

# QBJoins

## Property Summary

<a href="#">QBTableClause</a>	<a href="#">parent</a> Get query builder parent table clause, this may be a query or a join clause.
<a href="#">QBSelect</a>	<a href="#">root</a> Get query builder parent.

## Method Summary

<a href="#">QBJoin</a>	<a href="#">add(dataSource)</a> Add a join with join type IQueryBuilderJoin#LEFT_OUTER_JOIN and no alias for the joining table.
<a href="#">QBJoin</a>	<a href="#">add(dataSource, joinType)</a> Add a join with no alias for the joining table.
<a href="#">QBJoin</a>	<a href="#">add(dataSource, joinType, alias)</a> Add a join clause from the parent query builder part to the specified data source.
<a href="#">QBJoin</a>	<a href="#">add(dataSourceOrRelation, alias)</a> Add a join based on relation or add a manual join.

## Property Details

### parent

Get query builder parent table clause, this may be a query or a join clause.

### Returns

[QBTableClause](#)

### Sample

```
/** @type {QBSelect<db:/example_data/person>} */
var query = databaseManager.createSelect('db:/example_data/person')
query.where.add(query.joins.person_to_parent.joins.person_to_parent.columns.name.eq('john'))
foundset.loadRecords(query)
```

### root

Get query builder parent.

### Returns

[QBSelect](#)

### Sample

```
/** @type {QBSelect<db:/example_data/order_details>} */
var subquery = databaseManager.createSelect('db:/example_data/order_details')

/** @type {QBSelect<db:/example_data/orders>} */
var query = databaseManager.createSelect('db:/example_data/orders')
query.where.add(query
    .or
        .add(query.columns.order_id.not.isIn([1, 2, 3]))
        .add(query.exists(
            subquery.where.add(subquery.columns.orderid.eq(query.columns
order_id)).root
        ))
    )
foundset.loadRecords(query)
```

## Method Details

### add

**QBJoin add** (dataSource)

Add a join with join type IQueryBuilderJoin#LEFT\_OUTER\_JOIN and no alias for the joining table.

#### Parameters

{String} dataSource - data source

#### Returns

QBJoin

#### Sample

```
/** @type {QBSelect<db:/example_data/orders>} */
var query = databaseManager.createSelect('db:/example_data/orders')
/** @type {QBJoin<db:/example_data/order_details>} */
var join = query.joins.add('db:/example_data/order_details', JSRelation.INNER_JOIN, 'odetail')
join.on.add(join.columns.orderid.eq(query.columns.orderid))
// to add a join based on a relation, use the relation name
var join2 = query.joins.add('orders_to_customers', 'cust')
query.where.add(join2.columns.customerid.eq(999))
foundset.loadRecords(query)
```

### add

**QBJoin add** (dataSource, joinType)

Add a join with no alias for the joining table.

#### Parameters

{String} dataSource - data source

{Number} joinType - join type, one of JSRelation.LEFT\_OUTER\_JOIN, JSRelation.INNER\_JOIN, JSRelation.RIGHT\_OUTER\_JOIN, JSRelation.FULL\_JOIN

#### Returns

QBJoin

#### Sample

```
/** @type {QBSelect<db:/example_data/orders>} */
var query = databaseManager.createSelect('db:/example_data/orders')
/** @type {QBJoin<db:/example_data/order_details>} */
var join = query.joins.add('db:/example_data/order_details', JSRelation.INNER_JOIN, 'odetail')
join.on.add(join.columns.orderid.eq(query.columns.orderid))
// to add a join based on a relation, use the relation name
var join2 = query.joins.add('orders_to_customers', 'cust')
query.where.add(join2.columns.customerid.eq(999))
foundset.loadRecords(query)
```

### add

**QBJoin add** (dataSource, joinType, alias)

Add a join clause from the parent query builder part to the specified data source.

#### Parameters

{String} dataSource - data source

{Number} joinType - join type, one of IQueryBuilderJoin#LEFT\_OUTER\_JOIN, IQueryBuilderJoin#INNER\_JOIN, IQueryBuilderJoin#RIGHT\_OUTER\_JOIN, IQueryBuilderJoin#FULL\_JOIN

{String} alias - the alias for joining table

#### Returns

QBJoin

**Sample**

```

/** @type {QBSelect<db:/example_data/orders>} */
var query = databaseManager.createSelect('db:/example_data/orders')
/** @type {QBJoin<db:/example_data/order_details>} */
var join = query.joins.add('db:/example_data/order_details', JSRelation.INNER_JOIN, 'odetail')
join.on.add(join.columns.orderid.eq(query.columns.orderid))
// to add a join based on a relation, use the relation name
var join2 = query.joins.add('orders_to_customers', 'cust')
query.where.add(join2.columns.customerid.eq(999))
foundset.loadRecords(query)

```

**add**

**QBJoin add** (dataSourceOrRelation, alias)

Add a join based on relation or add a manual join.

When dataSourceOrRelation is a relation name, a join will be added based on the relation.

When dataSourceOrRelation is a data source, an empty join will be added with join type IQueryBuilderJoin#LEFT\_OUTER\_JOIN.

**Parameters**

{String} dataSourceOrRelation - data source

{String} alias - the alias for joining table

**Returns**

[QBJoin](#)

**Sample**

```

/** @type {QBSelect<db:/example_data/orders>} */
var query = databaseManager.createSelect('db:/example_data/orders')
/** @type {QBJoin<db:/example_data/order_details>} */
var join = query.joins.add('db:/example_data/order_details', JSRelation.INNER_JOIN, 'odetail')
join.on.add(join.columns.orderid.eq(query.columns.orderid))
// to add a join based on a relation, use the relation name
var join2 = query.joins.add('orders_to_customers', 'cust')
query.where.add(join2.columns.customerid.eq(999))
foundset.loadRecords(query)

```