

Nested JSON with Dynamic Query

 stackoverflow.com/questions/46315577/nested-json-with-dynamic-query



I finally solved the issue. For those who are interested here is the solution:

First step is to create a new table type which can be used in the recursive routine:

```
CREATE TYPE [dbo].[JSONCTETableType] AS TABLE(  
    [Sequence] [uniqueidentifier] NULL,  
    [ParentSequence] [uniqueidentifier] NULL,  
    [JSON] [nvarchar](max) NULL,  
    [IndentLevel] [int] NULL,  
    [WBS] [nvarchar](512) NULL,  
    [ChildCount] [int] NULL  
)
```

In the second step I changed the original recursive function to use this table definition:

```

-- =====
-- Author:      Arno Voerman
-- Create date: 2017, September
-- =====
CREATE FUNCTION [dbo].[TREEIFY]
(
    @ParentId uniqueidentifier, @JSONTABLE JSONCTETableType READONLY
)
RETURNS NVARCHAR(MAX)
AS
BEGIN
    DECLARE @Json NVARCHAR(MAX);

    If @ParentId Is NULL
        Begin
            SET @Json = (
                Select JSON +
                    Case When ChildCount > 0
                        THEN ', "children":[' + dbo.treeIfy(Sequence,@JSONTABLE) +
'    ]]'
                        ELSE ', "leaf":true}'
                    END + ', ' AS 'data()'
                FROM @JSONTABLE
                Where ParentSequence Is Null
                FOR XML PATH('')
            )
        END
    Else
        Begin
            SET @Json = (
                Select JSON +
                    Case When ChildCount > 0
                        THEN ', "children":[' + dbo.treeIfy(Sequence,@JSONTABLE) +
'    ]]'
                        ELSE ', "leaf":true}'
                    END + ', ' AS 'data()'
                FROM @JSONTABLE
                Where ParentSequence = @ParentId
                FOR XML PATH('')
            )
        End
    RETURN @Json
END

```

To get the nested JSON-result from the query the query-result must be transformed into JSONCTETableType. This is done in the following CTE which also generates the JSON and some extra data which my program needs:

```

DECLARE @JSONTT as JSONCTETableType;
WITH JSONCTETable AS (
    SELECT  Sequence, IDParentSequence As ParentSequence,
            (SELECT * FROM YourTable Where Sequence=anchor.Sequence FOR JSON AUTO,
WITHOUT_ARRAY_WRAPPER) As JSON,
            0 as IndentLevel,
            Cast(ROW_NUMBER() OVER(ORDER BY mnu_Order ASC) As NVARCHAR(512)) AS
WBS
    FROM    YourTable AS anchor
    WHERE   IDParentSequence is null
UNION  ALL -----

SELECT  recur.Sequence, recur.IDParentSequence As ParentSequence,
        (SELECT * FROM YourTable Where Sequence=recur.Sequence FOR JSON AUTO,
WITHOUT_ARRAY_WRAPPER) As JSON,
        cte.IndentLevel + 1,
        Cast(cte.WBS + '.' + Cast(ROW_NUMBER() OVER(ORDER BY mnu_Order ASC) As
NVARCHAR(10)) AS NVARCHAR(512)) AS WBS
FROM    YourTable AS recur
        INNER JOIN JSONCTETable AS cte ON cte.Sequence = recur.IDParentSequence

)INSERT INTO @JSONTT (Sequence, ParentSequence, JSON, IndentLevel, WBS,
ChildCount)
SELECT *, (Select count(*) From JSONCTETable tmpCTE Where
tmpCTE.ParentSequence=JSONCTETable.Sequence) as ChildCount
FROM    JSONCTETable
ORDER BY WBS;

--Remove last character ("}") of all JSON. Needed to simplify the add "CHILD"
Update @JSONTT SET JSON = stuff(JSON,len(JSON),1,'');

```

The last step is simple:

```

Select '[' + dbo.treeIfy(NULL,@JSONTT) + ']' as data, (Select count(Sequence) From
@JSONTT) as totRows;

```

Good luck!

Arno

answered Sep 24 '17 at 6:44



Arno Voerman

4722 silver badges88 bronze badges

[add a comment](#)

Your Answer

Sign up or [log in](#)

Sign up using Google

Sign up using Facebook



Sign up using Email and Password

Post as a guest

Required, but never shown

By clicking “Post Your Answer”, you agree to our [terms of service](#), [privacy policy](#) and [cookie policy](#).

Not the answer you're looking for? Browse other questions tagged [sql-server](#) [json](#) [vb.net](#) [nested](#) [dynamic-sql](#) or [ask your own question](#).
