



JS Доступ к входным наборам данных

Для доступа к данным портов `Входной источник данных[N]` используются объекты `InputTables[]` и `InputTable`. Обращение к источнику данных порта происходит через его номер:

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

Поскольку первый порт присутствует в узле *JavaScript* по умолчанию, для доступа к его данным существует отдельный объект `InputTable`.

Свойства источника данных

▼ Columns

Columns

Содержит доступную для чтения коллекцию столбцов. Возвращает объект, реализующий интерфейс `IInputColumns`. Элементы коллекции — объекты, реализующие интерфейс `IInputColumn` (см. [Полное описание API](#)).

▼ ColumnCount

ColumnCount

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

▼ RowCount

RowCount

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

Методы источника данных

▼ Get

Get(row, col)

- `row` — индекс строки. Принимает значение типа `number`.
- `col` — индекс или имя столбца. Принимает значение типов `number` или `string`.

Метод возвращает значение заданного столбца в заданной строке. Возвращаемое значение может иметь типы: `boolean` , `number` , `string` , `Date` , `undefined` .

▼ IsNull

IsNull(row, col)

- `row` — индекс строки. Принимает значение типа `number` .
- `col` — индекс или имя столбца. Принимает значение типа `number` или `string` .

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 `number` или `string` .

Метод возвращает объект столбца, реализующий интерфейс `IIntputColumn` (см. [Полное описание API](#)).

Примеры

```
import { InputTable, InputTables } from "builtIn/Data";

let inputTable0 = InputTables[0],           // Источник данных с
порта №1
    inputTable1 = InputTables[1];           // Источник данных с
порта №2
let colOutlook0 = inputTable0.Columns.OUTLOOK, // Получение ссылки на
столбец по имени
    colDefault0 = inputTable0.GetColumn("DEFAULT");
let colOutlook1 = inputTable1.Columns[0],     // Получение ссылки на
столбец по индексу
    colDefault1 = inputTable1.GetColumn(1);

// Получение из объекта Columns массива столбцов
let arrayOfColumns = Array.from(InputTable.Columns);
// Вывод значений свойств столбцов в консоль
arrayOfColumns.forEach(column => {
    console.log("Index: ", column.Index);
    console.log("Name: ", column.Name);
    console.log("DisplayName: ", column.DisplayName);
    console.log("DataType: ", column.DataType);
    console.log("DataKind: ", column.DataKind);
    console.log("UsageType: ", column.UsageType);
    console.log("DefaultUsageType: ", column.DefaultUsageType);
    console.log("RowCount: ", column.RowCount);
    console.log("");
});

// Получение из столбца "CLASS" массива значений
let arrayOfColumnValues = Array.from(InputTable.Columns["CLASS"]);
// Вывод значений столбца "CLASS"
arrayOfColumnValues.forEach((value, index) => {
    console.log(index, ":", value);
});

// Чтение значений из входной таблицы методом Get
for (let i = 0, c = InputTable.RowCount; i < c; i++) {
    // Вывод значений столбца с индексом 0
    console.log(`InputTable.Get(${i}, 0) = `, InputTable.Get(i, 0));
}
```

```
// Вывод значений столбца с именем "CLASS"
console.log(`InputTable.Get(${i}, "CLASS") = `, InputTable.Get(i,
"CLASS"));
}
```