XML Export Method (Handled In UtilityController)

The structure we are trying to export is (i.e. the input to exportToXML in UtilityController) :

[{"bookID":22,

"title":"HALLO GILBERT ONLY1",

"authors":[{"authorID":2,"firstName":"Gilbert","lastName":"Vincenta"}]

…

Into :

<?xml version="1.0"?>

<books> (0)

<data> (1)

<bookID>22</bookID> (2)

<title>HALLO GILBERT ONLY 1</title> (3)

<authors> (4)

<data> (5)

<authorID>2</authorID> (6)

<firstName>Gilbert</firstName> (7)

<lastName>Vincenta</lastName> (8)

</data>

</authors> (9)

</data> (10)

….

</books>

Notice that in the code:

$rootTag is the root element tag, in this case, "books".

$nestedTag, in this case is "authors", is used to check whether the JSON contains the highlighted data that needs to be parsed like the <authors> <data> …. </data> </authors> (a.k.a child objects) as shown above.

The algorithm goes as follow:

1. Initialise XML root tag (this step is done in exportToXML function).

For each JSON element found in the array:

2. (this step is done in constructChild function):
    a. If the object has the highlighted data

i. Parse from (1) to (3)
ii. Repeat 2(a) for the highlighted data. Notice that as nesting only occurs once, i.e. the child object doesn't have another potential grandchild objects to be parsed like <authors> <data> …. </data> </authors>, so basically, the recursion is useless for now (but might be useful in the future), and in this stage, (4) through to (9) will be parsed into XML.
iii. Then, parse (5).
iv. Go to step 4.

b. if the object has no highlighted data
i. go to step 3.

3. Parse the data from (1) to (10). Note that as the object has no highlighted data, part (4) to (9) would not be shown in this case.
4. Continue doing this until all JSON elements have been looped through.

Note that:

- for Books / Authors only XML, the procedure is the same. In these 2 cases, part (4) to (9) will always be missing.
- The function needs to know how to retrieve the highlighted data by specifying its key, a.k.a $childKeys in the export. (in this case, the childKeys is "authors").
- The function needs to know what (0) and (4) are, specified in $nestedTags.
- The function needs to know what tags go into (2), (3), (6),(7),(8), through the $attributes parameter. '
- Testing this output is handled in similar fashion.
- There are 2 versions of Authors and Books XML export:
  - api/authors/export/XML/with-books , as shown in the Figure 2
  - api/books/export/XML/with-authors, as shown in the Figure 1
  - For simplicity on the frontend, the one in Figure 1 isn't shown, but is maintained in case it is needed in the future.

```
<?xml version="1.0"?>
<books>
 <data>
  <bookID>22</bookID>
  <title>HALLO GILBERT ONLY 1</title>
  <authors>
   <data>
    <authorID>2</authorID>
    <firstName>Gilbert</firstName>
    <lastName>Vincenta</lastName>
   </data>
  </authors>
 </data>
</books>
```

Figure 1

```
<?xml version="1.0"?>
<authors>
 <data>
  <authorID>2</authorID>
  <firstName>Gilbert</firstName>
  <lastName>Vincenta</lastName>
  <books>
   <data>
    <bookID>22</bookID>
    <title>HALLO GILBERT ONLY 1</title>
   </data>
   </authors>
```

Figure 2