

ClickBlocks\DB\Reader

General information

Inheritance	no
Child classes	no
Interfaces	Countable, Iterator
Source	Framework/db/core/reader.php

The class allows to iterate rows from a query result set. Since **ClickBlocks\DB\Reader** implements **Iterator** interface you can get the rows of the fetched data by using foreach:

```
foreach ($reader as $row)
    // $row represents a row of the fetched data
```

You can also set a specific mode of data fetching for more flexible iterative process. To retrieve an instance of the class you can use method **query** of class **ClickBlocks\DB\DB**:

```
foreach ($db->query('SELECT * FROM MyTable') as $row)
    // $row represents a row of the fetched data
```

Public non-static methods

__construct()

```
public void __construct(PDOStatement $statement, integer $mode = PDO::FETCH_ASSOC, mixed $arg = null, array $ctorargs = null)
```

\$statement	PDOStatement	represents a prepared SQL statement and, after the statement is executed, an associated result set.
\$mode	integer	fetch mode for this SQL statement.
\$arg	mixed	this argument have a different meaning depending on the value of the \$mode parameter.
\$ctorargs	array	arguments of custom class constructor when the \$mode parameter is <code>PDO::FETCH_CLASS</code> .

The class constructor. Sets the specific mode of data fetching.

getStatement()

```
public PDOStatement getStatement()
```

Returns SQL statement for more low-level operating.

setFetchMode()

```
public void setFetchMode(integer $mode = PDO::FETCH_ASSOC, mixed $arg = null, array $ctorargs = null)
```

\$mode	integer	the fetch mode should be one of the <code>PDO::FETCH_*</code> constants.
\$arg	mixed	this argument have a different meaning depending on the value of the \$mode parameter.
\$ctorargs	array	arguments of custom class constructor when the \$mode parameter is <code>PDO::FETCH_CLASS</code> .

Set the default fetch mode for the current SQL statement.

count()

```
public integer count()
```

Returns the number of rows in the result set.

rewind()

```
public void rewind()
```

Resets the iterator to the initial state.

key()

```
public integer key()
```

Returns the index of the current row.

next()

```
public void next()
```

Moves the internal pointer to the next row.

current()

```
public mixed current()
```

Returns the current row. If the internal pointer points beyond the end of the row set or the row set is empty, the method returns FALSE

valid()

```
public boolean valid()
```

Returns whether there is a row of data at the current position.

Protected non-static properties

st

```
protected PDOStatement $st = null
```

An instance of **PDOStatement** that represents a prepared SQL statement and an associated result set.

row

```
protected array $row = null
```

The current row of the fetched data.

index

```
protected integer $index = null
```

The index of the current row.