tars-row-request

October 9, 2023

1 Row request

In summary, this script is designed to retrieve and print data for a specified planet from a CSV file using the 'Consultor' class. It handles the case where the specified planet name is not found in the data by catching and reporting a 'KeyError'.

1.1 Imports:

The script imports the 'Consultor' class from the 'plugin_io.consultor' module and the 'sys' module for handling command-line arguments.

[1]:
```
# Import the 'Consultor' class from the 'api_io.consultor' module and the 'sys'␣
 ↪module for handling command-line arguments.
from plugin_io.consultor import Consultor
import sys
```

- Command-Line Argument: It retrieves the name of a planet from the command-line arguments and stores it in the 'planet_name' 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").
- Retrieving and Printing Data: The script attempts to retrieve and print flattened row data for the specified planet using the 'get_flat_row' method of the 'Consultor' class. If the specified planet name is found in the CSV data, the flattened row data is printed. However, if the planet name is not found, a 'KeyError' is caught, and a message is printed indicating that no records were found for the specified planet.

[ ]:
```
# Retrieve the name of the planet from the command-line arguments.
planet_name = 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")

# Try to retrieve and print the flattened row data for the specified planet␣
 ↪using the 'get_flat_row' method.
try:
    flattened_row_data = consultor_instance.get_flat_row(planet_name)
```

```
    print(flattened_row_data)

# Catch a KeyError if the specified planet name is not found in the data.
except KeyError:
    print("No records found for the specified planet.")
```