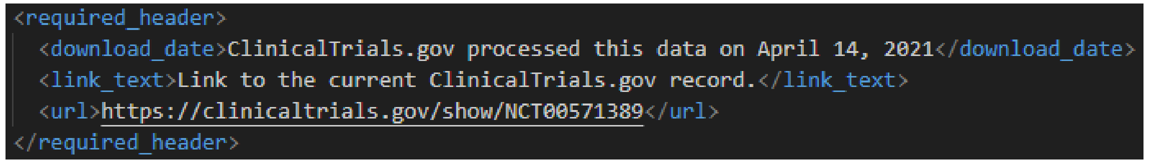
# Differences between v1 and v2

Main changes for v2

- Reworked node function --> will create a list of every child
- New clean_dict function, will clean up the converted dict and increase readability and simplicity of the output

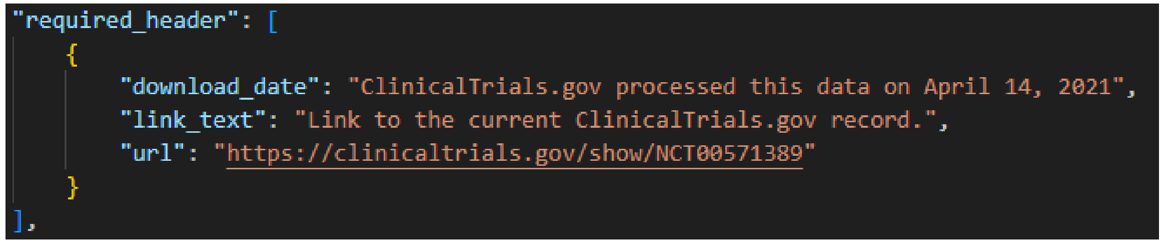
- Json Output

- XML Input



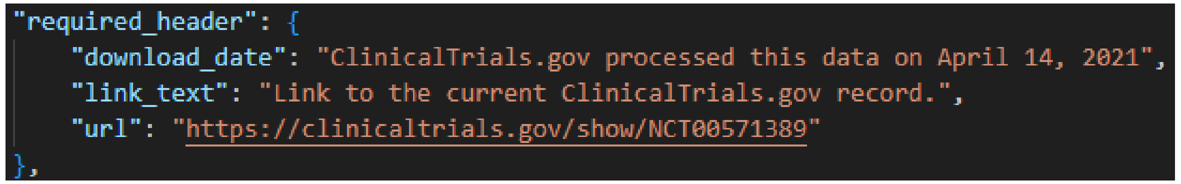
```
<required_header>
  <download_date>ClinicalTrials.gov processed this data on April 14, 2021</download_date>
  <link_text>Link to the current ClinicalTrials.gov record.</link_text>
  <url>https://clinicaltrials.gov/show/NCT00571389</url>
</required_header>
```

v1:



```
"required_header": [
    {
        "download_date": "ClinicalTrials.gov processed this data on April 14, 2021",
        "link_text": "Link to the current ClinicalTrials.gov record.",
        "url": "https://clinicaltrials.gov/show/NCT00571389"
    }
],
```

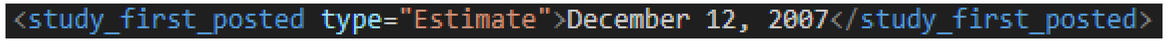
v2:



```
"required_header": {
    "download_date": "ClinicalTrials.gov processed this data on April 14, 2021",
    "link_text": "Link to the current ClinicalTrials.gov record.",
    "url": "https://clinicaltrials.gov/show/NCT00571389"
},
```

v2 will replace the list with the dict of the first element if it only has one element

- XML Input



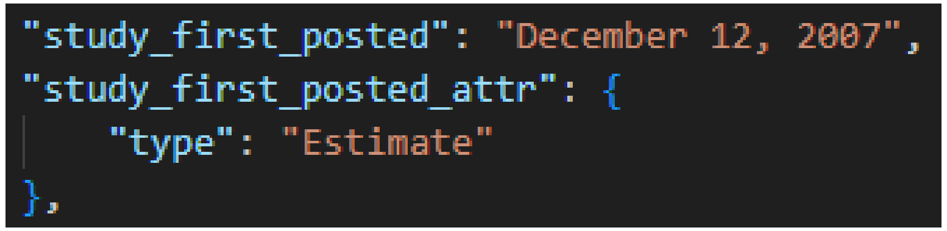
```
<study_first_posted type="Estimate">December 12, 2007</study_first_posted>
```

For each element and child, the attributes will be a sub-node/dict rather than one on the same level

v1 output might be preferable, I find it slightly more readable

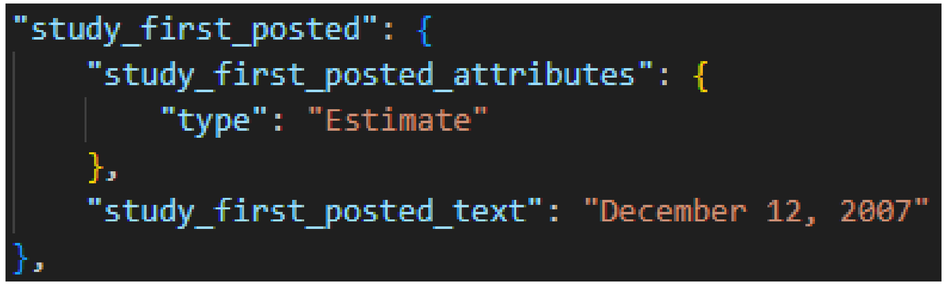
- Json Output

v1:



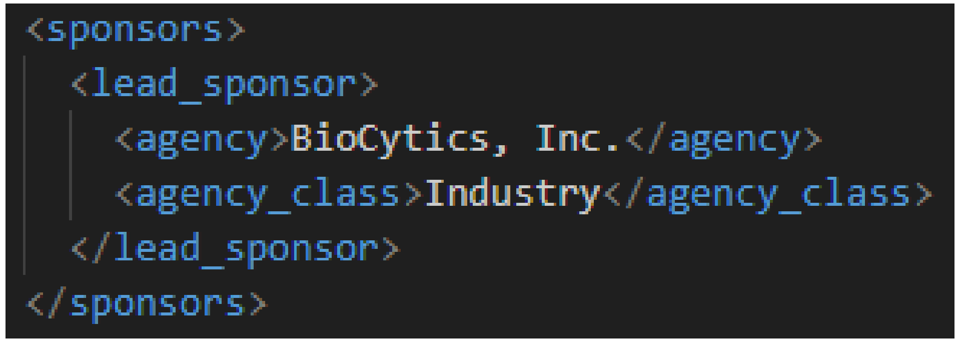
```
"study_first_posted": "December 12, 2007",
"study_first_posted_attr": {
    "type": "Estimate"
},
```

v2:



```
"study_first_posted": {
    "study_first_posted_attributes": {
        "type": "Estimate"
    },
    "study_first_posted_text": "December 12, 2007"
},
```

- XML Input

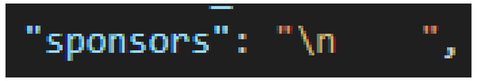


```xml
<sponsors>
  <lead_sponsor>
    <agency>BioCytics, Inc.</agency>
    <agency_class>Industry</agency_class>
  </lead_sponsor>
</sponsors>
```

v1 bug: would only take the child.text as the output if that child had a length of one. Here, <sponsors> only has a length of one, but the relevant data is not child.text

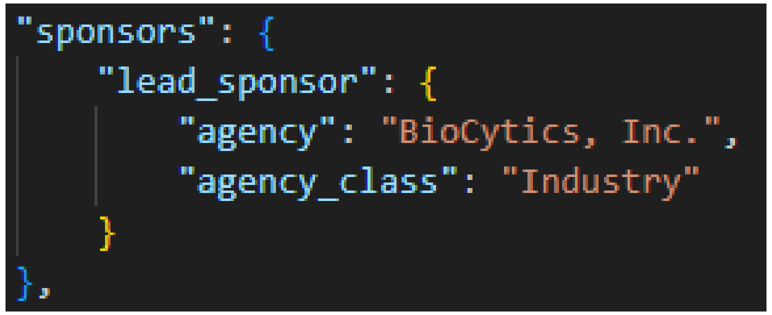
- Json Output

v1:



```json
"sponsors": "\n    ",
```

v2:

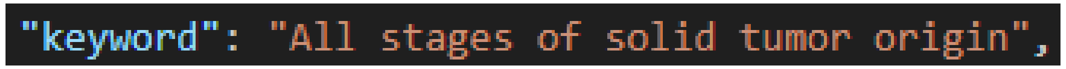


```json
"sponsors": {
    "lead_sponsor": {
        "agency": "BioCytics, Inc.",
        "agency_class": "Industry"
    }
},
```

v2 solves this with more robust code; it will always reference child.text but only if it’s relevant. It will also make the child list for every unique child, not just a child with length greater than 1.

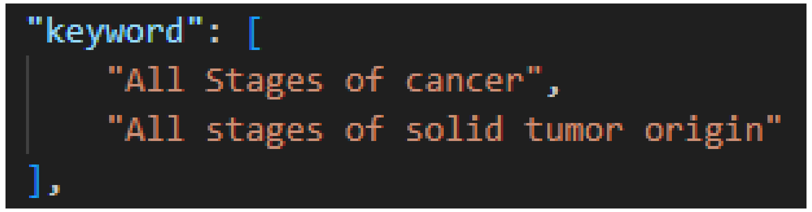
- Json Output

v1:



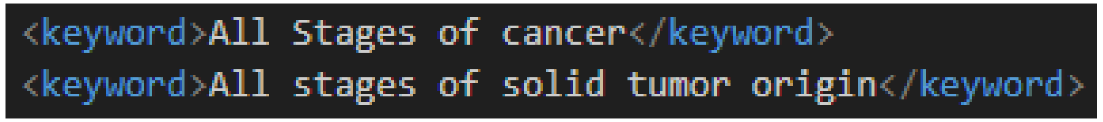
```
"keyword": "All stages of solid tumor origin",
```

v2:



```
"keyword": [
    "All Stages of cancer",
    "All stages of solid tumor origin"
],
```

- XML Input



```
<keyword>All Stages of cancer</keyword>
<keyword>All stages of solid tumor origin</keyword>
```

In v1, it would see that the length is less than or equal to 1, so it would update keyword with new data.

As v2 makes a child list for all children, even length 1, it can get the output of all keywords. Since it cleans itself up, there won't be any lists of data with only one instance.