

Go - Data Loader

Write a program that meets the following requirements:

1. Command-line Arguments:

- `-source-dir`: The path to the directory to read files from
- `-out-file`: The path to the file to write the final data to
- `-ignore-duplicates`: Whether or not to ignore duplicate keys

2. Logic:

- Parse the source directory and load any file (single level directory, does not need to traverse sub-directories)
 - Assume all files are valid JSON
 - If the source directory is not found, exit with an error:
 - Code: 10
 - Message: Source directory not found
- Load the data for each file found in the source directory, and store in memory
 - If `ignore_duplicates` is true, any duplicate data will be ignored
 - If `ignore_duplicates` is false (default), any duplicate data will raise an exception causing the program to exit with an error
 - Code: 20
 - Message: Duplicate data found
- If no errors are encountered, write the final resulting data as JSON to the `out_file` path which contains all of the loaded data (without duplicates)

Sample JSON Data File:

```
1 {
2     "data" : [
3         {
4             "id" : 100,
5             "first_name" : "John",
6             "last_name" : "Doe"
7         },
8         {
9             "id" : 101,
10            "first_name" : "Jane",
11            "last_name" : "Doe"
12        },
13        {
14            "id" : 102,
```

```
15         "first_name" : "Jackson",
16         "last_name" : "Doe"
17     }
18 ]
19 }
```

Example Execution:

```
1 ### without ignoring handling duplicates
2
3 $ ./data_loader -source-dir /path/to/source/dir -out-file /path/to/outfile.json
4
5
6 ### with ignoring handling duplicates
7
8 $ ./data_loader -source-dir /path/to/source/dir -out-file /path/to/outfile.json -ignore-
  duplicates
```