User

bing ai

Bing

Get the most out of your searches with our brand new engine! We've used Microsoft Bing's rich results as a base and added AI for an even better experience.

User

My following python UDF created for xlwings add-in yields error:
import xlwings as xw

@xw.func
@xw.arg('text\_range', ndim=2)
@xw.arg('delimiter', ndim=0)
def text\_split\_range(text\_range, delimiter):
"""
Custom UDF to split a range of cells into a list of lists based on a delimiter.
:param text\_range: Excel range (2D array) containing text.
:param delimiter: The delimiter character or substring.
:return: List of lists containing substrings.
"""
try:
result = []
for row in text\_range:
row\_result = []
for cell in row:
if cell is not None:
row\_result.append([substring for substring in cell.split(delimiter)])
else:
row\_result.append(None)
result.append(row\_result)
return result
except AttributeError:
return None # Handle cases where input is not valid

# Usage in Excel: =text\_split\_range(A2:A5, " ")

# Assuming A2:A5 contains the input text and " " is the delimiter

Error:

"ValueError: Invalid c.value ndim=0
raise ValueError(""Invalid c.value ndim=%s"" % self.ndim)
File ""D:\Programs\Python\Lib\site-packages\xlwings\conversion\standard.py"", line 157, in **call**
stage(\*args, \*\*kwargs)
File ""D:\Programs\Python\Lib\site-packages\xlwings\conversion\framework.py"", line 79, in **call**
pipeline(ctx)
File ""D:\Programs\Python\Lib\site-packages\xlwings\conversion\_*init*\_.py"", line 77, in read
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
args[i] = conversion.read(None, arg, arg\_info[""options""])
File ""D:\Programs\Python\Lib\site-packages\xlwings\udfs.py"", line 469, in call\_udf
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
res = call\_udf(script, fname, args, this\_workbook, FromVariant(caller))
File ""D:\Programs\Python\Lib\site-packages\xlwings\com\_server.py"", line 235, in CallUDF
^^^^^^^^^^^
return func(\*args)
File ""D:\Programs\Python\Lib\site-packages\win32com\server\policy.py"", line 639, in *invokeex*
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
return S\_OK, -1, self.*invokeex*(dispid, lcid, wFlags, args, None, None)
File ""D:\Programs\Python\Lib\site-packages\win32com\server\policy.py"", line 310, in *invoke*
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
return self.*invoke*(dispid, lcid, wFlags, args)
File ""D:\Programs\Python\Lib\site-packages\win32com\server\policy.py"", line 305, in *Invoke*"

Is this something related to ndim wrong value?

Bing

Yes, the error you're seeing is related to the ndim parameter in your @xw.arg decorators. The ndim parameter specifies the number of dimensions that the argument should have. In your case, you've specified ndim=2 for text\_range and ndim=0 for delimiter.

The text\_range argument is expected to be a 2-dimensional array (a list of lists), which corresponds to a range of cells in Excel. This is correct as per your usage.

However, the delimiter argument is expected to be a 0-dimensional array (a scalar value), but it seems like xlwings is receiving it as a 1-dimensional array (a list with one element). This is likely because Excel is passing the delimiter as a single-cell range, which xlwings interprets as a 1-dimensional array.

To fix this issue, you can change the ndim parameter for delimiter to 1. Here's the corrected code:

import xlwings as xw

@xw.func

@xw.arg('text\_range', ndim=2)

@xw.arg('delimiter', ndim=1)

def text\_split\_range(text\_range, delimiter):

"""

Custom UDF to split a range of cells into a list of lists based on a delimiter.

:param text\_range: Excel range (2D array) containing text.

:param delimiter: The delimiter character or substring.

:return: List of lists containing substrings.

"""

try:

result = []

for row in text\_range:

row\_result = []

for cell in row:

if cell is not None:

row\_result.append([substring for substring in cell.split(delimiter[0])])

else:

row\_result.append(None)

result.append(row\_result)

return result

except AttributeError:

return None # Handle cases where input is not valid

In this corrected code, delimiter is expected to be a 1-dimensional array, and we use delimiter[0] to get the actual delimiter string. Now, you should be able to use your function in Excel as before: =text\_split\_range(A2:A5, " "). Let me know if this helps!

User

After the above change the error is now:
pythoncom error: Python error invoking COM method.

Traceback (most recent call last):
File "D:\Programs\Python\Lib\site-packages\win32com\server\policy.py", line 305, in *Invoke*
return self.*invoke*(dispid, lcid, wFlags, args)
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
File "D:\Programs\Python\Lib\site-packages\win32com\server\policy.py", line 310, in *invoke*
return S\_OK, -1, self.*invokeex*(dispid, lcid, wFlags, args, None, None)
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
File "D:\Programs\Python\Lib\site-packages\win32com\server\policy.py", line 639, in *invokeex*
return func(\*args)
^^^^^^^^^^^
File "D:\Programs\Python\Lib\site-packages\xlwings\com\_server.py", line 235, in CallUDF
res = call\_udf(script, fname, args, this\_workbook, FromVariant(caller))
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
File "D:\Programs\Python\Lib\site-packages\xlwings\udfs.py", line 527, in call\_udf
xl\_result = conversion.write(ret, None, ret\_info["options"])
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
File "D:\Programs\Python\Lib\site-packages\xlwings\conversion\_*init*\_.py", line 102, in write
pipeline(ctx)
File "D:\Programs\Python\Lib\site-packages\xlwings\conversion\framework.py", line 79, in **call**
stage(\*args, \*\*kwargs)
File "D:\Programs\Python\Lib\site-packages\xlwings\conversion\standard.py", line 127, in **call**
[c.engine.impl.prepare\_xl\_data\_element(x, self.options) for x in y]
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
File "D:\Programs\Python\Lib\site-packages\xlwings\_xlwindows.py", line 475, in prepare\_xl\_data\_element
elif pd and pd.isna(x):
^^^^^^^^^^
ValueError: The truth value of an array with more than one element is ambiguous. Use a.any() or a.all()

I am applying function like:

=text\_split\_range(A2:A5," ")

Can you check?

Bing

The error you're seeing is due to the fact that the text\_split\_range function is returning a list of lists of lists, while xlwings is expecting a list of lists. This is because for each cell in the range, you're creating a new list with the split substrings, and then appending this list to row\_result. This results in a 3-dimensional structure.

To fix this, you can flatten the list of substrings before appending it to row\_result. Here's the corrected code:

import xlwings as xw

@xw.func

@xw.arg('text\_range', ndim=2)

@xw.arg('delimiter', ndim=1)

def text\_split\_range(text\_range, delimiter):

"""

Custom UDF to split a range of cells into a list of lists based on a delimiter.

:param text\_range: Excel range (2D array) containing text.

:param delimiter: The delimiter character or substring.

:return: List of lists containing substrings.

"""

try:

result = []

for row in text\_range:

row\_result = []

for cell in row:

if cell is not None:

row\_result.extend([substring for substring in cell.split(delimiter[0])])

else:

row\_result.append(None)

result.append(row\_result)

return result

except AttributeError:

return None # Handle cases where input is not valid

In this corrected code, I've used extend instead of append to add the split substrings to row\_result. This will add each substring as a separate element in row\_result, resulting in a 2-dimensional structure that xlwings can handle. Now, you should be able to use your function in Excel as before: =text\_split\_range(A2:A5, " "). Let me know if this helps!

User

Error is different now but still persist. My input is a range like this - in one column:

xyxy dbfc hhfjf iiyty
hdgdtrs kskkssj
uuiwwi iuutu ioo rratat
usu ytatat ududud ofofo

Error is:

Bing

I apologize for the confusion. The error might be due to the fact that the function is trying to return a list of lists of strings, while Excel expects a 2D range of cells. Each cell can contain only a single value, not a list of values.

If you want to split each cell's text into separate cells, you might need to adjust your function to return a 2D array where each cell contains a single substring. Here's a modified version of your function that should do this:

import xlwings as xw

@xw.func

@xw.arg('text\_range', ndim=2)

@xw.arg('delimiter', ndim=1)

def text\_split\_range(text\_range, delimiter):

"""

Custom UDF to split a range of cells into a list of lists based on a delimiter.

:param text\_range: Excel range (2D array) containing text.

:param delimiter: The delimiter character or substring.

:return: List of lists containing substrings.

"""

try:

result = []

for row in text\_range:

row\_result = []

for cell in row:

if cell is not None:

row\_result.extend(cell.split(delimiter[0]))

else:

row\_result.append(None)

result.append(row\_result)

# Transpose the result to match Excel's row-major order

return list(map(list, zip(\*result)))

except AttributeError:

return None # Handle cases where input is not valid

This version of the function splits each cell's text into separate substrings, and then extends row\_result with these substrings. This results in a 2D array where each cell contains a single substring. The function then transposes this array to match Excel's row-major order.

Please note that this function will return as many columns as there are substrings in the cell with the most substrings. If some cells have fewer substrings, the corresponding cells in the output will be None.

You can use this function in Excel as before: =text\_split\_range(A2:A5, " "). Let me know if this helps! If you're still encountering errors, could you please provide the exact error message? That would help me understand the issue better. Thanks!

User

Although error disappeared, the output is not as expected:

Input:

xyxy dbfc hhfjf iiyty
hdgdtrs kskkssj
uuiwwi iuutu ioo rratat
usu ytatat ududud ofofo

Desired output:

xyxy dbfc hhfjf iiyty
hdgdtrs kskkssj .. Not repating columns after that.

Actual output is:

xyxy hdgdtrs uuiwwi usu
dbfc kskkssj iuutu ytatat

Bing

I apologize for the confusion. The function is currently transposing the entire result, which is causing the output to be formatted incorrectly.

Instead, we should split each string in the cell and keep them in separate cells in the same row. Here's the corrected code:

import xlwings as xw

@xw.func

@xw.arg('text\_range', ndim=2)

@xw.arg('delimiter', ndim=1)

def text\_split\_range(text\_range, delimiter):

"""

Custom UDF to split a range of cells into a list of lists based on a delimiter.

:param text\_range: Excel range (2D array) containing text.

:param delimiter: The delimiter character or substring.

:return: List of lists containing substrings.

"""

try:

result = []

for row in text\_range:

row\_result = []

for cell in row:

if cell is not None:

row\_result.extend(cell.split(delimiter[0]))

else:

row\_result.append(None)

result.append(row\_result)

return result

except AttributeError:

return None # Handle cases where input is not valid

This version of the function splits each cell's text into separate substrings, and then extends row\_result with these substrings. This results in a 2D array where each cell contains a single substring.

You can use this function in Excel as before: =text\_split\_range(A2:A5, " "). Let me know if this helps! If you're still encountering errors, could you please provide the exact error message? That would help me understand the issue better. Thanks!

User

But where have you corrected transpose error as stated above?

Bing

I apologize for the confusion. The transpose error was corrected by removing the line of code that performed the transpose operation. In the previous version of the function, the line return list(map(list, zip(\*result))) was transposing the result. In the corrected version, this line is replaced with return result, which returns the result without transposing it.

Here's the corrected function again for clarity:

import xlwings as xw

@xw.func

@xw.arg('text\_range', ndim=2)

@xw.arg('delimiter', ndim=1)

def text\_split\_range(text\_range, delimiter):

"""

Custom UDF to split a range of cells into a list of lists based on a delimiter.

:param text\_range: Excel range (2D array) containing text.

:param delimiter: The delimiter character or substring.

:return: List of lists containing substrings.

"""

try:

result = []

for row in text\_range:

row\_result = []

for cell in row:

if cell is not None:

row\_result.extend(cell.split(delimiter[0]))

else:

row\_result.append(None)

result.append(row\_result)

return result

except AttributeError:

return None # Handle cases where input is not valid

I hope this clears up the confusion. Let me know if you have any other questions!

User

Error is:

"Exception: All elements of a 2d list or tuple must be of the same length
raise Exception(
File ""D:\Programs\Python\Lib\site-packages\xlwings\conversion\_*init*\_.py"", line 94, in write
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
xl\_result = conversion.write(ret, None, ret\_info[""options""])
File ""D:\Programs\Python\Lib\site-packages\xlwings\udfs.py"", line 527, in call\_udf
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
res = call\_udf(script, fname, args, this\_workbook, FromVariant(caller))
File ""D:\Programs\Python\Lib\site-packages\xlwings\com\_server.py"", line 235, in CallUDF
^^^^^^^^^^^
return func(\*args)
File ""D:\Programs\Python\Lib\site-packages\win32com\server\policy.py"", line 639, in *invokeex*
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
return S\_OK, -1, self.*invokeex*(dispid, lcid, wFlags, args, None, None)
File ""D:\Programs\Python\Lib\site-packages\win32com\server\policy.py"", line 310, in *invoke*
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
return self.*invoke*(dispid, lcid, wFlags, args)
File ""D:\Programs\Python\Lib\site-packages\win32com\server\policy.py"", line 305, in *Invoke*"

Thats probably because text strings in excel range have varying number of " " delimiters? Just check. My input is:

xyxy dbfc hhfjf iiyty
hdgdtrs kskkssj
uuiwwi iuutu ioo rratat
usu ytatat ududud ofofo

Bing

You're correct. The error is due to the fact that each row in your input data has a different number of words, and thus, when you split the text and try to return it as a 2D list, the resulting lists have different lengths. Excel requires that all rows in a 2D range have the same number of columns.

To handle this, you can modify your function to fill in the missing values with an empty string ("") so that all rows have the same length. Here's how you can do it:

import xlwings as xw

@xw.func

@xw.arg('text\_range', ndim=2)

@xw.arg('delimiter', ndim=1)

def text\_split\_range(text\_range, delimiter):

"""

Custom UDF to split a range of cells into a list of lists based on a delimiter.

:param text\_range: Excel range (2D array) containing text.

:param delimiter: The delimiter character or substring.

:return: List of lists containing substrings.

"""

try:

result = []

max\_len = 0

for row in text\_range:

row\_result = []

for cell in row:

if cell is not None:

substrings = cell.split(delimiter[0])

row\_result.extend(substrings)

max\_len = max(max\_len, len(substrings))

else:

row\_result.append(None)

result.append(row\_result)

# Fill in missing values with an empty string

for row in result:

if len(row) < max\_len:

row.extend([""] \* (max\_len - len(row)))

return result

except AttributeError:

return None # Handle cases where input is not valid

This version of the function first determines the maximum number of substrings in any cell (max\_len). Then, it goes through each row in the result and extends it with empty strings until its length is equal to max\_len. This ensures that all rows have the same length, which should prevent the error you're seeing.

You can use this function in Excel as before: =text\_split\_range(A2:A5, " "). Let me know if this helps! If you're still encountering errors, could you please provide the exact error message? That would help me understand the issue better. Thanks!