Документация

Доступ к выходным наборам данных

Для доступа к данным выходных портов *Выходной набор данных[N]* используются объекты OutputTables[] и OutputTable. Обращение к источнику данных порта происходит через его номер:

OutputTables[N], где N — номер (индекс) порта. Первый порт имеет индекс 0.

Поскольку первый порт присутствует в узле Python по умолчанию, для доступа к его данным существует отдельный объект OutputTable.

Свойства OutputTable

▼ Columns

Columns

Содержит доступную для чтения итерируемую коллекцию столбцов. Реализует протоколы *Mapping* и *Sequence*. Возвращает значение типа `ColumnsClass`. Доступ к элементам может осуществляться через скобочную нотацию [] по именам и по индексам. При установке флага *Разрешить формировать выходные столбцы из кода* элементы коллекции имеют тип `ConfigurableOutputColumnClass`, иначе — `OutputColumnClass`. Оба этих типа унаследованны от `ColumnClass`, и реализуют протокол *Sequence* (см. Полное описание API).

▼ ColumnCount

ColumnCount

Содержит доступное для чтения количество столбцов выходного набора данных. Возвращает значение типа `int`.

▼ RowCount

RowCount

Содержит доступное для чтения количество строк выходного набора данных. Возвращает значение типа `int`.

Методы OutputTable

Get

Get(row, col)

- row — индекс строки. Принимает значение типа `int`.
- col — индекс или имя столбца. Принимает значение типов `int` или `str`.

Метод возвращает значение заданного столбца в заданной строке. Возвращаемое значение может иметь типы: `bool`, `int`, `float`, `str`, `datetime.datetime`, `None`.

IsNull

IsNull(row, col)

- row — индекс строки. Принимает значение типа `int`.
- col — индекс или имя столбца. Принимает значение типов `int` или `str`.

Метод возвращает булево значение `true`, если столбец в заданной строке имеет пропущенное значение. В противном случае возвращается `false`.

GetColumn

GetColumn(col)

- col — индекс или имя столбца. Принимает значение типов `int` или `str`. При установке флага *Разрешить формировать выходные столбцы из кода* возвращается значение типа `ConfigurableOutputColumnClass`, иначе — `OutputColumnClass`. Оба этих типа унаследованны от `ColumnClass`, и реализуют протокол *Sequence* (см. Полное описание API).

Append

Append()

Метод добавляет новую строку в выходной набор данных. Не имеет аргументов.

Set

Set(col, value)

- col — индекс или имя столбца. Принимает значение типов `int` или `str`.
- value — значение. Принимает значения следующих типов: `bool`, `int`, `float`, `str`, `datetime.datetime`, `None`.

Метод задает значение заданного столбца в строке, добавленной методом `Append()`.

AssignColumns

AssignColumns(array)

- array — итерируемый объект, содержащий элементы типа ColumnInfo (см. Полное описание API). Метод создает столбцы выходного набора из коллекции элементов типа ColumnInfo.

AddColumn

AddColumn(ColumnInfo, Name, DisplayName, DataType, DataKind, DefaultUsageType)

Принимает аргументы по ключевым словам:

- ColumnInfo — значение типа ColumnInfo (см. Полное описание API). Необязательный аргумент.
- Name — имя столбца, значение типа `str`. Необязательный аргумент.
- DisplayName — метка столбца, значение типа `str`. Необязательный аргумент.
- DataType — тип данных столбца, значение типа `int` (см. Перечисления). Необязательный аргумент.
- DataKind — виды данных столбца, значение типа `int` (см. Перечисления). Необязательный аргумент.
- DefaultUsageType — назначение по умолчанию столбца, значение типа `int` (см. Перечисления). Необязательный аргумент.

Метод добавляет столбец в конец списка столбцов выходного набора. Возвращает значение типа OutputColumnClass (см. Полное описание API).

InsertColumn

InsertColumn(Index, ColumnInfo, Name, DisplayName, DataType, DataKind, DefaultUsageType)

Принимает аргументы по ключевым словам:

- Index — индекс столбца в коллекции столбцов, принимает значение типа `int`.
- ColumnInfo — значение типа ColumnInfo (см. Полное описание API). Необязательный аргумент.
- Name — имя столбца, значение типа `str`. Необязательный аргумент.
- DisplayName — метка столбца, значение типа `str`. Необязательный аргумент.
- DataType — тип данных столбца, значение типа `int` (см. Перечисления). Необязательный аргумент.
- DataKind — виды данных столбца, значение типа `int` (см. Перечисления). Необязательный аргумент.
- DefaultUsageType — назначение по умолчанию столбца, значение типа `int` (см. Перечисления). Необязательный аргумент.

Метод вставляет столбец по заданному индексу в выходной набор. Возвращает значение типа OutputColumnClass (см. Полное описание API).

DeleteColumn

DeleteColumn(col)

- col — индекс или имя столбца. Принимает значение типов `int` или `str`. Метод удаляет столбец по имени или индексу.

ClearColumns

ClearColumns()

Не имеет аргументов. Метод очищает список столбцов.

Использование модуля builtin_pandas_utils

Если включена опция "Разрешить формировать выходные столбцы из кода", доступны следующие методы (см. Пример №2):

prepare_compatible_table

prepare_compatible_table(table, dataframe, with_index)

Метод задает структуру полей `OutputTable` по `pandas.DataFrame`. Аргументы:

- table — ссылка на выходной набор OutputTableClass;
- dataframe — ссылка на `pandas.DataFrame`, структура которого используется для создания столбцов выходного набора;
- with_index — если аргумент принимает `True`, то индексы `pandas.DataFrame` включаются в структуру выходного набора. Необязательный аргумент. Значение по умолчанию `False`.

fill_table

fill_table(table, dataframe, with_index)

Метод осуществляет запись из `pandas.DataFrame` в `OutputTable`. Аргументы:

- table — ссылка на выходной набор. Принимает значение типа `OutputTableClass`;
- dataframe — ссылка на `pandas.DataFrame`;
- with_index — если аргумент принимает `True`, то индексы `pandas.DataFrame` выгружаются в выходной набор. Необязательный аргумент. Значение по умолчанию `False`.

Примеры

Пример №1

```
from builtin_data import InputTable, OutputTable, OutputTables, DataType,
DataKind, UsageType
import datetime

#Копирование столбцов входного набора
OutputTable.AssignColumns(InputTable.Columns)
#Копирование структуры первой выходной таблицы во вторую выходную таблицу
if len(OutputTables) > 1:
    OutputTables[1].AssignColumns(OutputTables[0].Columns)
else:
    print("Таблица [1] отсутствует (всего таблиц:
{})".format(len(OutputTables)))
#Удаление столбца по индексу
OutputTable.DeleteColumn(0)
#Удаление столбца по имени
OutputTable.DeleteColumn("Test1")
#Удаление всего списка столбцов
OutputTable.ClearColumns()
#Добавление столбца в конец списка столбцов выходного набора
OutputTable.AddColumn(Name="COL0",
                      DisplayName="Дата/Время",
                      DataType=DataType.DateTime,
                      DataKind=DataKind.Continuous,
                      DefaultUsageType=UsageType.Active)
#Вставка столбца по заданному индексу в список столбцов выходного набора
OutputTable.InsertColumn(Index=0,
                         Name="COL1",
                         DisplayName="Признак",
                         DataType=DataType.Boolean)
#Получение ссылки на столбец по имени
COL0 = OutputTable.GetColumn("COL0")
COL1 = OutputTable.GetColumn("COL1")
#Вывод значений свойств столбца
print("Index: ", COL1.Index)
print("Name: ", COL1.Name)
print("DisplayName: ", COL1.DisplayName)
print("DataType: ", COL1.DataType)
```

```
print("DataKind: ", COL1.DataKind)
print("DefaultUsageType: ", COL1.DefaultUsageType)

#Добавление строки в выходной набор данных
OutputTable.Append()
#В поле с индексом 0 записываются текущие Дата/Время
OutputTable.Columns[1].Set(datetime.datetime.now())
#В поле с индексом 1 записывается значение true
OutputTable.GetColumn(0).Set(True)
#Копирование значений первой строки во вторую
OutputTable.Append()
for i in range(OutputTable.ColumnCount):
    value = OutputTable.Get(0, i)
    OutputTable.Set(i, value)

#Проверка, что значение в строке с индексом 0 в столбце с индексом 1 не
определено
print(OutputTable.IsNull(0, 1))
print(OutputTable.Get(0, 1) is None)

print("RowCount = ", OutputTable.RowCount)
#Вывод: RowCount =  2
```

Пример №2

Применение модуля `builtin_pandas_utils`

```
from builtin_data import InputTable, OutputTable,
ConfigurableOutputTableClass
from builtin_pandas_utils import to_data_frame, prepare_compatible_table,
fill_table

#Входной порт необязательный и может не содержать данные
if InputTable:
    #Создать pd.DataFrame по входному набору
    input_frame = to_data_frame(InputTable)
    #Группировка input_frame
    output_frame = input_frame.groupby(["Class"]).sum()
    #Если включена опция "Разрешить формировать выходные столбцы из кода",
    #структуру выходного набора можно подготовить по pd.DataFrame
    assert isinstance(OutputTable, ConfigurableOutputTableClass)
    #Определение структуры выходного набора
    prepare_compatible_table(OutputTable, output_frame, with_index=True)
    #Заполнение выходного набора
    fill_table(OutputTable, output_frame, with_index=True)
```