# Using Excel files as sources for ConfiForms fields

Since version 1.53.4 we have introduced couple of **experimental APIs** which help you to make a bridge between ConfiForms web-services backed fields and Excel files attached to your Confluence page.

#### 3 services:

- excel-2json
- excel2007-2json
- csv-2json (starting from ConfiForms 2.27.17 Release Notes#Version2.27.17)

## These 2 APIs are available at the following URLs:

File format	API URL	Method
Excel 97-2004 (xls)	https://YOUR_CONFLUENCE_SERVER/ajax/confiforms/rest/excel-2json.action	GET
Excel (xlsx)	https://YOUR_CONFLUENCE_SERVER/ajax/confiforms/rest/excel2007-2json.action	GET
CSV (default format)	https://your_confluence_server/ajax/confiforms/rest/csv-2json.action	GET

#### Services expect/support the following parameters:

Parameter	Description	Туре	Required
attachmentId	Attachment ID of the Excel file attachment to use as source	Numeric	Yes
skipRows	Number of rows to skip (from the top). Defaults to 0 (rows to skip, from the top of the sheet)	Numeric	No
idCellNum	Which column to use as ID	Numeric	No
labelCellNum	Which column to use as label	Numeric	No
lookupValue	Can supply a lookup value for the service to lookup. The subset returned will be filtered to match (as substring) the given value	Text	No
Since 2.27.21			
lookupField	Limiting the matching logic to specific field (cell)	Text	No
Since 2.27.23	Example: cell1 to limit to cell1 value only. If not specified then the match for given lookupValue is done against the whole row (all cells)		
exactMatch Since 2.27.24	You can set the search to use "exact match", by default (when exactMatch is unspecified or false) the search is done as text substring match (case insensitive)	Boolean	No





NB! Only the first sheet from Excel file is currently supported!

#### Examples:

https://YOUR\_CONFLUENCE\_SERVER/ajax/confiforms/rest/excel-2json.action?attachmentId=2195459&skipRows=1&idCellNum=0&labelCellNum=1 https://YOUR\_CONFLUENCE\_SERVER/ajax/confiforms/rest/excel2007-2json.action?attachmentId=2195460&skipRows=1

Last example reads ALL the rows and columns (skipping just the first one) and creates a structure similar to the one listed below

```
{
    "cell2": "uno",
    "cell3": "in cell 3",
    "cell1": "1"
},

{
    "cell2": "duo",
    "cell1": "2"
},

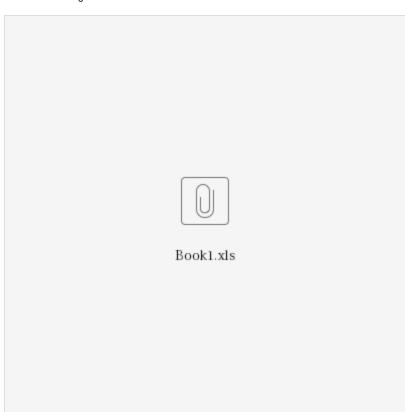
{
    "cell2": "tres",
    "cell3": "here is cell three",
    "cell1": "3"
},

{
    "cell2": "four",
    "cell1": "4"
},

{
    "cell2": "five",
    "cell1": "5"
},

{
    "cell2": "six",
    "cell1": "6"
}
```

### From the following excel file



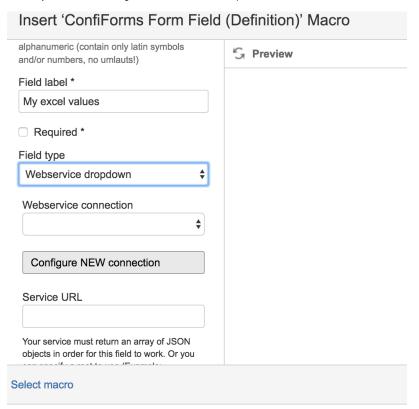
As you can see, we can create a JSON Array structure that perfectly fits our ConfiForms web-service enabled fields

So, here is how you can create a ConfiForms Form which uses this file and shows it's values in the dropdown

### Configuration for ConfiForms looks like this



With "myvalues" field configured as web-service dropdown

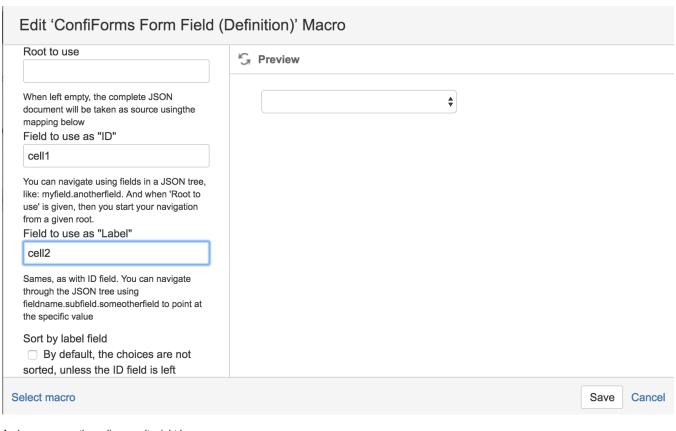


We have defined new connection to use the above mentioned excel to JSON APIs

# WS Connection Configuration

Name:*	excel-file-demo			
	The name for your configuration, will be visible	to users		
WS URL	/ajax/confiforms/rest/excel-2json.action?attachmentId=29392959&skipRows=1			
	' ' ' ' ' ' ' ' ' ' ' ' ' ' ' ' ' ' ' '	if you want to use this as a local proxy (in this can ner with the service URL you setup. For WS-back ests and return JSON as a result		to :
Default request				
headers		6		
	In HeaderName:HeaderValue format, use new	line for each new header/value pair		
Username				
	Leave empty, if the connection requires no aut	hentication		
Password				
When NTLM auth shall be used				
			Save CI	lose

And set the mapping to use cell1 as ID column for the dropdown and cell2 as a label



And you can see the online results right here

My excel values \* unoduotresfourfivesix

Save

All the registrations so far (and yes, we can access ANY other field, like cell3 in our views, see below)

My excel values	myvalues.id	myvalues.cell1	myvalues.label	myvalues.cell3
duo	c81e728d9d4c2f636f067f89cc14862c	2	duo	
five	e4da3b7fbbce2345d7772b0674a318d5	5	five	
six	1679091c5a880faf6fb5e6087eb1b2dc	6	six	
four	a87ff679a2f3e71d9181a67b7542122c	4	four	
uno	c4ca4238a0b923820dcc509a6f75849b	1	uno	in cell 3
uno	c4ca4238a0b923820dcc509a6f75849b	1	uno	in cell 3
uno	c4ca4238a0b923820dcc509a6f75849b	1	uno	in cell 3
uno	c4ca4238a0b923820dcc509a6f75849b	1	uno	in cell 3
duo	c81e728d9d4c2f636f067f89cc14862c	2	duo	

four	a87ff679a2f3e71d9181a67b7542122c	4	four	
six	1679091c5a880faf6fb5e6087eb1b2dc	6	six	
uno	c4ca4238a0b923820dcc509a6f75849b	1	uno	in cell 3
six	1679091c5a880faf6fb5e6087eb1b2dc	6	six	
uno	c4ca4238a0b923820dcc509a6f75849b	1	uno	in cell 3
four	a87ff679a2f3e71d9181a67b7542122c	4	four	
uno	c4ca4238a0b923820dcc509a6f75849b	1	uno	in cell 3
uno	c4ca4238a0b923820dcc509a6f75849b	1	uno	in cell 3
uno	c4ca4238a0b923820dcc509a6f75849b	1	uno	in cell 3
uno	c4ca4238a0b923820dcc509a6f75849b	1	uno	in cell 3
duo	c81e728d9d4c2f636f067f89cc14862c	2	duo	
uno	c4ca4238a0b923820dcc509a6f75849b	1	uno	in cell 3
uno	c4ca4238a0b923820dcc509a6f75849b	1	uno	in cell 3
uno	c4ca4238a0b923820dcc509a6f75849b	1	uno	in cell 3
duo	c81e728d9d4c2f636f067f89cc14862c	2	duo	
duo	c81e728d9d4c2f636f067f89cc14862c	2	duo	
uno	c4ca4238a0b923820dcc509a6f75849b	1	uno	in cell 3
uno	c4ca4238a0b923820dcc509a6f75849b	1	uno	in cell 3
tres	eccbc87e4b5ce2fe28308fd9f2a7baf3	3	tres	here is cell three
tres	eccbc87e4b5ce2fe28308fd9f2a7baf3	3	tres	here is cell three
uno	c4ca4238a0b923820dcc509a6f75849b	1	uno	in cell 3
uno	c4ca4238a0b923820dcc509a6f75849b	1	uno	in cell 3
tres	eccbc87e4b5ce2fe28308fd9f2a7baf3	3	tres	here is cell three