

Spreadsheet

Definition

Spreadsheet Form Element allow users to perform many actions in an excel-like environment, depending on the use cases. A few examples of uses of spreadsheets are:

- Database editing
- Configuration controlling
- Data merging
- Workforce planning
- Sales reporting
- Financial analysis

New Feature

This is a new feature in Joget Workflow v6.

Expense Items

	Date	Category	Purpose	Amount
1	2017-03-06	Travel Expenses	Bus rental	3000.00
2	2017-03-06	Entertainment Claims	Catering	12000.00
3				

Tip: Right click mouse on grid area to Insert Row or Delete Row.

Total Amount \$ 15000.00

Figure 1 : Sample Spreadsheet Form Element in the Userview

Edit Spreadsheet

Edit Spreadsheet > UI > Validation & Data Binder > Load Binder (Multirow Form Binder) > Store Binder (Multirow Form Binder)

Spreadsheet *

Label

Columns

spECD

Value *	Label	Format Type	Format	Regex Validation	Formula	Readonly
date	date	Date	YYYY-MM-DD			☐
category	category	Dropdown	ExpensesClaimE			☐
purpose	purpose	Text				☐
amount	amount	Numeric	0.00			☐

Figure 2 : Spreadsheet Form Element Properties

Name	Description
Spreadsheet	Element ID of the form element.

Label	Spreadsheet label/title.						
Columns	The spreadsheet column(s) is defined here. <table> <tr> <th>Sub-element</th><th>Description</th></tr> <tr> <td>Value</td><td>Field ID of the column.</td></tr> <tr> <td>Label</td><td> Column header title/label. You can also include a tooltip that shows up when someone hovers over the column label.  Sample Example label: <pre>Column Label A</pre> </td></tr> </table>	Sub-element	Description	Value	Field ID of the column.	Label	Column header title/label. You can also include a tooltip that shows up when someone hovers over the column label.  Sample Example label: <pre>Column Label A</pre>
Sub-element	Description						
Value	Field ID of the column.						
Label	Column header title/label. You can also include a tooltip that shows up when someone hovers over the column label.  Sample Example label: <pre>Column Label A</pre>						
Format Type	Defines the type of input to store. Default format type is Text. Available Data Types: • Text - format the value as text • Numeric - accepts a numeric value (See Format documentation) • Date - accepts a date type value • Time - accepts a time type value • Checkbox - appears as a checkbox to check • Dropdown - appears as a dropdown menu to select options • Autocomplete - autocompletes any known phrases • Password - obscures any alphanumerical value into a password • File - Accept file upload and render a file download link • Image - Accept image file upload and render a thumbnail • URL - Convert value to an URL • Hidden - Hidden column for storing formular value • Custom - Use custom setting to configure the column. Examples: Spreadsheet Custom Formats						
Format	Defines the format to show data on the spreadsheet based on the chosen format type. • Numeric - Key in format accepted by numbrojs library. (For example, 0.00 is to create 2 decimal places) • Date - Key in format accepted by moment.js library. (Default value: DD/MM/YYYY) • Time - Key in format accepted by moment.js library.(Default value: h:mm:ss a) • Dropdown - Key in the Form ID that contains the equivalent dropdown element.  Dynamic Cascading Select Box Dynamic Cascading Select Box is supported when the target select box is making use of Options Binder and with the attribute "Use AJAX for cascade options?" is checked. • Autocomplete - See handsontable for samples. • File - Key in Form Def ID that contains File Upload form element. • Image - Key in Form Def ID that contains Image Upload form element. • URL - Provide an URL syntax. Example: <code>http://www.joget.org?id={id}&name={name}</code> • Hidden - Key in default value to be saved. • Custom - See handsontable for samples. Please put the setting json in {{ and }}. Example: <code>{{"editor";"text"}}</code> Please note that any string within {{ and }} will be treat as setting json and used to configure the column.						
Regex Validation	Validates input value to match the defined Regular Expression pattern.						

Formula	Defines a formula to perform processing. Sample Example: sales + profit-loss Edit Spreadsheet Edit Spreadsheet > UI > Validation & Data Binder Spreadsheet * salesReport Label Simple Sales Report Columns <table border="1"> <thead> <tr> <th>Value *</th> <th>Label</th> <th>Format Type</th> <th>Format</th> <th>Regex Validation</th> <th>Formula</th> <th>Readonly</th> </tr> </thead> <tbody> <tr> <td>sales</td> <td>Sales</td> <td>Text</td> <td></td> <td></td> <td></td> <td>☐</td> </tr> <tr> <td>profit</td> <td>Profit</td> <td>Text</td> <td></td> <td></td> <td></td> <td>☐</td> </tr> <tr> <td>loss</td> <td>Loss</td> <td>Text</td> <td></td> <td></td> <td></td> <td>☐</td> </tr> <tr> <td>total</td> <td>Total</td> <td>Text</td> <td></td> <td></td> <td>sales+profit-lo</td> <td>☒</td> </tr> </tbody> </table>	Value *	Label	Format Type	Format	Regex Validation	Formula	Readonly	sales	Sales	Text				☐	profit	Profit	Text				☐	loss	Loss	Text				☐	total	Total	Text			sales+profit-lo	☒
Value *	Label	Format Type	Format	Regex Validation	Formula	Readonly																														
sales	Sales	Text				☐																														
profit	Profit	Text				☐																														
loss	Loss	Text				☐																														
total	Total	Text			sales+profit-lo	☒																														
	Please refer to formula .																																			
Readonly	Defines if the column is editable.																																			

UI
 Edit Spreadsheet > **UI** > Validation & Data Binder > Load Binder (Multirow Form Binder) > Store Binder (Multirow Form Binder)

Sorting
 Enable Header Sorting? ☒
 Data Order Field ID

Grid
 Readonly ☐
 Disable Add Feature ☐
 Disable Delete Feature ☒
 Show Row Numbering? ☒
 Number of columns to fixed on left *
 Number of spare rows *
 Custom Settings (JSON)

1

Figure 3 : Spreadsheet Form Element Properties - UI

Name	Description
Enable Header Sorting?	Determines if users can sort spreadsheet data by column in ascending or descending order.

Data Order Field ID	Field to keep the ordering sequence. Must correspond with a field id in the target form.																				
Readonly	Defines if the entire spreadsheet is editable.																				
Disable Add Feature	Determines if a new row can be added.																				
Disable Delete Feature	Determines if a row can be removed.																				
Show Row Numbering?	Show additional column on the leftmost to denote numbering. Sample Row numbering checked. <table><tr><th></th><th>Date</th><th>Category</th><th>Purpose</th><th>Amount</th></tr><tr><td>1</td><td>2017-03-06</td><td>Travel Expenses</td><td>Bus rental</td><td>3000.00</td></tr><tr><td>2</td><td>2017-03-06</td><td>Entertainment Claims</td><td>Catering</td><td>12000.00</td></tr><tr><td>3</td><td></td><td></td><td></td><td></td></tr></table>		Date	Category	Purpose	Amount	1	2017-03-06	Travel Expenses	Bus rental	3000.00	2	2017-03-06	Entertainment Claims	Catering	12000.00	3				
	Date	Category	Purpose	Amount																	
1	2017-03-06	Travel Expenses	Bus rental	3000.00																	
2	2017-03-06	Entertainment Claims	Catering	12000.00																	
3																					
Number of columns to fixed on left	Allows to specify the number of fixed (or frozen) columns on the left of the table. Default Value: 0																				
Number of spare rows	Number of spare row to be added automatically after lines with values. Sample 1 spare row <table><tr><th></th><th>Date</th><th>Category</th><th>Purpose</th><th>Amount</th></tr><tr><td>1</td><td>2017-03-06</td><td>Travel Expenses</td><td>Bus rental</td><td>3000.00</td></tr><tr><td>2</td><td>2017-03-06</td><td>Entertainment Claims</td><td>Catering</td><td>12000.00</td></tr><tr><td>3</td><td></td><td></td><td></td><td></td></tr></table>		Date	Category	Purpose	Amount	1	2017-03-06	Travel Expenses	Bus rental	3000.00	2	2017-03-06	Entertainment Claims	Catering	12000.00	3				
	Date	Category	Purpose	Amount																	
1	2017-03-06	Travel Expenses	Bus rental	3000.00																	
2	2017-03-06	Entertainment Claims	Catering	12000.00																	
3																					
Custom Settings (JSON)	Refer to plugins and APIs from https://handsontable.com/docs/6.2.2/tutorial-introduction.html for more customizations. Example: <pre>{ selectionMode: 'single' }</pre>																				

Validation & Data Binder

Edit Spreadsheet > UI > Validation & Data Binder

Validation

Validator

Min Number of Row Validation (Integer)

Max Number of Row Validation (Integer)

Error Message

Invalid number of rows

Data Binder

Load Binder

Store Binder

Figure 4 : Spreadsheet Form Element Properties - Validation & Data Binder

Name	Description
Validator	Attach a Validator plugin to validate the input value. Please see Form Validator. When will validation takes place? Validation will takes place whenever form is submitted except when it is submitted as "Save as Draft".
Min Number of Row Validation (Integer)	Defines the minimum number of rows required for input.
Max Number of Row Validation (Integer)	Defines the maximum number of rows possible for input.
Error Message	Error message to be shown when row requirements set above is not met.
Load Binder	Option by default. Grid data will be saved/loaded in JSON format in its defined database cell.
Store Binder	Option by default. Grid data will be saved/loaded in JSON format in its defined database cell.

Supported Formula Functions

The Spreadsheet element implemented partial of Excel formula functions as follows. For detail usage, please refer to [Excel Formula Functions](#) .

ABS(number)
ACOS(number)
ACOSH(number)
ACOT(number)
ACOTH(number)
ADD(num1, num2)
AGGREGATE(function_num, options, ref1, ref2)
AND(logical1, [logical2], ...)
ARABIC(text)
ARGS2ARRAY(arg1, [arg1], ...)
ASIN(number)

ASINH(number)
ATAN(number)
ATAN2(number_x, number_y)
ATANH(number)
AVEDEV(number1, [number2], ...)
AVERAGE(number1, [number2], ...)
AVERAGEA(number1, [number2], ...)
BASE(number, radix, min_length)
BESSELI(x, n)
BESSELJ(x, n)
BESSELK(x, n)
BESSELY(x, n)
BETA.DIST(x, alpha, beta, cumulative, A, B)
BETA.INV(probability, alpha, beta, A, B)
BETADIST(x, alpha, beta, cumulative, A, B)
BETAINV(probability, alpha, beta, A, B)
BIN2DEC(number)
BIN2HEX(number, places)
BIN2OCT(number, places)
BINOM.DIST(successes, trials, probability, cumulative)
BINOM.DIST.RANGE(trials, probability, successes, successes2)
BINOM.INV(trials, probability, alpha)
BINOMDIST(successes, trials, probability, cumulative)
BITAND(number1, number2)
BITLSHIFT(number, shift)
BITOR(number1, number2)
BITRSHIFT(number, shift)
BITXOR(number1, number2)
CEILING(number, significance, mode)
CEILINGMATH(number, significance, mode)
CEILINGPRECISE(number, significance, mode)
CHAR(number)
CHISQ.DIST(x, k, cumulative)
CHISQ.DIST.RT(x, k)
CHISQ.INV(probability, k)
CHISQ.INV.RT(p, k)
CHOOSE(index_num, value1, [value2], ...)
CLEAN(text)
CODE(text)
COMBIN(number, number_chosen)
COMBINA(number, number_chosen)
COMPLEX(real, imaginary, suffix)
CONCATENATE(arg1, [arg1], ...)
CONFIDENCE(alpha, standard_dev, size)
CONFIDENCE.NORM(lpha, standard_dev, size)
CONFIDENCE.T(lpha, standard_dev, size)
CONVERT(number, from_unit, to_unit)
CORREL(array1, array2)
COS(number)
COSH(number)
COT(number)
COTH(number)
COUNT(arg1, [arg1], ...)
COUNTA(arg1, [arg1], ...)
COUNTBLANK(arg1, [arg1], ...)
COUNTUNIQUE(arg1, [arg1], ...)
COVARIANCE.P(array1, array2)
COVARIANCE.S(array1, array2)
CSC(number)
CSCH(number)
CUMIPMT(rate, periods, value, start, end, type)
CUMPRINC(rate, periods, value, start, end, type)
DATE(year, month, day)
DATEVALUE(date_text)
DAY(serial_number)
DAYS(end_date, start_date)
DAYS360(start_date, end_date, method)
DB(cost, salvage, life, period, month)
DDB(cost, salvage, life, period, factor)
DEC2BIN(number, places)
DEC2HEX(number, places)
DEC2OCT(number, places)
DECIMAL(number, radix)

DEGREES(number)
DELTA(number1, number2)
DEVSQ(number1, [number2], ...)
DIVIDE(dividend, divisor)
DOLLAR(number, decimals)
DOLLARDE(dollar, fraction)
DOLLARFR(dollar, fraction)
E()
EDATE(start_date, months)
EFFECT(rate, periods)
EOMONTH(start_date, months)
EQ(value1, value2)
ERF(lower_bound, upper_bound)
ERFC(x)
EVEN(number)
EXACT(text1, text2)
EXPON.DIST(x, lambda, cumulative)
EXPONDIST(x, lambda, cumulative)
F.DIST(x, d1, d2, cumulative)
F.DIST.RT(x, d1, d2)
F.INV(probability, d1, d2)
F.INV.RT(p, d1, d2)
FACT(number)
FACTDOUBLE(number)
FALSE()
FDIST(x, d1, d2, cumulative)
FDISTRT(x, d1, d2)
FIND(find_text, within_text, position)
FINV(probability, d1, d2)
FINVRT(p, d1, d2)
FISHER(x)
FISHERINV(y)
FIXED(number, decimals, no_commas)
FLOOR(number, significance)
FORECAST(x, data_y, data_x)
FREQUENCY(data, bins)
FV(rate, periods, payment, value, type)
FVSCHEDULE(principal, schedule)
GAMMA(number)
GAMMA.DIST(value, alpha, beta, cumulative)
GAMMA.INV(probability, alpha, beta)
GAMMADIST(value, alpha, beta, cumulative)
GAMMAINV(probability, alpha, beta)
GAMMALN(number)
GAMMALN.PRECISE(x)
GAUSS(z)
GCD(GCD)
GEOMEAN(number1, [number2], ...)
GESTEP(number, step)
GROWTH(known_y, known_x, new_x, use_const)
GTE(num1, num2)
HARMEAN(number1, [number2], ...)
HEX2BIN(number, places)
HEX2DEC(number)
HEX2OCT(number, places)
HOUR(serial_number)
HTML2TEXT(value)
HYPGEOM.DIST(x, n, M, N, cumulative)
HYPGEOMDIST(x, n, M, N, cumulative)
IF(test, then_value, otherwise_value)
IMABS(inumber)
IMAGINARY(inumber)
IMARGUMENT(inumber)
IMCONJUGATE(inumber)
IMCOS(inumber)
IMCOSH(inumber)
IMCOT(inumber)
IMCSC(inumber)
IMCSCH(inumber)
IMDIV(inumber1, inumber2)
IMEXP(inumber)
IMLN(inumber)
IMLOG10(inumber)

IMLOG2(number)
IMPOWER(number, number)
IMPRODUCT(number1, number2, ...)
IMREAL(number)
IMSEC(number)
IMSECH(number)
IMSIN(number)
IMSINH(number)
IMSQRT(number)
IMSUB(number1, number2)
IMSUM(number1, number2, ...)
IMTAN(number)
INT(number)
INTERCEPT(known_y, known_x)
INTERVAL(second)
IPMT(rate, period, periods, present, future, type)
IRR(values, guess)
ISBINARY(number)
ISBLANK(value)
ISEVEN(number)
ISLOGICAL(ISNONTEXT)
ISNONTEXT(ISNONTEXT)
ISNUMBER(value)
ISODD(number)
ISOWEENUM(date)
ISPMT(rate, period, periods, value)
ISTEXT(value)
JOIN(array, separator)
KURT(number1, [number2], ...)
LCM(number1, [number2], ...)
LEFT(text, number)
LEN(text)
LINEST(data_y, data_x)
LN(number)
LOG(number, base)
LOG10(number)
LOGEST(data_y, data_x)
LOGNORM.DIST(x, mean, sd, cumulative)
LOGNORM.INV(probability, mean, sd)
LOGNORMDIST(x, mean, sd, cumulative)
LOGNORMINV(probability, mean, sd)
LOWER(text)
LT(num1, num2)
LTE(num1, num2)
MATCH(lookupValue, lookupArray, matchType)
MAX(number1, [number2], ...)
MAXA(number1, [number2], ...)
MEDIAN(number1, [number2], ...)
MID(text, start, number)
MIN(number1, [number2], ...)
MINA(number1, [number2], ...)
MINUS(num1, num2)
MINUTE(serial_number)
MIRR(values, finance_rate, reinvest_rate)
MOD(dividend, divisor)
MODE.MULT(number1, [number2], ...)
MODE.SNGL(number1, [number2], ...)
MODEMULT(number1, [number2], ...)
MODESNGL(number1, [number2], ...)
MONTH(serial_number)
MROUND(number, multiple)
MULTINOMIAL(number1, [number2], ...)
MULTIPLY(factor1, factor2)
NE(value1, value2)
NEGBINOM.DIST(k, r, p, cumulative)
NEGBINOMDIST(k, r, p, cumulative)
NETWORKDAYS(start_date, end_date, holidays)
NOMINAL(rate, periods)
NORM.DIST(x, mean, sd, cumulative)
NORM.INV(probability, mean, sd)
NORM.S.DIST(z, cumulative)
NORM.S.INV(probability)
NORMDIST(x, mean, sd, cumulative)

NORMINV(probability, mean, sd)
NORMSDIST(x, mean, sd, cumulative)
NORMSINV(probability)
NOT(logical)
NOW()
NPER(rate, payment, present, future, type)
NPV(arg1, [arg2], ...)
NUMBERS(arg1, [arg2], ...)
NUMERAL(number, format)
OCT2BIN(number, places)
OCT2DEC(number)
OCT2HEX(number, places)
ODD(number)
OR(logical1, [logical2], ...)
PDURATION(rate, present, future)
PEARSON(data_x, data_y)
PERMUT(number, number_chosen)
PERMUTATIONA(number, number_chosen)
PHI(x)
PI()
PMT(rate, periods, present, future, type)
POISSON.DIST(x, mean, cumulative)
POISSONDIST(x, mean, cumulative)
POW(base, exponent)
POWER(number, power)
PPMT(rate, period, periods, present, future, type)
PRODUCT(number1, [number2], ...)
PROPER(text)
PV(rate, periods, payment, future, type)
QUOTIENT(numerator, denominator)
RADIANS(number)
RAND()
RANDBETWEEN(bottom, top)
RATE(periods, payment, present, future, type, guess)
REFERENCE(context, reference)
REGEXEXTRACT(text, regular_expression)
REGEXMATCH(text, regular_expression, full)
REGEXREPLACE(text, regular_expression, replacement)
REPLACE(text, position, length, new_text)
REPT(text, number)
RIGHT(text, number)
ROMAN(number)
ROUND(number, digits)
ROUNDDOWN(number, digits)
ROUNDUP(number, digits)
RRI(periods, present, future)
RSQ(data_x, data_y)
SEARCH(find_text, within_text, position)
SEC(number)
SECH(number)
SECOND(serial_number)
SERIESSUM(x, n, m, coefficients)
SIGN(number)
SIN(number)
SINH(number)
SKEW(number1, [number2], ...)
SKEW.P(number1, [number2], ...)
SKEWP(number1, [number2], ...)
SLN(cost, salvage, life)
SLOPE(data_y, data_x)
SPLIT(text, separator)
SQRT(number)
SQRTPI(number)
STANDARDIZE(x, mean, sd)
STDEV.P(number1, [number2], ...)
STDEV.S(number1, [number2], ...)
STDEVA(number1, [number2], ...)
STDEV.P(number1, [number2], ...)
STDEVPA(number1, [number2], ...)
STDEV.S(number1, [number2], ...)
STEYX(data_y, data_x)
SUBSTITUTE(text, old_text, new_text, occurrence)
SUBTOTAL(function_code, ref1)

SUM(number1, [number2], ...)
 SUMPRODUCT(array1, [array2], [array3], ...)
 SUMSQ(number1, [number2], ...)
 SUMX2MY2(array_x, array_y)
 SUMX2PY2(array_x, array_y)
 SUMXMY2(array_x, array_y)
 SWITCH(expression, value1, result1, [value2, result2], [value3, result3], ..., [default])
 SYD(cost, salvage, life, period)
 T(value)
 T.DIST(x, df, cumulative)
 T.DIST.2T(x, df)
 T.DIST.RT(x, df)
 T.INV(probability, df)
 T.INV.2T(probability, df)
 TAN(number)
 TANH(number)
 TBILLEQ(settlement, maturity, discount)
 TBILLPRICE(settlement, maturity, discount)
 TBILLYIELD(settlement, maturity, price)
 TDIST(x, df, cumulative)
 TDIST2T(x, df)
 TDISTRT(x, df)
 TEXT(value, format)
 TIME(hour, minute, second)
 TIMEVALUE(time_text)
 TINV(probability, df)
 TINV2T(probability, df)
 TODAY()
 TRANSPOSE(matrix)
 TREND(data_y, data_x, new_data_x)
 TRIM(text)
 TRUE()
 TRUNC(number, digits)
 UNICHAR(number)
 UNICODE(text)
 UNIQUE(arg1, [arg2], ...)
 UPPER(text)
 VALUE(text)
 VAR.P(number1, [number2], ...)
 VAR.S(number1, [number2], ...)
 VARA(number1, [number2], ...)
 VARP(number1, [number2], ...)
 VARPA(number1, [number2], ...)
 VARS(number1, [number2], ...)
 WEEKDAY(serial_number, return_type)
 WEEKNUM(serial_number, return_type)
 WEIBULL.DIST(x, alpha, beta, cumulative)
 WEIBULLDIST(x, alpha, beta, cumulative)
 WORKDAY(start_date, days, holidays)
 XNPV(rate, values, dates)
 XOR(logical1, [logical2], ...)
 YEAR(serial_number)
 YEARFRAC(start_date, end_date, basis)

Special Function

FORMDATA(formDefId, primaryKey, fieldName) in Spreadsheet "Formula" column.

where:

- formDefId: The form id of the source form containing the setup data to 'pull' from.
- primaryKey: The dependent pulldown field id in your spreadsheet design, that provides the WHERE condition to search the data to populate.
- fieldName: The field id of the source form to populate in this spreadsheet column based on the above 'primaryKey' value.

Use this function to 'pull' and populate a spreadsheet cell based on available data in other forms.

For example, to make spreadsheet pull and display the population value after the user picks the city name (pulldown menu), use FORMDATA("city_formId",select_city,"population") where 'select_city' is the first field id in your spreadsheet.

Below is a sample app to demonstrate the use of Spreadsheet form element:

- Example app using Spreadsheet in Form [APP_spreadsheet app.jwa](#).