

Motors - API Technical Documentation

Document Version	Author	Date Published	Notes
v1.0.0	Manjula Chetty	05/04/2023	Initial
v1.0.1	Daniel Perry	27/04/2023	Initial
v1.0.2	Daniel Perry	10/05/2023	Added Sandbox Firewall info. Removed Deleted status references (move to ListingStatus). Updated Metadata names. Added refresh token info. Added support for the motorcycle category.
v1.0.3	Max McLaren	02/06/2023	Added finance offer info.
v1.0.4	Daniel Perry	20/06/2023	Added car category. Removed 'IsWof', 'IsNew' (replaced with ConditionId), 'IsDamaged', 'IsSpecial' flags.
v1.0.5	Daniel Perry	07/07/2023	Added Manufacturer/Model/Type names to MakeDetails response. Corrected naming of ManufactureId -> ManufacturerId
v1.1.0	Daniel Perry	13/07/2023	Introduced Branch Service.
v1.1.1	Max McLaren	25/07/2023	Added truck category. Added Conditions and ExcludesGstSupportedCategories to metadata resource. Added category restriction notes to PriceDetails.
v1.1.2	Daniel	02/08/2023	Added the Bus category.

	Perry		
v1.1.3	Daniel Perry	11/08/2023	Added info about caching clean car attributes.
v1.1.4	Daniel Perry	21/08/2023	Added info about Dealerbase environments. Metadata <i>conditions</i> are now provided with category context. Metadata service Manufacturer endpoint now will return a list of Certified Pre-Owned Programs if supported by the Manufacturer. Added 'CertifiedPreOwnedProgramId' to CarRequest so that a listing that has a condition of Certified Pre-Owned is mandatorily associated with a Pre Owned program.
v1.1.5	Martin Hing	22/08/2023	Added the Boat category.
v1.1.6	Ernesto Orellana	18/09/2023	Added information for vehicle response (external media) and Vehicle External Media delete method.
v1.1.7	Daniel Perry	20/09/2023	Added Branch resource. Added Delete 'All' Images endpoint. Added 'Replace an image' endpoint. Added 'Compact' mode to Get Vehicles endpoint. Added 'CategoryId' query to Metadata endpoint to scope metadata by category.
v1.1.8	Martin Hing	22/09/2023	Added the machinery category.
v1.1.9	Daniel Perry	09/11/2023	Added Checksum to Image Responses.
v1.2.0	Daniel Perry	16/11/2023	Added Settings, SuperFeatureTotal, SuperFeatureUsed, ShowroomCount to Branch endpoint.
v1.2.1	Max McLaren	24/11/2023	Removed the company-level stock number uniqueness validation.
v1.2.2	Daniel Perry	08/02/2024	Added information about transferring a vehicle to another branch.

v1.2.3	Daniel Perry	27/02/2024	Document Special Offers resource now supports image upload.
v1.2.4	Daniel Perry	04/03/2024	Information about Category 0 (Unassigned).
v1.2.5	Daniel Perry	03/05/2024	Added information around properties sensitive to package / product information and where to look on the Branch Service to govern your decisions around setting the appropriate values. Added PHP Basic Sample to demonstrate basic functionality.
v1.2.6	Daniel Perry	16/05/2024	Added info around new conversational metadata endpoints.
v1.2.7	Daniel Perry	14/06/2024	Important: VIN Number system constraints.
v1.2.8	Daniel Perry	25/06/2024	Notes on Deactivation.
v1.2.9	Daniel Perry	04/07/2024	Notes on the 'Sync' process and Listing Status'.
v1.2.10	Max McLaren	14/08/2024	Updated odometer requirements for Brand New and Ex-Demo vehicles.
v1.3	Daniel Perry	27/08/2024	Added information about requesting models not present when requesting metadata.
v1.4	Daniel Perry	14/10/2024	Notes around vehicle counts, partner program and active partner vehicles. (See Branch resource). Crossed out mentions of SuperFeature (Deprecated 01/07/2024)
v1.4.1	Max McLaren	20/01/2025	Add information for authenticating with the client credentials grant.
v1.4.2	Max McLaren	09/04/2025	Clarify swagger access.
v1.4.3	Dan Perry	16/07/2025	Add new Feature Collection information.
v1.4.4	Dan Perry	21/10/2025	Added more Feature Collection information. Added new properties supported on Vehicle

			Request that were considered Features.
v1.4.5	Dan Perry	04/06/2026	Added Watchlist Offer information.

Purpose

The purpose of this document is to describe the locations, capabilities and limitations of the Trade Me Motors REST API as well as provide a general overview of listing a vehicle and have it appear on the main Trade Me site.

The author assumes the reader is familiar with basic software development principles and methodologies.

Motors - API Technical Documentation.....	0
Purpose.....	3
Locations.....	8
Services.....	8
Service Versions.....	8
Resources.....	8
Sandbox Access.....	9
Production Access.....	9
Authentication.....	10
Authentication Grants.....	10
password.....	10
refresh_token.....	10
client_credentials.....	11
Authentication Body / Credentials.....	11
Authentication Response.....	12
Client Generation.....	12
Dealerbase.....	13
Listing Service.....	14
Design.....	14
Vehicle Resource.....	15
Description.....	15
Endpoints.....	16
Add / Update flow.....	17
The 'Sync' Process.....	18
VIN (Vehicle Identification Number) System Constraints.....	19
The Vehicle Structure.....	20
Terms and Conditions Checks.....	24
Auxiliary Details.....	25
Delivery Details.....	26
Links.....	27
Vehicle Make.....	27
Price Details.....	28
Listing a Campervan.....	29
Listing a Motorcycle.....	31
Listing a Car.....	32
Franchise Dealer and Cars.....	33

Listing a Truck.....	34
Listing a Bus.....	35
Listing a Boat.....	37
Listing Machinery.....	38
Identification Details.....	39
Weight Details.....	40
Vehicle Measurements.....	40
Vehicle Response.....	41
External Media.....	42
Listing Status.....	42
Vehicle Make.....	43
Getting a listing by stock number.....	43
Getting vehicle listings for your branch.....	43
Response.....	44
Updating a listing.....	44
Patching a listing.....	44
Activating/Deactivating a listing.....	45
Transferring a vehicle to another branch.....	45
Adding an image to a listing.....	46
Payload:.....	46
Response.....	46
Getting an image.....	47
Updating an image's data.....	47
Getting all images.....	48
Deleting an image.....	48
Deleting all images for a listing.....	48
Deleting external media.....	48
Updating image orders.....	48
New Vehicle Features.....	49
Creating a new feature collection.....	50
Updating an existing feature collection.....	52
Deleting an existing feature collection.....	52
Retrieving an existing feature collection.....	52
Metadata Resource.....	53
Description.....	53
Endpoints.....	53
Retrieving metadata.....	54

Retrieving a paged list of manufacturers, models or vehicle types.....	55
Response.....	56
Retrieving detailed metadata information.....	57
Proposed Usage:.....	58
Feature Collection Metadata.....	58
ModelQueue Resource.....	60
Description.....	60
Endpoints.....	60
Requesting to add a model to the system.....	60
Branch Service.....	62
Description.....	62
Branch Resource.....	62
Description.....	62
Endpoints.....	62
Retrieving information about your branch.....	62
Branch Properties that dictate Vehicle Payload Composition.....	64
Vehicle Allowance.....	65
Watchlist Offer Resource.....	66
Description.....	66
Endpoints.....	66
Creating a Watchlist Offer.....	67
Retrieving Watchlist Offer Summaries.....	68
Special Offers Resource.....	69
Description.....	69
Endpoints.....	69
Creating/Updating a Special Offer.....	69
Response.....	70
Finance Offers Resource.....	70
Description.....	70
Endpoints.....	71
Creating/Updating a Finance Offer.....	71
Response.....	71
API Errors.....	72
Error Status Codes.....	72
Code Samples.....	73
PHP.....	73

Locations

The Trade Me Motors REST API is located in the following publicly accessible environments. Access to the Sandbox environment is by request.

Environment	URI
Production	https://restapi.dealerbase.co.nz/
Sandbox	https://restapi.sbx.dealerbase.co.nz/

Services

Within each environment exist one or many services available on the API.

Service	Purpose	Relative Uri
Listing	Manage vehicle listings and their images.	listing
Branch	Manage branch related data.	branch

An OpenAPI tool is available at [URI]/[Service]/swagger/index.html i.e. the swagger for the Listing Service in prod is: <https://restapi.dealerbase.co.nz/listing/swagger/index.html>

Service Versions

Within each service there are one or many API versions.

Service	Version	Relative Uri
Listing	1	v1
Branch	1	v1

Resources

Within each service/version there are one or many resources available for management.

Service	Version	Resource	Accepts	Produces
Listing	1	vehicles	application/json	application/json

Listing	1	metadata	application/json	application/json
Branch	1	special-offers	application/json	application/json
Branch	1	finance-offers	application/json	application/json
Branch	1	<i>empty</i>	application/json	application/json

Example:

To access version 1 of the vehicles resource available in the listing service in the sandbox environment the resulting url would be: <https://restapi.sbx.dealerbase.co.nz/listing/v1/vehicles>

Sandbox Access

Email motorsapi@trademe.co.nz and provide Trade Me with the following:

- A static IP range we can add to our whitelist,
- The dealer branch code/s you wish to begin testing with,
- Your contact details including key contact person, email and phone

We will set up your authentication to the sandbox and advise you of your **client ID**, **password** and **client secret**. This will give you access to the sandbox.

Production Access

Once you've completed testing in the Sandbox environment and are ready to promote to production there are a few things you'll need to do:

- Obtain a production client ID and secret from us.
- The dealer branch code/s you wish to go live with and the date you wish to move to Production. If you are doing a gradual rollout branch by branch then you will need to let us know before you promote each branch so we can ensure a smooth transition.
- Stop sending FTP files.

Production doesn't require a whitelisted IP.

Authentication

In order to access any resource on the Trade Me Motors Rest API a caller must first authenticate with a *separate* service in order to obtain an access token. This token is then used on all subsequent requests to the REST API for 30 minutes until it expires.

Below is the Authentication Service locations for each environment:

Environment	URI
Production	https://auth.dealerbase.co.nz/
Sandbox	https://auth.sbx.dealerbase.co.nz/

To authenticate - an HTTP POST request must be made to this 'token' resource containing your credentials. The request must have a content type of application/x-www-form-urlencoded and credentials are posted in the body of the request. The server will produce an application/json result (so remember to add the Accepts header to your request).

Authentication Grants

A token can be retrieved using the following grants: password, refresh_token, and client_credentials.

password

The 'password' grant provides the caller with a token that can be used to access resources that are owned by a specific branch; most importantly, vehicles. This grant requires that the caller provides the username and password of the branch, alongside the caller's client id and secret. The response contains the access token to be used in calls to the main services, as well as a refresh token for use in obtaining a new access token. **Note: Only 1 refresh token is issued per client_id / branch.**

Upon expiration the consumer will receive a 401 UNAUTHORISED HTTP code. At this point the consumer should generate a new token using the 'refresh_token' grant as described below.

refresh_token

The 'refresh_token' grant provides the caller with a way to generate a new access token for a branch without needing to resupply the branch's username and password. This grant requires that the caller provide their client id and secret, alongside a refresh_token which is acquired

from the 'password' grant. The 'refresh_token' grant can be repeatedly used to generate a new access token until the refresh token expires.

Upon expiration of the refresh token, the request to refresh the token will result in a 400 BAD REQUEST. At this point, the caller will need to authenticate with the 'password' grant again to acquire a new refresh token.

client_credentials

The 'client_credentials' grant provides the caller with a token that can be used on resources not owned by a branch, most notably, metadata resources. This grant only requires that the caller provide their client id and secret. Attempting to use a token generated with this grant to access a branch-owned resource, such as a vehicle, will result in a 403 FORBIDDEN HTTP code.

Upon expiration the consumer will receive a 401 UNAUTHORISED HTTP code. At this point the consumer should generate a new token, again using the 'client_credentials' grant.

Authentication Body / Credentials

Below is a table of the required body properties when posting to the 'token' endpoint.

Property	Value	Description
grant_type	password / refresh_token / client_credentials	Must be 'password' when using username/password, must be 'refresh_token' when using your refresh token, must be client_credentials when authenticating without a branch - describes the specific OAuth flow.
username	'Your branch number'	Important - the value of this property must be the number associated with the branch that you are wanting to manage data for. See your account manager for this value. This is only required when using the 'password' grant
password	'Your branch FTP password'	This value is the password used for your branch when FTPing vehicle listing data to Trade Me. This is only required when using the 'password' grant
client_id	'Your API client id'	This value is your provided client ID - see your account manager for this information.

client_secret	'Your API client secret'	This value is your provided client secret - see your account manager for this information.
refresh_token	'The refresh token from your password flow response'	This value is required when refreshing your access token. This is only required when using the 'refresh_token' grant

Authentication Response

As a result of calling the 'token' endpoint successfully you will receive the following JSON structure denoting your token for you to redeem when accessing the REST API.

Property	Description
access_token	This is the token you will use in your Authorization HTTP header for API access. Valid for 30 minutes.
expires_in	A value (in seconds) denoting how long the token is valid for before you will receive a 401 (Unauthorised) HTTP response code attempting to use it.
refresh_token	A token that may be used to redeem a new access token after access token expiration. Valid for 7 days. This is not returned when using the 'client_credentials' grant.
token_type	A value denoting the authentication scheme.

Client Generation

Please note that you can download the OpenAPI definition from the mentioned OpenAPI endpoints and use these to generate the model definitions required when interacting with the REST API. This is particularly useful for helping with vehicle category determination.

Please use: <https://editor.swagger.io/>

Dealerbase

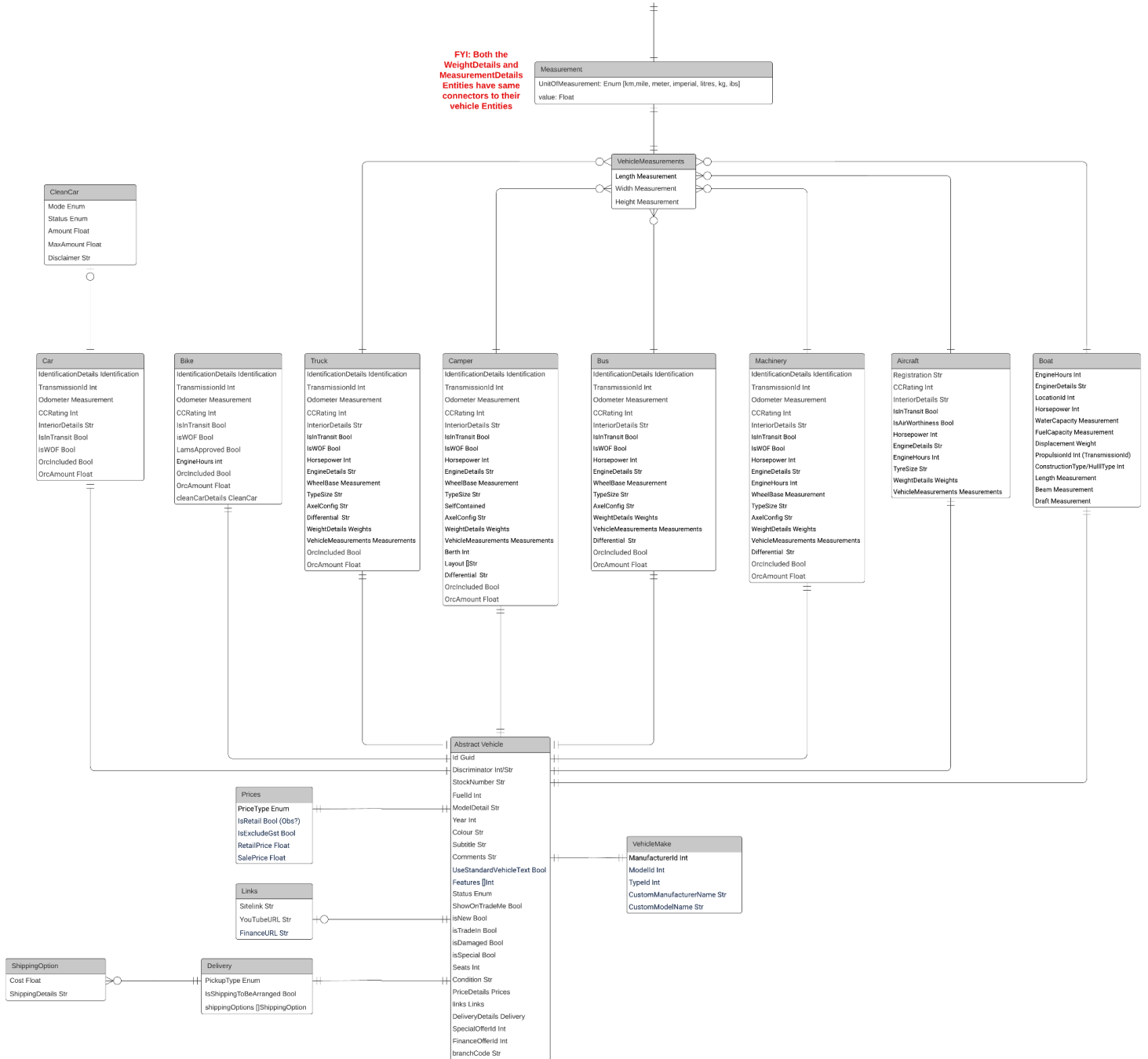
You can verify that a listing that you have added/modified via the API has made it into the API *visually* by visiting a Dealerbase environment. For most cases, if the API accepts your listing details and provides a 200 response then it will meet most of the requirements for a listing to be in Dealerbase. You can then poll the HTTP GET `/listing/v1/vehicles/{stockNumber}` endpoint and pay close attention to the *ListingStatus* property on the response to determine where abouts in the Trade Me syncing process a listing is. **If you do wish to see the listing in Dealerbase then you will need Staff credentials for the branch you are managing via the API. These can be requested via our Motors Customer Support team for a specific branch you are testing with.**

Environment	URI
Production	https://www.dealerbase.co.nz/
Sandbox	https://sbx.dealerbase.co.nz/

Listing Service

Design

FYI: Both the WeightDetails and MeasurementDetails Entities have same connectors to their vehicle Entities



Vehicle Resource

Description

The purpose of the Listing Service is to allow a branch/consumer to add new Motor listings to Trade Me as well as manage existing listings. Listing is a two part process consisting of: 1) The vehicle is created and 2) Images are 'added' to the listing - the listing *must* have at least one image on it at all times after the first image has been added. Once the two part process has been completed the listing will be submitted to a 'queue' where in which it will be processed and pushed to the Trade Me site - this means that there is a delay between submitting those images (in step two) and seeing your listing on the Trade Me site.

Listing a new motor vehicle is done by posting a vehicle payload into the 'vehicles' resource. As there are multiple types of vehicles - the API requires you send in a structure unique to the type of vehicle you are wanting to list as *well as* setting the appropriate category value on the request.

The category of your vehicle is determined by the Manufacturer, Model and Vehicle Type combination. E.g A Toyota Rav 4 - 4 Door SUV is a Car type. When sending in this combination - all properties specific to a Car type will be accepted. The payload type you are sending in is validated against the calculated category, so sending in a Campervan structure with a Car category value will result in an error.

To determine the category value for your payload you need to request reference data from the metadata service. Doing so will provide you with (amongst other things) a list of manufacturers.

You must search the manufacturer data for your desired manufacturer. You can search the manufacturer list and compare the *Name* property. Only if no manufacturers are found by *Name* - you can search the manufacturer list and compare the *ExternalNames* list that is provided on each manufacturer. This mapping list contains a list of known variations in spellings for the manufacturer name.

After resolving a manufacturer you need to issue an HTTP GET request to the 'metadata/manufacturers/{id}' resource (passing the manufacturer ID you resolved) to get the complete manufacturer details for your given manufacturer (the manufacturer details from the 'metadata' resource are rudimentary due to scalability reasons).

Once you have obtained the manufacturer details from the manufacturer endpoint you must use the same resolution strategy in discovering the *model* structure as you did with the manufacturer by interrogating the manufacturer's model collection.

Once you have obtained the model details from the manufacturer structure you must use the same resolution strategy in discovering the *vehicle* type structure as you did with the model.

Once you have obtained the *vehicle* type you have access to the 'SupportedCategories' for your vehicle type. There will only ever be one supported category and this is the category value you use when setting the category value on the payload as well as what properties will be recognized and acknowledged by the Listing API.

The metadata resolution strategy defined above will aid in resolving many of the reference identifier requirements present on the vehicle payload.

If you find that you can't find a Manufacture, Model, VehicleType, Transmission or Feature either by Name or ExternalName then you should let our Customer Experience team know. It may be possible to add it to the system, or at least add the external name to the system so that mapping is possible for you. [New models can be added to the system via the Model Queue resource.](#)

Endpoints

HTTP Verb	URI	Effect
POST	vehicles	Creates a new vehicle for your branch.
GET	vehicles	Retrieves a paged list of vehicles belonging to your branch.
PUT	vehicles/{stockNumber}	Updates an existing vehicle for your branch with the given stock number. (Important: State replacement). Can specify query string <i>?ignoreAuxiliaryDetails=true</i> to omit AuxiliaryDetails from the state.
GET	vehicles/{stockNumber}	Retrieves the vehicle for the given stock number within your branch.
PATCH	vehicles/{stockNumber}	Patches specific attributes of a listing (limited to a subset of properties - not a state replacement)
POST	vehicles/{stockNumber}/images	Adds a new image to an existing listing.
PUT	vehicles/{stockNumber}/images/{id}	Replaces the image data for the given image id.
GET	vehicles/{stockNumber}/images	Gets all image metadata associated with an

		existing listing.
GET	vehicles/{stockNumber}/images/{id}	Get metadata for an image with the given id for an existing listing with the given stock number.
DELETE	vehicles/{stockNumber}/images	Deletes all but n images for a given listing.
DELETE	vehicles/{stockNumber}/images/{imageId}	Delete an existing image (by imageId) from a listing (by stock number)
PUT	vehicles/{stockNumber}/image-orders	Re-order the existing order of images for a given listing.
GET	vehicles/{stockNumber}/features-collection	Get the curated features for a vehicle.
POST	vehicles/{stockNumber}/features-collection	Curate a new features collection..
PUT	vehicles/{stockNumber}/features-collection	Update an existing features collection curation.
DELETE	vehicles/{stockNumber}/features-collection	Delete an existing feature collection curation.

Note: For all endpoints accepting a stock number - your stock number must be URL encoded. All stock numbers with a forward slash in them need to be *double-encoded*. All stock numbers with a space character in them need to be encoded with a replacement of '%20'. **No stock numbers are allowed to have %2F contained in their value.**

Add / Update flow.

When creating or updating a vehicle listing you should follow this flow.

- 1) Formulate your vehicle payload based on category (Using the metadata resource).
- 2) Check if the vehicle exists first (for your given stock number), if so then call the PUT with your payload, otherwise call the POST
- 3) If your vehicle didn't exist then add your images to the listing via the POST. If your vehicle **did** exist then you can choose to clear out your images (except 1) via the DELETE images endpoint and update the last one with the PUT Images/{id} endpoint OR you can retrieve all the images for the vehicle from the GET Images endpoint and

enumerate over them calling the PUT Images/{id} to replace the image data for that image/display order.

The 'Sync' Process.

After you have sent a created/modified a vehicle in some way (whether it be the vehicle itself, or adding images etc) the change is applied to the record in Dealerbase and the change enters a 'queue' to be 'synced to Trade Me' so we can either get the listing up on site (if it's new or an activation), modify the existing listing in place (if it is an edit) or withdraw the listing (if it's a deactivation). So your changes in Dealerbase are instant, however **please allow roughly 10 minutes** (on average, depending on traffic) for the change to appear on the main Trade Me site.

A successful 'sync' is subject to a few pieces of criteria:

- The Listing API endeavours to enforce the core rules that Trade Me enforce for listings in motor vehicle categories - sometimes time-based rules can cause listings updates to not sync if a listing errors have not been addressed in a timely manner in Dealerbase. An example of this is our rule around not being able to modify the odometer on a listing by more than 1000kms after 7 days. If a listing is created with the wrong odometer value but fails to sync for another reason and that reason is not addressed in a timely manner then this may affect the listing odometer being changed in the future. For most cases a call to our Customer Experience team should be able resolve the issue.
- The availability of the Trade Me API. **The sync process is resilient and employs a retry mechanism over the next hour.**

To work out the status of a listing we provide a "ListingStatus" property on a vehicle response from the API. This structure provides information about the status of a listing based on the state of the listing combined with its attempt to sync to Trade Me. The following are possible Status' that can be returned and their meaning in the Dealerbase world:

- Registered - The listing is in Dealerbase but has yet to be synced to Trade Me. A listing may remain in this state if you have set *ShowOnTradeMe* to *false* (and a message will be returned in the Message property for this particular state)
- Active - The listing is live on Trade Me. The *ListingId* property will be populated in this state and you can navigate to the listing by using the interpolation <https://www.trademe.co.nz/a/{listingId}>
- Ended - The listing has been deactivated. The *ListingId* is populated with the ID at time of deactivation. To end a listing you must issue a deactivation (described further down in this document)

- Reactivated - The listing was previously deactivated and now has been activated again and is live on Trade Me. **Please Note: Activating a listing after it has been deactivated will change the ListingId.**
- Failed - The most recent update has failed to sync to Trade Me. The reason why will be in the *Message* property. It's important to display this status to a user in a UI so that they can manually try to fix issues. This will decrease the need for intervention which can take time.

There are scenarios where a listing will be *Failed* but only for a finite amount of time as the sync process tries to auto-heal the type of error - these scenarios are as follows (and not limited to):

- Dealerbase has a model in its cache that it sends to Trade Me for the listing but Trade Me itself doesn't know about the model. The error message returned for the next 10 minutes is "*For model : {model name} is not valid for this category*". In the interest of the Dealer we attempt to self heal here by placing it in an *Other* model bucket under the manufacturer url space on Trade Me.
- If a Listing has been withdrawn from Trade Me for a certain amount of time and you attempt to activate it again you will see a failed status of "*Listing not found*". We will attempt to self heal in this situation to get the listing back on line.

VIN (Vehicle Identification Number) System Constraints.

Please read: <https://www.nzta.govt.nz/vehicles/vehicle-registration/vin/>

For the Listing Service - when you add a vehicle to your branch, or update an existing vehicle, the primary identifier that we leverage is the **stock number**. A stock number must be unique across your whole branch for active and inactive listings.

However, a VIN number is a vehicle fingerprint that uniquely identifies a vehicle in NZ and across the world. We **also** require that VIN's entered for your branch's vehicles also be unique, even for deactivated vehicles.

There are scenarios where you may run into conflicts with VIN numbers:

- A dealership/branch sells a vehicle with Stock Number *QRSTUV*, VIN *ABC123*, the dealer/VMS deactivates the listing as it has sold. The customer changes their mind and either returns the vehicle or trades it in and the dealer wants to add the vehicle back to the system but *under a different stock number*. The Listing Service **will reject** the request to add the new vehicle to the branch with a different stock number **as a vehicle already exists (deactivated) for the branch with the existing VIN.**

- The branch has sent in **more than 1** vehicle with the same VIN (but different stock numbers) via the recently decommissioned FTP system where validation was more relaxed. This isn't a very common situation but still arises.

How to handle the above scenarios:

- **First and foremost** - let us know when you're ready to move into the production environment - we will provide a list of vehicles with the same VIN number that have crept in via the FTP system, you tell us what ones aren't used and we will get rid of them. This solves the second bullet point above.
- **Secondly** - Apply the following flow that meets your situation.
 - As per standard process do a GET on the Stock Number to determine if you need to POST or PUT.
 - If it doesn't exist, issue a POST.
 - If it does exist:
 - If it is deactivated then issue a PUT with an Status of *Activate* and the rest of your payload.
 - If it is active, just apply your payload.
 - It's at this point that we will tell you if the VIN is a duplicate. You will receive a 409 Conflict status code.
 - You can inspect the *conflictingStockNumbers* property on the returned error details to determine what *other* vehicle has the conflict.
 - Issue a GET passing the conflicting stock number.
 - If it is deactivated then issue a PUT with an Status of *Activate* and the rest of your payload.
 - If it is active, just apply your payload.
 - You can update the stock number to be the new stock number the dealer has requested.
- **Note:** Listings that have been deactivated for over 120 days will be deleted from the system, the above flow should compensate for that check.

The Vehicle Structure.

When creating or updating a vehicle listing a vehicle payload is required. The vehicle itself is abstract, and its specific categories are determined by its descendent structures (and category value).

The following are the currently supported categories, further categories will be released as the API evolves:

CategoryId	Description	Supported
------------	-------------	-----------

0	Unassigned. Note: This has entered the system via FTP and the category is unable to be determined.	Read Only
2	Car	Yes
3	Bus	Yes
5	Motorcycle	Yes
6	Truck	Yes
7	Camper / Caravan / Motorhome	Yes
8	Machinery	Yes
9	Boat	Yes

Below is a table of the properties common to all vehicle payloads. All data validation requirements can be found at the OpenAPI location.

Important: Some properties on the vehicle request payload are sensitive to the package (a group of products) the branch is on. The Extra Info/Rules section will be highlighted if this is the case.

The Branch Service base resource provides you with available products/attributes that you can then use to make decisions on whether a vehicle payload property can be set or not, or what value it is restricted to.

Name	Type	Purpose	Extra Info / Rules
AaAppraised	Primitive	Is the vehicle AA Appraised?	Not supported for 'Boat' and 'Motorcycle' categories.
AuxiliaryDetails	Object	See table below	See table below. Note that during PUT requests this field can be ignored from the request by specifying <code>ignoreAuxiliaryDetails=true</code>.
AvailabilityStatus	Primitive	To set the 'status' of the	For Franchise Dealers this property is sensitive for Car categories.. Please

		vehicle.	use the Branch Service Products property to govern your decisions.
CategoryId	Primitive	To denote the vehicle category.	All categories discoverable using 'metadata' resource, only some supported so far. This is resolved by determining your Manufacturer/Model/VehicleType from the 'metadata' resource - the VehicleType object will provide the category.
Color	Primitive	Vehicle color.	
Comments	Primitive	Main body part of listing.	<p>Can't contain contact information.</p> <p>Important:</p> <ul style="list-style-type: none"> Carriage Returns are replaced with empty strings, and line feeds are replaced with '
' meaning a difference of 3 characters. Instances of two digit numbers followed immediately by a double quote are replaced with the same 2 digit numbers followed by a space and then the word "inch", meaning a character difference of 4. <p>Comments can't be longer than 3000 characters total, including the normalisation differences above. To determine count use formula: $Length = Input.Length - (A \times 2) + (B \times 3) + (C \times 4)$ where A = # of CR, B = # of LF, C = # of matches for regex "[0-9]{2}(\")"</p> <p>Can't have words longer than 20 characters.</p> <p>Please see the Terms and Conditions Checks section below.</p>
Condition	Primitive	The condition description of the vehicle.	This is a free text description for outlier cases that don't adhere to the conditions that can be provided in the

			ConditionId field.
ConditionId	Primitive	The id of the condition for the vehicle.	For Franchise Dealers this property is sensitive for Car categories.. Please use the Branch Service Products property to govern your decisions. All conditions discoverable using 'metadata' resource. Defaults to 0 (Used). For non-car categories only 'Used' and 'Brand New' can be used.
DeliveryDetails	Object	See table below	Please use the Branch Service Products property to govern your decisions. (ShippingOptions). This may only be set if the product is present.
Doors	Primitive	How many doors does the vehicle have?	Min 2, Max 5. Not supported for 'Boat' or 'Bike' categories.
Features	Array	Vehicle features	All features discoverable using 'metadata' resource - features are category specific. Note: These features are not the same as the features curated on the feature-collection endpoints.
FuelId	Primitive	Fuel Type	All fuel types discoverable using the 'metadata' resource - fuel types are category specific.
FourWheelDrive	Primitive	Is the vehicle a Four Wheel Drive?	Not supported for the 'Boat' category.
Links	Object	See table below	See table below
MakeDetails	Object	See table below	See table below
ModelDetails	Primitive	Extra model details	
PriceDetails	Object	See table below	See table below
Seats	Primitive	Number of	Can be 0 if unsure.

		seats	
ShowOnTradeMe	Primitive	If false, the listing will only be held in Dealerbase, and not 'synced' to the Trade Me site.	This mechanism allows you to use Dealerbase as a stock management system.
StockNumber	Primitive	UID for vehicle listing.	Must be unique across the branch. Can't contain %2F.
Subtitle	Primitive	Extra subtitle on the listing	Please use the Branch Service VehicleSubtitleEnabled property to govern your decisions. If your branch has a default set then setting this to null will instruct the API to use your branch default. Please see the Terms and Conditions Checks section below.
UseStandardVehicleText	Primitive		Text prepended to the comments that has been designated by the branch.
VtnzAppraised	Primitive		Not supported by 'Boat' category.
Year	Primitive	The model year	Must be greater than 1908, and less than next year's numerical value.

Terms and Conditions Checks

The *Subtitle* and *Comments* fields both are scanned for words or numbers that may violate our terms and conditions checks. Please do not include any contact information in these fields. False positives may be raised if you are adding multiple digits in succession.

Examples of contact details are:

- Phone numbers - e.g 3124567, False positive may be: 1000011kg
- Email addresses - e.g bob@snailmail.com, False positive may be: 2@90HP
- Website addresses - e.g www.mydealership.com, <http://google.com>.

Auxiliary Details

The Auxiliary Details property/structure is used to set/unset certain supplementary details on the listing. It is also used to activate or deactivate the listing. **Please note that during PUT requests to the Vehicle resource this field must be set to how you want the state of the listing. If you wish to have the PUT method ignore these details then you must set the query string *ignoreAuxiliaryDetails=true* on the URL.**

Name	Type	Purpose	Extra Info / Rules
FeatureProducts	Map	To set a listing as a top pick.	The numerical value for the mentioned products should be obtained from the metadata service. Top Picks assignment - A branch can have no more than 10 Top Picks at any given time. The total amount of Top Picks currently assigned can be retrieved from the Branch endpoint.
FinanceOfferId	Primitive	To assign a finance offer to a listing.	Finance offers for your branch are obtainable from the Finance service. Please use the Branch Service Products property to govern your decisions. (Finance). This may only be set if the product is present.
SpecialOfferId	Primitive	To assign a special offer to a listing.	A full list of Special Offers for your branch can be retrieved from the Listing service 'special-offers' resource. Please use the Branch Service Products property to govern your decisions. (SpecialOffers). This may only be set if the product is present.
Status	Primitive	To set a listing to be activated or deactivated.	This property is used to activate or deactivate your listing. Once deactivated, a listing can only be activated. Deactivating a listing will remove it from the Trade Me site. Deactivated listings remain for 120 days and then are removed from the system.

BranchCode	Primitive	To transfer a vehicle to another branch within the same company.	See Transfer section. Ignored when adding a new vehicle.
------------	-----------	--	--

Delivery Details

The Delivery Details property/structure is used to set/unset delivery/shipping options on a listing. Your branch may have default options for shipping and setting vehicle shipping options will override those default options. **You will need the Shipping Options product enabled for your branch. Please use the Branch Service Products property to govern your decisions. (ShippingOptions). This may only be set if the product is present.** Furthermore, if you opted to not display Shipping Options on listings via Dealerbase then these delivery details will be ignored on payloads and will not be returned on responses. Please talk to your account manager if you run into any issues setting this Delivery Details property.

Name	Type	Purpose	Extra Info / Rules
PickupType	Primitive	To configure pickup options for a listing	
IsShippingToBeArranged	Primitive	To set if shipping will be arranged with the seller.	
ShippingOptions	Array	To set shipping options for the listing.	<p>Maximum of 5 per listing. This property must be an empty array if IsShippingToBeArranged = true or PickupType = Demand - otherwise it can be empty if there are default options set for the branch.</p> <p>If you have set up branch level defaults then this array can be empty. Please use the Branch Service HasDefaultShippingOptions property to govern your decisions. This may only be set if the product is present.</p>

Links

The Links property/structure is used to set/unset important links that appear on the listing.

Name	Type	Purpose	Extra Info / Rules
YoutubeUrl	Primitive	To set the video for the listing.	Must be in the Youtube url format and actually exist.
FinanceUrl	Primitive	A separate finance url.	
SiteLink	Primitive	A link back to the dealers site.	Must have this feature enabled for your branch. Please use the Branch Service ShowUrlOnTradeMe property to govern your decisions.

Vehicle Make

The VehicleMake property/structure is used to set/unset important details about the make of the vehicle.

Name	Type	Purpose	Extra Info / Rules
ManufacturerId	Primitive	The ID of the manufacturer.	All manufacturers are discoverable using the 'metadata' resource - manufacturers are category specific. If you can't find your manufacturer by the Name property, use the ExternalNames property.
ModelId	Primitive	The ID of the model.	The model must be supported by the manufacturer and so querying the 'metadata' resource 'manufacturer' endpoint will allow you to discover what models are available for your manufacturer. Use the same name resolution strategy as you do with manufacturers.
TypeId	Primitive	The ID of the vehicle type.	The vehicle type must be supported by the model and so your resolved model

			from the 'metadata' resource 'manufacturer' endpoint will allow you to discover what vehicle types are available for your model. Use the same name resolution strategy as you do with manufacturers.
--	--	--	--

Note: The 'make' data is cascading, so a vehicle type must be related to a model, and a model related to a manufacturer.

There is a process to request new models to be added to the Dealerbase system, which is documented in the [ModelQueue resource](#).

For especially niche models, or urgently uploading a vehicle while awaiting the addition of a new model, the Make 'Other' and Model 'Other' can be used. When using the 'Other' make and model, the desired name of the make and model should be supplied in the ModelDetails field. When a vehicle with the 'Other' make and model is synced to Trade Me the title will contain the ModelDetails field and omit the 'Other' make and model name as long as the ModelDetails field is populated.

Price Details

The PriceDetails property/structure is used to set/unset important details about price information on the listing..

Name	Type	Purpose	Extra Info / Rules
ExcludesGst	Primitive	Show information relating to whether or not the price includes or excludes GST.	This is restricted to certain categories. Allowed categories are discoverable via the metadata resource.
PriceType	Primitive	The pricing type of the listing.	If a vehicle is not new, then the price type must be 'AskingPrice'. Allowed categories for each price type are discoverable via the metadata resource.
RetailPrice	Primitive	The price the	

		vehicle usually retails at.	
SalePrice	Primitive	The sale price of the vehicle.	<p>Must be less than the Retail Price.</p> <p>Please use the Branch Service Products property to govern your decisions. (DiscountedPricing). This may only be set if the product is present.</p>

Listing a Campervan

A campervan listing inherits all base properties described above, and extends with properties specific to a campervan. A campervan must have its Category value set to '7' and must be a campervan (Manufacturer, Model and Vehicle Type).

The following properties are present on a campervan:

Name	Type	Purpose	Extra Info / Rules
AxleConfig	Primitive	The axle configuration of the vehicle.	
Berth	Primitive	The berth of the vehicle (bedding).	
CylinderCapacity	Primitive	Volume of cylinders in cubic centimetres	
Differential	Primitive	The type of differential the vehicle uses.	
EngineDetails	Primitive	Extra info about the engine.	
HorsePower	Primitive	The horsepower of the vehicle.	

IdentificationDetails	Object	See table below	See table below
InteriorDetails	Primitive	Details about the interior.	
Layout	Array	The interior layout of the campervan.	Accepts multiple values, values discoverable using the 'metadata' resource.
Odometer	Primitive	The odometer reading.	
OrcAmount	Primitive	The dollar figure amount for On Road Costs	
OrcIncluded	Primitive	Are On Road Costs included in the sale price?	
SelfContained	Primitive	Is the camper self contained? (plumbing, cooking etc)	
TransmissionId	Primitive	The ID of the transmission type for the camper.	All transmission types are discoverable using the 'metadata' resource. Transmission types are category specific.
TyreSize	Primitive	The size of the tyre.	
WeightDetails	Object	See table below.	See table below.
Wheelbase	Object	The wheelbase measurement.	Accepts Meter.
VehicleMeasurements	Object	See table below.	See table below.

Listing a Motorcycle

A motorcycle listing inherits base all properties described above, and extends with properties specific to a motorcycle. A motorcycle must have its Category value set to '5' and must be a motorcycle (Manufacturer, Model and Vehicle Type).

The following properties are present on a motorcycle:

Name	Type	Purpose	Extra Info / Rules
CylinderCapacity	Primitive	Volume of cylinders in cubic centimetres	This is a required field for motorcycles.
EngineHours	Primitive	The number of hours an engine has run over its lifetime.	
IdentificationDetails	Object	See table below.	See table below.
LamsApproved		Is the motorcycle approved for LAMS?	Learner Approved Motorcycle Scheme (LAMS) is an initiative that identifies motorcycles that are considered suitable for motorcycle learners and for all riders in their first year after progressing from a motorcycle learner licence.
Odometer	Primitive	The odometer reading.	
OrcAmount	Primitive	The dollar figure amount for On Road Costs.	
OrcIncluded	Primitive	Are On Road Costs included in the sale price?	
TransmissionId	Primitive	The ID of the transmission type for the motorcycle.	All transmission types are discoverable using the 'metadata' resource. Transmission types are category specific.

Listing a Car

A car listing inherits all base properties described above, and extends with properties specific to a car. A car must have its Category value set to '2' and must be a car (Manufacturer, Model and Vehicle Type).

The following properties are present on a car:

Name	Type	Purpose	Extra Info / Rules
CacheCleanCarDetails	Primitive	Internal use only.	Internal Use Only
CertifiedPreOwnedProgramId	Primitive	The ID of the Certified Pre-Owned Program	Associate the listing with a Certified Pre Owned Program. The Program must be for the manufacturer provided in the MakeDetails. All programs can be discovered for a manufacturer on the Metadata resource Manufacturer endpoint. ConditionId must be 3 (Certified Pre Owned)
CleanCarDetails	Object	Provide overrides for clean car information. Affects listing price.	See table below. Required for listings with a status of 'Available to Order' that do not have IdentificationDetails and have a Year greater than 1999, otherwise optional.
CylinderCapacity	Primitive	Volume of cylinders in cubic centimetres	
IdentificationDetails	Object	See table below.	See table below.
InteriorDetails	Primitive	Details about the interior.	
Odometer	Primitive	The odometer reading.	After creating a car you have 7 days to change this value. You can then only modify less than 1000 once.

OrcAmount	Primitive	The dollar figure amount for On Road Costs.	
OrcIncluded	Primitive	Are On Road Costs included in the sale price?	
TransmissionId	Primitive	The ID of the transmission type for the car.	All transmission types are discoverable using the 'metadata' resource. Transmission types are category specific.

Franchise Dealer and Cars

If you have purchased the appropriate package for your branch you are afforded some visual benefits when listing [Cars](#).

When listing a Car (as a Franchise Dealer with the appropriate packages) there are a set of rules around the AvailabilityStatus, ConditionId and Odometer values that aren't enforced for branches that are not classed as Franchise Dealers or the listing is not not a car listing.

As a package is a group of products you can use the Branch Service Product property to make your decision around what AvailabilityStatus' can be set on your vehicle payload.

For 'Available To Order' check for 'FdAvailableToOrder'.

For 'In Transit' check for 'FdInTransit'.

The following are rules pertaining to Franchise Dealers listing Cars:

- You must have at least the **Lite** package to use the **'InStock'** status (this is by default) to have visual differentiation of your listing on Trade Me.
- You must have the **Franchise** package to use the **'Available to Order'** and **'InTransit'** statuses to have visual differentiation of your listing on Trade Me.
- When using the **'InStock'** status you can use **any** condition for the ConditionId field.
- When using the **'Available to Order'** status you can only use the 'Brand New' condition.
- When using the **'In Transit'** status you can only use the 'Brand New' or 'Used' status'.
- When using a Condition of **'Brand New'** your odometer value must **be less than 500**.
- When using a Condition of **'Ex Demo'** your odometer value must **be less than 12500**.
- When using a Condition of **'Certified Pre Owned', or 'Used'** your odometer value has no limit (within data type).

- When using a status of **'Available to Order'** you **do not need to provide IdentificationDetails**. This is due to the chance that the vehicle may not be in NZ yet and not known by NZTA. **If you omit identification details** then you must populate the CleanCarDetails field with clean car information.
- When using a condition of **'Certified Pre-Owned'** you **must** provide a program identifier (*CertifiedPreOwnedProgramId*). This is available on the Metadata resource Manufacturer endpoint and is specific to a Manufacturer.

Listing a Truck

A truck listing inherits all base properties described above, and extends with properties specific to a truck . A truck must have its Category value set to '6' and must be a truck (Manufacturer, Model and Vehicle Type).

The following properties are present on a truck :

Name	Type	Purpose	Extra Info / Rules
AxleConfig	Primitive	The axle configuration of the vehicle.	
CylinderCapacity	Primitive	Volume of cylinders in cubic centimetres	
Differential	Primitive	The type of differential the vehicle uses.	
EngineDetails	Primitive	Extra info about the engine.	
HorsePower	Primitive	The horsepower of the vehicle.	
IdentificationDetails	Object	See table below	See table below
InteriorDetails	Primitive	Details about the interior.	
Odometer	Primitive	The odometer reading.	

OrcAmount	Primitive	The dollar figure amount for On Road Costs	
OrcIncluded	Primitive	Are On Road Costs included in the sale price?	
TransmissionId	Primitive	The ID of the transmission type for the camper.	All transmission types are discoverable using the 'metadata' resource. Transmission types are category specific.
TyreSize	Primitive	The size of the tyre.	
WeightDetails	Object	See table below.	See table below.
Wheelbase	Object	The wheelbase measurement.	Accepts Meter.
VehicleMeasurements	Object	See table below.	See table below.

Listing a Bus

A bus listing inherits all base properties described above, and extends with properties specific to a bus. A bus must have its Category value set to '3' and must be a bus (Manufacturer, Model and Vehicle Type).

The following properties are present on a bus:

Name	Type	Purpose	Extra Info / Rules
AxleConfig	Primitive	The axle configuration of the vehicle.	
CylinderCapacity	Primitive	Volume of cylinders in cubic centimetres	

Differential	Primitive	The type of differential the vehicle uses.	
EngineDetails	Primitive	Extra info about the engine.	
HorsePower	Primitive	The horsepower of the vehicle.	
IdentificationDetails	Object	See table below	See table below
InteriorDetails	Primitive	Details about the interior.	
Odometer	Primitive	The odometer reading.	
OrcAmount	Primitive	The dollar figure amount for On Road Costs	
OrcIncluded	Primitive	Are On Road Costs included in the sale price?	
SelfContained	Primitive	Is the camper self contained? (plumbing, cooking etc)	
TransmissionId	Primitive	The ID of the transmission type for the camper.	All transmission types are discoverable using the 'metadata' resource. Transmission types are category specific.
TyreSize	Primitive	The size of the tyre.	
WeightDetails	Object	See table below.	See table below.
Wheelbase	Object	The wheelbase	Accepts Meter.

		measurement.	
VehicleMeasurements	Object	See table below.	See table below.

Listing a Boat

A boat listing inherits all base properties described above, and extends with properties specific to a boat. A boat must have its Category value set to '9' and must be a boat (Manufacturer, Model and Vehicle Type).

The following properties are present on a boat:

Name	Type	Purpose	Extra Info / Rules
Beam	Object	The beam measurement	Accepts Meter.
ConstructionTypeId	Primitive	Identifier of what the boat is made out of.	
Displacement	Object	Displacement for the boat.	Accepts Kg.
Draft	Object	Draft of the boat.	Accepts Meter.
EngineDetails	Primitive	Extra info about the engine.	
EngineHours	Primitive	The number of hours an engine has run over its lifetime.	
FuelCapacity	Object	Fuel capacity of the boat.	Accepts Liter
Horsepower	Primitive	The horsepower of the vehicle.	
Length	Object	Length of the boat.	Accepts Meter.

LocationId	Primitive	The location of the boat.	
PropulsionId	Primitive	Transmission of the boat.	
WaterCapacity	Object	Boat water capacity.	Accepts Liter.

Listing Machinery

A machinery listing inherits all base properties described above, and extends with properties specific to a piece of machinery . A machine must have its Category value set to '8' and must be a machine (Manufacturer, Model and Vehicle Type).

The following properties are present on a piece of machinery:

Name	Type	Purpose	Extra Info / Rules
AxleConfig	Primitive	The axle configuration of the vehicle.	
CylinderCapacity	Primitive	Volume of cylinders in cubic centimetres	
Differential	Primitive	The type of differential the vehicle uses.	
EngineDetails	Primitive	Extra info about the engine.	
EngineHours	Primitive	The number of hours an engine has run over its lifetime.	
HorsePower	Primitive	The horsepower of the vehicle.	
IdentificationDetails	Object	See table below	See table below

InteriorDetails	Primitive	Details about the interior.	
Odometer	Primitive	The odometer reading.	
OrcAmount	Primitive	The dollar figure amount for On Road Costs	
OrcIncluded	Primitive	Are On Road Costs included in the sale price?	
TransmissionId	Primitive	The ID of the transmission type for the machine.	All transmission types are discoverable using the 'metadata' resource. Transmission types are category specific.
TyreSize	Primitive	The size of the tyre.	
WeightDetails	Object	See table below.	See table below.
Wheelbase	Object	The wheelbase measurement.	Accepts Meter.
VehicleMeasurements	Object	See table below.	See table below.

Identification Details

The IdentificationDetails property/structure is used to set/unset important details pertaining to identification.

Name	Type	Purpose	Extra Info / Rules
Chassis	Primitive	The vehicle chassis.	If the vehicle year is greater than 1999 and no other identification details have been set then this is required. No other vehicle for your branch must have this chassis value.

Registration	Primitive	The vehicle registration.	If the vehicle year is greater than 1999 and no other identification details have been set then this is required. No other vehicle for your branch must have this registration.
Vin	Primitive	The Vehicle Identification Number	If the vehicle year is greater than 1999 and no other identification details have been set then this is required. No other vehicle for your branch must have this vin. Please see the VIN flow for instructions on how to handle this situation.

Weight Details

The WeightDetails property/structure is used to set/unset details pertaining to the weight of the vehicle.

Name	Type	Purpose	Extra Info / Rules
GVM	Object	The Gross Vehicle Mass of the vehicle.	Accepts Kg.
Payload	Object	The payload vehicle.	Accepts Kg.
Tare	Object	The tare of the vehicle.	Accepts Kg.

Vehicle Measurements

The VehicleMeasurements property/structure is used to set/unset details pertaining to the wheel measurements of the vehicle.

Name	Type	Purpose	Extra Info / Rules
Height	Object	The height of the vehicle.	Accepts Meter.
Width	Object	The width of	Accepts Meter.

		the vehicle.	
Length	Object	The length of the vehicle.	Accepts Meter.

Vehicle Response.

After a successful request to add/update/patch an existing listing the API will send back the related vehicle categorical structure as well as a few other details:

Name	Type	Purpose	Extra Info / Rules
BackgroundCheck	Primitive	Enumeration pertaining to the background check status.	This process is asynchronous and is only completed if your branch has a subscription with Motorweb (and you have provided the authentication details to us) as well as the vehicle having a valid VIN or REGISTRATION. A background check is only 'valid' for 90 days.
ExternalMedia	Object	External media.	See table below.
RecordAdded	Primitive	The date the record was added.	In NZ time.
RecordModified	Primitive	The date the record was last modified.	In NZ time.
Image	Primitive		The Url for the image with the smallest display order.
ImageCount	Primitive	The amount of images associated with the listing.	
ListingStatus	Object	See table below.	See table below.

External Media

The ExternalMedia property (ReadOnly) will be returned when provided by Instavid and associated with a vehicle listing.

Name	Type	Purpose	Extra Info / Rules
Instavid360SpinnerId	Primitive	Spinner 360 ID.	
Instavid360SpinnerThumbnailUrl	Primitive	Spinner 360 thumbnail url.	
InstavidVideoThumbnailUrl	Primitive	Instavid video thumbnail url.	
InstavidVideoUrl	Primitive	Instavid video url	

Listing Status

The ListingStatus property is an informational structure relating to the overall status of the listing within the system.

Name	Type	Purpose	Extra Info / Rules
Status	Primitive	Enumeration pertaining to the state of the listing.	The listing may go through many states. This enumeration describes the current state.
ListingId	Primitive	The ID of the listing.	This will be populated when the listing has been synced to the main Trade Me site. This can be used in an URL format to locate the listing within a browser.
Message	Primitive	Error message	If there are any issues with the listing the error will be present in this

			property.
--	--	--	-----------

Vehicle Make

The MakeDetails property will be returned with the provided ids as well as the names of the associated entities.

Name	Type	Purpose	Extra Info / Rules
ManufacturerName	Primitive		The name of the manufacturer.
ModelName	Primitive		The name of the model.
TypeName	Primitive		The name of the vehicle type.

Getting a listing by stock number.

Issuing a HTTP GET request to the 'vehicles/{stockNumber}' endpoint will retrieve for you a vehicle if it exists for the given stock number.

Getting vehicle listings for your branch.

Issuing a HTTP GET request to the 'vehicles' endpoint will retrieve for you a page of vehicle listings for your branch. By default you will receive a page of 10 listings. The following values in query string format can be used to modify the request:

Name	Type	Purpose	Extra Info / Rules
Compact	Primitive	Return the core properties for a listing	Will return a subset of properties for a listing, allowing for a larger page size (500) to be requested.
PageSize	Primitive	Request a specific number of listings per page.	Max is 50 when Compact=false/null, otherwise 500. Default 10.
Page	Primitive	The page	This will be populated when the

			listing has been synced to the main Trade Me site. This can be used in an URL format to locate the listing within a browser.
--	--	--	--

Response.

Name	Type	Purpose	Extra Info / Rules
Items	Array	The page of your items	
CurrentPage	Primitive	The page number you request.	
PageSize	Primitive	The page size you requested.	
TotalPages	Primitive	The total number of pages.	Based on your page size.
TotalRecords	Primitive	The total count of records.	

Updating a listing.

Issuing a HTTP PUT request to the 'vehicles/{stockNumber}' endpoint using any related vehicle categorical structure (as described above) will update the vehicle for the provided stock number. (Note that this includes updating the stock number for a vehicle). As this is a state replacement - the provided vehicle data, if valid, will be the new vehicle data (as opposed to a HTTP PATCH).

Patching a listing.

Currently, only the Auxiliary Details structure is supported when PATCHING.

Issuing a HTTP PATCH request to the 'vehicles/{stockNumber}' endpoint will replace the properties you set (and not modify any other properties on the supported request structure).

Activating/Deactivating a listing.

Activating or Deactivating a listing is done by issuing an HTTP PUT or PATCH modifying the Auxiliary Details property / root structure to have a 'Status' of Activate / Deactivate.

Activating a deactivated listing will asynchronously 'push' the listing to the main Trade Me site (given that it still meets all business requirements).

Deactivating a listing will remove it from your promoted list of products, and tell the main Trade Me site to remove the listing. Note that when deactivating a listing via the HTTP PUT path no other data will be updated (the vehicle will just be deactivated).

To determine if a vehicle is activated or deactivated - check the 'ListingStatus.status' property on the Vehicle Response - if it is Ended then it has been deactivated.

Deactivated listings remain in the system (just not on Trade Me site) for 120 days, then they are deleted.

Important: When deactivating a listing the Trade Me members that have Watchlisted the listing are removed from the listing. You must be certain a deactivation is the action you want to take before sending it. Furthermore, deactivations take at least 10 minutes to process.

Transferring a vehicle to another branch.

A vehicle may be transferred to another branch however there are rules and assumptions that are in place when requesting this operation:

- The transfer operation is a **singular operation** and **nothing else will occur in the transaction** - this means that other data within the request will be ignored. ie Other PUT details will be ignored.
- The operation request will be ignored when *adding* a new vehicle.
- A vehicle may only be transferred to another branch within the same company.
- A vehicle may only be transferred to the target branch if the stock number is available.
- When transferring a car - if the car has an AvailabilityStatus or VehicleCondition that requires a Franchise Dealer product then the target branch needs to be allocated that product/be on the correct package.
- If the vehicle is ~~SuperFeatured~~ or a Top Pick in the source branch - after transfer the vehicle **will not be** ~~SuperFeatured~~ **or be** a Top Pick - this is to reduce complexity around slot allocations. Please make a subsequent update request to assign this.
- If a vehicle is assigned to a Special Offer or a Finance option then after transfer they **will not be assigned anymore.**
- If the vehicle has a subtitle set and the target branch does not have this setting then the subtitle will be removed after transfer.

- If the vehicle has a site link set and the target branch does not have this setting then the site link will be removed after transfer.
- If the vehicle has comments that would be in violation after transfer then the transfer is rejected.

The best way to transfer a vehicle is to send a PATCH to the `listing/v1/{stockNumber}` endpoint with the `branchCode` set.

Adding an image to a listing.

After a listing has been created you can add an image to a listing. A listing is not 'valid' until it has at least one image. Only then will it be asynchronously pushed to the main Trade Me site.

Issuing a HTTP POST to the 'vehicles/{stockNumber}/images' endpoint with the below structure will add an image to a listing.

As the content is a mix of binary and text, the content type for the endpoint must be `multipart/form-data`.

Only 20 images are supported for a listing.

Payload:

Name	Type	Purpose	Extra Info / Rules
DisplayOrder	Primitive	The order property for displaying this image on the main Trade Me site.	Must be unique for this listing's images and greater than 0 and less than or equal to 20.
Image	Binary Stream	The image / stream data.	Must be gif, jpeg (exactly this spelling) and png. Must be no larger than 10MB.

Response

The response from adding a new image or retrieving an existing image is not the associated binary data. It is the image's metadata associated with two URI's.

Name	Type	Purpose	Extra Info / Rules
------	------	---------	--------------------

Checksum	Primitive	For file comparison.	You can utilise this property to determine if the image you're updating is the same as the one you're updating it with. https://stackoverflow.com/questions/10520048/calculate-md5-checksum-for-a-file
DisplayOrder	Primitive	The order property for displaying this image on the main Trade Me site.	
ImageId	Primitive	The unique ID of the image.	Must be used in PUT requests.
PhotoUrls	Object	URI's denoting the image locations.	

Getting an image.

After you have added an image to an existing listing you may retrieve the image metadata information by issuing a HTTP GET to the 'vehicles/{stockNumber}/images/{imageId}' endpoint, where *imageId* is the *ImageId* property on the return structure that is received from the API after an image has been added.

The response is the same as the response received when adding an image.

Updating an image's data.

After you have added an image to an existing listing you may update the image binary data by issuing a HTTP PUT to the 'vehicles/{stockNumber}/images/{imageId}' endpoint, where *imageId* is the *ImageId* property on the return structure that is received from the API after an image has been added. **Note: Display order can't be updated with this endpoint.**

Getting all images.

Issuing an HTTP GET to the 'vehicles/{stockNumber}/images' endpoint will return all images for a listing.

Deleting an image.

After you have added an image to an existing listing you may delete the image by issuing a HTTP DELETE to the 'vehicles/{stockNumber}/images/{imageId}' endpoint, where *imageId* is the *ImageId* property on the return structure that is received from the API after an image has been added.

Deleting all images for a listing.

After you have added images to an existing listing you may delete all but one of the images by issuing a HTTP DELETE to the 'vehicles/{stockNumber}/images' endpoint. The endpoint **also takes a query string parameter of 'leave'** - allowing you to specify how many images you want to remain (and thus the difference will be deleted). This query string is useful if your listing is 'SuperFeatured' and requires at least 3 images at all times.

Deleting external media.

After seeing external media associated to an existing listing you may delete the external media fields by issuing a HTTP DELETE to the 'vehicles/{stockNumber}/external-media/{type}' endpoint, where *type* is the external media fields enumeration (Instavid360spinner or InstavidVideo).

Updating image orders.

As you have control of the image ordering on a listing you may want to update this. Issuing a HTTP PUT request to the 'vehicles/{stockNumber}/image-orders' endpoint with the correct payload will update image ordering.

The endpoint receives a list of ImageOrder's. The list must be the entirety of all images for the listing (the absolute state). Each image order is composed of:

Name	Type	Purpose	Extra Info / Rules
DisplayOrder	Primitive	The order property for displaying this image on the main Trade Me site.	Must be unique across the entire set of images for the listing.
ImageId	Primitive	The unique ID	

		of the image.	
--	--	---------------	--

New Vehicle Features.

A new collection of endpoints are available for consumers listing vehicles in the Car category. These endpoints allow the consumer to curate and manage custom feature collections for a specific listing which will more accurately represent the unique aspects of the vehicle being sold.

A feature collection can contain different types of features, these feature types are:

- Standard Features: These are features that are 'standard' to most cars.
- Aftermarket Features: These are features that have been installed on a vehicle after its initial sale by the original manufacturer (OEM).
- Bespoke features: These are user defined features, and represent a customization or enhancement that is uniquely tailored to an individual customer's specifications for the vehicle.

Standard and Aftermarket feature metadata is available on the Metadata resource to facilitate feature collection curation. As Bespoke features are user defined, the API interface accepts string input for these features.

Furthermore, the consumer can optionally dictate what features in the collection are considered to be the 'Top Features'. A total of 9 Top Features can be selected. If no Top Features are selected then we will determine the top features, favouring Aftermarket and Bespoke features. Top features will be initially displayed to the user on the Listing page, with the full set viewable with an extra user click. If your vehicle has features that sets it apart from other car listings for the same manufacturer/model then it will be beneficial to select these Top Features.

Important: When creating or updating a feature collection using the new endpoints it is important that you tell us you aren't using the 'old' features anymore. You can do this by ensuring your core vehicle data contains an empty *Features* property - please ensure you clear this collection before updating features. For the old AA Appraised, 4WD, Doors, VTNZ Appraised features you can now use the properties available on the request model.

Important: Standard features do not require the target vehicle to have a VIN. However, when curating aftermarket or bespoke features a valid VIN known by the NZTA is required, the VIN must also not be in use by another active vehicle in the system.

The core endpoints are:

HTTP Verb	URI	Effect
GET	vehicles/{stockNumber}/features-collection	Get an existing feature collection for a vehicle.
POST	vehicles/{stockNumber}/features-collection	Create a new features collection..
PUT	vehicles/{stockNumber}/features-collection	Update an existing features collection. Note: This is state replacement and not a PATCH.
DELETE	vehicles/{stockNumber}/features-collection	Delete an existing feature collection.

Creating a new feature collection.

Vehicle Feature Collection Payload

Name	Type	Purpose	Extra Info / Rules
AftermarketFeatures	Collection	The collection of Aftermarket Features	Must have at least one item if <i>Features</i> and <i>BespokeFeatures</i> is empty. Must not contain duplicates.
BespokeFeatures	Collection	The collection of Bespoke Features	Must have at least one item if <i>AftermarketFeatures</i> and <i>Features</i> is empty. Must not contain duplicates. A maximum of 20 bespoke features can be presented.
Features	Collection	The collection of Standard Features	Must have at least one item if <i>AftermarketFeatures</i> and <i>BespokeFeatures</i> is empty. Must not contain duplicates.

AftermarketFeature Payload:

Name	Type	Purpose	Extra Info / Rules
Id	Primitive	The identifier for the Aftermarket feature.	Must be a valid identifier that relates to an Aftermarket feature that is discoverable on the Metadata resource
TopFeatureOrder	Primitive	The order this feature should appear in a Top Feature set.	As this property is nullable, setting this will tell us you want this feature to be a 'Top Feature' and this is the order it should appear.

BespokeFeature Payload:

Name	Type	Purpose	Extra Info / Rules
Name	Primitive	The identifier for the Bespoke feature and the primary identifier.	Must be no longer than 255 characters, and must be the same as an existing Standard or Aftermarket feature name.
TopFeatureOrder	Primitive	The order this feature should appear in a Top Feature set.	As this property is nullable, setting this will tell us you want this feature to be a 'Top Feature' and this is the order it should appear.

StandardFeature Payload:

Name	Type	Purpose	Extra Info / Rules
Id	Primitive	The identifier for the Bespoke feature and the primary identifier.	
TopFeatureOrder	Primitive	The order this feature should appear in a Top Feature set.	As this property is nullable, setting this will tell us you want this feature to be a 'Top Feature' and this is the order it should appear.

Value	Primitive	The value associated with the feature	The input value for the feature. Note: Only 'Number of speakers' feature is supported for input currently.
-------	-----------	---------------------------------------	---

When creating or updating a feature collection there are a few general rules:

- This is only available for vehicles in the **Car** category.
- The vehicle must have a valid VIN and the VIN must not already be in use.
- At least one feature (of any type) must be presented in the payload.
- A maximum of 9 Top Features can be selected.
- If no Top Features are selected, then we will select them for the vehicle.
- The same feature can't be presented twice in the payload.
- A maximum of 20 Bespoke Features can be presented in the payload.

Updating an existing feature collection.

In order to determine if a vehicle has an existing feature collection you can call the endpoint to retrieve vehicle details for a specific vehicle (GET vehicles/{stockNumber}). On the response for the vehicle there will be a **"HasFeatureCuration"** property that tells the consumer if a vehicle has a feature collection curation currently available. If this value is **True** then you can issue an update to the **PUT** endpoint. If this is **FALSE** then you must create a new feature collection by issuing a request to the **POST** endpoint.

NOTE: The PUT endpoint is full state replacement. What the consumer provides us will be the new state of the feature collection.

Deleting an existing feature collection.

If you would like to delete an existing feature collection for a vehicle then you can issue a request to the DELETE endpoint in the table above.

Retrieving an existing feature collection.

If a given listing has an associated feature collection then it can be retrieved at the GET endpoint denoted in the table above. The response structure will be almost identical to the POST/PUT structures, but more information to aid in display purposes will be provided:

- Standard Features will be returned with name and category information.

- Aftermarket Features will be returned with name information.

More comprehensive information about the properties returned can be found in the OpenAPI documentation for these endpoints.

Metadata Resource

Description

The metadata resource provides reference data for populating vehicle listings. It can be thought of as ‘the resource you call to populate your dropdown boxes on a UI’.

The following endpoints are available for the Metadata resource.

All endpoints within the metadata resource can be accessed with a token generated with the `client_credentials` grant OR the `password` grant.

Endpoints

HTTP Verb	URI	Effect
GET	metadata	Retrieves vehicle metadata..
GET*	metadata/manufacturers	Retrieve a paged list of manufacturers.
GET	metadata/manufacturers/{manufacturerId}	Retrieves detailed manufacturer data.
GET*	metadata/manufacturers/{manufacturerId}/models	Retrieve a paged list of models for a given manufacturer.
GET	metadata/models/{id}	Retrieve detailed information about a model.
GET*	metadata/models/{id}/vehicle-types	Retrieve a paged list of vehicle types for a given model.
GET*	metadata/vehicle-types	Retrieve a paged list of vehicle types.
GET	metadata/vehicle-types/{id}	Retrieve detailed information about a vehicle type.
GET	metadata/vehicles/features	Retrieve feature curation metadata for use in the new feature-collection endpoints .

GET	metadata/vehicles/aftermarket-features	Retrieve aftermarket feature curation metadata for use in the new feature-collection endpoints .
-----	--	---

Retrieving metadata.

All reference data required for a listing can be discovered by issuing an HTTP GET on the metadata resource. The following is details about the response.

As this data isn't transactional and is very rarely changed it is advised that you cache it on your side. This will require you to implement a cache renewal strategy.

Note: The /metadata endpoint can take a 'categoryId' query string allowing you to scope all metadata to a particular category (where applicable). It can also take a 'returnManufacturers' boolean (default true) that allows you to omit the manufacturer list from the payload.

Name	Type	Purpose	Extra Info / Rules
Branch	Object	Branch specific properties.	Provides branch defaults.
Categories	Array	Master list of vehicle categories.	
Conditions	Array	Master list of vehicle conditions.	
ConstructionTypes	Array	Master list of boat construction types.	Only for the boat category.
ExcludesGstSupportedCategories	array	List of categories which support the ExcludesGst flag.	
Features	Array	Master list of vehicle features.	Each feature has a list of vehicle categories it is allowed to be associated with. NOTE: These

			features are not the same features that are used in the new <i>feature-collection</i> endpoints.
FuelTypes	Array	Master list of vehicle fuel types.	Each fuel type has a list of vehicle categories it is allowed to be associated with.
Layouts	Array	Master list of vehicle layout options.	Each layout has a list of vehicle categories it is allowed to be associated with.
Locations	Array	Master list of boat location options.	Only for the boat category.
Manufacturers	Array	Master list of manufacturers.	Each manufacturer has a list of vehicle categories it is allowed to be associated with. No model information is provided here - please use the 'metadata/manufacturers/{manufacture rld}' endpoint for this information.
PricingTypes	Array	List of price types.	Each price type has a list of vehicle categories it is allowed to be associated with.
Products	Array	Master list of API supported listing products.	Provides ID's to be used for associated products that are used on the AuxiliaryDetails structure.
Transmissions	Array	Master list of vehicle transmissions.	Each transmission type has a list of vehicle categories it is allowed to be associated with.

Retrieving a paged list of manufacturers, models or vehicle types.

When requesting items from the endpoints that return a list of metadata items you can use query strings to scope your result set. This applies to the endpoints denoted in the endpoint table for this resource that have an asterisk beside them. The endpoints return the same structure representing your query.

Name (Query)	Type	Purpose	Extra Info / Rules
--------------	------	---------	--------------------

Page	Primitive	The page number to request.	Defaults to 1.
PageSize	Primitive	The amount of items to retrieve.	Defaults to 100.
CategoryId	Primitive	Only return items that support a given category	By default items in all categories are returned - must be a category within the supported categories provided in the /metadata endpoint
SearchText	Primitive	Only return items that match the given string based on the Comparison Type	Based on ComparisonType your search will either return a single exact match for the item, or items that contain the provided search.
ComparisonType	Enum	Controls how items are matched.	If "Exact" then a single item will be returned, otherwise defaults to "Like" and items will be matched based upon if the search text appears in the item's name.
SortDescending	Primitive	Sort by name descending or not.	By default, items are returned in ascending order.

Response.

Name	Type	Purpose	Extra Info / Rules
Items	Array	The page of your items	
CurrentPage	Primitive	The page number you request.	
PageSize	Primitive	The page size you requested.	
TotalPages	Primitive	The total	Based on your page size.

		number of pages.	
TotalRecords	Primitive	The total count of records.	

Items will be of the following structure:

Name	Type	Purpose	Extra Info / Rules
Id	Primitive	The ID of your item.	
Name	Primitive	The name of your item.	
CategoriesSupported	Array		Important: Not populated.
ExternalNames	Array		Important: Not populated on list resources, populated on ID resources.

Notes:

- '/metadata/manufacturers': Only manufacturers that are connected to at least one model that is connected to at least 1 vehicle type will be returned.
- '/metadata/manufacturers/{id}/models': Only models that are connected to at least 1 vehicle type will be returned.
- '/metadata/models/{id}/vehicle-types': Only connected vehicle types are returned.

Retrieving detailed metadata information.

Each metadata resource type has an associated endpoint that allows you to receive more detailed information about the item, including categories it supports, external names (allowing you to match your internal string identifier to our record).

- For the '/metadata/manufacturers/{id}' endpoint you will receive all connected models and their associated vehicle types, with each level having external names and supported categories populated.
- For the '/metadata/models/{id}' endpoint you will receive all connected vehicle types, with each level having external names and supported categories populated.

- For the '/metadata/vehicle-types/{id}' endpoint you will receive the associated category with the type.

Proposed Usage:

- If you are doing batch processing you should call the '/metadata' endpoint to retrieve all base level data. Subsequent calls can then be made per manufacturer to '/metadata/manufacturers/{id}' to retrieve the full metadata tree.
- If you are building a UI you can call the '/metadata' endpoint scoped to the category of vehicle you are creating by passing ?categoryId= to the endpoint, you can also add returnManufacturers=false to omit manufacturer data if you plan to use the paged endpoints. You can then use the paged endpoints to populate your tables/dropdowns scoped to the category of the vehicle you are creating. Further detail about each metadata item can be found by calling the individual detail endpoints.

Feature Collection Metadata.

When adding or updating a feature collection against the new /vehicle/{stockNumber}/feature-collection endpoints the *Feature* and *AftermarketFeature* collections require unique identifiers. The response from the /metadata/vehicles/(features | aftermarket-features) endpoints will provide these identifiers. The response payloads will return collections of this associated metadata and each item response is as follows.

Feature:

Name	Type	Purpose	Extra Info / Rules
Id	Primitive	The primary identifier for this feature.	
CategoryId	Primitive	The primary identifier for the associated category for the feature	
CategoryOrder	Primitive	The order this category should appear visually in.	
MetalInfo	Object	Provide	This object will be populated when

		validation / control information for rendering.	<i>RequiresUserInput is True.</i> Use this object to determine the type of control and validation to implement. Currently only 'Number of speakers' feature requires input. See info below.
Name	Primitive	The name of the feature.	
Order	Primitive	The order this feature should appear visually in.	
RequiresUserInput	Primitive	Determines if this feature requires an associated <i>Value</i> .	This is used to tell the consumer if the feature does require an input value.
Unit	Primitive	The unit associated with this feature, if any.	Note: Not supported yet.

MetalInfo

Name	Type	Purpose	Extra Info / Rules
Max	Primitive	The maximum numerical value for the feature.	
Min	Primitive	The minimum numerical value for the feature.	
Placeholder	Primitive	The control placeholder if applicable.	
Type	Primitive	The control type discriminator.	Currently only 'Number of speakers' is supported - the only Type available is 'Integer'.

AftermarketFeature:

Name	Type	Purpose	Extra Info / Rules
Id	Primitive	The primary identifier for this aftermarket feature.	
Name	Primitive	The name of this aftermarket feature.	

ModelQueue Resource

Description

The model-queue resource provides the ability for a consumer to request a new model (for an existing manufacturer) be added to the system and associated with existing vehicle types OR an existing model be updated to be associated with existing vehicle types.

Endpoints

HTTP Verb	URI	Effect
POST	model-queue	Request to add a model to the system.

Requesting to add a model to the system.

Invoking the POST endpoint on the 'model-queue' resource with the correct payload will put through a request to our Customer Support team to add a new model to the system. In your processing if you find that we do not provide a certain Manufacturer/Model/BodyType combination that you need and so this is the endpoint to request. **Note: Requests enter a queue to be processed manually and are not instant, please allow up to 48 hours for processing and you will receive an email when the request has been processed.**

Your request must make sense and be spelled correctly to be approved. I.e. Suzuki Swft will be rejected, Holden Commodore Catamaran will be rejected.

Payload.

Name	Type	Purpose	Extra Info / Rules
Callback	Primitive	The email to notify you when processing is complete.	Note that after receiving the email you will need to wait an hour before the model is available.
CallbackType	Primitive	The type of callback.	Currently only "Email" is available.
ManufacturerName	Primitive	The name of the manufacturer.	The manufacturer must already exist in the system. If you require a new manufacturer then please contact Customer Support.
ModelName	Primitive	The name of the new/existing model.	Must be less than 50 characters long. Any diacritics will be removed.
VehicleTypes	Array	The ids of the Vehicle Types to associate with this new/existing model.	All types must be within the same category. You can use the Metadata resource metadata/vehicle-types?categoryId= resource to discover types to add. A maximum of 20 types can be associated at one time.

Rules:

- **This endpoint is not available to all consumers - please send a request to Customer Support to be added to the allow list.**
- For some consumers - only a maximum of 20 requests can be submitted per branch per 24 hours.
- Currently adding models for the Car category is not supported. Any requests containing VehicleTypes that associate with Category 2 will be rejected from the API.
- Duplicate requests will be rejected.

- The identifier returned from a successful request is not the identifier of the new model.

Branch Service

Description

The branch service provides consumers non-listing related resources.

Branch Resource

Description

The endpoint for the Branch resource is accessed requesting the **root** url for the Branch service.

Endpoints

HTTP Verb	URI	Effect
GET	/	Retrieves information about your branch.

Retrieving information about your branch.

A consumer can request information about their branch. The following information is returned:

Name	Type	Purpose	Extra Info / Rules
ActivePartnerVehicles	Primitive	Vehicles meeting Partner Program criteria.	Not loaded by default. Please provide a query string of <i>activePartnerVehicles=true</i> to load this info.
HasShippingOptions	Primitive	Allows consumers to know if they have <i>default</i> shipping options.	

Number	Primitive	The UID for the branch.	
OrclIncludedDefault	Primitive	Internal use.	
PartnerProgramLevel	Primitive	The Partner Program Level for the branch.	Not many branches are a partner.
Products	String List	Branch products.	<p>A dealer package is composed of multiple products. This property represents products relevant to this system (not the entire list).</p> <ul style="list-style-type: none"> • SpecialOffers, • ShippingOptions, • Finance, • FranchiseDealer, • FdAvailableToOrder, • FdInTransit, • DiscountedPricing, • InstavidCore, • InstavidVideoEnhance
Settings	Dictionary	Branch specific settings.	Not loaded by default. Please provide a query string of <i>settings=true</i> to load this info.
ShowroomCount	Primitive	The total amount of vehicles set as 'Top Picks'	Not loaded by default. Please provide a query string of <i>showRoomInfo=true</i> to load this info.
ShowUrlOnTradeMe			Determines if the vehicle payload Links.SiteLink can be displayed.
StaffId	Primitive	Internal use.	
SubtitleDefault	Primitive	The default subtitle for the branch.	If a specific subtitle is not provided for a vehicle, this value will be used for that vehicle's subtitle.
SuperFeatureTotal	Primitive	The total amount of listings that can	Not loaded by default. Please provide a query string of <i>superFeatureInfo=true</i> to load this info.

		be SuperFeatured.	
SuperFeatured	Primitive	The amount of vehicles currently set as SuperFeatured	Not loaded by default. Please provide a query string of <i>superFeatureInfo=true</i> to load this info.
TradingName	Primitive	Branch trading name.	
VehicleLimit	Primitive	Vehicle allowance before overage.	
VehicleSubtitleEnabled	Primitive		Controls if the branch can have the listing subtitle set or not.
VehiclesOnTradeMeTotal	Primitive	The total vehicles live on Trade Me.	Not loaded by default. Please provide a query string of <i>vehiclesOnTradeMe=true</i> to load this info.

Branch Properties that dictate Vehicle Payload Composition.

- Products property
 - SpecialOffers - Determines if VehicleRequest.AuxiliaryDetails.SpecialOfferId can be set.
 - ShippingOptions - Determines if VehicleRequest.DeliveryDetails can be set.
 - Finance - Determines if VehicleRequest.AuxiliaryDetails.FinanceOfferId can be set.
 - FranchiseDealer - Information - Determines if the Branch is a Franchise Dealer
 - FdAvailableToOrder - Determines if the VehicleRequest.AvailabilityStatus can be set to 'AvailableToOrder' for Franchise Dealers on a Car listing
 - FdInTransit - Determines if the VehicleRequest.AvailabilityStatus can be set to 'In Transit' for Franchise Dealers on a Car listing
 - DiscountedPricing - Determines if the VehicleRequest.PriceDetails.SalePrice can be set to a value greater than 0.
 - InstavidCore - Determines if the Listing Service, Vehicle Resource DELETE external-media endpoint can be called.
- ShowUrlOnTradeMe property - Determines whether the VehicleRequest.Links.SiteLink property can be set.
- VehicleSubtitleEnabled property - Determines whether the VehicleRequest.Subtitle can be set.

Vehicle Allowance.

When a branch/dealer signs up to trade me they purchase a package (composition of many products) and they purchase a vehicle allowance separately. The vehicle allowance represents the total number of vehicles the dealer can have live on Trade Me at any given time before they are charged an overage fee.

This information can be determined by the *VehiclesOnTradeMeTotal* and *VehicleLimit* properties returned from the Branch resource.

Trade Me also runs a 'Partner Program' for certain dealerships. The details of what the program is are not relevant however it's important to note that the certain level of our Partner Program that the dealer is on affects the amount of vehicles they are allowed to have live on Trade Me.

The *PartnerProgramLevel* property on the Branch resource can be used to determine the program level.

There are 3 levels but 4 values:

- 0 - The branch is **not** on the Partner Program **and is not afforded certain vehicle limit discounts.**
- 1 - The branch is on level 1 of our program **and is afforded certain vehicle limit discounts.**
- 2 - The branch is on level 2 of our program **and is afforded certain vehicle limit discounts.**
- 3 - The branch is on level 3 of our program **and is not afforded certain vehicle limit discounts.**

To determine how many vehicles are discounted for a branch the *ActivePartnerVehicles* property can be leveraged on the Branch resource.

This property will be 0 if the branch **is not afforded any discounts OR has no vehicles that are classified as valid for discount.**

Therefore - to calculate whether a branch has breached their vehicle allowance you can apply the following calculation - $(VehiclesOnTradeMeTotal - ActivePartnerVehicles) > VehicleLimit$. (Please note that the properties of the calculation require you pass certain query strings on the resource in order to load that data - this is described in the Branch resource table).

Watchlist Offer Resource

Description

The Watchlist Offer resource is a CRUD resource for managing your branch's Watchlist Offers. Watchlist Offers are offers that get emailed to Trade Me members that have watchlisted your listing. An example of using this feature is to promote moving aged stock. The offers are valid for up to 7 days (to midnight NZT) from the provided start date and you are legally obligated to honor them. Only one offer per vehicle can be active or scheduled at any given time. An offer must last for at least a period of a day otherwise no email will be sent to watchlisters. The offer is emailed on a schedule and is not instant. **If a mistake has been made in setting the details then the listing must be withdrawn by deactivating it via the listing endpoint and reactivating it an hour later. The listing on Trade Me will be effectively a new listing at that point (a new Trade Me Listing ID is generated) and thus the old listing becomes void along with its offer. As described previously in the document, deactivating and reactivating a listing is not a 'reset' mechanism and has destructive effects on watchlisters.**

If you want to provide more comprehensive offers then please utilize the Vehicle Finance or Special Offers API features.

Note: Your branch must have access to this 'Watchlist Offer Product' in order to manage special offers. Please talk to your account manager if you have issues managing special offers via the API.

There are currently 2 types of offers:

- A discount based offer. (ie 10% off this vehicle for a limited time)
- A text offer (ie Free mechanical warranty)

Endpoints

HTTP Verb	URI	Effect
GET	watchlist-offers	Retrieves a page of watchlist offer summaries for your branch.
POST	watchlist-offers	Create a new watchlist offer.
GET	watchlist-offers/{id}	Retrieves a specific watchlist offer.
GET	watchlist-offers/types	Get watchlist offer type metadata.

Creating a Watchlist Offer.

The following structure is used when creating a watchlist offer. Note: Only 1 active or scheduled offer may be present on a listing at a given time.

Name	Type	Purpose	Extra Info / Rules
Attributes	Array	Attributes describing the offer.	Attribute ID's from the /watchlist-offers/types resource based on offer type.
ExpireAtUtc	Primitive	Date the offer expires in UTC	Must be at least a day after StartAtUtc. Maximum is 7 days (up to midnight NZT on the seventh day) after StartAtUtc.
OfferTypeId	Primitive	The ID of the offer type.	Type ID retrieved from /watchlist-offers/types.
StockNumber	Primitive	The stock number of the vehicle this offer applies to.	Must be an active listing and have a non-zero Listing ID.

The /watchlist-offers/types endpoint provides available types that are available to be offered. Each type has attributes. These need to be provided in the Attributes property on the main request structure. Each attribute is structured as follows:.

Name	Type	Purpose	Extra Info / Rules
NumericData	Primitive	Generic numeric data bucket for numeric type attributes	The /types attribute property 'DataType' can be used to determine whether or not this property needs to be populated for the attribute.
TextData	Primitive	Generic text data bucket for numeric type attributes	The /types attribute property 'DataType' can be used to determine whether or not this property needs to be populated for the attribute. Must not contain any email addresses, phone numbers or websites otherwise

			the request will be rejected. Maximum length = 2000 chars.
OfferAttributeType	Primitive	The ID of Offer Type Attribute	

Retrieving Watchlist Offer Summaries.

The GET /watchlist-offers endpoint can be used to retrieve a page of Watchlist Offers. Different query strings can be used to filter the returning offers.

Name	Type	Purpose	Extra Info / Rules
ExpireFromUtc	Primitive	Get offers that expire after this date	
ExpireToUtc	Primitive	Get offers that expire before this date	
StartFromUtc	Primitive	Get offers that start after this date	
StartToUtc	Primitive	Get offers that start before this date	
StockNumber	Primitive	Get offers linked to the vehicle with this stock number.	

Note: To retrieve all offers that are either active or scheduled then pass `ExpireFromUtc={nowUtc}` where {nowUtc} is the current date time in UTC. **It is currently not possible to have a scheduled offer and an active offer. This may or may not change in the future so assume that you may receive one or more active or scheduled offers for a given listing.**

Special Offers Resource

Description

The Special Offers resource is a CRUD resource for managing your branches Special Offers.

Note: Your branch must have access to this 'Special Offers Product' in order to manage special offers. Please talk to your account manager if you have issues managing special offers via the API.

Endpoints

HTTP Verb	URI	Effect
GET	special-offers	Retrieves all special offers for your branch..
POST	special-offers	Create a new special offer.
GET	special-offers/{id}	Retrieves a specific special offer.
PUT	special-offers/{id}	Updates an existing special offer (state replacement)
DELETE	special-offers/{id}	Deletes an existing special offer.
PUT	special-offers/{id}/images	Set the image for the special offer.
DELETE	special-offers/{id}/images	Delete the image for the special offer

Creating/Updating a Special Offer.

The following structure is used when creating / updating a special offer.

Name	Type	Purpose	Extra Info / Rules
Description	Primitive	The description of the special offer.	

DisplayName	Primitive	The name that is displayed in Dealerbase for this special offer. (Not visible on Trade Me)	Must be unique.
ImageUri	Primitive	The URL of the image associated with the special offer.	Optional. Null if no image has been set.
IsFinance	Primitive	Is this offer a finance offer?	Important: You must declare if this offer contains information relating to finance.
OfferUrl	Primitive	An URL relating to the offer.	
Subtitle	Primitive	The visible subtitle for the offer.	
TermsAndConditions	Primitive	Associated terms and conditions.	
Title	Primitive	The main visible title for the offer.	

Response

The request and response structures are almost identical, except the response will also return an Id property for you to use in subsequent requests.

Finance Offers Resource

Description

The Finance Offers resource is a resource for retrieving your branches Finance Offers.

Note: Your branch must have access to this Finance Offers Product' in order to use finance offers. Please talk to your account manager if you have issues managing finance offers. Currently Finance offers can only be managed via dealerbase web.

Endpoints

HTTP Verb	URI	Effect
GET	finance-offers	Retrieves all finance offers for your branch..

Creating/Updating a Finance Offer.

Currently, finance offers can only be managed via dealerbase web.

Response

Name	Type	Purpose
Id	Primitive	The Id of the offer. This can be used to apply the offer to a vehicle.
FinanceName	Primitive	The name that is displayed in Dealerbase for this finance offer. (Not visible on Trade Me)
InterestRate	Primitive	The estimated interest rate.
MinimumInterestRate	Primitive	The minimum possible interest rate.
MaximumInterestRate	Primitive	The maximum possible interest rate.
EstablishmentFee	Primitive	The establishment fee for the offer.
MonthlyFee	Primitive	The monthly maintenance fee for the offer.
DepositPercent	Primitive	The percent of the total value required for deposit.
TermInMonths	Primitive	The term of the offer, in months.
Pitch	Primitive	The description of the finance offer.
ApplicationFormUrl	Primitive	An URL used to apply for the offer.
TermsAndConditions	Primitive	Associated terms and conditions.

IsDefault	Primitive	Whether this is the default offer for the branch.
-----------	-----------	---

API Errors

Depending on the situation you may receive an error from the API.

The API will return two types of error responses depending on the situation.

For any error that is not a 400 BAD REQUEST you will receive the following Microsoft standard error structure -

<https://learn.microsoft.com/en-us/dotnet/api/microsoft.aspnetcore.mvc.problemdetails?view=aspnetcore-6.0>

The key properties to pay attention to on the structure are: Detail (a description of the error) and the Status (the status code)

For any error that is a 400 BAD REQUEST you will receive the following Microsoft standard error structure (which is based on the non-400 structure) -

<https://learn.microsoft.com/en-us/dotnet/api/microsoft.aspnetcore.mvc.validationproblemdetails?view=aspnetcore-6.0>

As well as the standard properties to pay attention to (as described above) - an extra property (Details) which is a Dictionary<string, string[]> will be returned. The key for the dictionary represents the property on the payload. The value represents one or many errors associated with the property. We will do our best to provide as many validation errors in the response as possible, but some functional/business logic errors may come back as a single error.

All errors will be extended with a 'traceld' property which can be used when submitting an issue to our Customer Experience team to help us troubleshoot your error.

Error Status Codes

Code	Description	Info
400	BAD REQUEST	The payload you provided, or query string parameter value, failed validation in one way or another.
401	UNAUTHORISED	Your authentication token has expired. Please retrieve a new one.
403	FORBIDDEN	You don't have access to that resource or

		product.
404	NOT FOUND	This will occur if you provide a reference id in a payload that doesn't exist physically or logically.
409	CONFLICT	The record you are trying to access already exists. (Will happen attempting to create a record for an ID that is already in use) or is locked.
422	UNPROCESSABLE ENTITY	The request was accepted, but you have exceeded a product slot quota.
500	INTERNAL SERVER ERROR	Something went wrong - please report this to customer support along with the trace id presented in your error message.
501	NOT IMPLEMENTED	As the API is a work in progress you may receive this code attempting to access a feature that hasn't been implemented yet. (ie some vehicle categories will be gradually rolled out)
503	SERVICE UNAVAILABLE	

Code Samples

Note: The below samples are just examples of calling our API to perform certain basic actions. Please read this document in its entirety in conjunction with each service's OpenAPI documentation to understand the full business and validation requirements needed for integration.

PHP

https://drive.google.com/file/d/1SJk3ILmHfL1mu9JGouWj8UnTw-fZQSN4/view?usp=drive_link