

JSRecord

Property Summery

[ServoyException](#) [#exception](#)
Returns last occurred exception on this record (or null).

[JSFoundset](#) [#foundset](#)
Returns parent foundset of the record.

Method Summery

[JSDataSet](#) [#getChangedData\(\)](#)
Returns a JSDataSet with outstanding (not saved) changed data of this record.

[Object\[\]](#) [#getPKs\(\)](#)
Returns an array with the primary key values of the record.

[Boolean](#) [#isEditing\(\)](#)
Returns true or false if the record is being edited or not.

[void](#) [#rollbackChanges\(\)](#)
If this record exists in underlying datasource it will do a re-query to fetch the latest data from the datasource.

Property Details

[exception](#)
Returns last occurred exception on this record (or null).

Returns

[ServoyException](#) – The occurred exception.

Sample

```
var exception = record.exception;
```

[foundset](#)
Returns parent foundset of the record.

Returns

[JSFoundset](#) – The parent foundset of the record.

Sample

```
var parent = record.foundset;
```

Method Details

[getChangedData](#)
[JSDataSet](#) **getChangedData()**
Returns a JSDataSet with outstanding (not saved) changed data of this record.

NOTE: To return an array of records with outstanding changed data, see the function `databaseManager.getEditedRecords()`.

Returns

[JSDataSet](#) – a JSDataSet with the changed data of this record.

Sample

```
var dataset = record.getChangedData()  
for( var i = 1 ; i <= dataset.getMaxRowIndex() ; i++ )  
{  
    application.output(dataset.getValue(i,1) + ' ' + dataset.getValue(i,2) + ' ' + dataset.getValue(i,3));  
}
```

[getPKs](#)

[Object\[\]](#) **getPKs()**

Returns an array with the primary key values of the record.

Returns

[Object\[\]](#) – an Array with the pk values.

Sample

```
var pks = foundset.getSelectedRecord().getPKs() // also foundset.getRecord can be used
```

isEditing

[Boolean](#) **isEditing()**

Returns true or false if the record is being edited or not.

Returns

[Boolean](#) – a boolean when in edit.

Sample

```
var isEditing = foundset.getSelectedRecord().isEditing() // also foundset.getRecord can be used
```

rollbackChanges

void **rollbackChanges()**

If this record exists in underlying datasource it will do a re-query to fetch the latest data from the datasource.

NOTE: If you use transactions then it will be the data of your last update of this record in the transaction, not the latest committed data of that record in the datasource.

Returns

void

Sample

```
var record= forms.customer.foundset.getSelectedRecord();  
record.rollbackChanges();
```