

pjs.data.update()

This API is used to update records of a database object.

Parameters

1. Database Connection Definition:
(Object / Optional) - A database connection definition object returned from `pjs.getDB()`. If not passed, the default database connection will be used.
2. Database Object:
(String) Database object name. The name can be qualified with a library/schema. If not qualified the `pathlist` will be used to resolve the object.
--or--
(String) DbDefn file name (with the extension). The file can be qualified with a directory. If not qualified the `pathlist` will be used to resolve the object.
--or--
(Object) DbDefn instance.
3. Filter:
Single condition - created using `pjs.data.createCondition` API
--or--
Array of conditions - created using `pjs.data.createCondition` API
--or--
(String) – SQL syntax, because `pjs.data.createCondition` does support every possible way of conditioning
4. Data: (object) - the object that contains the column names and values to be updated.
(Note, this object can contain more, or less, properties than the actual table. It only uses the matched names.

Exception Handling

An Error instance will be thrown along with these additional properties:

- `sqlstate` - The error state.
- `sqlcode` - The error code.
- `sqlMessage` - The message text.

Example

```
var newValues = {
  firstname: fname,
  dob: null
}

try {
  // This will result in only the firstname and dob columns being updated.
  let filter = pjs.data.createCondition("customerNumber", "=", "1");
  let result = pjs.data.update("customers", filter, newValues);
  console.log(result);
} catch(error) {
  console.error(error);
}
```