ZOHO CREATOR

DELUGE IN DETAIL

Adding Advanced Functionality

Workflow Triggers

Form Actions

1. On Create
    a. On Load
        i. The **On Load** form action script is used to dynamically alter a Form when it is loaded. The Form actions "On Load" are executed when the Form is loaded by a user to add a new record.
    b. On Validate
        i. In addition to the default validation, Zoho creator allows you to write action scripts for custom validation. **This script is executed when a user submits the form to the server, before the user data is stored in the database.**
        ii. The default action of the validate script in the "on add" or "on edit" block, is to submit the data, so you have to do a **cancel submit** if you want to stop a form from being submitted.
        iii. You can also choose to provide proper **alert message** to the user while cancelling. The same form is then shown to the user to re-enter the data.
    c. On Success
        i. On Success script in Deluge is used to execute script actions when a new record is successfully submitted to the database

The "**On Success**" script associated with a form will get triggered only in the following cases:

- When data is added from the browser or Zoho Creator App
- When data is added through API
- When data is added through Import
- When data is added through email data feature.

So, if the "InsertRecord task" is invoked from onClick action of stateless form, the "onSuccess" action associated with the form will not be executed.

2. On Edit
    a. On Load
    b. On Validate
    c. On Success

3. On Delete
    a. On Validate
        i. The *validate form action* script in the **Form Actions -> On Delete** block is executed when an existing record is submitted for deletion.
    b. On Success
        i. The on success action script added to the **Form Actions -> On Delete** block is invoked when an existing record is deleted from the database. The record is removed from the database when this script is run.

Field actions

4. Field Actions

a. On User Input

i. The *On User input* is a client side action which will be called whenever the value of a field is modified **either by the user or through script**. It is used to improve the usability of a form by validating field data even before it is submitted or display other field values based on the value specified in this field. The On user input script is executed before the changed data is persisted in the database.

b. On Update

i. If you want to perform action whenever the value of a field is modified, you can write an **on update** script for that field. The on update script is executed after the changed data is persisted in the database.

Sub Form Actions

5. Sub form Actions
    a. On Add row
        a. The *On Add Row Script* will be called whenever a new row is addded to the Subform.
    b. On delete row
        a. The *On Delete Row Script* will be called whenever a row is Deleted from the Subform.
    c. User input of subform Field
        a. The *On User Input Script* will be called whenever the value is modified in any field in the Subform.

Functions

A function is a set of statements grouped together under a name and can be invoked from anywhere within a program.

Advantages of using Deluge Functions

- **Write once and reuse utilities**
- **Custom action integration in views**
- **Serves as bridge between applications (created by the same user)**

Control Statements

Conditional Execution - If, else if, else

The 'If' construct in deluge is the same as that of other languages. It conditionally executes a set of statements depending on the value of the boolean expression.

Conditional IF statements

If *boolean-expression is TRUE,* returns *expression1*; otherwise it returns *expression2* .

Null check using conditional IF statements

For NULL check, the simplified version of conditional IF is given below. If *expression1* is NOT null, IFNULL() returns *expression1* ; otherwise it returns *expression2* .

Data Access

1. ## Add Record

   The *add record* Deluge task is used in Form action and Field action scripts to insert a new record to a given Form with values specified in the expression.

2. ## Delete Record

   The *delete records* Deluge task is used in Form action and Field action scripts to delete existing records from a form, based on the specified criteria.

3. ## For each record

   Deluge supports only data driven iterators that are purely based on Form data. You can iterate through all the rows in the Form or through selected rows based on the given criteria.

4. ## Fetch Records

   Forms are structures that contain data in Zoho Creator. The form data is stored in a relational database and Deluge provides an easy wrapper called **collection variable** for accessing these data. The collection variables hold one or more records fetched from a form, based on a given criteria, sort by and range.

5. Aggregate Records

The Aggregate Records task supports for calculating SUM/COUNT/AVG/MIN/MAX of a Form's field values. This will help creating Applications without iterating the records.

- **Count**: Returns the count of all or specific records in a form.
- **Sum**: Returns the sum of a numeric field for all or selected records in the form
- **Average**: Returns the average of a numeric field for all or selected records in the form
- **Maximum**: Returns the maximum value of a numeric /date-time field, from all or selected records in the form
- **Minimum**: Returns the minimum value of a numeric /date-time field, from all or selected records in the form
- **Distinct**: Returns the distinct values of a field in a form.

Client Side Functions

Hide/Show

Enable/Disable

Select/Deselect

Add to picklist dynamically

Clear Picklist

Alert Box

Reload Form

List Manipulation

1. Create List

The **Create List** Deluge syntax enables the creation of new list.

{} - Holds list values of any type. Handy syntax for creating list with some initial values.
List() - Holds list values of any type.
List:String() - Holds list values of type String.
List:Int() - Holds list values of type Integer.
List:Date() - Holds list values of type Date.
List:Bool() - Holds list values of type Boolean - true/false.
List:Float() - Holds list values of type Float.

2. Add

The *Add list* Deluge statement appends the specified element to the end of this list. The type of elements that can be added, depends on the list type.

3. Remove index

The *Remove Index* list syntax removes the element at the specified position from this list.

```
<list>.remove(<val>);
```

4. Remove element

The **removeElement()** list syntax removes the first occurrence of the specified element, from a given list. This function will be just ignored if the specified element is not present in the list.

```
sports_list.removeElement("Hockey");
```

5. Add all

The **Add all** list syntax appends all of the elements in the specified source list to the end of the target list, in the order, the elements are present in the source list.

```
<list1>.addAll(<list2>);
```

6. Remove all

The **Remove All** Deluge list syntax removes from a given list all the elements that are contained in another specified list.

```
<list1>.removeAll(<list2>);
```

7. Clear

The *Clear* Deluge list syntax, clears all the elements from the specified list and makes it empty.

```
<list>.clear();
```

8. Sort

The **Sort** Deluge list syntax sorts the elements in a list in the ascending or descending order, based on the returned boolean value. If the boolean expression returns **true** the list values will be sorted in the **ascending** order. If the boolean expression returns **false**, the list values will be sorted in the **descending** order.

9. For each element

The *For each element* Deluge syntax executes a set of tasks repeatedly, for each element in the list.

```
for each <elementname> in <listname>
{

}
```

Map Manipulation

1. Create Map

The **create map** Deluge task creates a new empty map expression.

2. Put Key

The *put key* Deluge task inserts a key - value pair into the map variable.

```
<map1>.put(<key>,<value>);
```

3. Put all key

Inserts all key - value pairs into the specified map variable, if key(s) already present in the map-variable the key's associated value is replaced with new value.

Remove key

This Deluge task removes the specified key and associated value from the target map variable.

```
<map-variable>.remove(<key-name>);
```

Clear Map

The clear map Deluge task removes all key-value mappings from the map and makes it empty.

```
<map>.clear();
```

Thank you