tars-key-words-reques

October 9, 2023

1 Key words request

This Python script demonstrates the use of the ‘Consultor’ class to retrieve and process data from a CSV file based on command-line arguments.

1.1 Imports:

The script begins by importing the ‘Consultor’ class from the ‘api_io.consultor’ module and the ‘sys’ module for handling command-line arguments.

[1]:
```python
from plugin_io.consultor import Consultor
import sys
```

- Command-Line Arguments: It retrieves a string of column names from the command-line arguments and stores it in the ‘column_names’ variable.
- Creating ‘Consultor’ Instance: An instance of the ‘Consultor’ class is created, providing the path to a CSV file containing data (e.g., “planets_system.csv”).
- Parsing Column Names: The provided column names string is split into a list of column names, which is stored in the ‘column_list’ variable.
- Printing Column Names: The script prints the list of column names to the console.

Retrieving and Printing Data: It attempts to retrieve and print flattened keys for the specified columns using the ‘get_flat_keys’ method of the ‘Consultor’ class. If the specified columns are found in the CSV data, the flattened keys and associated data are printed. Otherwise, it catches a ‘KeyError’ and prints a message indicating that no records were found for the specified columns.

In summary, this script serves as an example of how to use the ‘Consultor’ class to extract and process data from a CSV file based on user-provided column names via command-line arguments. It handles both the parsing of column names and the retrieval of data, gracefully handling situations where specified columns are not found.

[ ]:
```python
# Retrieve the column names from the command-line arguments.
column_names = sys.argv[1]

# Create an instance of the 'Consultor' class and provide the path to a CSV␣
 ↪file containing data (e.g., "planets_system.csv").
consultor_instance = Consultor("plugin_io\\planets_system.csv")

```

```
# Split the provided column names into a list.
column_list = column_names.split(",")

# Print the list of column names.
print(column_list)

# Try to retrieve and print the flattened keys for the specified columns using␣
 ↪the 'get_flat_keys' method.
try:
    flattened_keys = consultor_instance.get_flat_keys(*column_list)
    print(flattened_keys)

# Catch a KeyError if the specified columns are not found.
except KeyError:
    print("No records found for the specified columns.")
```