

# JSCalculation

## Property Summary

|        |   |
|--------|---|
| String | <a href="#">code</a><br>The full source code of this method (including doc and function declaration). |
| Number | <a href="#">variableType</a><br>Get or set the sql type of this variable.                             |

## Method Summary

|         |  |
|---------|--|
| String  | <a href="#">getName()</a><br>This method returns the name of the stored calculation.   |
| UUID    | <a href="#">getUUID()</a><br>Returns the UUID of the calculation.                      |
| Boolean | <a href="#">hasFlag(flag)</a><br>Check a flag of the calculation.                      |
| Boolean | <a href="#">isStored()</a><br>Returns whether this calculation is a stored one or not. |

## Property Details

### code

The full source code of this method (including doc and function declaration).

### Returns

[String](#)

### Sample

```
var calc = solutionModel.getDataSourceNode("db:/example_data/customers").getCalculation("myCalculation");
calc.code = "function myCalculation() { return 123; }";
```

### variableType

Get or set the sql type of this variable.

Type should be one of JSVariable.DATETIME, JSVariable.TEXT, JSVariable.NUMBER, JSVariable.INTEGER or JSVariable.MEDIA.

### Returns

[Number](#)

### Sample

```
var calc = solutionModel.getDataSourceNode("db:/example_data/customers").getCalculation("myCalculation");
calc.variableType = JSVariable.DATETIME;
```

## Method Details

### getName

[String](#) [getName\(\)](#)

This method returns the name of the stored calculation.

### Returns

[String](#) - the name of the stored calculation

**Sample**

```
var calc = solutionModel.newCalculation("function myCalculation() { return 123; }", JSVariable.INTEGER, "db:/example_data/customers");
application.output(calc.getName());
```

**getUUID****UUID** **getUUID ()**

Returns the UUID of the calculation.

**Returns****UUID****Sample**

```
var calc = solutionModel.getDataSourceNode("db:/example_data/customers").newCalculation("function myCalculation() { return 123; }", JSVariable.INTEGER);
application.output(calc.getUUID().toString());
```

**hasFlag****Boolean** **hasFlag (flag)**

Check a flag of the calculation.

The flags are a bit pattern consisting of 1 or more of the following bits:

- JSColumn.UUID\_COLUMN

**Parameters**{**Number**} flag**Returns****Boolean** - boolean whether flag is set.**Sample**

```
var datasourceNode = solutionModel.getDataSourceNode('db:/example_data/orders')
var calculation = datasourceNode.getCalculation('mycalculation')
if (calculation.hasFlag(JSColumn.UUID_COLUMN))
{
    // calculation was typed as UUID
}
```

**isStored****Boolean** **isStored ()**

Returns whether this calculation is a stored one or not.

**Returns****Boolean** - true if the calculation is stored, false otherwise**Sample**

```
var calc = solutionModel.getDataSourceNode("db:/example_data/customers").newCalculation("function myCalculation() { return 123; }", JSVariable.INTEGER);
if (calc.isStored()) application.output("The calculation is stored");
else application.output("The calculation is not stored");
```