

monkey
SOFTWARE

Supplier Catalogues API

Technical Specification

V1.1

Author: Marjan Ilkov

Published: 01.05.2025

Monkey Software Pty Ltd

10 Chelmsford Street, Williamstown North Vic 3016

P: 1300 650 295 | NZ Toll Free: 0800 666 539

Website: www.monkeysoftware.com.au

TABLE OF CONTENTS

Change History.....	3
Overview.....	4
Architecture.....	4
Details.....	4
Security.....	5
Authentication.....	5
Request Headers.....	5
Supplier Catalogues API.....	6
Details.....	6
Primitive JSON Value Formats.....	6
Request.....	6
Response.....	6
Resources.....	8
Contact Lenses Catalogue.....	8
Frames Catalogue.....	10
Other Items Catalogue.....	11
Packages Catalogue.....	12
Solutions Catalogue.....	13
Spectacle Lenses Catalogue.....	14
Spectacle Lens Add-Ons Catalogue.....	15
Spectacle Lens Tints Catalogue.....	16
Static Reference Data.....	17

CHANGE HISTORY

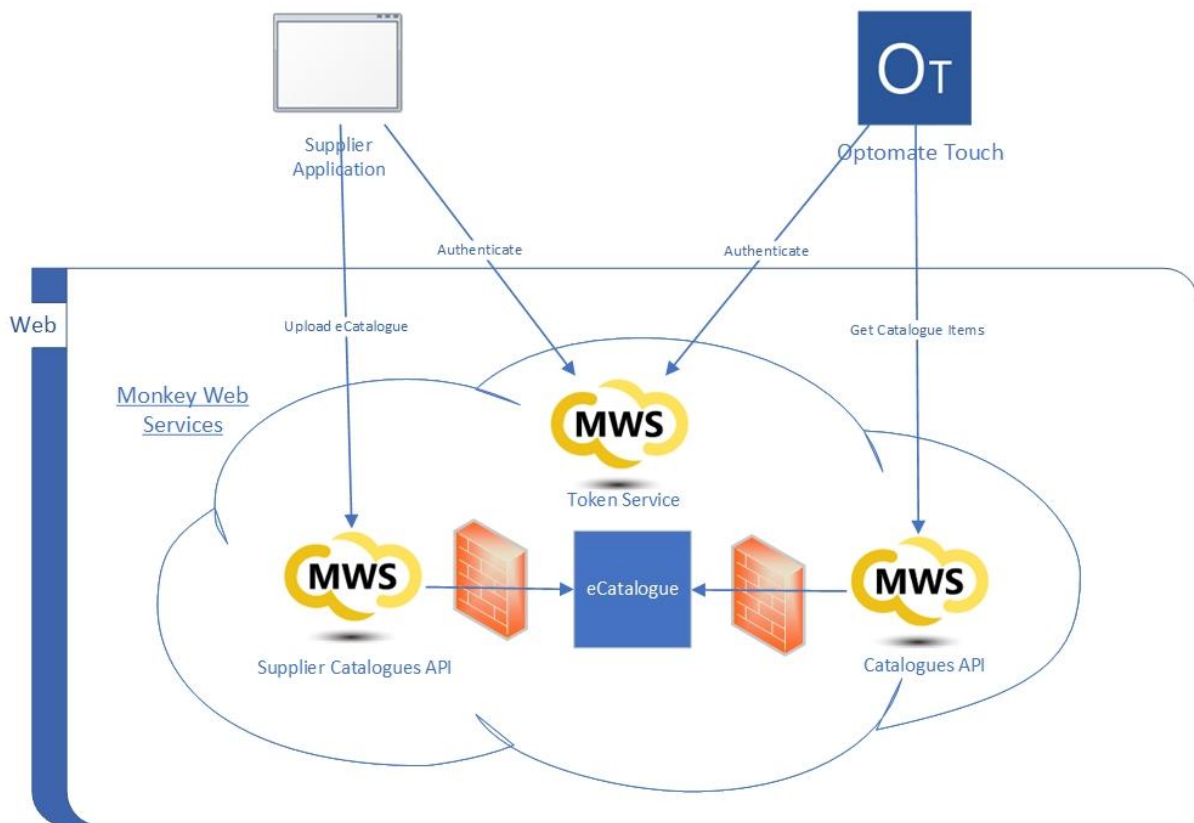
Potentially breaking changes are highlighted in red.

Version	Date	Changes
1.1	01.05.2025	New - Static Reference Frame Shapes resource Updates - Add Frame Shape and IsEcoFriendly
1.0	28.03.2023	Initial Release

OVERVIEW

The Supplier Catalogues API is a RESTful service that is hosted in the Monkey Web Services environment. It exposes several resources used for uploading data into the eCatalogue Manager Database. This document describes these resources in terms of their functionality, URIs and the method calls that can be performed on them. It also describes the format of the request and receive data for each method call and any associated input data validation.

ARCHITECTURE



The Supplier Catalogues API is an MWS API that allows for eCatalogue uploads into the Monkey eCatalogue Manager. These catalogues are available to be downloaded by Monkey clients' installed Optomate Touch instances.

DETAILS

Type	RESTful MWS Web Service
Name	Supplier Catalogues API
Description	API for uploading of Supplier eCatalogues into the Suppliers eCatalogue Portal
Authentication	Monkey Token



Specific details related to API authentication and URLs can be obtained from Monkey Software.

SECURITY



Every request sent to the API will need to contain an Authorization header containing a Monkey Token that can be retrieved from the Monkey Token service.

AUTHENTICATION

To be able to use the Supplier Catalogues API a supplier would need to be registered as a Monkey Software authorised supplier. The Monkey Token can be obtained by making a call to the MWS Token Service using the Supplier eCatalogue credentials.

REQUEST HEADERS

Authorization	Bearer <monkey_token>
---------------	-----------------------

SUPPLIER CATALOGUES API

DETAILS

PRIMITIVE JSON VALUE FORMATS

Null values are represented as the JSON literal null.

Values of type Boolean are represented as the JSON literals true and false

Values of type Byte, Int16, Int32, Int64, Decimal, Single and Double are represented as JSON numbers.

Values of type String, Binary, DateTimeOffset, Date, TimeOfDay and Guid are represented as JSON strings.

REQUEST

The Catalogues API implements the standard HTTP Request Methods as follows:

HTTP Method	Action
GET <Resource>	Return all records of the requested Static Reference data resource
POST <Resource>	Perform a full upload of the submitted eCatalogue matching on SuplplierStockCode <ul style="list-style-type: none"> - Create new items in the upload that do not exist on the MWS server - Update existing items if they have changes - Delete any existing items on the MWS server that are not in the upload If IgnoreErrorsAndUpload is set to true ignore any validation errors and upload the valid items
PATCH <Resource>	Perform a partial upload of the submitted eCatalogue matching on SuplplierStockCode <ul style="list-style-type: none"> - Create new items in the upload that do not exist on the MWS server - Update existing items if they have changes If IgnoreErrorsAndUpload is set to true ignore any validation errors and upload the valid items
DELETE <Resource>	Delete the submitted eCatalogue items matching on SuplplierStockCode

RESPONSE

The Catalogues API returns standard HTTP Responses such as:

HTTP Status Code	Response
200	OK
400	Bad Request
401	Unauthorized
404	Not Found
405	Forbidden
500	Internal Server Error

Common Record Types

Post Success Message

JSON

```
{  
  "Message": "Full upload of eCatalogue completed [ Inserted: {n}; Updated: {n}; Deleted: {n} ]"  
}
```

Patch Success Message

JSON

```
{  
  "Message": "Full upload of eCatalogue completed [ Inserted: {n}; Updated: {n} ]"  
}
```

Delete Success Message

JSON

```
{  
  "Message": " Deleting of {n} eCatalogue records completed"  
}
```

Error Message

When an error status code is returned, an error message may also be returned in the output.

Custom Errors

JSON

```
{  
  "Code": "string",  
  "Message": "string"  
}
```

The API will return the following custom errors when the request is unsuccessful:

Code	Message
100	Missing Credentials!
101	Invalid Credentials!
103	The supplied User does not have the required permissions for this request!
200	The supplied request is invalid!

Validation Errors

JSON

```
{
  "Code": "string",
  "Message": [
    {
      "SupplierStockCode": "string",
      "ValidationMessage": "string"
    }
  ]
}
```

If the request is accepted the API will perform validation on the supplied eCatalogue items and will return a Validation Errors message if at least one item fails validation. The Validation Errors will have a Code of "400" and will contain a list of 1 or more validation messages. Each Validation Message will be composed of one or more validation errors related to a given property.

Condition	Validation Text
Required	Missing {name}
Letters and Digits	Invalid {name} only letters and digits are allowed
Not Duplicate	Duplicate {name}
Valid Value (Allowed list)	Invalid {name}
String (Length)	Maximum length of {length} exceeded for {name}
Contains Valid Values (Allowed list)	{name} contains invalid value(s)



For POST and PATCH if the IgnoreErrorsAndUpload flag is set to true the API will ignore any validation errors and upload only the valid items.

RESOURCES

CONTACT LENSES CATALOGUE

POST /api/ContactLensesCatalogue

Input

JSON	Default	Conditions
"IgnoreErrorsAndUpload": boolean	false	
"Items": [
{		Required, Not Duplicate
"SupplierStockCode": "string (50)",		Required
"Description": "string (100)",		Not Duplicate
"ItemId": "string (20)",		Valid Value (Contact Lens Category Code)
"Category": "string",	"~MIS"	Not Duplicate, Letters and Digits
"Barcode": "string (14)",		
"EyetalkBookCode": "string (20)",		Valid Value (Contact Health Fund Item Single Health Fund Number)
"SingleHealthFundNumber": "string",		Valid Value (Contact Health Fund Item Multi Health Fund Number)
"MultiHealthFundNumber": "string",		
"IsTaxable": boolean,	true	
"ListPriceExcludingTaxAu": decimal,	0	

"ListPriceExcludingTaxNz": decimal,	0	
"RrpIncludingTaxAu": decimal,	0	
"RrpIncludingTaxNz": decimal,	0	
"Country": "string",		Valid Value (Country Code)
"Url": "string (300)",		
"OrderCode": "string (20)",		
"Brand": "string (20)",		
"BaseCurveMeasures": "string (200)",		*
"DiameterMeasures": "string (200)",		*
"HmlMeasures": "string (200)",		*, Contains Valid Values (HML Measures Code)
"DaysAtLab": int16,	0	
"Features": "string (50)",		
"Axis": "string (40)",		
"Power": "string (40)",		
"Tint": "string (40)",		
"WaterContent": double,	0	
"Oxygen": "string (10)",		
"IsDisposable": boolean,	false	
"ExtendedWear": boolean,	false	
"DaysUsagelnBox": int32,	0	
"Frequency": "string",		Valid Value (Frequency Code)
}		
]		
}		

* Item Counts Match - Message: BaseCurveMeasures, DiameterMeasures and HmlMeasures counts don't match

PATCH /api/ContactLensesCatalogue

Input

Same as POST.

DELETE /api/ContactLensesCatalogue

Input

JSON	Default	Conditions
<pre>{ "Items": [{ "SupplierStockCode": "string" }] }</pre>		

FRAMES CATALOGUE

POST /api/FramesCatalogue

Input

JSON	Default	Conditions
{		
"IgnoreErrorsAndUpload": boolean	false	
"Items": [
{		
"SupplierStockCode": "string (50)",		Required, Not Duplicate
"Barcode": "string (14)",		Not Duplicate, Letters and Digits
"Model": "string (20)",		Required
"Type": "string",		Required, Valid Value (Frame Type Code)
"Brand": "string (15)",		Required
"UvRating": null,		Valid Value (UV Rating Id)
"Gender": "string",	"UNISEX"	Valid Value (Gender Option Code/Name)
"Bridge": byte,		Valid Value (0-99)
"EyeSize": byte,	0	Valid Value (0-99)
"Depth": int32,		Valid Value (0-99)
"Diag": byte,		
"Temple": int32,	0	
"FrameCurve": int32,		
"PantoAngle": byte,		
"Construction": "string",	"~OTH"	Valid Value (Construction Type Code)
"Shape": "int16"		Valid Value (Frame Shape Id)
"Material": "string",	"~OTH"	Valid Value (Frame Material Code)
"ColourCode": "string (15)",		Required
"ColourDescription": "string (15)",		
"IsEcoFriendly": boolean	false	
"Gst": boolean,	true	
"Category": "string",	"~MIS"	Valid Value (Frame Category Code)
"ListPriceExAu": decimal,	0	
"ListPriceExNz": decimal,	0	
"RrplncAu": decimal,	0	
"RrplncNz": decimal,	0	
"Status": "string",		Valid Value (Statuses Code)
"Url": "string",		
"Packages": "string (100)",		
"MaxPlusPower": double,	0	
"MaxMinusPower": double,	0	
}		
]		
}		

PATCH /api/FramesCatalogue

Input

Same as POST.

DELETE /api/FramesCatalogue

Input

JSON	Default	Conditions
<pre>{ "Items": [{ "SupplierStockCode": "string" }] }</pre>		

OTHER ITEMS CATALOGUE

POST /api/OtherItemsCatalogue

Input

JSON	Default	Conditions
<pre>{ "IgnoreErrorsAndUpload": boolean "Items": [{ "SupplierStockCode": "string (50)", "Description": "string (100)", "ItemId": "string (20)", "Category": "string", "Barcode": "string (14)", "EyetalkBookCode": "string (20)", "HealthFundNumber": "string (5)", "IsTaxable": boolean, "ListPriceExcludingTaxAu": decimal, "ListPriceExcludingTaxNz": decimal, "RrpIncludingTaxAu": decimal, "RrpIncludingTaxNz": decimal, "Country": "string", "Url": "string", "OrderCode": "string (20)" }] }</pre>	<p>false</p> <p>true</p> <p>0</p> <p>0</p> <p>0</p> <p>0</p>	<p>Required, Not Duplicate Required Not Duplicate Valid Value (Other Item Category Code) Not Duplicate, Letters and Digits</p> <p>"~ACC"</p> <p>Valid Value (Country Code)</p>

PATCH /api/OtherItemsCatalogue

Input

Same as POST.

DELETE /api/OtherItemsCatalogue

Input

JSON	Default	Conditions
<pre>{ "Items": [{ "SupplierStockCode": "string" }] }</pre>		

PACKAGES CATALOGUE

POST /api/PackagesCatalogue

Input

JSON	Default	Conditions
<pre>{ "IgnoreErrorsAndUpload": boolean "Items": [{ "Code": "string (3)", "Description": "string (100)", "Price": decimal, "FramePrice": decimal, "LensPrice": decimal, "LensSupplierStockCode": "string (50)" }] }</pre>	<p>false</p> <p>0</p> <p>0</p> <p>0</p>	<p>Required, Not Duplicate Required</p>

PATCH /api/PackagesCatalogue

Input

Same as POST.

DELETE /api/PackagesCatalogue

Input

JSON	Default	Conditions
<pre>{ "Items": [{ "Code": "string" }] }</pre>		

SOLUTIONS CATALOGUE

POST /api/SolutionsCatalogue

Input

JSON	Default	Conditions
<pre>{ "IgnoreErrorsAndUpload": boolean "Items": [{ "SupplierStockCode": "string (50)", "Description": "string (100)", "ItemId": "string (20)", "Barcode": "string (14)", "EyetaikBookCode": "string (20)", "HealthFundNumber": "string (5)", "IsTaxable": boolean, "ListPriceExcludingTaxAu": decimal, "ListPriceExcludingTaxNz": decimal, "RrpIncludingTaxAu": decimal, "RrpIncludingTaxNz": decimal, "Country": "string", "Url": "string", "OrderCode": "string (20)" }] }</pre>	<p>false</p> <p>true</p> <p>0</p> <p>0</p> <p>0</p> <p>0</p>	<p>Required, Not Duplicate</p> <p>Required</p> <p>Not Duplicate</p> <p>Not Duplicate, Letters and Digits</p> <p>Valid Value (Country Code)</p>

PATCH /api/SolutionsCatalogue

Input

Same as POST.

DELETE /api/SolutionsCatalogue

Input

JSON	Default	Conditions
<pre>{ "Items": [{ "SupplierStockCode": "string" }] }</pre>		

SPECTACLE LENSES CATALOGUE

POST /api/SpectacleLensesCatalogue

Input

JSON	Default	Conditions
{		
"IgnoreErrorsAndUpload": boolean	false	
"Items": [
{		
"SupplierStockCode": "string (50)",		Required, Not Duplicate
"Description": "string (100)",		Required
"ItemId": "string (20)",		Not Duplicate
"Category": "string",	"~MIS"	Valid Value (Spectacle Lens Category Code)
"Barcode": "string (14)",		Not Duplicate, Letters and Digits
"EyetaalkBookCode": "string (20)",		
"SingleHealthFundNumber": "string",		Valid Value (Lens Health Fund Item Single)
"MultiHealthFundNumber": "string",		Valid Value (Lens Health Fund Item Multi)
"AddOnSingleHealthFundNumbers": "string (200)",		*, Contains Valid Values (Lens Health Fund AddOn Type)
"AddOnMultiHealthFundNumbers": "string (200)",		*, Contains Valid Values (Lens Health Fund AddOn Type)
"IsTaxable": boolean,	true	
"AccountDescription": "string (100)",		
"CostExcludingTaxAu": decimal,	0	
"CostExcludingTaxNz": decimal,	0	
"CostIncludingTaxAu": decimal,	0	
"CostIncludingTaxNz": decimal,	0	
"SellIncludingTaxAu": decimal,	0	
"SellIncludingTaxNz": decimal,	0	
"ListPriceExcludingTaxAu": decimal,	0	
"ListPriceExcludingTaxNz": decimal,	0	
"RrpIncludingTaxAu": decimal,	0	
"RrpIncludingTaxNz": decimal,	0	
"Country": "string",		Valid Value (Country Code)
"OrderCode": "string (20)",		
"Diameter": double,	0	
"BaseCurve": double,	0	
"SegSize": double,	0	
"HighestSph": double,	0	
"LowestSph": double,	0	
"RefractiveIndex": double,	0	
"StockGrind": "string",		Valid Value (Spectacle Lens Stock Grind Code)
"Material": "string",		Valid Value (Spectacle Lens Material Code)
"IsPolarised": boolean,	false	
"Transitions": boolean,	false	
"Type": "string",		Valid Value (Spectacle Lens Type Code)
"Url": "string (300)",		
"IsMultiCoatNz": boolean,	false	
"IsHardCoatNz": boolean,	false	
"IsHighIndexNz": boolean,	false	
"IsPhotoGlassNz": boolean,	false	
"IsPhotoResinNz": boolean,	false	
"MaxCylinder": double,	0	
"DaysAtLab": in32,	0	
"IsSunRx": boolean,	false	
"IsCustomised": boolean,	false	
"NeedHeights": boolean,	false	
"CorridorLength": "string (200)",		
"FocalRange": "string (200)",		
"MinimumHeight": int32,	0	
"BaseCurveMulti": "string (100)",		

Supplier Catalogues API

```
"IsAvailableInGrey": boolean,           false
"IsAvailableInBrown": boolean,         false
"IsAvailableInG15": boolean,           false
}
]
}
```

* Item Counts Match - Message: AddOnSingleHealthFundNumbers count does not match AddOnMultiHealthFundNumbers count

PATCH /api/SpectacleLensesCatalogue

Input

Same as POST.

DELETE /api/SpectacleLensesCatalogue

Input

JSON	Default	Conditions
<pre>{ "Items": [{ "SupplierStockCode": "string" }] }</pre>		

SPECTACLE LENS ADD-ONS CATALOGUE

POST /api/SpectacleLensAddOnsCatalogue

Input

JSON	Default	Conditions
<pre>{ "IgnoreErrorsAndUpload": boolean, "Items": [{ "SupplierStockCode": "string (50)", "Description": "string (100)", "ItemId": "string (20)", "Barcode": "string (14)", "EyetalkBookCode": "string (20)", "SingleHealthFundNumber": "string (5)", "MultiHealthFundNumber": "string (5)", "IsTaxable": boolean, "AccountDescription": "string (100)", "CostExcludingTaxAu": decimal, "CostExcludingTaxNz": decimal, "SellIncludingTaxAu": decimal, "SellIncludingTaxNz": decimal, "RrpIncludingTaxAu": decimal, "RrpIncludingTaxNz": decimal, "Country": "string", }] }</pre>	<pre>false true 0 0 0 0 0 0</pre>	<pre>Required, Not Duplicate Required Not Duplicate Not Duplicate, Letters and Digits Valid Value (Country Code)</pre>


```

"OrderCode": "string (20)",
"IsMultiCoatNz": boolean           false
}
]
}

```

PATCH /api/SpectacleLensAddOnsCatalogue

Input

Same as POST.

DELETE /api/SpectacleLensAddOnsCatalogue

Input

JSON	Default	Conditions
<pre> { "Items": [{ "SupplierStockCode": "string" }] } </pre>		

SPECTACLE LENS TINTS CATALOGUE

POST /api/SpectacleLensTintsCatalogue

Input

JSON	Default	Conditions
<pre> { "IgnoreErrorsAndUpload": boolean "Items": [{ "SupplierStockCode": "string (50)", "Description": "string (100)", "ItemId": "string (20)", "Barcode": "string (14)", "SingleHealthFundNumber": "string (5)", "MultiHealthFundNumber": "string (5)", "IsTaxable": boolean, "AccountDescription": "string (100)", "CostExcludingTaxAu": decimal, "CostExcludingTaxNz": decimal, "SellIncludingTaxAu": decimal, "SellIncludingTaxNz": decimal, "RrpIncludingTaxAu": decimal, "RrpIncludingTaxNz": decimal, "Country": "string", "OrderCode": "string (20)", "Type": "string" }] } </pre>	<p>false</p> <p>true</p> <p>0</p> <p>0</p> <p>0</p> <p>0</p> <p>0</p> <p>0</p> <p>0</p>	<p>Required, Not Duplicate</p> <p>Required</p> <p>Not Duplicate</p> <p>Not Duplicate, Letters and Digits</p> <p>Valid Value (Country Code)</p> <p>Valid Value (Spectacle Lens Tint Type Code)</p>

```
}
```

PATCH /api/SpectacleLensTintsCatalogue

Input

Same as POST.

DELETE /api/SpectacleLensTintsCatalogue

Input

JSON	Default	Conditions
<pre>{ "Items": [{ "SupplierStockCode": "string" }] }</pre>		

STATIC REFERENCE

GET StaticReference/<Resource>

Gets all records for the requested resource.

Resources

Name	Output Type
ConstructionTypes	Reference Code
Countries	Reference Code
GenderOptions	Reference Code
FrameCategories	Reference Code
FrameMaterials	Reference Code
FrameShapes	Reference Id
FrameTypes	Reference Code
LensHealthFundItems	Reference Add-On Type
LensHealthFundAddOnTypes	Reference Add-On Type
SpectacleLensCategories	Reference Code
SpectacleLensMaterials	Reference Code
SpectacleLensStockGrind	Reference Code
SpectacleLensTypes	Reference Code
SpectacleLensTintTypes	Reference Code
Statuses	Reference Code
UvRating	Reference Id

Output

Reference Code record.

JSON

```
[
  {
    "Code": "string",
    "Description": "string"
  }
]
```

Reference Id record.

JSON

```
[
  {
    "Id": byte,
    "Description": "string"
  }
]
```

Reference Add-On Type record.

JSON

```
[
  {
    "Name": "string",
    "SingleHealthFundNumber": "string",
    "MultiHealthFundNumber": "string"
  }
]
```
