Validation in Power apps

When building apps, you want to make sure your data is stored correctly into your Data Source. Especially when you're storing specific data types (such as email address, telephone number, etc.
The Validate function checks whether the value of a single column or a complete record is valid for a data source.

Description

Before a user submits a data change, you can provide immediate feedback on the validity of that submission, resulting in a better user experience.

Data sources can provide information on what constitutes valid values within a record. This information can include many constraints, such as these examples:

- whether a column requires a value
- how long a string of text can be
- how high and low a number can be
- how early and late a date can be

The Validate function uses this information to determine whether a value is valid and to return an appropriate error message if not. You can use the DataSourceInfo function to view the same information that Validate uses.

Data sources vary in how much validation information they provide, including not providing any at all. Validate can only verify values based on this information. Even if Validate doesn't find a problem, applying the data change may still fail. You can use the Errors function to obtain information about the failure.

If Validate finds a problem, the function returns an error message that you can show to the user of the app. If all values are valid, Validate returns blank. When you work with a collection that has no validation information, values are always valid.

Syntax

Validate(*DataSource*, *Column*, *Value*)

- *DataSource* – Required. The data source to validate with.
- *Column* – Required. The column to validate.
- *Value* – Required. The value for the selected column to be validated.

Validate(*DataSource*, *OriginalRecord*, *Updates*)

- *DataSource* – Required. The data source to validate with.
- *OriginalRecord* - Required. The record to which updates are to be validated.
- *Updates* - Required. The changes to apply to the original record.

Examples

For these examples, values in the Percentage column of the Scores data source must be between 0 and 100, inclusive. If the data passes validation, the function returns *blank*. Otherwise, the function returns an error message.

Validate with a single column

Formula	Description	Result
Validate(Scores, Percentage, 10)	Checks whether 10 is a valid value for the Percentage column in the Scores data source.	*blank*
Validate(Scores, Percentage, 120)	Checks whether 120 is a valid value for the Percentage column in the Scores data source.	"Values must be between 0 and 100."

Validate with a complete record

Formula	Description	Result
Validate(Scores, EditRecord, Gallery.Updates)	Checks whether values in all columns are valid for the Scores data source. In this example, the value in the Percentage column is 10.	*blank*

Validate(Scores, EditRecord, Gallery.Updates)	Checks whether values in all columns are valid for the Scores data source. In this example, the value in the Percentage column is 120.	"Values must be between 0 and 100."