to Prevent Caching>

PaymentData:

```
<Amount of payment>;  
<success 1=yes,0=no>;  
<Payment Type eq Paypal, Visa>;  
<Payment Date dd-MMM-yyyy HH:mm eq 05-Nov-2016 14:40>;  
<PaymentSupplierID eg 2, confirm with RCM >;  
<TransactionBillingID eg 033300030baf814c, rebilling token >;  
<PaymentTxnRef eg 000000020180916e >;  
<CardHolderName eg RA Stevens>;  
<PaymentSource eg VISA or online payment>;  
<TruncatedCardNumber eg 411111.....11 (do not pass in full credit card number)>;  
<CardExpiry eq 0717 (mmyy)>;  
<TransType eq Empty=purchase, Auth=allow zero amount>
```

PaymentScenario:

If this payment is part of the original booking, this can be left blank or omitted.

If this payment is at a later stage of the booking process, the following values must be used:

“convertquote” – for adding payment information when converting a quotation into a booking

“prehire” – for adding payment information at any other stage of the booking, for example as customer check in procedure.

Examples of Base64 data (pre encoding):

- 42.61;1;VISA;02-Nov-2017
10:16;2;transbillingID;transReference;CardHolderName;PaymentSource;111111.....11;1219;;0
|prehire|636452178688932194
- 42.61;1;VISA;02-Nov-2017 11:11;2;|636452178688932194
- 42.61;1;VISA;02-Nov-2017 11:11;2;

Example:

```
https://apis.rentalcarmanager.com/booking/v3.1/<key>/confirmpayment/BBCED2822BE54561B6E8F7F2FF/?<data>
```

Returned variables:

- rcmPaymentSaved=true

Using the Credit Card Vault

If you wish to incorporate the Auric credit card vault into your API implementation, there are two ways you can go about this.

The first is incorporating both server-side and client-side code from the API.

The second is only using server-side calls to the API.

Method 1: Implementing Vault by directly displaying the Vault Entry Screen.

Follow these instructions if you would like to use the secure Auric credit card vault in your API implementation. This way will give you, as the developer, the most control over the vault process, but requires some javascript on your client to listen and react to the event that is triggered by the vault frame.

Step 1

After the booking has been saved and you have the rcmReservationRef, make the following server-side call to the API to get the vault url:

```
https://apis.rentalcarmanager.com/booking/v3.1/<key>/geturl/<data>
```

where <data> is a base64 encoded string made up of the following:

```
rcmReservationRef + "|" + <Random value to Prevent Caching>
```

rcmReservationRef is the booking reference number referred to in previous API calls (also referred to as rcmReferenceID).

The API will return a variable rcmURL containing the url of the vault that the user needs to be redirected to.

Step 2

On the client-side, create an iframe and set the source to the url returned from the API call.

Step 3

Once the user has finished entering card details, the page will receive a message (event) from the iframe. Javascript on the client page needs to check for this message and if the card details were saved successfully then call the "vaultentry" method of the API.

Example:


```

<script>
  var eventMethod = window.addEventListener ? "addEventListener" : "attachEvent";
  var eventer = window[eventMethod];
  var messageEvent = eventMethod == "attachEvent" ? "onmessage" : "message";

  // Listen to message from child window needed for vault page
  eventer(messageEvent, function (e) {
    var key = e.message ? "message" : "data";
    var data = e[key];
    //run function//
    var split = data.split(',');
    if (split[5] == "ADD") {

      // Add code here to save payment info to API

    }
  }, false);
</script>

```

NOTE After the vault triggers the “ADD” event, the credit card vault token has not yet been linked to the booking/customer in RCM. You now need to call our API method “vaultentry” to record the data returned from the vault.

Sample data returned from vault = 8K7Q1w4mwA0Wj0B1122,Visa,R M Test,08/18,123,ADD

You need to use this data to call the “vaultentry” method on our API as follows:

<https://apis.rentalcarmanager.com/booking/v3.1/<key>/vaultentry/<refno>/<data>>

where <data> is a base64 encoded string made up of the following:

- data returned from vault + “|” + <Random value to Prevent Caching>

Note that if you are using this method, there is no need to call our separate API method to send an email to the customer. The email will be sent upon calling “vaultentry”, if the conditions have been met for sending emails (see next section “Sending Email Confirmations”).

Method 2: Implementing Vault using RCM’s Secure Vault Screen

This method of credit card vault entry relies on RCM’s own page to manage the vault events and api calls. This method is easier to set up, however using RCM’s secure vault screen will not send your customer a confirmation email. For this reason, there is a second step if you wish to send the email.

There are 2 steps detailed below, calling the secure vault screen and sending the confirmation email.

Step 1: Loading RCM’s Secure Vault screen

After collecting the customer’s personal information and calling the booking method, you will need to call a file on our server which will host the vault entry screen.

This will need to be done by using an iFrame, in order to incorporate the styling of your other web pages. The iFrame will need to load the following file, s_TokenPublic.aspx along with some querystring parameters.

Below is a sample of the URL to place in the iFrame. The <api-server> is the same value as all of the other calls you have made to the API.

https://<api-server>/ccvault2/s-TokenPublic.aspx?
CompanyKey=<key>&RefID=<refid>&redctUrl=<ReturnURL(must be encoded)>

There are three parameters for this URL:

- CompanyKey – this is the key you use for any API calls.
- RefID - this is the value returned from the booking method in the variable rcmReservationRef.
- redctUrl – This is the URL you wish to customer to be redirected to after they have successfully saved a credit card in the vault. It is important to note that this value must be URL-encoded.

After the card has been successfully saved to the vault, the customer will be re-directed to whatever URL you passed in as the redctURL value. The relevant vault token will have been automatically saved against the booking within RCM.

Step 2: Sending the Confirmation Email

If you wish to send a confirmation email out for the booking once the credit card has been successfully saved to the secure vault, then you will need to make a server-side call to the method “sendemail” from the page that the customer was redirected to in the redctURL.

For example, if your redctUrl above is <http://mysite.com/booking-complete>, then this page will need to make an api call to the “sendemail” method. For this reason, make sure that you encode the rcmReservationRef value within your redctUrl, for example <http://mysite.com/booking-complete/rcmReservationRefGoesHere>

Then pull the rcmReservationRef from the querystring, and use it to call our api method “sendemail”

Below is the syntax for the sendemail method. It only has one parameter, the rcmReservationRef.

```
https://apis.rentalcarmanager.com/booking/v3.1/<key>/sendemail/<rcmReservationRef>
```

Sending Email Confirmations

For quotes:

The confirmation email gets sent when either the method *booking* is successful or, in the case of an Agent booking, if the agent has been setup to receive emails.

For Unallocated Bookings:

The confirmation email gets sent in the following circumstances:

- the payment/card collection has been successful (through the *confirmpayment* method), or
- the Rental Car Company has been setup not to save cards, or
- in the case of an Agent booking, the agent has been setup to receive emails.

Setting the *sendemail* parameter value to 2 in the *booking* method will force the email to be sent after a successful booking.

There is a system parameter in Rental Car Manager that can be used to turn off email confirmations if you want to handle this yourself. Contact Rental Car Manager support if you would like to use this feature.

Booking Information

Once a booking has been made, you can use the *bookinginfo* method to get all of the finalised information about the booking in one call.

Method name:

bookinginfo

Method call Option 1:

If you are making this call straight after the booking has been entered, use the following:

```
https://apis.rentalcarmanager.com/booking/v3.1/<key>/bookinginfo/  
<rcm_reservation_reference>/-
```

Parameters:

- rcm_reservation_reference – this is the value received in the output of the “booking” step. Important - use the rcmReservationRef value rather than rcmReservationNo

Sample call:

```
https://apis.rentalcarmanager.com/booking/v3.1/<key>/bookinginfo/004298C7DCA57E  
F/-
```

Method call Option 2:

If you are making this call to get booking information at a later time (i.e. not at the time of original booking), you may not have the rcm_reservation_reference. In this case you can use the Reservation Number along with something to identify the customer.

```
https://apis.rentalcarmanager.com/booking/v3.1/<key>/bookinginfo/  
<rcm_reservation_number>/<data>
```

Parameters:

- rcm_reservation_number (rcmReservationNo): The customer’s booking number, which will appear on all correspondence between RCM and the customer, for e.g. confirmation email. This is an integer.
- data: Base64 data string built up of customer data. The customer data can be:
 - Email – use base64 encoded “eml:fred@mailserver.com”, OR
 - Last Name – use base64 encoded “lnm:Smith”
 - AND Optional random value to prevent caching – “|123456”

Sample call:

```
https://apis.rentalcarmanager.com/booking/v3.1/<key>/bookinginfo/1234/  
ZW1sOmZyZWRAbWFpbHN1cnZlci5jb20=  
Where "ZW1sOmZyZWRAbWFpbHN1cnZlci5jb20=" is the base64 encoding of  
"eml:fred@mailserver.com"
```

OR

```
https://apis.rentalcarmanager.com/booking/v3.1/<key>/bookinginfo/1234/  
bG5tO1NtaXRofDc4MjUzOTc3MzI2MjE=  
Where "bG5tO1NtaXRofDc4MjUzOTc3MzI2MjE=" is the base64 encoding of  
"lnm:Smith|7825397732621"
```

Returned datasets:

- rcmBookingInfo
- rcmCustomerInfo
- rcmCompanyInfo
- rcmRateInfo
- rcmExtraFees
- rcmPaymentInfo
- rcmExtraDrivers
- rcmAgentInfo (for Agent API only)
- rcmErrors

Editing a booking or Quotation

Once the booking has been made call this method to make adjustments to details such as optional extras, insurance and user information.

Method name:

editbooking

Method call:

```
https://apis.rentalcarmanager.com/booking/v3.1/<key>/editbooking  
/<rcm_reservation_reference>  
/<pickup_location_id>  
/<booking_type>  
/<insurance_id>  
/<extra_kms_id>  
/<transmission>  
/<sendemail>  
/?<data>
```

Parameters:

Name	Type	Size (max)	Mandatory	Description
<i>rcm_reservation_reference</i>	text		Yes	This is the value received in the output of the "booking" step or the "bookinginfo" method (also known as rcmreferenceid). <u>Important - use the rcmReservationRef value rather than rcmReservationNo</u>
<i>pickup_location_id</i>	Int		Yes	Location ID matches the ID of thatlocation within RCM system Retrieved from rcmLocationInfo dataset
<i>booking_type</i>	Int		Yes	Indicates the type of booking Quote or actual booking 1=quote 2=booking
<i>insurance_id</i>	Int		Yes	Insurance ID corresponding to the ID used within RCM Part of optional Extras Retrieved from rcmInsuranceOptions dataset
<i>extra_kms_id</i>	Int		Yes	Extra Kms ID corresponding with the ID in RCM system Part of optional extras Retrieved from rcmKmCharges

				dataset
<i>transmission</i>	Int		Yes	Indicates what the preference is for Car transmission 0 = No Preference 1 = Auto 2 = Manual
<i>sendemail</i>	Int		Yes	Determines the functionality when sending email confirmations. 0 - No Email 1 - Default Behaviour* 2 - Always send email
<i>data</i>	String	2048	Yes	Base64 data string build up of the next data groups: CustomerData OptionalData ReferralID Random value to Prevent Caching See below for specifications.

*Default Behaviour for Sending email confirmations:

Quotes - The confirmation email gets sent when either the method booking is successful or in the case of an agent booking, if the agent has been setup to receive emails.

Bookings - The confirmation email gets sent in the following circumstances; the payment/card collection has been successful, the rental car company has been setup not to save cards, or in the case of an agent booking, the agent has been setup to receive emails.

Customer Data:

```

fnm:<firstname: alphanumeric>,
lnm:<lastname: alphanumeric>,
eml:<email: email_string>,
phn:<phone number: alphanumeric>,
mob:<mobile phone number: alphanumeric>,
dob:<date of birth: date (dd/mm/yyyy)>,
lcn:<license number: number>,
lci:<country id where license was issues: number>, Retrieved from rcmCountries dataset
lce:<date license expires: date (dd/mm/yyyy)>,
adr:<address: text>,
cty:<city: alphanumeric>,
sta:<state: alphanumeric>,
pcd:<postal code: alphanumeric>,
cnt:<country id>, Retrieved from rcmCountries dataset

```

```
fax:<fax number>,
fus:<found us id: number>, Retrieved from rcmRentalSource - can be empty for Agent
bookings
rmk:<remarks: alphanumeric>,
not:<number of people travelling: number>,
fln:<flight no arrival> ,
flo:<flight no departure>,
flc:<collection point: text>,
flr:<return point: text>,
aru:<area of use id: number> Retrieved from rcmAreaOfUse dataset
```

!!!! ONLY Normal text characters are allowed, any strange character will cause a validation error
A-Z a-z 0-9 . , - # @ /

Plus Unicode: u0080-u0250 to allow for European characters

Example:

```
fnm:test,lnm:test,eml:email@email.com,phn:555555555,mob:,dob:01/01/1900,lcn:,lci:7,
lce:1/1/1900,adr:test,cty:test,sta:QLD,pcd:4301,cnt:7,fax:,fus:,rmk:,not:2,fln:TG123,flo:,
flc:Airport,flr:,aru:
```

OptionalData:

```
<Optional Extra ID>:<Quantity>,...
eq: 9:2,11:1,15:3
```

ExtraID is retrieved from rcmOptionalFees dataset

Leave empty if no optional extra fees

ReferralID:

ReferralID is a numeric value that is used to identify affiliate or referrer to the Web-site to credit that affiliate with the booking

eg: 123

Leave empty if no referrer

Random value to prevent caching:

Any value can be used here. It will be ignored by our server but prevents a cached response.

Returned variables:

- rcmReservationRef
- rcmReservationNo

Adding or editing Additional Drivers for an existing booking

This method can be called alongside “editbooking” in order to add, update or delete Additional Drivers.

Method name:

extradriver

Method call:

```
https://apis.rentalcarmanager.com/booking/v3.1/<key>/extradriver  
/<rcm_reservation_reference>  
/<customer_id>  
/?<data>
```

Action:

The parameters you use will depend on the action you wish to perform:

- Add – to add a new driver, set customer_id to 0 and pass the data as defined below
- Update – to update an existing driver, set customer_id to the applicable ad_customerid as returned in the rcmExtraDrivers dataset
- Delete – to delete an existing driver, set customer_id to the *negative* value of the applicable ad_customerid as returned in the rcmExtraDrivers dataset

Example:

```
Add:  
https://apis.rentalcarmanager.com/booking/v3.1/<key>/extradriver/0267B92D1/0/?<  
data>  
Update:  
https://apis.rentalcarmanager.com/booking/v3.1/<key>/extradriver/0267B92D1/3214  

```

Parameters:

Name	Type	Size (max)	Mandatory	Description
rcm_reservation_reference	Text		Yes	This is the value received in the output of the “booking” step or the “bookinginfo” method (also known as rcmreferenceid). <u>Important - use the rcmReservationRef value rather than rcmReservationNo</u>
customer_id	Int		Yes	CustomerID of the driver you are updating or deleting. New Driver – set this to 0 Update Driver – set this to the customerID from the existing Extra Drivers (rcmExtraDrivers.ad_customerid returned from “bookinginfo” method) Delete Driver – set this to be the <i>negative</i> value of the customer id of the driver you wish to delete (rcmExtraDrivers.ad_customerid returned from “bookinginfo”

				method)
<i>data</i>	String	2048	Yes	Base64 data string build up of the next data groups: CustomerData Random value to Prevent Caching See below for specifications.

Customer Data:

```

fnm:<firstname: alphanumeric>,
lnm:<lastname: alphanumeric>,
eml:<email: email_string>,
phn:<phone number: alphanumeric>,
mob:<mobile phone number: alphanumeric>,
dob:<date of birth: date (dd/mm/yyyy)>,
lcn:<license number: number>,
lci:<country id where license was issued: number; or free text such as
"NSW">, Retrieved from rcmCountries dataset if country is used.
lce:<date license expires: date (dd/mm/yyyy)>,
adr:<address: text>,
cty:<city: alphanumeric>,
sta:<state: alphanumeric>,
pcd:<postal code: alphanumeric>,
cnt:<country id>, Retrieved from rcmCountries dataset
fax:<fax number>

```

!!!! ONLY Normal text characters are allowed, any strange character will cause a validation error
 A-Z a-z 0-9 . , - # @ /

Plus Unicode: u0080-u0250 to allow for European characters

Example:

fnm:test,lnm:test,eml:email@email.com,phn:555555555,mob:,dob:01/01/1900,lcn:,lci:7,
 lce:1/1/1900,adr:test,cty:test,sta:QLD,pcd:4301,cnt:7,fax:

Returned variables:

- rcmCustomerID (new or updated (unchanged) customerid for the extra driver. 0 for deleted driver)

Canceling a booking (Agent API only)

Bookings can be cancelled by calling the cancelbooking API method. Before you invoke the cancelbooking method, you need to get the list of possible reasons for the cancellation by calling the cancelreasons method.

Method name:

cancelreasons

Method call:

```
https://apis.rentalcarmanager.com/booking/v3.1/<key>/cancelreasons
```

Returning datasets:

- rcmCancelReasons

Method name:

cancelbooking

Method call:

```
https://apis.rentalcarmanager.com/booking/v3.1/<key>/cancelbooking/<Rcm_reference_no>/<CancelReasonID>
```

Parameters:

Parameter	Type	Size (max)	Mandatory	Description
<i>Rcm_reference_no</i>	String	30	Yes	The unique Reference Key for the current Booking/quote. The Reference Key expires within 30 minutes of the initiation of the booking.
<i>CancelReasonID</i>	Int	-	Yes	ID that matches the ID returned from method CancelReasons

Returned datasets:

- rcmCancelInfo

Get Agent Bookings (Agent API only)

Agents can get a list of bookings that have been made for their agent code by calling the agentbookings method.

Method name:

agentbookings

Method call:

There are three options for calling the agentbookings method, depending on how you want to filter the results.

To get the information for a specific reservation:

```
https://apis.rentalcarmanager.com/booking/v3.1/<key>/agentbookings/<reservation no>
```

To get the reservations within a specified date range:

```
https://apis.rentalcarmanager.com/booking/v3.1/<key>/agentbookings/<startdate>/<enddate>
```

To get the reservations within a specified date range and with a specified booking status:

```
https://apis.rentalcarmanager.com/booking/v3.1/<key>/agentbookings/<startdate>/<enddate>/<bookingstatus>
```

Parameters:

Parameter	Type	Size (max)	Mandatory	Description
reservationno	Int		Yes	Reservation No for a booking.
startdate	Date		Yes	Date string with next format dd/mm/yyyy
enddate	Date		Yes	Date string with next format dd/mm/yyyy
bookingstatus	String	20	No	Status string one of the next options: 'ALL' 'ONREQUEST' 'CONFIRMED' 'CANCELLED'

Returned datasets:

- rcmAgentBookings

Other Useful API Methods

Get Location Details (locationdetails)

There is a method which can be used to get information about a particular location.

Method Name:

locationdetails

Method Call:

https://apis.rentalcarmanager.com/booking/v3.1/<key>/locationdetails/<Location_id>

Parameters:

Parameter	Type	Size (max)	Mandatory	Description
locationID	Int		Yes	Location ID matches the ID of the location within RCM system Retrieved from rcmLocationInfo dataset

Returned datasets:

- rcmLocationDetails
- rcmErrors

Dataset Reference

Below you will find details regarding the datasets that the API uses.

Notes:

All fieldnames are lower case! Make sure you use lower case to access the correct field.
Please note that <int_id> is not the same as <int>, <int_id> refers to an integer value that reflects the ID within the RCM system for that particular dataset, and in case this value is 0 that means it is valid for all.

Example:

If we have the next Driver's age table within the system:

ID	AGE
1	18
2	19
3	20
4	25
5	30+

An age_id of 4 equals an age of 25. An age_id of 0 passed to a method means any age, in this case above 18, 19, 20, 25 and 30+

Field types:

Possible field types / options are displayed in the dataset definition as <option1/option2/...>.

<True/False> means we have only 2 possibilities either

1. True
2. False

<0/1> means we have only 2 possibilities either

1. 0
2. 1

<Sunday/Monday/Tuesday/Wednesday/Thursday/Friday/Saturday> gives the next possibilities

1. Sunday
2. Monday
3. Tuesday
4. Etc...

rcmLocationInfo

```
rcmLocationInfo = [{
  "id": "<int>",
  "location": "<string>",
  "webdefault": "<True/False>",
  "pickupavailable": "<True/False>",
  "dropoffavailable": "<True/False>",
  "flightnoreqd": "<True/False>",
  "noticerequired": "<real>",
  "daysquotevalid": "<int>",
  "officeopeningtime": "<24hourtime - HH:mm>",
  "officeclosingtime": "<24hourtime - HH:mm>",
  "afterhourbooking": "<True/False>",
  "unattendeddropoffs": "<True/False>",
  "minimunage": "<int_id>",
  "afterhourfeeid": "<int_id>",
  "unattendeddropofffeesid": "<int_id>"
}];
```

rcmOfficeTimes

```
rcmOfficeTimes = [{
  "locid": "<int_id>",
  "wd": "<int>",
  "afterhour": "<True/False>",
  "unattended": "<True/False>",
  "openingtime": "<24hourtime - HH:mm>",
  "closingtime": "<24hourtime - HH:mm>",
  "amstart": "<24hourtime - HH:mm>",
  "amend": "<24hourtime - HH:mm>",
  "pmstart": "<24hourtime - HH:mm>",
  "pmend": "<24hourtime - HH:mm>",
  "startpickup": "<24hourtime - HH:mm>",
  "endpickup": "<24hourtime - HH:mm>",
  "startdropoff": "<24hourtime - HH:mm>",
  "enddropoff": "<24hourtime - HH:mm>"
}];
```

1 = Sunday, 2 = Monday, 3 = Tuesday etc

rcmCategoryTypeInfo

```
rcmCategoryTypeInfo = [{
  "id": "<int_id>",
  "categorytype": "<string>",
  "ordering": "<int>",
  "rentingtype": "<True/False>",
  "raimage": "<string>",
  "rate12hour": "<True/False>",
  "ratepernight": "<True/False>"
}];
```

rcmDriverAgesInfo

```
rcmDriverAgesInfo = [{  
  "id": "<int_id>",  
  "driverage": "<int>",  
  "defaultage": "<True/False>"  
}];
```

rcmLocationFees

```
rcmLocationFees = [{  
  "cattypeid": "<int_id>",  
  "loctypeid": "<int_id>",           1 = pickup, 2 = dropoff  
  "loctype": "<pickup/dropoff>",  
  "locationid": "<int_id>",  
  "currencyname": "<string>",  
  "currencysymbol": "<string>",  
  "locdate": "<date dd-MMM-yyyy>",  
  "loctime": "<24hourtime HH:mm:ss>",  
  "dwname": "<string>",  
  "days4minbookingdaycheck": "7",  
  "location": "<string>",  
  "locdatelocetime": "<date dd-MMM-yyyy>",   location current time  
  "tstavavailable": "<0/1>",  
  "ierrorcode": "<int>",  
  "availablemsg": "<string>"  
  "iafterhourpickup": "<0/1>",  
  "iafterhourdropoff": "<0/1>",  
  "tstminbookingday": "<int>",  
  "flightnoreqd": "<0/1>"  
}];
```

rcmAvailableCarDetails

```
rcmAvailableCarDetails = [{  
  "cattypeid": "<int_id>",  
  "column1": "<string>",  
  "available": "<0/1/2>", 0 = not available, 1 = available, 2 = might be  
  available (place a request)  
  "availablemsg": "<string>",  
  "statusid": "<1/2/3>",  
  "minbookingday": "<int>",  
  "ierrorcode": <int>,  
  "carsizeid": "<int_id>",  
  "categoryfriendlydescription": "<string>",  
  "bookingday": <date/time dd-mon-yyy hh:mm>  
  "carratestructureid": "<int_id>",  
  "discountname": "<string>",  
  "seasonid": <int>,  
  "season": "<string>",  
  "numofhours": "<real>",  
  "hourlyrate": "<money>",  
  "discounttype": "<p/d/f>",  
  "discountrate": "<money>",  
  "numofdays": "<real>",
```

```

"dailyrate": "<money>",
"rate": "<money>",
"rateafterdiscount": "<money>",
"wknddailyrate": "<money>",
"wkndrateafterdiscount": "<money>",
"weekendrate": "<money>",
  discountid:<int>,
  discountcode:<string>
}];

```

rcmAvailableCars

```

rcmAvailableCars =[{
  "cattypeid": "<int_id>",
  "column1": "<string>",
  "statusid": "<1/2/3>",
  "available": "<0/1/2>", 0 = not available, 1 = available, 2 = might be
  available (place a request)
  "availablemsg": "Available for booking",
  ierrorcode:<int>,
  minbookingday:<int>,
  minimumage:<int>,
  maximumage:<int>,
  "carsizeid": "<int_id>",
  "vehiclecategory": "<string>",
  "categoryfriendlydescription": "<string>",
  "numofhours": "<real>",
  "hourlyrate": "<money>",
  "numofdays": "<real>",
  "avgrate": "<money>",
  "totrate": "<money>",
  "discounteddailyrate": "<money>",
  "totalrateafterdiscount": "<money>",
  "totaldiscount": "<money>",
  "discontrate": "<money>",
  "discounttype": "<string>",
  "total": "<money>",
  "freedays": "<int>",
  "freedaysamount": "<money>",
  "imagenam": "<string>",
  "mobileimage": "<string>",
  "noadults": "<int>",
  "nochildren": "<int>",
  "nolargecase": "<int>",
  "nosmallcase": "<int>",
  "sippcodes": "<string>",
  "vehicledescription1": "<string>",
  "vehicledescription2": "<string>",
  "vehicledescription3": "<string>",
  "vehicledescurl": "<string>",
}]

```


rcmMandatoryFees

```
rcmMandatoryFees = [{
  "cattypeid": "<int_id>",
  "id": "<int_id>",
  "locationid": "<int_id>",
  "vehiclesizeid": "<int_id>",
  "numofdays": "<real>",
  "qty": "<int>",
  "fees": "<money>",
  "name": "<money>",
  "type": "<Daily/Fixed/Percentage>",
  "maxprice": "<money>",
  "stampduty": "<True/False>",
  "gst": "<True/False>",
  "percentagetotalcost": "<True/False>",
  "excessfee": "<money>",
  "merchantfee": "<True/False>",
  "fromage": "<int>",
  "toage": "<int>",
  "qtyapply": "<True/False>",
  "extradesc": "<string>",
  "extradesc1": "<string>",
  "extradesc2": "<string>",
  "extradesc3": "<string>",
}];
```

rcmOptionalFees

```
rcmOptionalFees = [{
  "cattypeid": "<int_id>",
  "id": "<int_id>",
  "orderno": "<string>",
  "igroupid": "<int_id>",
  "sgroupname": "<string>",
  "locationid": "<int_id>",
  "vehiclesizeid": "<int_id>",
  "numofdays": "<real>",
  "fees": "<money>",
  "name": "<string>",
  "type": "<Daily/Fixed/Percentage>",
  "maxprice": "<money>",
  "stampduty": "<True/False>",
  "gst": "<True/False>",
  "percentagetotalcost": "<True/False>",
  "excessfee": "<money>",
  "merchantfee": "<True/False>",
  "fromage": "<int>",
  "toage": "<int>",
  "qtyapply": "<True/False>",
  "extradesc": "<string>",
  "extradesc1": "<string>",
  "extradesc2": "<string>",
  "extradesc3": "<string>"
}];
```

rcmInsuranceOptions

```
rcmInsuranceOptions = [{
  "cattypeid": "<int_id>",
  "id": "<int_id>",
  "locationid": "<int_id>",
  "vehiclesizeid": "<int_id>",
  "numofdays": "<real>",
  "fees": "<money>",
  "name": "<string>",
  "type": "<Daily/Fixed/Percentage>",
  "maxprice": "<money>",
  "stampduty": "<True/False>",
  "gst": "<True/False>",
  "percentagetotalcost": "<True/False>",
  "excessfee": "<money>",
  "default": "<True/False>",
  "merchantfee": "<True/False>",
  "fromage": "<int>",
  "toage": "<int>",
  "qtyapply": "<True/False>",
  "extradesc": "<string>",
  "extradesc1": "<string>",
  "extradesc2": "<string>",
  "extradesc3": "<string>"},
  ]};
```

rcmKmCharges

```
rcmKmCharges = [{
  "cattypeid": "<int_id>",
  "numofdays": "<real>",
  "mileagedesc": "<string>",
  "id": "<int_id>",
  "kmsfree": "<int>",
  "addkmsfee": "<money>",
  "defaultkm": "<True/False>",
  "vehiclesizeid": "<int_id>",
  "fromday": "<int>", "today": "<int>",
  "locationid": "<int_id>",
  "maxprice": "<money>",

  "dailyrate": "<money>",
  "webavaliabile": "<True/False>",
  "daterange": "<True/False>",
  "pickupdatefrom": "<datetime>",
  "dropoffdateto": "<datetime>",
  "bookingdatefrom": "<datetime>",
  "bookingdateto": "<datetime>",
  "kfdropofflocationid": "<int_id>",
  "kmnotes": "<string>"
  }];
```

rcmRentalSource

```
rcmRentalSource = [{  
  "id": "<int_id>",  
  "rentalsource": "<string>",  
  "locationid": "<int_id>",  
  "default": "<True/False>"  
}];
```

rcmCountries

```
rcmCountries = [{  
  "id": "<int_id>",  
  "country": "<string>",  
  "default": "<True/False>"  
}];
```

rcmAreaOfUse

```
rcmAreaOfUse = [{  
  "id": "<int_id>",  
  "areaofused": "<string>",  
  "locationid": "<int_id>",  
  "defaulted": "<True/False>"  
}];
```

rcmUserData

```
rcmUserData = [  
  {"id": "<int_id>",  
   "firstname": "<string>",  
   "lastname": "<string>",  
   "license": "<string>",  
   "licenseexpiry": "<datetime>",  
   "licenseissued": "<string>",  
   "address": "<string>",  
   "suburb": "<string>",  
   "city": "<string>",  
   "state": "<string>",  
   "postcode": "<string>",  
   "email": "<email>",  
   "phone": "<string>",  
   "countryid": "<int_id>",  
   "dob": "<date>",  
   "loyaltycardno": "<string>"  
  }  
];
```

rcmHolidays

```
rcmHolidays = [{  
  "id": "<int_id>",  
  "locationid": "<int_id>",  
  "startdate": "<date dd-MMM-yyyy>",  
  "enddate": "<date dd-MMM-yyyy>",  
  "type": "<D/P P -> Pickup,D -> Dropoff>",  
  "weekdays": "<int>",  
  "holidayname": "<string>"  
}];
```

rcmAgentInfo

```
rcmAgentInfo = [{  
  "agentid": "<int_id>",  
  "agentcode": "<string>",  
  "agentbranchid": "<int_id>",  
  "agency": "<string>",  
  "agentbranch": "<string>",  
  "agentcollected": "<money>"  
}];
```

rcmErrors

```
rcmErrors = [{  
  "error": "<string>"  
}];
```

rcmPaymentInfo

```
rcmPaymentInfo = [{  
  "paymentinfo": "paymentinfo",  
  "citycode": "<string>",  
  "cashreceiptname": "<string>",  
  "otherextradesc": "<string>",  
  "paymenttype": "<string>",  
  "paid": " <money> ",  
  "paymentrefno": "<string>",  
  "sdmycreated": "<date dd/MMM/yyyy eq. 01/Sep/2016>",  
  "spaymentdate": "<date dd/MMM/yyyy eq. 01/Sep/2016>",  
  "paymentloginname": "<string>"  
}];
```

rcmLocationDetails

```
rcmLocationDetails = [{  
  "location": "<string>",  
  "address": "<string>",  
  "city": "<string>",  
  "postcode": "<string>",  
  "phone": "<string>",  
  "freecallocal": "<string>",  
  "email": "<string>",  
  "suburb": "<string>",  
  "statename": "<string>",  
  "fax": "<string>",  
  "country": "<string>",  
  "pickupavailable": "<True/False>",  
  "dropoffavailable": "<True/False>",  
  "minimunage": "<int>",  
  "minbookingday": "<int>",  
  "noticerequired": "<1/0>",  
  "officeopeningtime": "<True/False>",  
  "officeclosingtime": "<24hourtime - HH:mm>",  
  "afterhourbooking": "<True/False>",  
  "afterhourfeeid": "<int_id, 0=no fee>",  
  "unattendeddropoffs": "<True/False>",  
  "unattendeddropofffeesid": "<int_id, 0=no fee >",  
  "airportcodeid": "<int_id>"  
}];
```

rcmAgentBookings

```
rcmAgentBookings = [{  
  "bookingstatus": "<string:ONREQUEST/CONFIRMED/CANCELLED>",  
  "unallocatedbookingno": "<int>",  
  "quotation": "<True/False>",  
  "allocatereservationno": "<int>",  
  "referenceno": "<string>",  
  "pickupdate": "<date:Day dd/MMM/yyyy>",  
  "dropoffdate": "<date:Day dd/MMM/yyyy>",  
  "dateentered": "<date: dd/MMM/yyyy>",  
  "dropofflocation": "<string>",  
  "pickuplocation": "<string>",  
  "acfirstname": "<string>", "aclastname": "<string>"  
}];
```

rcmCancelReasons

```
rcmCancelReasons = [{  
  "id": "<int_id>",  
  "reason": "<string>"  
}];
```

rcmCancelInfo

```
rcmCancelInfo = [{  
  "success": "<True/False>",  
  "successmsg": "<string>",  
  "reservationref": "<string>"  
}];
```

rcmBookingInfo

```
rcmBookingInfo = [{  
  "bookinginfo": "bookinginfo",  
  "currencyname": "<string eq. USD>",  
  "currencysymbol": "<string eq. $>",  
  "reservationtypeid": "<int_id>",  
  "reservationno": "<int>",  
  "rcmreferenceid": "<string>",  
  "squotation": "<Yes/No>",  
  "quotation": "<True/False>",  
  "carsizeid": "<int_id>",  
  "driverage": "<int>",  
  "driverageid": "<int_id>",  
  "pickuplocationid": "<int_id>",  
  "dropofflocationid": "<int_id>",  
  "rentalsource": "<string>",  
  "sreservationtype": "<string>",  
  "bookrefno": "<string>",  
  "customerid": "<int>",  
  "spickupdate": "<date dd/MMM/yyyy eq. 01/Sep/2016>",  
  "pickuptime": "<24hourtime: HH:mm eq. 17:00>",  
  "sdropoffdate": "<date dd/MMM/yyyy eq. 01/Sep/2016>",  
  "dropofftime": "<24hourtime: HH:mm eq. 17:00>",  
  "sdateentered": "<date dd/MMM/yyyy eq. 01/Sep/2016>",  
  "pickupcity": "<string>",  
  "pickuplocationaddress": "<string>",  
  "collectionpoint": "<string>",  
  "arrivalflight": "<string>",  
  "dropoffcity": "<string>",  
  "returnpoint": "<string>",  
  "flightout": "<string>",  
  "dropofflocationaddress": "<string>",  
  "customernote": "<string>",  
  "notravelling": "<int>",  
  "areaofused": "<string>",  
  "campaigncode": "<string>",  
  "docprfix": "<string>",  
  "referenceno": "<string>",  
  "brandid": "<int_id>",  
  "squoteconverturl": "<string>",  
  "transmission": "<string>",  
  "imagenam": "<string>",  
  "category": "<string>",  
  "categoryspecial": "<string>",  
}
```

```

"vehicledesc": "<string>",
"vehicledesc1": "<string>",
"webimage": "<string>",
"mobileimage": "<string>",
"sippcodes": "<string>",
"sratetypedesc": "<string>",
"rentaldays": " <real> ",
"averagerate": " <money> ",
"totalcost": " <money> ",
"payment": " <money> ",
"balancedue": " <money> ",
"taxname1": "<string>",
"taxname2": "<string>",
"stampduty": " <money> ",
"gst": " <money> ",
"mileagedesc": "<string>",
"kmfeeid": "<int_id>",
"sselectedkmsoptions": "<string>",
"addkmscost": " <money> ",
"kmsfree": " <money> ",
"addkmsfee": " <money> ",
"kmddailyrate": " <money> ",
"maxkmscharge": " <money> ",
"totalkmddailyrate": " <money> ",
"totalkmsfree": "<int>",
"referralcommission": " <money> ",
"cuscompanyid": "<int_id>",
"imgdbpath": "<string>",
"carimage": "<string>",
"gstinclusive": "<Yes/No>",
"quotelettfile": "<string>",
"quotelettfile1": "<string>",
"quotelettfile2": "<string>",
"quotefile": "<string>",
"quotefile1": "<string>",
"quotefile2": "<string>",
"confirmatlett": "<string>",
"confirmatlett1": "<string>",
"confirmatlett2": "<string>",
"confirmfile": "<string>",
"confirmfile1": "<string>",
"confirmfile2": "<string>",
"agency": "<string>",
"agentbranch": "<string>",
"agencyname": "<string>",
"agentbranchid": "<int_id>",
"agentcommission": " <money> ",
"createemailtoagent": "<True/False>",
"agentemail": "<string>",
"noadults": "<int>",
"nochildren": "<int>",
"nolargecase": "<int>",
"nosmallcase": "<int>"
}];

```

rcmCustomerInfo

```
rcmCustomerInfo = [{  
  "customerinfo": "customerinfo",  
  "bookingcompanyname": "<string>",  
  "accountno": "<string>",  
  "billinggstno": "<string>",  
  "cbillingaddress": "<string>",  
  "cbillingphone": "<string>",  
  "cbillingemail": "<string>",  
  "cbillingcontact": "<string>",  
  "licenseexpirydate": "<date dd/MMM/yyyy eq. 01/Sep/2016>",  
  "dob": "<date dd/MMM/yyyy eq. 01/Sep/2016>",  
  "licenseno": "<string>",  
  "aclicenseissued": "<string>",  
  "acfirstname": "<string>",  
  "aclastname": "<string>",  
  "acphone": "<string>",  
  "acemail": "<string>",  
  "acfax": "<string>",  
  "localaddress": "<string>",  
  "acaddress": "<string>",  
  "acstreet": "<string>",  
  "acsuburb": "<string>",  
  "accity": "<string>",  
  "acstate": "<string>",  
  "acpostcode": "<string>",  
  "accountry": "<string>",  
  "cc_type": "<string>",  
  "acid": "<int_id>",  
  "ratediscountid": "<int_id>",  
  "acwebsite": "<string>",  
  "nosalestax": "<True/False>",  
  "loyaltycardno": "<string>",  
  "paymenttermsdesc": "<string>",  
}];
```


rcmCompanyInfo

```
rcmCompanyInfo = [{  
  "companyinfo": "companyinfo",  
  "logo": "<string>",  
  "companyemail": "<string>",  
  "companyname": "<string>",  
  "companycolour": "<hex colour code>",  
  "branchaddress": "<string>",  
  "branchcity": "<string>",  
  "branchpostcode": "<string>",  
  "branchphone": "<string>",  
  "branchfax": "<string>",  
  "website": "<string>",  
  "freephone": "<string>",  
}];
```

rcmRateInfo

```
rcmRateInfo = [{  
  "rateinfo": "rateinfo",  
  "ratename": "<string>",  
  "standardrate": " <money> ",  
  "days": "<int>",  
  "rate": " <money> ",  
  "dailyrate": " <money> ",  
  "discountperc": "<real>",  
  "discountid": "<int_id>",  
  "discounttype": "<string>",  
  "discountname": "<string>",  
  "triprates": " <money> ",  
  "nohours": "<real>",  
  "cost": " <money> ",  
  "totalratesvalue": " <money> ",  
  "season": "<string>",  
}];
```

rcmExtraFees

```
rcmExtraFees = [{  
  "extrafees": "extrafees",  
  "extrafeesid": "<int_id>",  
  "fees": " <money> ",  
  "extrarate": " <money> ",  
  "name": "<string>",  
  "extravalue": " <money> ",  
  "extradesc": "<string>",  
  "itemsupplied": "<True/False>",  
  "bond": "<True/False>",  
  "insuranceextra": "<True/False>",  
  "merchantfee": "<True/False>",  
  "singleexcess": " <money> ",  
}];
```

```
"msfapplyamount": " <money> ",
"extrafeesqty": "<int>",
"iextrafeedays": "<int>",
"qty": "<int>",
"optionalfee": "<1/0>",
"type": "<Daily/Percentage/Fixed>",
"percentagetotalcost": "<True/False>",
"cancellationfee": "<True/False>",
"merchantfee1": "<True/False>",
}];
```

rcmExtraDrivers

```
rcmExtraDrivers = [{
"ad_customerid": "<int>",
"ad_additionaldriverid": "<int>",
"ad_firstname": "<string>",
"ad_lastname": "<string>",
"ad_dob": "<date dd/MMM/yyyy eq. 01/Sep/1996>",
"ad_email": "<string>",
"ad_phone": "<string>",
"ad_mobile": "<string>",
"ad_address": "<string>",
"ad_city": "<string>",
"ad_state": "<string>",
"ad_countryid": "<int>",
"ad_countryname": "<string>",
"ad_licenseo": "<string>",
"ad_licenseexpiry": "<date dd/MMM/yyyy eq. 01/Sep/2016>",
"ad_licenseissued": "<string>"
}];
```

Troubleshooting

The first stage of your testing should be to make sure that your key is valid and working. You can do this at https://sandbox.rentalcarmanager.com/developer/livetest_3_1. Select your key from the list and click the Test button. You should receive a message saying “The key works!”. If you don’t, or if there is an error returned, then please contact support@rentalcarmanager.com as it is possible that there is an issue with your key.

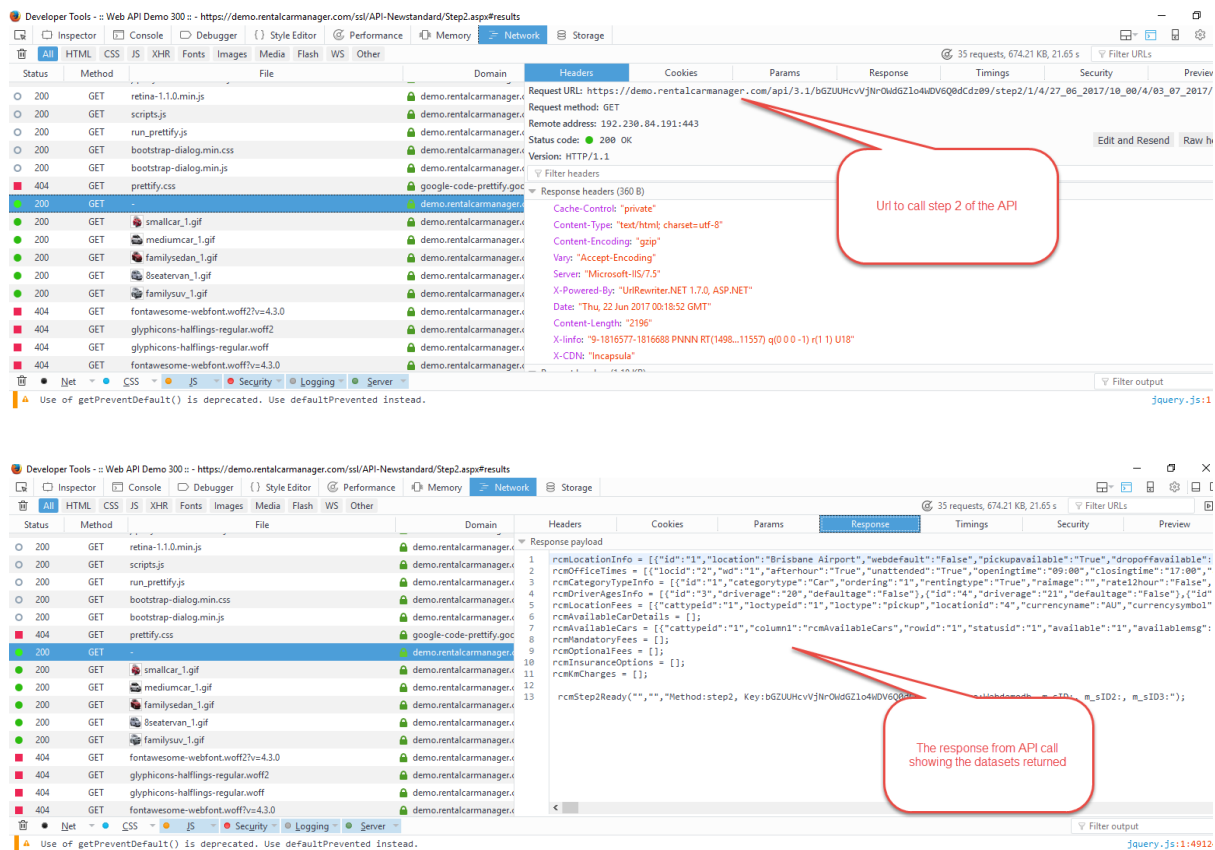
Any errors generated from your API calls will be returned in the rcmErrors dataset.

For more help with API implementation problems, please contact support@rentalcarmanager.com.

If you are having problems with the API please ensure that you include debugging information in your email.

For server side debugging, include details of the response that you are getting from our API server.

For client side debugging, you can use the developer tools in your browser. Here is an example of the network tool in Firefox showing an API call to step 2:



For more information about browser developer tools go to:

Internet Explorer: [How to use F12 Developer Tools to Debug your Webpages](#)

Firefox: [Firefox Developer Tools](#)

Chrome: [Chrome DevTools](#)

Google Analytics Ecommerce Tracking

For those developers that want to add Google Analytics Ecommerce tracking to the web booking module, here is some sample code to show how you can get the booking information and send it on to Google.

Google Tag Manager

1. Add code in the `$(document).ready` function to get the booking details and then call the function that creates the GTM datalayer:

```
$(document).ready(function () {  
  
    // Get the booking details  
    var resRef = document.getElementById('ReservationRef').value;  
    oAPI.OnReadyGetBookingInfo(getGoogleEcommerceData);  
    oAPI.GetBookingInfo(resRef);  
  
});
```

2. Create the `dataLayer` object and send it to GTM:

```
function getGoogleEcommerceData() {  
  
    var totalItems = rcmRateInfo.length + rcmExtraFees.length;  
    window.dataLayer = window.dataLayer || [];  
  
    var products = [];  
  
    // Add the rate items  
    for (var i in rcmRateInfo) {  
        products.push({  
            'sku': 'rate' + i,  
            'name': (rcmRateInfo[i]["name"] == null || rcmRateInfo[i]["name"] == '') ? 'Rental charges' : rcmRateInfo[i]["name"],  
            'price': rcmRateInfo[i]["dailyrate"],  
            'quantity': rcmRateInfo[i]["days"]  
        });  
    }  
  
    //Add the extra fee items  
    for (var i in rcmExtraFees) {  
        products.push({  
            'sku': 'extrafee' + i, // Product SKU - Type:String - Required  
            'name': rcmExtraFees[i]["name"], // Product Name - Type:String - Required  
            'price': rcmExtraFees[i]["extrarate"], // Product Price - Type:Numeric - Required  
            'quantity': rcmExtraFees[i]["qty"] // Product Quantity - Type:Numeric - Required  
        });  
    }  
  
    dataLayer.push({  
        'transactionId': rcmBookingInfo[0]["reservationno"],  
        'transactionAffiliation': rcmCompanyInfo[0]["companyname"],  
        'transactionTotal': rcmBookingInfo[0]["totalcost"],  
        'transactionTax': rcmBookingInfo[0]["gst"],  
        'transactionShipping': '0',  
        'transactionProducts': products  
    });  
  
    (function (w, d, s, l, i) {  
        w[l] = w[l] || []; w[l].push({  
            'gtm.start':  
                new Date().getTime(), event: 'gtm.js'  
        }); var f = d.getElementsByTagName(s)[0],  
            j = d.createElement(s), dl = l != 'dataLayer' ? '&l=' + l : ''; j.async = true; j.src =  
            'https://www.googletagmanager.com/gtm.js?id=' + i + dl; f.parentNode.insertBefore(j, f);  
    })(window, document, 'script', 'dataLayer', 'GTM-XXXXXX');
```

Google Analytics

1. Add code in the `$(document).ready` function to get the booking details and then call the function that creates the GA transaction:

```
$(document).ready(function () {  
  
    // Get the booking details  
    var resRef = document.getElementById('ReservationRef').value;  
    oAPI.OnReadyGetBookingInfo(getGoogleEcommerceData);  
    oAPI.GetBookingInfo(resRef);  
  
});
```

2. Add the transaction and items to the GA object and send it to Google:

```
function getGoogleEcommerceData() {  
  
    ga('require', 'ecommerce');  
  
    // Add a transaction to the ecommerce object  
    ga('ecommerce:addTransaction', {  
        'id': rcmBookingInfo[0]["reservationno"], // Transaction ID. Required.  
        'affiliation': rcmCompanyInfo[0]["companyname"], // Affiliation or store name.  
        'revenue': rcmBookingInfo[0]["totalcost"], // Grand Total.  
        'shipping': '0', // Shipping.  
        'tax': rcmBookingInfo[0]["gst"] // Tax.  
    });  
  
    // Add the rate items  
    for (var i in rcmRateInfo) {  
        ga('ecommerce:addItem', {  
            'id': rcmBookingInfo[0]["reservationno"], // Transaction ID. Required.  
            'name': rcmRateInfo[i]["ratename"], // Product name. Required.  
            'sku': 'rate' + i,  
            'price': rcmRateInfo[i]["dailyrate"], // Unit price.  
            'quantity': rcmRateInfo[i]["days"] // Quantity.  
        });  
    }  
  
    // Add the extra fee items  
    for (var i in rcmExtraFees) {  
        ga('ecommerce:addItem', {  
            'id': rcmBookingInfo[0]["reservationno"], // Transaction ID. Required.  
            'name': rcmExtraFees[i]["name"], // Product name. Required.  
            'sku': 'extrafee' + i,  
            'price': rcmExtraFees[i]["extrarate"], // Unit price.  
            'quantity': rcmExtraFees[i]["qty"] // Quantity.  
        });  
    }  
  
    // Send the transaction info to Google  
    ga('ecommerce:send');
```

API Error Codes

The following table lists the error codes that may be returned from the RCM APIs. An error message is also included in this listing.

Error Code	Error Messages
001	Invalid Agent Code
002	Invalid Date Format for Pickup/DropOff Date. Format dd_MM_yyyy

003	Invalid Time Format for Pickup/DropOff Time. Format HH_mm
004	Return Date is earlier then Pick up date.
005	This location has a minimum renting age of <age> years.
006	Reservation requests made for <pickup location> must be made <min notice period > days or...prior to vehicle pick up.
007	The minimum Rental period for <pickup location> Internet booking is <minimum booking> days.
008	Minimum rental period for and booking is <days>...in details
009	Minimum rental period for and booking is <days>
010	One way booking not possible
011	Requested pickup date is unavailable
012	Requested DropOff date is unavailable
013	Pickup/DropOff Locatlion <PickupLocation>/<DropOffLocation> will not take bookings outside office hours(<PickupOfficeOpeningTime>/<DropOffOfficeOpeningTime> - <PickupOfficeClosingTime>/<DropOffOfficeClosingTime>).Please contact the bookings office directly.
014	Driver's Age cannot be empty
015	Pick up Location cannot be empty
016	Drop Off Location cannot be empty
017	Category type ID cannot be empty
018	Error:18. Please contact customer care in case you receive this error
019	Agent Code Cannot be blank
020	Minimum age to rent this category of vehicle is <age>
021	Invalid Reference Key or Session Timed Out.
022	First Name cannot be blank.
023	Last Name cannot be blank.
024	County ID cannot be blank
025	Email ID cannot be blank.
026	Invalid Email ID.
027	No of persons travelling cannot be blank or it should be a number.
028	Invalid Country ID
039	Reference Key cannot be blank
040	CarSizeID cannot be blank.
041	Invalid CarSizeID.
042	You must specify a valid email address of the Customer.
043	Invalid Secure Key
44	Invalid Company ID
45	Please enter either BookingDateFrom-BookingDateTo or ReservationNo.
46	ReservationNo needs to be a number.
47	BookingDateFrom or BookingDateTo is of invalid dates format. Correct format is dd_MM_yyyy.
48	ReservationStatus cannot be blank.
49	Sorry no records found.
50	CancellationReasonID cannot be blank.
51	CancellationReasonID has to be a number. Please check the Cancel Reason List.
52	ReservationNo cannot be blank.
53	ReservationNo has to be a number.
54	This is already cancelled booking.
55	Sorry, this AgencyCode is not authorised to cancel this booking.
56	Sorry, cannot cancel this booking. Booking should be 1 day before pickup.
55	SecureKey cannot be blank
58	Pickup Date cannot be blank
59	Dropoff Date cannot be blank

60	Agent Name cannot be blank
61	Pickup time cannot be blank
62	Dropoff time cannot be blank
63	CarID is not Available
64	Please enter ReturnedDateFrom and ReturnedDateTo
65	ReturnedDateFrom or ReturnedDateTo is of invalid date format. Correct format is dd_MM_yyyy.
66	ReturnedDateTo is earlier than ReturnedDateFrom.
67	CarID is not Available. Please call the method 'confirmVehicleSelection' before calling this.
68	Please call this method after confirmVehicleSelection
69	InsuranceID is not Available.
70	Please enter given InsuranceID
71	ExtraFeeID is not Available
72	KmsID is not Available
73	Please enter given KmsID
74	Unfortunately the pickup or dropoff time you have selected is outside our opening hours [Weekday] [OfficeOpeningTime] - [OfficeClosingTime]. Please select a new time then click continue.
75	Please insert correct 'BookingType' (integer). It could be either '1' for quote or '2' for booking.
76	Country ID should be a number.
77	Please call the 'requestCustomerCountryDetails' method for correct Country ID.
78	Sorry this Vehicle is Not Available.
79	Please select a car.
80	AreaofUse is not Available.
81	Please enter given AreaofUsedID
82	You do not have the permission to access this web-service method.
83	BrandID should be a number.
84	Country Id should be in the Country List provided
85	Location Id cannot be blank.
86	Location is not a renting location.
87	LocationID should be a number.