



A Graphical JSON Query and Conversion Tool

CONTEXT

JSON is a wide-spread data exchange format that is very popular to represent structured data. In particular, it is well supported to serialize and de-serialize objects to JSON in JavaScript and Java, making it easy to persist objects in a human-readable form. If the data stored as JSON is to be further processed by humans, it would be more suitable to transform it to another, record-based format such as Excel spreadsheets.

The research group has a concrete use case where historic and current information on courses is stored in a JSON array. The information is organized in a tree-structure: The file contains an array of semesters; a semester has an array of courses with a title, instructor, etc.; a course has an array of appointments with a time and a room; and so on. This data is supposed to be used in different ways, e.g., get the schedule of lectures in a certain semester or get a list of all course titles over all semesters.

ASSIGNMENT

In this project, a graphical user interface should be developed that allows users to filter and transform JSON files and to export them as Excel spreadsheets. It can be assumed that the JSON document does not contain references, so that the JSON data indeed has a tree-structure and it is straight-forward to generate table-based data from it: Fields simply become columns and if an object contains sub-objects, their fields also become columns.

```
{ "title" : "OOP",
  "basicInfo" : {
    "lvNr" : "LV-12-079-094", "einheit" : "FB12"}}
```

⇒

title	basicInfo.lvNr	basicInfo.einheit
OOP	LV-12-079-094	FB12

For an array, naturally, each element becomes a row. If nested arrays are used one row is created for each element in the nested array and the field values of the outer objects are repeated.

```
[ { "term" : "winter", "year" : "2023",
  "courses" :
    [ { "title" : "OOP",
      "appts" :
        [ { "day" : "monday", "time" : "12:00 - 14:00" },
          { "day" : "wednesday", "time" : "12:00 - 14:00" } ] } ] },
  { "term" : "summer", "year" : "2024",
  "courses" :
    [ ... ] }
]
```

⇒

term	year	courses.title	courses.appts.day	courses.appts.time
winter	2023	OOP	monday	12:00 - 14:00
winter	2023	OOP	wednesday	12:00 - 14:00
summer	2024

It should be possible to configure the conversion to the table-based spreadsheet in several ways. Firstly, the column titles should be editable, and it should be possible to select which columns are to be included. Furthermore, it should be

INFO



Utilities



Java



Apache POI, Jackson, GUI library (e.g., SWT, JavaFX)



Queries (e.g., SQL, XQuery)



Research 
Develop 



Thesis (BSc, MSc), FoPra



1-2 people



Prof. Christoph Bockisch

possible to specify which rows to include, as well. This should be facilitated in terms of queries, e.g., to restrict output to rows where the value of *term* or *title* matches a certain pattern. But queries can also be more complex and be based on the JSON object structure, e.g., *"include courses and all their appointments, when at least one appointment is on a Monday"*.

Furthermore, it should be possible to use queries to re-write data. Simple examples would be simply text replacements (like replacing *"Monday"* with *"Mo"*) or trimming leading and trailing whitespaces.

There are several libraries available to perform queries and transformations on JSON objects. A comprehensive overview can be found here: <https://www.jsonquerytool.com/credits>. Using such a query approach provides a lot of power and flexibility, but also is difficult to use by end-users who are not familiar with the query language. For these reasons, it may be a good idea to internally use an existing JSON query approach, but not to require users to directly define the queries—at least not for the most common use cases. Instead a graphical user interface should be provided to interactively define a query, possible based on the values in the currently loaded JSON object.

The result should be previewed in a table viewer, which can optionally also facilitate to select single rows manually to exclude them from the export to the Excel file. Besides exporting the data to an Excel file, it should also be possible to store the filtered and possibly re-written data as a JSON file again. It should also be possible to store and load query and transformation configurations in order to facilitate applying the same transformation to new data, i.e., new JSON files.

The data on courses and course instances, which is the primary motivation for this assignment can become very large. A JSON file can easily be several a hundred or more megabyte large. Therefore, the tool to be developed must support such file sizes.