# Documentation for back-end code

Inter IIT Tech Meet 2021

March 24, 2021

## `datamanip.py`:

Contains the **class** `dataCleaner`. It provides methods to read data files using the given instructions, to query `SIMBAD` using the data files, and to export a modified data file with information that is required for the frontend.

## Methods:

1. `__init__`: Instantiates an object of the class `dataCleaner`.

2. `inputInterface`: Interface for inputting all the data required for generating modified data file. Prints a message confirming the storage of inputted data.
   It uses two built-in functions to collect the inputs.
   (a) `addByteInfo` takes and stores the path of the ReadMe file along with information about the line and column location of the data.
   (b) `addFiles` takes and stores the path of our relevant source files.

3. `makeCatalog`: Creates and stores the catalog (stored as a `pandas DataFrame`) of combined low-mass X-ray binary and high-mass X-ray binary source data. Individual DataFrames are constructed first (using `_makeCat`) and then concatenated together. Prints a message if construction of catalog is successful.
   (a) `__byteIndexDict` reads the ReadMe file for the byte-to-byte description of the .dat files and makes a dictionary of column labels and their indices.
   (b) `_makeCat` is a transient function which is used in `makeCatalog`. It constructs a DataFrame for a given data file and dictionary of column labels.

4. `makeObsCatalog`: Creates and stores a DataFrame object of the AstroSat catalog of observation.

5. `makeBibCatalog`: Creates and stores a DataFrame object of the AstroSat publication list.

6. `combCatalog`: Creates a new DataFrame using the existing catalog generated by `makeCatalog` that contains the following additional columns.
   i. **lat** - Declination in degrees. To be used by frontend when creating the plot.
   ii. **lng** - Right Ascension in degrees. To be used by frontend when creating the plot.
   iii. **isObserved** - A Boolean indicating whether the source was observed by AstroSat.
   iv. **isReferred** - A Boolean indicating whether the source was referred to in the publication list.
   v. **references** - A list of bibcodes of the publications that match our source.

vi. **identifiers** - A list of identifiers associated to each source.

The following built-in functions are used.

(a) `coordinatesQuery`: This function queries SIMBAD for data at n-th row of the catalog choosing a suitable radius of query so as to obtain a single result.
The queries are done through the `requests` module.

- `__idQuery` is a transient function used inside `coordinatesQuery` for querying SIMBAD using identifiers.
- `__coordsQuery` is a transient function used inside `coordinatesQuery` for querying SIMBAD using the coordinates of sources.

(b) `getQueryInfo`: Retrieves the list of identifiers and bibcodes for n-th datapoint in the catalog using `coordinatesQuery`.

(c) `filterCatalog`: This function is used to check if the object of interest is observed by AstroSat, and whether any paper in the publication has been referred to.

(d) `__CelestialToGeo`: This function converts Right-Ascension and Declination values to degrees for using them as longitude and latitude respectively.

7. `exportNewCatalog`: Exports the new and final DataFrame in .csv format to the given path for frontend's utility.

## backend.py:

Calls on the `dataCleaner` methods to generate the .csv/JSON file.