Db 1.0

May 1, 2024

Contents

Db

The `Db` module has functionality for connecting to databases. It currently supports `sqlite`, `mssql`, `msaccess`, `oracle` and `postgresql` databases.

Connect

The `connect` method initialises a connection to a given database and returns a Database object.

Parameters

- `type` the type of the database, currently this should be “mssql”, “sqlite”, “msaccess”, “oracle” or “postgresql”.
- `connection` the connection-string which contains information about how to connect to the database in question

Example

```
1 var db = Db.connect('sqlite', 'Data Source=C:\\MyFolder\\Test.db;
    Version=3;');
```

Database

The database object returned from a `Db.connect(...)` invocation represents a database connection. It has two primary methods for interacting with a database; `query` and `exec`.

Exec

The `exec` method will execute a non-query (e.g. `INSERT`, `UPDATE`) and return the number of affected rows.

Example

```
var affectedRows = db.exec('CREATE TABLE Test (id int, name string)');
```

Also supports db parameters:

```
Db.exec(
  "INSERT INTO Mammals (name, species) VALUES (@name, @species)",
  { "@name": "John", "@species": "Seacow" }
);
```

The arguments in the 2nd argument **must** be prefixed with "@".

Query

The `query` method is used for queries (e.g. `SELECT` etc) and returns an array of objects representing the result of the query.

Example

```
var rows = db.query('SELECT id, name from Test');
for (var i=0; i<rows.length; i++) {
  Debug.showDialog("id="+rows[i].id+", name="+rows[i].name);
}
```

Begin

The `begin()` method is used to initiate a transaction.

Example

```
var tx = db.begin();
```

Close

You can use the `close()` method to close the database connection.

```
db.close();
```

Transaction

A `transaction` object is conceptually similar to the database object. It has the same `query` and `exec` methods, but will delay the execution of the query or command until `commit()` is invoked and of course maintains transactional integrity. If the `rollback()` method is invoked the `query` and `exec` operations already made are discarded.

Commit

A `commit()` invocation will commit the tx to the db.

Example

```
tx.exec("INSERT INTO Test (id, name) VALUES (1, 'John')");
tx.exec("INSERT INTO Test (id, name) VALUES (2, 'Jane')");
// Commit John and Jane
tx.commit();
```

Rollback

A `rollback()` invocation will rollback the tx.

Example

“‘javascript tx.exec(“INSERT INTO Test (id, name) VALUES (1, ‘John’)”); tx.exec(“INSERT INTO Test (id, name) VALUES (2, ‘Jane’)”); // John and Jane are not needed anyways tx.rollback();