19.09. mSupply REST API

What is it?

mSupply has a REST API that enables certain actions to be carried out externally, by external applications for example. This API allows you to:

- Send an order (response requisition) to mSupply
- Retrieve customer invoices that have been confirmed but not yet been acknowledged or received
- Mark a customer invoice as acknowledged (shipped)
- Mark a customer invoice as received (arrived)
- Retrieve stock information

Response requisitions will be processed in mSupply as normal in the supplying store, and can be fulfilled by generating and processing one or more customer invoices, depending on stock availability.

Basic usage

- Communication is over the HTTP protocol. HTTPS is supported and you should use it to keep your passwords and other information secure. If you want to restrict to only HTTPS, then block the HTTP port mSupply is using on your router.
- All data is submitted and returned as JSON.
- You must always login first. This will return a JWT token that you use in further requests to authenticate each request.

The details of the requests you can make and what information you need to send with each request are in the section below.

Authentication

For all endpoints except /storeCustomerInvoice

Logins (user name and password) are set up per store as contacts of store names. This uses exactly the same mechanism as customer web ordering. See the 19.03. mSupply customer web interface page for details.

All requests relate to transactions, stock and orders **only** in the store that the user used to authenticate with is a contact of. The customer and supplier's details are derived from the log-in details i.e. the customer will be the store the logged in user is a contact of and the supplier will be the supplying store of that store.

For the /storeCustomerInvoice endpoint

Logins are setup as normal mSupply users. A search for matching customer invoices will be made in the user's default store.

API definition

See the Examples section below for payload details for each request.

All below URLs should be prefixed with `api/v4`:

URL	HTTP Verb	Description
/login	POST	Login with username and password setup in mSupply. Returns a JWT cookie to be used in following requests (send it in an Authorization header with scheme Bearer i.e. `Authorization: Bearer <jwt_token>`). Include the `loginType = "invoice"` element in the request body or, for the /storeCustomerInvoice endpoint, include the `loginType = "storeInvoice"` element.
/stock	GET	Returns an array of stock details for all items in the customer's master list(s). * Quantities are specified in number of packs and pack size. * Separate lines for each different batch, expiry date or pack size.
/stock?code=xxx&name=yyy	GET	Returns an array of stock details for all visible items, optionally restricted to those starting with the specified code/name. * Quantities are specified in number of packs and pack size. * Separate lines for each different batch, expiry date or pack size.

URL	HTTP Verb	Description
/customerOrder	POST	Create a new customer order with the supplied array of order lines. Returns the ID of the order created. * The customer and supplier's details are derived from the log-in details i.e. the customer will be the store the logged in user is a contact of and the supplier will be the supplying store of that store. * An order (serial) number will be automatically assigned by mSupply (unique per store). * Item codes must match the codes (the mSupply item code or the catalogue code) of items already existing in mSupply. * Requested quantities are specified in number of packs and pack size. * Customer stock on hand quantities (if available) are specified in units. * Customer orders are created as response (customer) requisitions in mSupply. These are then processed manually in the supplying store to generate one or more customer invoices to fulfil the order (see the 8.07. Stock control methods for your customers (requisitions) page for details). * Customer invoice numbers will also be automatically assigned by mSupply (unique per store). * Generated customer invoices will be linked back to the original customer order via the order number. * Customer invoices are confirmed when the stock has been picked and packed for shipment. * Comments can be entered for the order and also for individual lines.
/customerOrder/{order number}	GET	Return the details of the specified order number.

URL	HTTP Verb	Description
/customerInvoice	GET	Returns an array of customer invoices which have been confirmed but not acknowledged, received or cancelled. * Invoices are linked to the original order via the order number. * Quantities are specified in number of packs and pack size. * Prices are per-pack and a line total, with an invoice total which includes any extras (e.g. shipping).
/customerInvoice/{invoice number}	GET	Returns a particular customer invoice. * Invoices are linked to the original order via the order number. * Quantities are specified in number of packs and pack size. * Prices are per-pack and a line total, with an invoice total which includes any extras (e.g. shipping).
/customerInvoiceReceived/{invoice number}	PATCH	Set the specified customer invoice's received (arrival) date to the date specified (in ISO format).
/customerInvoiceAcknowledged/{invoice number}	PATCH	Set the specified customer invoice's acknowledged (shipped) date to the date specified (in ISO format).
/storeCustomerInvoice?startDate=YYYY-MM-DD&endDate=YYYY-MM-DD	GET	Return all confirmed (and not cancelled) customer invoices in the default store of the user logged in with, which were confirmed between the startDate and endDate.

Examples

URL	HTTP Verb	Response

api/v4/login	POST	Request:
		`{`
		`"username": "Admin",`
		`"password": "Admin",`
		`"loginType": "invoice"`
		`}`
		Response:
		`// success (HTTP status 200)`
		`{`
		`"status": "success",`
		`"authenticated": true,`
		`"username": "Admin",`
		`"userFirstName": "Minnie",`
		`"userLastName": "Streator",`
		`"userJobTitle": "Boss",`
		`"userType": "contact",`
		`"service": "invoice",`
		`"storeName": "General Warehouse"`
		`}`
		`// failure (HTTP status 400)`
		`{`
		`"status": "error",`
		`"error": "Username/password/login type missing"`
		`}`
		`// failure (HTTP status 401)`
		`{`
		`"status": "error",`
		`"error": "Failed to authenticate/No store found for user"`
		`}`

api/v4/stock	GET	Response:
		`// success (HTTP status 200)`
		`[`
		`{`
		`"itemCode": "ABC012",`
		`"itemName": "Amoxycillin 250mg tab",`
		`"batchName": "SD34567",`
		`"expiryDate": "2021-05-05T12:00:00.000Z",`
		`"unit": "Tab",`
		`"barcode": "0123456778899",`
		`"packSize": 100,`
		`"quantity": 23, // in packs`
		`"costPricePerPack": 1.23, // in default currency`
		`"sellPricePerPack": 1.33, // in default currency`
		`"storeName": "General Warehouse"`
		`},`
		`...`
		`]`
		`// failure (HTTP status 401)`
		`{`
		`"status": "error",`
		`"error": "JWT token/user ID/store ID not found"`
		`}`
		The stock levels will be those of the supplying store for the logged in customer, with separate lines for different batches, expiry dates or pack sizes. By default, the items included will be those in the customer's master list(s), unless overridden by the optional item name/code parameters, in which case the results will be filtered by items with a name/code starting with the specified value(s).

URL	HTTP Verb	Response

api/v4/customerOrder	POST	Request:
		`{`
		`  "orderReference": "VS2345",`
		`  "comment": "Test order",`
		`  "lines":[`
		`  {`
		`  "itemCode": "ABC012",`
		`  "itemName": "Amoxycillin 250mg tab",`
		`  "packSize": 100,`
		`  "quantity": 3, // in packs`
		`  "SOH": 3500, // current customer stock on hand (in units)`
		`  "comment": "something about this line"`
		`  },`
		`  {`
		`  "itemCode": "CYZ456",`
		`  "itemName": "Paracetamol 500mg tab",`
		`  "packSize": 100,`
		`  "quantity": 5, // in packs`
		`  "comment": ""`
		`  },`
		`  ...`
		`  ]`
		`}`
		Response:
		`// success (HTTP status 200)`
		`{`
		`  "status": "success",`
		`  "numberOfRecordsUpdated": 1,`
		`  "orderNumber": 23425`
		`}`
		`// failure (HTTP status 404)`
		`{`
		`  "status": "error",`
		`  "error": "Item code not found"`
		`}`
		`// failure (HTTP status 409)`
		`{`
		`  "status": "error",`
		`  "error": "Duplicate line for item"`
		`}`
		`// failure (HTTP status 403)`
		`{`
		`  "status": "error",`
		`  "error": "Order already exists"`
		`}`
		`// failure (HTTP status 403)`
		`{`
		`  "status": "error",`
		`  "error": "Item is not available to order"`
		`}`
		`// failure (HTTP status 403)`
		`{`
		`  "status": "error",`
		`  "error": "Invalid pack size/quantity"`
		`}`
		`// failure (HTTP status 400)`
		`{`
		`  "status": "error",`
		`  "error": "Order reference/order lines/item code/item name/quantity missing"`
		`}`
		`// failure (HTTP status 401)`
		`{`
		`  "status": "error",`
		`  "error": "JWT token/user ID/store ID not found"`
		`}`
		The customer order (response requisition) will be created in the logged in customer's supplying store, set to confirmed with the current date, and the new (automatically assigned) customer order number returned

URL	HTTP Verb	Response

api/v4/customerOrder/23425	GET	Response:
		// success (HTTP status 200)
		{
		"ID": "UID23425",
		"confirmedDate": "2020-12-27T17:51:00.000Z",
		"orderNumber": 23425,
		"orderReference": "VS2345",
		"comment": "Test order",
		"storeName": "General Warehouse",
		"lines":[
		{
		"itemCode": "ABC012",
		"itemName": "Amoxycillin 250mg tab",
		"packSize": 100,
		"quantity": 2, // in packs
		"comment": "Reduced quantity supplied"
		},
		{
		"itemCode": "CYZ456",
		"itemName": "Paracetamol 500mg tab",
		"packSize": 100,
		"quantity": 5, // in packs
		"comment": ""
		},
		...
		]
		}
		// failure (HTTP status 404)
		{
		"status": "error",
		"error": "Order not found"
		}
		// failure (HTTP status 409)
		{
		"status": "error",
		"error": "More than one order found"
		}
		// failure (HTTP status 400)
		{
		"status": "error",
		"error": "Order number missing"
		}
		// failure (HTTP status 401)
		{
		"status": "error",
		"error": "JWT token/user ID/store ID not found"
		}
		Returns the specified customer order in the logged in customer's supplying store

URL	HTTP Verb	Response

api/v4/customerInvoice	GET	Response:
		// success (HTTP status 200)
		[
		{
		"ID": "UID8567546AS",
		"invoiceNumber": 8567546,
		"confirmedDate": "2020-12-29T12:45:00.000Z",
		"receivedDate": "",
		"acknowledgedDate": "",
		"cancelledDate": "", // blank unless the invoice has been manually cancelled after it has been confirmed
		"invoiceReference": "REF222",
		"comment": "From order reference VS2345",
		"orderNumber": 23425, // original order number (can have multiple invoices from a single order)
		"storeName": "General Warehouse",
		"invoiceTotal": 135.55, // sum of all lines plus any extras e.g. shipping
		"lines": [
		{
		"itemCode": "ABC012",
		"itemName": "Amoxycillin 250mg tab",
		"batchName": "SD34567",
		"expiryDate": "2021-05-05T12:00:00.000Z",
		"unit": "Tab",
		"barcode": "0123456778899",
		"packPrice": 3.65,
		"packSize": 100,
		"quantity": 2, // in packs
		"comment": "Reduced quantity supplied; Requested pack size 100",
		"lineTotal": 7.3 // pack price x quantity
		},
		...
		]
		},
		...
		]
		// failure (HTTP status 401)
		{
		"status": "error",
		"error": "JWT token/user ID/store ID not found"
		}
		Returns all customer invoices for the logged in customer in their supplying store and confirmed, but which haven't yet been acknowledged, received or cancelled. Note that a single customer order can be split into multiple customer invoices

URL	HTTP Verb	Response

api/v4/customerInvoiceReceived/8567546	PATCH	Request:
		`{`
		`"receivedDate": "2020-12-31T09:23:00.000Z"`
		`}`
		Response:
		`// success (HTTP status 200)`
		`{`
		`"status": "success",`
		`"numberOfRecordsUpdated": 1`
		`}`
		`// failure (HTTP status 404)`
		`{`
		`"status": "error",`
		`"error": "Invoice not found"`
		`}`
		`// failure (HTTP status 409)`
		`{`
		`"status": "error",`
		`"error": "More than one invoice found"`
		`}`
		`// failure (HTTP status 503)`
		`{`
		`"status": "error",`
		`"error": "Unable to update invoice"`
		`}`
		`// failure (HTTP status 503)`
		`{`
		`"status": "error",`
		`"error": "receivedDate is invalid"`
		`}`
		`// failure (HTTP status 403)`
		`{`
		`"status": "error",`
		`"error": "Invoice is not yet ready for dispatch"`
		`}`
		`// failure (HTTP status 403)`
		`{`
		`"status": "error",`
		`"error": "Invoice has been already been received/cancelled"`
		`}`
		`// failure (HTTP status 400)`
		`{`
		`"status": "error",`
		`"error": "Invoice number/received date missing"`
		`}`
		`// failure (HTTP status 401)`
		`{`
		`"status": "error",`
		`"error": "JWT token/user ID/store ID not found"`
		`}`
		Updates the received date for the specified customer invoice → it will no longer appear in the GET response above

URL	HTTP Verb	Response

api/v4/customerInvoiceAcknowledged/8567546	PATCH	Request:
		{
		"acknowledgedDate":
		"2020-12-31T09:23:00.000Z"
		}
		Response:
		// success (HTTP status 200)
		{
		"status": "success",
		"numberOfRecordsUpdated": 1
		}
		// failure (HTTP status 404)
		{
		"status": "error",
		"error": "Invoice not found"
		}
		// failure (HTTP status 409)
		{
		"status": "error",
		"error": "More than one invoice found"
		}
		// failure (HTTP status 503)
		{
		"status": "error",
		"error": "Unable to update invoice"
		}
		// failure (HTTP status 503)
		{
		"status": "error",
		"error": "acknowledgedDate is invalid"
		}
		// failure (HTTP status 403)
		{
		"status": "error",
		"error": "Invoice is not yet ready for
		dispatch"
		}
		// failure (HTTP status 403)
		{
		"status": "error",
		"error": "Invoice has been already been
		acknowledged/cancelled"
		}
		// failure (HTTP status 400)
		{
		"status": "error",
		"error": "Invoice number/acknowledged date
		missing"
		}
		// failure (HTTP status 401)
		{
		"status": "error",
		"error": "JWT token/user ID/store ID not
		found"
		}
		Updates the acknowledged date for the specified customer invoice → it will no longer appear in the GET response above

URL	HTTP Verb	Response

api/v4/customerInvoice/8567546	GET	Response:
		// success (HTTP status 200)
		{
		"ID": "UID8567546AS",
		"invoiceNumber": 8567546,
		"confirmedDate": "2020-12-29T12:45:00.000Z",
		"receivedDate": "2020-12-31T09:23:00.000Z",
		"cancelledDate": "", // blank unless the invoice has been manually cancelled after it has been confirmed
		"invoiceReference": "REF222",
		"comment": "From order reference VS2345",
		"orderNumber": 23425, // original order number
		"storeName": "General Warehouse",
		"invoiceTotal": 135.55, // sum of all lines plus any extras e.g. shipping
		"lines": [
		{
		"itemCode": "ABC012",
		"itemName": "Amoxycillin 250mg tab",
		"batchName": "SD34567",
		"expiryDate": "2021-05-05T12:00:00.000Z",
		"unit": "Tab",
		"barcode": "0123456778899",
		"packPrice": 3.65,
		"packSize": 100,
		"quantity": 2, // in packs
		"comment": "Requested pack size 100",
		"lineTotal": 7.3 // pack price x quantity
		},
		...
		]
		}
		// failure (HTTP status 404)
		{
		"status": "error",
		"error": "Invoice not found"
		}
		// failure (HTTP status 409)
		{
		"status": "error",
		"error": "More than one invoice found"
		}
		// failure (HTTP status 401)
		{
		"status": "error",
		"error": "JWT token/user ID/store ID not found"
		}
		Returns the specified customer invoice for the logged in customer in their supplying store

URL	HTTP Verb	Response

api/v4/storeCustomerInvoice?startDate=2023-07-23&endDate=2023-09-23	GET	Response:
		`// success (HTTP status 200)`
		`{`
		`"ID": "ABCD1234EFGH5678",`
		`"customerName": "Customer01",`
		`"customerCode": "CT32695",`
		`"invoiceNumber": 25387,`
		`"orderNumber": 4578, // Requisition number`
		`"confirmedDate": "2023-07-30T09:32:00.000Z",`
		`"receivedDate": "2023-07-22T19:11:00.000Z",`
		`// Invoice entry date`
		`"invoiceReference": "NEW3756",`
		`"comment": "From order reference ORD768",`
		`"storeName": "General Warehouse",`
		`"invoiceTotal": 742.56, // sum of all lines plus any extras e.g. shipping`
		`"lines": [`
		`{`
		`"itemCode": "ABC012",`
		`"itemName": "Amoxycillin 250mg tab",`
		`"batchName": "SD34567",`
		`"expiryDate": "2026-05-31T12:00:00.000Z",`
		`"unit": "Tab",`
		`"barcode": "0123456778899",`
		`"packPrice": 3.65, // Sell price of one pack`
		`"packSize": 100,`
		`"quantity": 2, // in packs`
		`"comment": "Requested pack size 100",`
		`"lineTotal": 7.3 // pack price x quantity`
		`},`
		`...`
		`]`
		`}`
		`// failure (HTTP status 401)`
		`{`
		`"status": "error",`
		`"error": "Selected store is not default for user"`
		`}`
		`// failure (HTTP status 401)`
		`{`
		`"status": "error",`
		`"error": "JWT token/user ID/store ID not found"`
		`}`
		`// failure (HTTP status 401)`
		`{`
		`"status": "error",`
		`"error": "End date/Start date is mandatory"`
		`}`
		Returns customer invoices in the logged in user's default store that were confirmed and not cancelled between the startDate and endDate

// Previous: **19.08. Stock web app** | Next: **19.10. mSupply legacy REST APIs** //

From:
https://docs.msupply.org.nz/ - **mSupply documentation wiki**

Permanent link:
https://docs.msupply.org.nz/web_interface:rest_api?rev=1757503007

Last update: **2025/09/10 11:16**