

# pjs.rowCount()

This API returns the number of database rows returned from the most recent fetch from the passed SELECT statement handle.

The API can also be called in the form of a method attached to a statement object as follows: `stmt.rowCount()`.

## Parameters

1. Statement handle returned from `pjs.prepare()` or `pjs.allocStmt()`. The statement handle must reference a SELECT statement that has been successfully executed using either `pjs.execute()` or `pjs.executeDirect()` and fetched from at least one time using `pjs.fetch()` or one of its variants. This parameter is omitted if the API is called as a statement method.

## Return Value

The number of rows returned. Zero is returned if the statement handle does not reference a SELECT statement, or if the statement has not been successfully executed and fetched from.

## Example

### Select 100 rows from SYSTABLES

```
pjs.define("systables", { type: "data structure", qualified: true, dim: 100, elements: {
  "table_name": { type: "char", varying: true, length: 128},
  "table_text": { type: "char", varying: true, length: 50 }
}});

var stmt = pjs.allocStmt();
stmt.executeDirect("select table_name, table_text from qsys2/systables");
pjs.fetch(stmt, systables, 100);
for (var idx = 1; idx <= stmt.rowCount(); idx++) {
  console.log("(%d) %s %s", idx, systables[idx].table_name, systables[idx].table_text);
}
stmt.close();
```

## Requirements

When using an IBM i database, this API requires the [Profound.js Connector](#) module.