JSRelationItem

Apr 04, 2024 08:48

Return Types

JSRelationItem

Supported Clients

SmartClient WebClient NGClient

Constants Summary

String	LITERAL_PREFIX	Constant for using literals in solution model in relations.

Property Summary

String	foreignColumnName	The name of the column from the destination table that this relation item is based on.
String	operator	The operator that defines the relationship between the primary dataprovider and the foreign column.
String	primaryDataProviderID	The name of the column from the source table that this relation item is based on.
Object	primaryLiteral	Get the literal.

Methods Summary

String	getComment()	Returns the comment of this component.
UUID	getUUID()	Returns the UUID of this component.

Constants Details

LITERAL_PREFIX

```
Constant for using literals in solution model in relations.
Strings must be passed as quoted value to make a distinction between string '5' and number 5.
```

Returns

String

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
relation.newRelationItem(JSRelationItem.LITERAL_PREFIX + "'hello'",'=', 'mytextfield');
```

Property Details

foreignColumnName

```
The name of the column from the destination table
that this relation item is based on.
```

Returns

String

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
var relation = solutionModel.newRelation('parentToChild', 'db:/example_data/parent_table', 'db:/example_data
/child_table', JSRelation.INNER_JOIN);
var criteria = relation.newRelationItem('parent_table_id', '=', 'child_table_parent_id');
criteria.primaryDataProviderID = 'parent_table_text';
criteria.foreignColumnName = 'child_table_text';
criteria.operator = '<';
```

operator

The operator that defines the relationship between the primary dataprovider and the foreign column.

Returns

String

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
var relation = solutionModel.newRelation('parentToChild', 'db:/example_data/parent_table', 'db:/example_data
/child_table', JSRelation.INNER_JOIN);
var criteria = relation.newRelationItem('parent_table_id', '=', 'child_table_parent_id');
criteria.primaryDataProviderID = 'parent_table_text';
criteria.foreignColumnName = 'child_table_text';
criteria.operator = '<';
```

primaryDataProviderID

The name of the column from the source table that this relation item is based on.

Returns

String

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
var relation = solutionModel.newRelation('parentToChild', 'db:/example_data/parent_table', 'db:/example_data
/child_table', JSRelation.INNER_JOIN);
var criteria = relation.newRelationItem('parent_table_id', '=', 'child_table_parent_id');
criteria.primaryDataProviderID = 'parent_table_text';
criteria.foreignColumnName = 'child_table_text';
criteria.operator = '<';
```

primaryLiteral

Get the literal.

Returns

Object

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
var relation = solutionModel.newRelation('parentToChild', 'db:/example_data/parent_table', 'db:/example_data
/child_table', JSRelation.INNER_JOIN);
var criteria = relation.newRelationItem(JSRelationItem.LITERAL_PREFIX + "'hello'",'=', 'myTextField');
criteria.primaryLiteral = 'literal_text';
//criteria.primaryLiteral = number;
var primaryLiteral = criteria.primaryLiteral;
```

Methods Details

getComment()

Returns the comment of this component.

Returns

String

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
var comment = solutionModel.getForm("my_form").getButton("my_button").getComment();
application.output(comment);
```

getUUID()

Returns the UUID of this component.

Returns

UUID

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
var button_uuid = solutionModel.getForm("my_form").getButton("my_button").getUUID();
application.output(button_uuid.toString());
```