

How to make a QuickBase buttons (or URL formulas for Dummies)

OK, so you figure you are halfway decent at the basics of QuickBase and are now ready to give yourself or your users a better experience by adding some buttons to forms or reports to automate actions. For example, you might want to have a user click a button to "Approve" a record or maybe even click a button right from an email to Approve, without going through the steps of edit, scroll down to locate the Approve field, and save.

It's really not difficult at all! Let try a few easy dives and then we will work our way up to a triple half gainer in no time. There are three reasons you should have confidence that you can do these.

- The biggest reason is that virtually everything that QuickBase does is driven and is visible in the "URL" at the top of your Browser screen. So that means that all we need to do is to make a URL formula field to mimic what the URL would be showing for the usual user interface actions.
- There is a ["API" guide in the Help](#) which is a cheat sheet to give you hints when you are stuck.
- If all else fails you can post a question to the QuickBase Community forum when many of us hang out and are willing to help.

Let's start easy. Let's say we just want to redisplay the same record on a different form.

```
URLRoot() & "db/" & Dbid() & "?a=dr&rid=" & ToText([Record ID#]) & "&dfid=10"
```

But let's break that down into pieces.

urlroot()	this says to go to https://mycompany.quickbase.com
& "db/"	this adds on the "db/" you see in all URLs
& dbid()	this adds on the Table ID you are currently on
& "?a=dr&rid="	this says "Action = Display Record and the Record ID is"
& [record ID#]	this gives the record ID to display
& "&dfid=10"	this says to use an alternate form ID# of form # 10.

Ok, that was easy. Let's now do the more common situation. Let's say that you want to push a button, edits a field on the record and then redisplay the record. I am also going to assume for this example that you have gone into your Application Properties and unchecked "Require [Application Tokens](#)"

```
var text URLONE = URLRoot() & "db/" & Dbid()  
& "?act=API_EditRecord&rid=" & [Record ID#]  
& "&_fid_15=1";
```

```
var text URLTWO = URLRoot() & "db/" & Dbid() & "?a=dr&rid=" & [Record ID#];
```

```
$URLONE  
& "&rdr=" & URLEncode($URLTWO)
```

Let's have a look at what is happening here. The first thing you will notice is the use of [formula variables](#). This is highly recommended as it keeps the formula elements short, easier to read and hence easier to debug for syntax errors.

I have chosen to call the first formula variable URLONE (We are not allowed to use numbers or special characters in their names) and after the usual identification of your company ID and the Table ID, it says to "edit" the record we are on and set the Field with ID number 6 to be a 1. We must always end a formula variable definition with a semicolon.

When you look at your list of fields and enable the option to show the fields ID's (or look at the Usage tab for a field) it will show you the field ID#. In my case this is a checkbox field and I am setting it to be a "1" which means true (a "0" would be to uncheck it or make it false).

The next formula variable called URLTWO simply says to display the record I am on. Whenever we Add or Edit records, the last action we need to do is to display a record or display a report. If we don't, then QuickBase will fill the screen with a message attesting to the success of your Edit or Add in a format which is most unfriendly to your users.

The last step is to string the two actions together. For technical reasons which we don't need to understand but do need to follow, when we string successive actions together we need to successively URLEncode them.

For example in the above example we were doing two things, an Edit and then Displaying a record. If we were stringing four actions together the format would be

```
$URLONE  
& "&rdr=" & URLEncode($URLTWO)  
& URLEncode("&rdr=" & URLEncode($URLTHREE))  
& URLEncode(URLEncode("&rdr=" & URLEncode($URLFOUR)))
```

So, in fact now you have quite a lot of power that you can build into a single button click.

Let do one final example which does three actions. Let's say that we are sitting on a child record which is a Line Item on a Parent Purchase Order.

1. The PO Line item has a field we want to set to have the value "PO Line Approved", and also we want to capture the name of who Approved.
2. Then let's say that we want to trigger an email notification off the PO Header (Parent record) by jiggling a toggle field whose job it is to trigger an Email Notification.
3. Lastly, let's we want to display the Parent PO record and leave the user there as opposed to the child record.

```
// edit the Child record
var text URLONE = URLRoot() & "db/" & Dbid()
& "?act=API_EditRecord&rid=" & [Record ID#]
& "&_fid_15=" & URLEncode("PO Line Approved")
& "&_fid_16=" & URLEncode(UserToName(User()))
& "&apptoken=d7emakdb32ctedqz22hdb29du2q";

// determine the new value of the toggle field which will be used to trigger an email
Notification.
var number NewToggleValue = If ([Purchase Order Toggle to trigger email
Notification]=true,0,1);

// edits the Parent Purchase Order record
var text URLTWO = URLRoot() & "db/" & [_DBID_PURCHASE_ORDERS]
& "?act=API_EditRecord&rid=" & [Related Purchase Order]
& "&_fid_14=" & $NewToggleValue
& "&apptoken=ch67fndkskdf83ch5bdihra8z";

// display the Parent PO Record
var text URLTHREE = URLRoot() & "db/" & [_DBID_PURCHASE_ORDERS]
& "?a=dr&rid=" & ToText([Related Purchase Order]);

// now string them all together and execute them
$URLONE
& "&rdr=" & URLEncode($URLTWO)
& URLEncode("&rdr=" & URLEncode($URLTHREE))
```

In the first formula variable URLONE you can see that we are editing the record to update fields with ID # 13 and 14. You will also see that use of that URLEncode

again. That is necessary whenever a text value may have spaces or special characters.

The next formula variable called NewToggleValue is setting up to set a toggle fields to the opposite of what it currently is. I'm assuming here that we want to "jiggle" the parent record to trigger an email Notification.

The URLTWO is editing a record which is in a different table. In this case rather than referring to the current table with dbid() we need to specify the name in square brackets. The table name is available in the Advanced Properties of each table's settings.

URLTHREE is simply going to display the Parent record.

Lastly we string them all together in that nested URLEncode format.

Hopefully this little tutorial will give you the confidence to try a few URL formula "buttons" yourself for simple actions at the click of a button.

There is an app Shared in the QuickBase Exchange called "URL Formula Buttons for Dummies (v1)" which has these URL formula buttons working. Just go to the Exchange and search on "dummies" and it should come up.

Mark Shnier is the principal at the one man consultancy [Your QuickBase Coach](#) which specializes in creating user friendly apps by leveraging native QuickBase, URL formula buttons, and integration with third add on tools for QuickBase. He can generally found hanging out at the [QuickBase Community Forum](#).