

# Managely to Manitou Integration

05/24/2024 8:07 am EDT

## Scope

This describes the models, queries, and mutations for use with integrating Manitou services in Managely.

This allows the user to push a full Manitou customer with links to a Managely site, push a Manitou system with links to a Managely system, edit customer addresses, add/edit/delete customer contact points, contact list contacts, customer passwords, areas, and zones.

This also allows the user to sync limited data between a Manitou Customer and Managely Site, and a Manitou Customer System and a Managely System.

Also, the user can view customer event history, and set a customer site on or off test.

Example queries/mutations can be found at <https://bold-group.postman.co/workspace/SedonaX~70b03fbe-938c-4e36-bf0a-717763c1016a/folder/15420852-8906a73f-d0ec-4e2c-ab43-68f4c63effe5>

See Toby for instructions on connecting to a Manitou testing system.

## Manitou Definitions

**Customer:** Entity for which alarm monitoring is contracted. This is associated with a Managely Site

**Contact Point:** The type of contact for this containing entity. For example, phone contact points could be Site, Home, Business, or Mobile. Email contact types could be Email, Email 2, or Email 3. A customer has contact points directly associated with it, and a customer's contact list has contacts that themselves can have multiple contact points.

**Contact List:** These are entities available for contact while handling an event. This contains information about the contact such as name, time zone, etc. Each contact in the contact list has one or more contact points.

**Customer System:** These are associated with Managely systems. Customer systems contain data such as the panel type, areas, and zones.

## Mutation Models

### Customer Model

**SerialNo:** The customer serialNo. Null for Customer push, as this will be generated at the time the customer is created.

**CustomerId:** String(32) identifier for the customer. This must be unique in the Manitou database.

**DealerSerialNo:** serialNo from Dealers query (Dealer query takes CountryId from Countries)

**BranchSerialNo:** serialNo from Branches query

Name: String(50) Customer's name

SubtypeId: 0 for a residential customer, 1 for a commercial customer.

CountryId: countryId from Countries query

LocalId: localId from Locales query

TimeZoneId: timeZoneId from the TimeZones query

CrossStreet: String(35) Free form data

Subdivision: String(35) Free form data

PoliceSerialNo: serialNo from the Authorities query with a type of POLICE

FireSerialNo: serialNo from the Authorities query with a type of FIRE

MedicalSerialNo: serialNo from the Authorities query with a type of MEDICAL

Address: Address model for this customer.

ContactPoints: List of contact point models associated with the for the customer (not to be confused with contact list contacts).

ContactList: List of ContactList models for customer contact list

CustomerSystems: List of CustomerSystem models. These associate with Managely Systems.

CustomerPasswords: List of Passwords associated with the customer. Can be flagged as duress or not duress. This is different from contact list passwords.

## Address Model

SerialNo: The customer serialNo for this address. Null for Customer push.

Address1: String(50) Address line 1

Address2: String(50) Address line 2

Address3: String(50) Address line 3

Address4: String(50) Address line 4

CityId: cityId from Cities query

CountryId: countryId from Countries query.

RegionId: regionId from Regions query (equivalent to states)

Postcode: String(15) postal (zip) code

## ContactPoint Model

SerialNo: The serialNo of the containing type.

- If this is a customer contact point, this is the serialNo of the customer.
- If this is a contact list contact point, this is the serialNo of the contact.
- Null if this is in a Customer push or a Contact List contact create.

ParentSerialNo: The serialNo of the customer.

- If this is a customer contact point, this will be the same as this contact point's serialNo.
- Null if this is in a Customer push or a Contact List contact create.

ContactType: From the ManitouContactType enum.

- CUSTOMERS for a customer contact point.
- PEOPLE for a contact list contact point.
- Null for a Customer push or a Contact List contact create

Sequence: The database sequence number this contact point in the contact point list for the customer or contact list contact.

- Null for Customer push, Contact Point create, or Contact List create.

DisplaySequence: The display sequence number for this contact point in the contact point list for the customer or contact list contact.

CountryId: countryId from the Countries query.

ContactPointTypeId: contactPointTypeId from the ContactPointTypes query. This determines if it's a phone, pager, or email type.

- From the ContactPointTypes query, attributes of "P" are phone types, "M" are pager types, and "E" are e-mail types.

ContactPointText: String(256) the phone number or the e-mail address for this contact point.

Extension: String(10) The phone number extension for phone contact point types.

DocFormat: From the ManitouDocFormat enum. This is for e-mail contact point types only. Null for non-email contact point types.

OutputDeviceId: outputDeviceTypeId from the OutputDeviceTypes query. Only used for Email and Pager contact point types.

- If this is an email field, filter on attribute = E for the OutputDeviceTypes query. If this is a pager, filter on attribute = M.

ServiceId: serviceProviderDeviceCode from ServiceProviderTypes query. If this is an e-mail or pager contact point, filter on the protocol property from the matching outputDeviceType if the outputDeviceType has serviceCodeRequired = true.

## CustomerPassword Model

Note: This is for passwords attached at the customer level, not passwords for contact list contacts.

SerialNo: The Customer's serialNo. Null for Customer push.

Sequence: The sequence number for this password.

- This with the serialNo make a unique key.

Password: String(25) The password

Description: String(35) Description of this password.

IsDuress: Flag indicating if this is a duress password.

## ContactList Model

Note: This is the contact list of contacts that will be called during an event. Not to be confused with the Customer contact points.

SerialNo: The serialNo of the customer this contact is linked to.

- Null for Customer push.

MemberSerialNo: The serialNo specific to this contact.

- Null for Customer push and Contact List contact create.

MemberSubtype: suptypeld from the ContactPointSubTypes query.

Sequence: Sequence number this contact in the database.

- Null for Customer push and Contact List contact create.

DisplaySequence: The display sequence number for this contact in the contact list.

Name: String(50) Name of this contact

CountryId: countryId from the Countries query.

LocalId: localId from Locales query

TimeZoneld: timeZoneld from the TimeZones query

Password: String(25) Password associated with this contact.

ContactPoints: List of contact points associated with this contact list contact. Only used when creating a contact list contact.

## CustomerSystem Model

SerialNo: The customer's serialNo.

- Null for Customer push.

SystemNo: The system number.

- Must be unique for this customer.

MonitoringServiceSequence: The sequence number for this system in the monitoring services

- Null for Customer push and Customer System push.

Description: String(35) Description for this system.

PanelTypeId: panelTypeId from the PanelTypes query.

ListOfAreas: The list of areas associated with this system.

ListOfZones: The list of zones associated with this system

## Area Model

SerialNo: The customer serialNo for this system. Null for Customer push or Customer System push.

SystemNo: The systemNo for the system this area is under. Null for Customer push or Customer System push.

AreaId: String(4) The ID of this area

Description: String(35) The description of this area

## Zone Model

SerialNo: The customer serialNo for this system. Null for Customer push or Customer System push.

SystemNo: The systemNo for the system this zone is under. Null for Customer push or Customer System push.

AreaId: String(4) The area ID from the Area, or \* for all areas.

- If not "\*", the areaId must match an existing area or one being pushed alongside this zone.

ZoneId: String(12) The ID of this zone

Description: String(50) The description of this zone.

## Out Of Service Model

SerialNo: The customer's serialNo for the system to be put on test

SystemNo: The system's systemNo to be put on test

LogSequence: The log sequence number for the test to be canceled.

Description: String(80) Reason this customer system is being put on or taken off test.

From: DateTime when the on test will start in the customer's local time. Must be in the past.

To: DateTime when the on test will end in the customer's local time. Must be in the future, and after the from datetime.

## Queries

### Queries for Dropdowns

Countries: Gets the list of countries defined in Manitou.

- CountryId is used for reference in the other models.

Regions: Gets the list of states/provinces defined in Manitou.

- Filter by countryId.
- RegionId is used for reference in the other models.

Cities: Gets the list of cities defined in Manitou.

- Requires passing in a regionId from the Regions query.
- CityId is used for reference in the other models.

Dealers: Gets the list of dealers defined in Manitou.

- Requires passing in a countryId from the Countries query or 0 to get all dealers regardless of country.
- SerialNo is used for reference in the other models.

Branches: Gets the list of branches defined in Manitou.

- SerialNo is used for reference in the other models.

TimeZones: Gets the list of time zones defined in Manitou.

- TimeZoneId is used for reference in the other models.

Locales: Gets the list of language locals defined in Manitou.

- LocalId is used for reference in the other models.

Authorities: Gets the list of authorities defined in Manitou.

- Requires passing in a countryId from the Countries query and an authority type (FIRE, POLICE, MEDICAL) from the ManitouAuthorityType enum to get the list for that type.
- SerialNo is used for reference in the other tables.

DealerCustomers: Gets a simple list of customers for a particular dealer from Manitou.

- Requires passing in a dealer serialNo from the Dealers query. Pass in 0 for customers that aren't associated with a dealer.
- serialNo is used for references in other tables.
- contactId is equivalent to the CustomerId in the Customer model.

ContactPointTypes: Gets a list of contact point types for use in the ContactPoint model.

- Defines the type that contact point is in categories of phone, pager, or email. Attribute P indicates a phone type, M indicates a pager type, and E indicates an email type.
- ContactPointTypeId is used for reference in the other tables.

OutputDeviceTypes: Gets a list of output device types defined in Manitou for use with Email and Pager type contact points.

- If using for a dropdown with Email types, filter on attribute = "E". If using with pager, filter on attribute = "M".
- OutputDeviceTypeid is used for reference in the other tables.

ServiceProviderTypes: Gets a list of service provider types defined in Manitou for use with Email and pager contact point types where the output device for that contact point has a particular protocol.

- For the dropdown, only list service provider types if the output device type for that contact point has serviceCodeRequired = true and where the protocol matches the output device type protocol.

ContactPointSubTypes: Gets a list of contact point subtypes defined in Manitou. These define what sort of contact this is (e.g. Keyholder, Janitor, Technician, etc.)

- subTypeid is used for reference in the other tables.

PanelTypes: Gets a list of panel types defined in Manitou for setting up a Manitou customer system.

- panelTypeid is used for reference in the other tables.

## Customer Queries

CustomerById: Gets a full customer stack by customerID. The customer ID is the general readable ID for a Manitou customer and is unique across customers.

- Requires a customerID

CustomerBySerialNo: Gets a full customer stack by serialNo. The customer serialNo is the primary key for the customer in the database and is used by other models when they need a link to the customer.

- Requires a customer serialNo

CustomerPasswords: Gets the list of passwords associated with the customer. Note: This is not for contact list contact's passwords.

- Requires a customer serialNo

CustomerAddress: Gets the address for a customer.

- Requires a customer serialNo

HistoryForCustomer: Gets the event history for a customer.

- Requires a customer serialNo and the number of days of history to fetch

TestForCustomer: Gets all out of service (on test) records associated with a customer. Include a filter on the query on the systemNo field if you need to get tests for a particular customer system.

- Requires a customer serialNo.

CustomerContactPoints: Gets the contact points for the customer. These are not the contact points associated with contact list contacts.

- Requires a customer serialNo.

## Contact List Queries

CustomerContacts: Gets a list of contacts in the customer's contact list with their contact points.

- Requires a customer serialNo.

CustomerContact: Gets a specific contact in the customer's contact list with their contact points.

- Requires a customer serialNo and the sequence number for that contact.

## Customer System Queries

CustomerSystems: Gets a list of customer's systems including area and zone data.

- Requires a customer serialNo.

CustomerSystem: Gets a specific system on a customer including area and zone data.

- Requires a customer serialNo and systemNo.

CustomerSystemAreas: Get a list of areas attached to a Customer's System.

- Requires a customer serialNo and systemNo.

CustomerSystemZones: Gets a list of zones attached to a Customer's System.

- Requires a customer serialNo and systemNo.

## Mutations

### Customer Mutations

PushCustomer: This creates a full Manitou customer and links it to a Managely Site and system.

- In the mutation model, you should include the entityId and entityKey for the Managely site, and in the ManitouCustomerSystemMutation model, include the entityId and entityKey for the Managely systems to link.
- This creates a Manitou customer with address, the customer contact points, customer passwords, contacts for the contact list with their contact points, and customer systems with their area/zones. See the respective model descriptions above for what the fields do.

SyncCustomer: Syncs customer name and address between Managely and Manitou.

- Note: Manitou does not store city and state by name, but rather by an ID in respective tables. Sync will look up the cityId and regionId based on the name, but the names must match the names stored in Manitou (for regions, it can match on either name or shortName) or an exception will be thrown.

EditAddress: Updates a Manitou address without any changes to Managely.

- In the mutation model, the serialNo field should be the customer's serialNo.

CreateCustomerPassword: Adds a customer level password to Manitou. This is not a contact list contact password.

- In the mutation model, the serialNo should be the customer's serialNo and the sequence should be the next available number from the list of passwords from the CustomerPasswords query.

EditCustomerPassword: Updates a customer level password to Manitou. This is not a contact list contact password.

- In the mutation model, the serialNo should be the customer's serialNo, and the sequence field should match the sequence field of the record you wish to edit.

DeleteCustomerPassword: Deletes a customer level password to Manitou. This is not a contact list contact password.

- The mutation model should match with the full CustomerPassword record you wish to delete.

## Customer System Mutations

PushCustomerSystem: This creates a Manitou system and links it to a Managely System.

- In the mutation model, serialNo is the Customer's serialNo, and systemNo must be unique to all customer systems.
- In the mutation model, you should include the entityId and entityKey for the Managely systems to link.
- Areas and Zones can be added in the ListOfAreas and ListOfZones properties as in that section of the PushCustomer mutation.

SyncCustomerSystem: This syncs between Manitou customer system and Managely system. This can go both ways between Manitou and Managely.

CreateSystemArea: Creates a Manitou customer system area.

- In the mutation model, serialNo is the Customer's serialNo and systemNo is the Customer System's systemNo.

EditSystemArea: Updates a Manitou customer system area.

- In the mutation model, serialNo is the Customer's serialNo, systemNo is the Customer System's systemNo, and areald is the area record areald to be updated.

DeleteSystemArea: Deletes a customer system area in Manitou.

- In the mutation model, serialNo, systemNo, and areald define the area to be deleted.

CreateSystemZone: Creates a Manitou customer system zone.

- In the mutation model, serialNo is the Customer's serialNo, systemNo, is the Customer System's systemNo.
- The areald should either be a valid areald for an existing area, or \*.

EditSystemZone: Updates a Manitou customer system zone.

- In the mutation model, serialNo is the Customer's serialNo, systemNo is the Customer System's systemNo, areald is the areald, and zoneld is the zone to be updated. Areald can not be changed once the zone record is created.

DeleteSystemZone: Deletes a customer system zone in Manitou.

- In the mutation model, serialNo, systemNo, areald, and zoneld together define the zone to be deleted.

ActivateTest: Puts a Customer System on test (out of service) for a specified amount of time.

- In the mutation model, serialNo and systemNo are the customer system's serialNo and systemNo. Description is the reason this system is being put on test.
- LogSequence should be 0 (this gets assigned when the out of service record is created).
- From and To are the datetime range are the range of date times the test will be active (in the customer system's local time).

CancelTest: Takes a Customer System off test.

- In the mutation model, serialNo and systemNo are the customer system's serialNo and systemNo. Description is the reason this system is being taken off test.
- logSequence is the log sequence of the out of service record to be taken off test. This can be retrieved from the testsForCustomer query.

## Contact List Mutations

CreateContactListItem: Creates a contact in the Customer's contact list. Contact points can be created at the same

time.

- In the mutation model, serialNo is the Customer's serialNo. Sequence will be automatically assigned.

**EditContactListItem:** Updates a contact in the customer's contact list. Contact points can not be added, edited, or deleted with this mutation. Use the corresponding contact point mutations to handle CRUD operations on an existing contact list contact.

- In the mutation model, serialNo, memberSerialNo, and sequence define the contact list contact you wish to update.

**DeleteContactListItem:** Deletes a contact from the customer's contact list. This will also delete any contact points below this contact.

- In the mutation model, serialNo, memberSerialNo, and sequence define the contact you wish to delete. The name field is required but can be empty.

## Contact Point Mutations

Note: These mutations operate on both customer contact points and contact list contact points.

**CreateContactPoint:** Adds a contact point to either the customer, or to a contact list contact.

- ContactPointTypeId should always be unique against all contact points associated with its containing object. For example, a customer can only have one Home phone contact point, but might also have Business phone or Site phone contact points. The same applies for a contact list contact's contact points.

- ParentSerialNo is always the customer's serialNo

- SerialNo is the customer's serialNo for a customer contact point, or the contact list's contact's serialNo for a contact's contact point.

- ContactType is the ManitouContactType.Customers enum for customer contact types, and ManitouContactType.People enum for the contact list contact point types.

**EditContactPoint:** Updates an existing contact point for either a customer or a contact list contact.

- In the mutation model, the serialNo, parentSerialNo, sequence, and contactType should match the contact point record you wish to update.

**DeleteContactPoint:** Deletes an existing contact point for either a customer or a contact list contact.

- In the mutation model, the serialNo, parentSerialNo, sequence, and contactType define the contact point you wish to delete.