SheetFlow Requirements

You are required to implement an HTTP server that manages a spreadsheet with the functionality described below. Each function should be represented by an HTTP endpoint.

Example:
GET http://localhost/sheet/{sheetId} – Retrieves a sheet by its ID.

1. Create a New Sheet Endpoint
This endpoint receives a JSON schema for the new sheet to be created.

Example schema:

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

The endpoint returns the ID of the newly created sheet.

2. Set Cell Value Endpoint
Sets the value of a specific cell in a specific sheet. The value must match the defined type in the column schema; otherwise, an error should be thrown.

3. Get Sheet by ID Endpoint
Fetches a full sheet using its ID. The structure should clearly represent all cell values with their respective types.

4. Support Lookup Function
The set cell endpoint should also support a `lookup(columnName, rowIndex)` function.

Example (pseudo-code):

```
sheetId = createSheet({...})
setCell(sheetId, "A", 10, "hello")
setCell(sheetId, "B", 11, true)
setCell(sheetId, "C", 1, "lookup(A,10)")
```

Result:

("A",10) -> "hello"
("B",11) -> true
("C",1) -> "hello"

Notes:
- Type validation must occur for lookup values.
- Cycles in references (even of size 1) are not allowed.

Examples of invalid cycles:
setCell(sheetId, "C", 1, lookup("C",1)) – cycle of size 1
setCell(sheetId, "C", 1, lookup("A",1))
setCell(sheetId, "A", 1, lookup("B",1))
setCell(sheetId, "B", 1, lookup("C",1)) – cycle of size 3

Final Pointers:
1. Write clean, structured, and modular code following best practices.
2. Include both unit tests (logic testing) and integration tests (using HTTP client).
3. If anything is unclear, make reasonable assumptions and document them.
4. Include instructions to start the server and run the tests in the README.