TRAC DB-Reset- New TABLE Creation

1. Database Setup.
   1. Need to create a TRAC admin user that is different from the postgresql admin account. From psql you can run the following: CREATE USER tracpostgres WITH CREATEDB CREATEROLE ENCRYPTED PASSWORD ‘<PASSWORD>’;
   2. Need to create a dbConfig.json file in the config directory based on the template found there.
   3. Run db-reset with nx run db-reset:run-script to populate the database. The database is filled using the data files found in apps/db-reset/src/assets.
2. Adding Your Module to DB-Reset.
   1. TRAC\apps\db-reset\src\assets Add your excel sheet into the assets folder and name the file that follows the existing files and make sure you import the correct excel file-format. Because TRAC-api throws multer ERROR! while parsing it.

1. \*\*Create Your Module\*\*:

- Create a TypeScript file, e.g., `yourModule.ts`.

- Export a class representing your table structure.

2. \*\*Module Data Mappings\*\*:

- Create `yourModuleDataMappings.ts`.

- Define mappings between Excel columns and table fields.

3. \*\*Import into Postgres TL Main\*\*:

- In `index.ts` of Postgres TL main:

- Import your module and data mappings.

4. \*\*Verify Fields and Functions\*\*:

- Ensure table fields, signature fields, and unique ID functions are correct.

5. \*\*Import into DB-reset/main.ts\*\*:

- In `main.ts` of DB-reset:

- Import your module.

- Import `dirname\_` for assets directory.

6. \*\*Write Code in DB-reset/main.ts\*\*:

- Reset database and populate table using Excel file.

- Replace placeholders with actual module names and file paths.

7. \*\*Run DB-reset Command\*\*:

- Execute `db-reset` in the terminal.

- Database resets and table is populated from Excel file.

Ensure to customize filenames and paths accordingly.