

Bedingungen

Under construction ...

Wählen Sie die Verfahrensweise anhand der Bedingung

Mit Ultradox können Sie erweiterte Workflows erstellen, um verschiedene Aktionen auszuführen, die auf die von Ihnen festgelegten Bedingungen basieren.

Dabei bietet Ultradox verschiedene Möglichkeiten Bedingungen zu nutzen und gleichzeitig Ihre Automation einfach zu halten.

Um Ihnen beim Erstellen von Bedingungen zu helfen, wird Ultradox mit dem **Condition Builder** geliefert, mit dem Sie leicht Bedingungen erstellen können.

Einfache Bedingungen

Wenn Sie auf das **if** im on/off Schalter, rechts oben im Baustein klicken, öffnet sich ein Dialog, in dem Sie die Bedingung mit dem Condition Builder erstellen können. Alle verfügbaren Variablen finden Sie in der Dropdown-Liste und können Ihre Bedingung mit den entsprechenden Operatoren aus dieser Liste zusammenstellen. Im Eingabefeld können Sie - wie in allen blauen Eingabefeldern - feste Werte oder Variablen eingeben.

Ausführen wenn...

Einfach Fortgeschritten

email ist ein Text enthält @gmail.com und x

name ist ein Text beginnt mit \${givenName} + x

+

Übernehmen Abbrechen

Mit dem **+** Zeichen **auf der rechten Seite** lassen sich weitere Teilbedingung hinzufügen, so dass die vollständige Bedingung nur dann wahr ist, wenn der erste **und** der zweite Teil erfüllt sind.

Wenn Sie auf das **+** **unter** der Bedingung klicken, können Sie eine weitere Teilbedingung hinzufügen, so dass die vollständige Bedingung wahr ist, wenn der erste **oder** der zweite Teil zutrifft.

JavaScript conditions

If you need even more sophisticated conditions, you can switch to the **Advanced** tab and enter the conditions as JavaScript code.

If you have created a simple condition using the condition builder, you will find a JavaScript version of the condition in the **Advanced** tab that you can use as a starting point.

As indicated by the green background of the input field you can enter JavaScript code that may contain variables.

If you for example want to execute the block only if the variable `plan` has the value `Gold` and the variable `age` is greater than 21 enter the following condition:

```
'\${plan}' == 'Gold' && \${age} > 21
```

Ultradox will first replace all variables with their values, so if your variables contain the values `Silver` and 30 your condition will look like this after evaluating the variables.

```
'Silver' == 'Gold' && 30 > 21
```

Ultradox will then evaluate this expression as JavaScript code

Playground

To learn how Ultradox handles advanced conditions, you can use the interactive playground. Adjust the condition in the `Template` tab or the data in the `Data` tab and see how the merged JavaScript looks like.

The `Result` tab contains the evaluated JavaScript code and should contain just `true` or `false`.

```
tab:Template[interactive-ultradox]
'\${plan}' == 'Gold' && \${age} > 21
tab:Data[interactive-json]
{
  "plan" : "Gold",
  "age" : 50
}
tab:JavaScript[transformed-javascript]
tab:Result[evaluated-json]
```

Conditional execution

Each block comes with an **on / if / off** switch. If you just want to execute a single building block based on a condition, this is the easiest way to go.

If you are creating a form with multiple pages (=multiple subsequent form blocks) you can show **forms only if a certain option has been selected**.

You could also set a condition on a stop block to end the flow if a condition is met. This can make sense if you have a **newsletter campaign with timed follow up emails** and want to stop the flow **if the user unsubscribed** in the meantime.

Skip blocks

If you have a number of steps that shall be skipped under certain conditions, the Skip block is the right choice. You'll not have to specify the same condition on each individual building block, but you can simply skip a number of steps and specify the condition right on the skip block.

If the specified condition is met, the nested blocks will be skipped and are **not** getting **executed**. So you will have to specify the opposite condition on the skip block as if you would use the **if** button on the nested blocks to get the same result!

Überspringe Bausteine
fs-control icon-forward

Ende der übersprungenen Blöcke
fs-control icon-skipEnd

If / Else blocks

If you want to execute a different sequence of steps based on conditions, you can use the **If / Else** blocks. They are the most flexible solution and will make your flow more readable if things are getting more complex.

Die folgenden Bausteine ausführen falls **Bedingung erfüllt ist**
fs-control icon-if

Anderenfalls die folgenden Bausteine ausführen falls **keine andere Bedingung zutrifft**
fs-control icon-else

Ende der bedingten Ausführung
fs-control icon-ifEnd