Home Task

You are required to implement an HTTP server that manages a spreadsheet with the functionality that is described below. Each function should be represented by an http endpoint. For example
GET http://localhost/sheet/{sheetId} - the endpoint that gets a sheet by id.

1. Write an endpoint for creating a new sheet: The endpoint will receive a schema for the new sheet to be created. The schema will be in JSON format. Here is an example:

```
{
 "columns":[
   {
     "name":"A",
     "type":"boolean"
   },
   {
     "name":"B",
     "type":"int"
   },
   {
     "name":"C",
     "type":"double"
   },
   {
     "name":"D",
     "type":"string"
   }
 ]
}
```

   The endpoint will return the ID of the newly created sheet.

2. Write an endpoint to set a specific cell's value in a specific sheet

   You can only set values to cells of the appropriate type. If the value doesn't match the schema definition for the given column - an error should be thrown

3. Write an endpoint to get a sheet by id

   You can design the sheet data structure as you see fit.
   The structure should represent the cells with the various values

4. Support the lookup function

   Set cell endpoint can also accept a lookup function definition:
   lookupFunction(columnName, rowIndex)
   Here is a pseudo code example:

```
sheetId = createSheet({
 "columns": [
   {"name": "A", "type": "string"},
   {"name": "B", "type": "boolean"},
   {"name": "C", "type": "string"}
 ]
 }
)

setCell(sheetId, "A", 10, "hello")
setCell(sheetId, "B", 11, true)
setCell(sheetId, "C", 1, "lookup(A,10)")

getSheet(sheetId) should return a sheet with the following 3
cells:
("A",10) -> "hello"
("B", 11) ->  true
("C",1) -> "hello"
```

   Please note that type validation should occur when setting the lookupFunction as a value. For example, in the example sheet above, the following would result in an error:
   setCell(sheetId, “B”, 1, “lookup(“A”,10)”) because the type of column “B” (boolean) doesn’t fit the type of lookup(“A”,10) which is the same type as column “A” (string instead of boolean)

   Please also notice that creating a circle of references is not allowed. For example:

   setCell(sheetId, “C”, 1, lookup(“A”,10)) -> works ok
   setCell(sheetId, “A”, 10, lookup(“C”,1)) -> should return an error - cycles are not allowed

   Please also notice that a cycle can be of size 1 or more:

   setCell(sheetId, “C”, 1, lookup(“C”,1)) -> cycle of size 1 - not allowed

   Or

   setCell(sheetId, “C”, 1, lookup(“A”,1)) -> ok
   setCell(sheetId, “A”, 1, lookup(“B”,1)) -> ok
   setCell(sheetId, “B”, 1, lookup(“C”,1)) -> cycle of size 3 - not allowed

Final pointers:

1. Please write clean and clear code that represents your standard of coding and best practices. Make sure the code is structured to packages and modules in a logical way.

2. The code should be fully tested:
   Please write one or more integration tests using an http-client against a running server instance. You can start the server from the test itself or manually start it before running the tests.
   Please write unit-tests as well. These do not require a running server or using an http client. They should test the logic directly.

3. In case something is missing from the requirements please take any working assumptions that you need in order to complete the task. If you do take any assumptions please write them and attach to the solution (can be a readme file in the repo)

4. Please keep in mind that we will want to discuss the implementation details of the proposed solution on a separate frontal interview

When you are done please send us back a link to the repo, and provide read access to the repo for:

- Daniel Coldham (@britishdan)
- Elad Benitah (@eladb2011)
- Anna Itzhaki (@annaitzhaki)

Please also provide instructions on how to start the server as well as how to run the tests (readme in the repo will be great).

Good luck!