

# How To Add a Lead Automatically To The CRM

Last Modified on 01/29/2019 9:21 pm CST

## HIGH-LEVEL OVERVIEW

### MILESTONE 1 - ADD A LEAD TO THE CRM

- **STEP 1** - Lookup the Contact to See if exist
- **STEP 2** - If contact exists then UPDATE it if not then ADD the contact.  
*\* Don't forget to also include the vehicle inventory when you add the lead!*

### MILESTONE 2 - ASSIGN A SALES PERSON

- **STEP 3** - Lookup Team Member Full Name
- **STEP 4** - Title Case The Team Member's Full Name
- **STEP 5** - Understanding Sources vs Team Members
- **STEP 6** - Confirm The Team Member Has a Valid Source
- **STEP 7** - Your Ready To Add Sales Person Now

### MILESTONE 3 - TRIGGER A FOLLOW-UP

- **STEP 8** - Lookup the list of actions and ask the user what type of follow-up they want to do via your app. By default the first action is called "LEAD", but if you want to play it safe just list them all in a drop down.
- **STEP 9** - Fire off an "Actions" as the "Team Member" so any required follow-up that is configured by the dealership will automatically happen within the contact record after the lead is added. Remember an "Actionset" is a predefined set of rules that store configures as to how they wish to follow up with customers. If you do not trigger an actionset then automatic follow-up rules will not apply and the contact will just be added with no follow-up.

For example, an action set may say.. call within 24 hours, then email after 48 hours. The

dealership creates the "Actions" and each one has a unique ID, you can ask them "which action do you want us to fire off" and store that ID or you can just "get all the actions ID " via the API and display list that the dealer can choose from when the log the lead.

---

## **BEFORE WE BEGIN - PREREQUISITE**

### **1st - YOU'LL NEED THE CONTACTID FROM YOUR UPDATE**

\* At this point, we'll assume you have done step 1 and 2 and you now have a '**CONTACTID**'. If you'll get this ID after you add it or update it, if you need help [adding the contact see here](#) , if you need help [updating the contact you can see here](#) .

### **DEVELOPER TIP**

- **What is the Sales PersonTeam Member ID? THIS VALUE IS "BIGINT"**  
This is the unique ID of the Sales Person who uses the CRM, these CRM users are called Team Members and each have their own unique ID.
- **What is the Sales Person Display\_Full\_Name? THIS VALUE IS "TEXT STRING"**  
This is part of the Team Member Object and is the full name of the sales person/user. This will also match the "Source" as the source an Alias of the Team Member (aka Sales Person)
- **What is the Sales PersonSource? THIS VALUE IS "TEXT STRING"**  
This is alias for the Team Members (aka Sales Person) "Full Name", for example "John Doe"
- **What is the Sales PersonActionSet ID? THIS VALUE IS "BIGINT"**  
These are predefined follow-up rules that are setup by the dealership. For example when a lead is added an action will be fired off. That action may be something like "call them customer tommrow", "next week" and "send this email". **If you DO NOT assign an action set ID after you add a contact record then no follow-up will be scheduled.**

---

## **LET'S BEGIN TO CODE..**

### **MILESTONE 1 - HOW TO ADD A LEAD TO THE**

# CRM

## **BEFORE YOU BEGIN - YOU'LL NEED THE STORE ID AND API KEY**

At this point, you'll need to get the store ID and API key from the dealership. Dealers can request this information via filling out API [request form here](#). On the form above, enter the first email of [support@morethanrewards.com](mailto:support@morethanrewards.com), then the dealer email, then the follow-up email of [support@morethanrewards.com](mailto:support@morethanrewards.com) and submit the form. [SignNow.com](#) will automatically send out the request.

## **STEP 1 & 2 - LOOKUP THE "CONTACT ID"**

At this point, you should have the "ContactID" of the record you want to assign the sales person to. We'll assume you did step 1 and 2 above at this point.

### **DEV TIP - DO NOT FORGET TO ADD THE INVENTORY INFORMATION FOR THE LEAD!**

Remember our CRM stores things like Year, Make, Model, VIN, Trade-In and more! This is all stored in the custom fields 1 to 20. Instead of just adding their basic contact info why not also include the bike information that they customer is interested in.

#### **Custom Field Mapping**

Custom2 = Deal Value, Custom3 = Deal Inventory Date, Custom4 = TradeValue, Custom5 = TradeVIN, Custom6 = TradeMake, Custom7 = TradeModel, Custom8 = TradeYear, Custom9 = TradeColor, Custom10 = TradeOdemeter, Custom11 = StockNumber, Custom12 = VIN, Custom13 = Make, Custom14 = Model, Custom15 = Year, Custom16 = Color, Custom17=Odemeter, Custom18 = New/Used, Custom19 = LeadCreateDate

## **API REFERENCE**

LOOKUP CONTACT - <http://developers.morethanrewards.com/docs/getcontacts>

UPDATE CONTACT - <http://developers.morethanrewards.com/docs/updatecontact>

ADD CONTACT - <http://developers.morethanrewards.com/docs/addcontact>

## **MILESTONE 2 - ASSIGN A SALES PERSON TO A LEAD**

### **BEFORE YOU BEGIN - YOU'LL NEED THE CONTACTID FROM THE UPDATE ABOVE**

At this point, we'll assume you have done step 1 and 2 and you now have a '**CONTACTID**'. If you'll get this ID after you add it or update it, if you need help [adding the contact see here](#), if

you need help updating the contact you can see here .

---

## STEP 3 - LOOKUP THE "TEAM MEMBER"

Depending on how you store the current logged in user of your application you'll need to figure out who the corresponding "sales person" is in our CRM. You can do this by looking them up by an email, name, phone or just have a drop down selector. The easiest way is to just get a list of all sales people and display it in your GUI. Let's assume tho that you have a person "logged in" to your application and you have access to their email. Maybe that's what your using to log them in your app with.

### FIRST - GET A LIST OF ALL THE "SALES PEOPLE"

We'll assume you have read the prerequisite and have the StoreID and APIKey, you'll use these two parms to do GET to our API and which will return back an JSON object of all the sales people here.

#### API REFERENCE

<http://developers.morethanrewards.com/docs/getteammembers>

### SECOND - FIND THE ACTUAL LOGGED IN USER BY THEIR EMAIL

We'll assume that you have the "logged in" sales persons email from your app, if you don't have this you can just display a drop down of sales people but to make it easy we'll assume you have the authenticated users email on your app. You'll need to take the JSON object from the call **getteammembers()** and loop through to find the record that matches email of your authenticated user.

---

## STEP 4 - FORMAT THE USING "TITLE CASE"

You'll want to take the field "display\_fullname" from the getteammembers() API call, Trim() it and TitleCase() it. So you'll remove any leading and trailing spaces, then also capitalize the first letter of the first name, and the first letter of the second name. Then lower case the others.

- For example... **JOHN SMITH** will be come **John Smith**
- For example... **JOHN smith** will be come **John Smith**

This is very important, if you do not trim and title case the name you risk creating duplicate names when you assign a sales person to a contact. So Please make sure you format your string correctly! The reason why we have to do this is because we need to confirm that this record is in listed in the "Sources" drop down before we try to assign it to the contact record. If

it's not in the drop down of sources then the sales person will NOT be assigned.

---

## STEP 5 - UNDERSTANDING SOURCES VS TEAM MEMBERS

Before we can assign the sales person we need to know lookup their corresponding "Source Name" as well as their normal "Team Member ID". So we'll need both to successfully assign a Team Member (aka Sales Person).

### WHAT ARE "SOURCES" vs "TEAM MEMBERS":

Think of "Sources" as an alias to the Team Member (aka Sales Person), the reason this is important is because you can share "logins" with our CRM to save money if you're not willing to purchase a user for each sales person you can simply purchase 1 Team Member (aka Sales Person) and then just add unlimited "Sources" (aka Sales People) on a contact. For example you may have a Team Member that the Dealership Owner and the GM log in with called "admin" but then each of them are listed as a "Source" in the CRM as ... "Jim Brown" and "John Doe", while the Team Member login is called "Admin". We do this to allow the customer to share logins and save money if they are not willing to purchase a "Team Member" for each sales person.

---

## STEP 6 - CONFIRM THEIR IS A VALID SOURCE

Every single Team Member's "**display\_fullname**" that was pulled back from the **getteammember()** API call should be also listed as a "Source" within our CRM by default. However, customers can rename and delete sources, this is common is if they want to give someone a nick name or get married and have a name change. So we recommend BEFORE going to step 6 and actually assigning the sales person you first make sure the "Source" is listed on the contact. **If you don't and you just assign the sales person without doing this software check then the sales person may not be assigned since their won't be a source. Remember sources need to be EXACT and are case sensitive.**

### FIRST - CHECK THE SOURCE

Simply fetch all the sources just like you did with the Team Members and loop through and confirm that the "**display\_fullname**" that you found in the **getteammmember()** API from Step 3 is found in the source list, just fetch all the sources, easy as 123.

### API REFERENCE

<http://developers.morethanrewards.com/docs/getsources>

### SECOND - IF THE SOURCE IS MISSING ADD IT

Most of the time you'll find the source and you can move to the next step, however if you don't

find the source in the JSON object from the `getsources()` API call then that means the dealer must have deleted the sales person alias (aka Source). So all you have to do is add it.

## API REFERENCE

<http://developers.morethanrewards.com/docs/addsource>

**WARNING - If you do not do this step and just blindly assign a sales person in the next step and there is no source then it will NOT assign a sales person via the API because a source basically an Alias to your Team Member which is your Sales Person.**

---

## STEP 7 - YOU'RE READY TO ADD THE SALES PERSON NOW

### WE'RE READY - LET'S DO A FINAL SPOT CHECK

- \* You should have an int called `contactid` of the contact that was captured from Step 1 and Step 2
- \* You should have an int called `teammemberid` of the Sales Person that is accessing your app from Step 3, Step 4
- \* You should also have a string called `display_fullname` which is the full name of the sales person from Step 4 AND it also matches the EXACT same name of the Source that in Step 5 and Step 6 and if one was not found you added it so we 100% for sure have a source that matches this display name of the team member.

So now we're going to do an **UPDATE** on the contact record with the Team Member (aka Sales Person) information. This makes it so that the TeamMember (aka the Sales Person / aka Deal Owner) appears on the contact record as the person who is responsible for the deal.

**WARNING - If the Source is not EXACTLY as it appears in the drop down (case sensitive) then it will be ignored, which is why we recommended you add a source if one is not found.**

### REQUIRED FIELDS YOU'LL NEED TO UPDATE NOW

`accountid = "YOU'LL GET THIS FROM THE DEALER"`

`apikey = "YOU'LL GET THIS FROM THE DEALER"`

`response_type = "JSON"`

`identifymethod = "1"`

`identifyvalue = "ContactID" (from above from STEP 1 and 2)`

```
source = "Source" (from above from STEP 6)
```

```
othersystemid = "TeamMemberID" (from above from Step 3)
```

## API REFERENCE

<http://developers.morethanrewards.com/docs/updatecontact>

# MILESTONE 3 - TRIGGER THE FOLLOW-UPS

## BEFORE YOU BEGIN - YOU'LL NEED THE CONTACTID FROM THE UPDATE ABOVE

At this point, we'll assume you have done all the steps above and are ready trigger the follow-up actions.

## STEP 8 - LOOKUP THE "ACTIONSETID" OF "SALES PERSON"

We recommend you just fetch a list of all the actionsets and display them in the UI so the customer can choose what kind of follow-up they want. In many cases the default follow-up is called "LEAD", so if it's easier for you just fetch all the actions and loop through looking for the one called "LEAD" and use that as the default action. Since that's the first action in the set typically. For example when leads come in via other sources this is the action that is triggered.

## API REFERENCE

<http://developers.morethanrewards.com/docs/getactionsets>

=====

**QUICK CHECK - At this point you'll have the "ContactID", "ActionSetID", "TeamMemberID" and "Source"**

=====

## STEP 9 - FIRE OFF AN "ACTIONSET" AS THE TEAM MEMBER

While this step is technically optional it's very important and should be done. What this does is trigger an Actionset which causes a series of predefined follow-ups to happen with the contact. If you do not fire off the Action then no follow-up rules will be generated for this contact. Every store can define their own follow-up method. By default most locations have something called "Internet Leads". We recommend having some kind of a config value that is set in your preferences which stores the ID of the "Actionset" since every store has a different ActionsetID that is unique to them. So you'll either need to get a list of all the Actionset then

ask them which one they want to fire off or make it part of your config setup. We recommend just making it part of the config setup. This way you can ask the store "what's your storeID, what's your API key, what's your ActionsetID you want us to trigger when the contact is added?"

### **API REFERENCE**

FIRE ACTIONSET - <http://developers.morethanrewards.com/docs/runactionset>

\* You'll pass over the contactID and TeamMemberID to this API to trigger it.

---

**IF YOU NEED DEVELOPMENT HELP PLEASE GIVE US A CALL AT 414-326-4100**

**DISCLAIMER - WE ONLY PROVIDE DEV SUPPORT FOR SELECT PARTNERS**