Документация

Доступ к входным наборам данных

Для доступа к данным портов `Входной источник данных[N]` используется кортеж `InputTables[]`, в котором обращение к источнику данных порта происходит через его номер:

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

Элементы кортежа имеют тип `DataSourceClass` (см. Полное описание API). Поскольку первый порт присутствует в узле *Python* по умолчанию, для доступа к его данным выделен отдельный класс `InputTable`.

Свойства источника данных

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. Элементы коллекции имеют тип `InputColumnClass`, унаследованный от `ColumnClass`, реализующий протокол *Sequence* (см. Полное описание API).

ColumnCount

ColumnCount

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

RowCount

RowCount

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

Методы источника данных

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`.

GetColumn

GetColumn(col)

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

Метод возвращает столбец входного набора. Возвращаемое значение имеет тип `InputColumnClass`, унаследованный от `ColumnClass`, реализующий протокол *Sequence*. (см. Полное описание API).

IsNull

IsNull(row, col)

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

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

Использование модуля builtin_pandas_utils

В модуле builtin_pandas_utils реализован метод `to_data_frame` для создания `pandas.DataFrame` по набору данных входного порта (см. пример ниже).

to_data_frame

to_data_frame(table, useNullableArrays)

- table — ссылка на входной набор (значение типа `DataSourceClass`).
- useNullableArrays — если значение аргумента `True`, то преобразование таблицы в `pandas.DataFrame` происходит без изменения типов данных полей, в которых имеются пропущенные значения. Необязательный аргумент. Значение по умолчанию `False`.

Примеры

```
from builtin_data import InputTable, InputTables
from builtin_pandas_utils import to_data_frame
import numpy as np

inputTable0 = InputTables[0]                    #Источник данных с порта
№1
inputTable1 = InputTables[1]                    #Источник данных с порта
№2
colOutlook0 = inputTable0.Columns["OUTLOOK"]    #Получение ссылки на
столбец по имени
colOutlook1 = inputTable1.Columns[1]            #Получение ссылки на
столбец по индексу

#Вывод значений свойств столбцов
for column in InputTable.Columns:
    print("Index: ", column.Index)
    print("Name: ", column.Name)
    print("DisplayName: ", column.DisplayName)
    print("DataType: ", column.DataType)
    print("DataKind: ", column.DataKind)
    print("UsageType: ", column.UsageType)
    print("DefaultUsageType: ", column.DefaultUsageType)
    print("RowCount: ", column.RowCount, "\n")

#Вывод значений столбца
for index, value in enumerate(colOutlook0):
    print("Индекс {}, значение {}".format(index, value))

#Чтение значений из входной таблицы методом Get
for i in range(InputTable.RowCount):
    #Вывод значений столбца с индексом 0
    print("InputTable.Get({}, 0) = {}".format(i, InputTable.Get(i, 0)))
    #Вывод значений столбца с именем "CLASS"
    print("InputTable.Get({}, 'CLASS') = {}".format(i, InputTable.Get(i,
"CLASS")))

#Получение из столбца "CLASS" массива значений
arrayOfColumnValues = np.array(InputTable.Columns["CLASS"])
print(arrayOfColumnValues)
```

```
#Создание pd.DataFrame по входному набору
data = to_data_frame(InputTable)
print(data)
```