

LIBPQ wrapper c++

Generated by Doxygen 1.8.11



# Contents

<b>1</b>	<b>Class Index</b>	<b>1</b>
1.1	Class List	1
<b>2</b>	<b>Class Documentation</b>	<b>3</b>
2.1	libpqwrp::libPQwrp Class Reference	3
2.1.1	Detailed Description	5
2.1.2	Member Function Documentation	5
2.1.2.1	begin()	5
2.1.2.2	cfield(int nfield)	5
2.1.2.3	clearRes()	6
2.1.2.4	closeDB()	6
2.1.2.5	commit()	6
2.1.2.6	connectDB(std::string info)	6
2.1.2.7	count_format_specifiers(std::string const &format)	6
2.1.2.8	countfield()	6
2.1.2.9	countrow()	7
2.1.2.10	DoubleToChar(double _X_, unsigned _precision_=0)	7
2.1.2.11	end()	7
2.1.2.12	fechsql(std::string cursor=""mycursor"")	7
2.1.2.13	fetch(int row, int column)	7
2.1.2.14	fetchAll(std::string, std::string cursor=""mycursor"")	7
2.1.2.15	is_Table(const char *table)	8
2.1.2.16	nfield(std::string field)	8
2.1.2.17	opensql(std::string sql, std::string cursor=""mycursor"")	8
2.1.2.18	prepare(std::string const &format, Head &&head, Args &&...args)	8
2.1.2.19	qexec(std::string sql)	9
2.1.2.20	query(std::string sql, bool binary=true)	9
2.1.2.21	rollback()	9
2.1.2.22	savpoint()	9
2.1.2.23	savpointRelease()	9
2.1.2.24	savpointRollback()	9
2.1.3	Friends And Related Function Documentation	9
2.1.3.1	operator>>	9



# Chapter 1

## Class Index

### 1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

<a href="#">libpqwrp::libPQwrp</a>	
Gestion parametre sql . . . . .	<a href="#">3</a>



## Chapter 2

# Class Documentation

### 2.1 libpqwrp::libPQwrp Class Reference

gestion parametre sql

```
#include <libpqwrp.h>
```

#### Public Member Functions

- void [connectDB](#) (std::string info)  
*connect to the database*
- void [qexec](#) (std::string sql)  
*connect to the database*
- void [query](#) (std::string sql, bool binary=true)  
*PQexec.*
- char \* [fetch](#) (int row, int column)  
*PQquery.*
- double [fetchDbf](#) (int row, int column)  
*get the value of the row and the column*
- int [fetchInt](#) (int row, int column)  
*get the value of the row and the column*
- int [nfield](#) (std::string field)  
*get the value of the row and the column*
- const char \* [cfield](#) (int [nfield](#))  
*number column of the field*
- int [countfield](#) ()  
*name of the field*
- int [countrow](#) ()  
*number of columns*
- bool [is\\_Table](#) (const char \*table)  
*number of rows*
- void [closeDB](#) ()  
*if exist table of the database*
- void [begin](#) ()  
*close to the database*

- void `commit` ()  
*start transaction for commit*
- void `rollback` ()  
*commit for transaction init*
- void `end` ()  
*rollback for transaction init*
- void `clearRes` ()  
*end transaction for init*
- void `savpoint` ()  
*clear PGresult*
- void `savpointRollback` ()  
*point de sauvegarde*
- void `savpointRelease` ()  
*rollback base savepoint*
- void `fetchAll` (std::string, std::string cursor="mycursor")  
*delete savepoint*
- void `openSql` (std::string sql, std::string cursor="mycursor")  
*fetch ALL include requete*
- void `fetchSql` (std::string cursor="mycursor")  
*query for fetchsql record / record*
- void `prepare` (std::string const &format)  
*fetch record use openSQL*
- template<typename Head , typename... Args>  
std::string `prepare` (std::string const &format, Head &&head, Args &&...args)  
*format without parameters causes an error*
- std::stringstream `result` ()  
*retrieve values on the fields*
- char \* `DoubleToChar` (double \_X\_, unsigned \_precision\_=0)  
*out buffer;*

## Public Attributes

- bool `fetch0`  
*count the number of parameters in the format*
- bool `update0`  
*pas / plus d'eng en lecture*
- bool `select0`  
*pas / plus d'eng en update*
- bool `delete0`  
*pas / plus d'eng select*
- bool `insert0`  
*pas / plus d'eng delete*
- bool `fetchEOF`  
*pas / plus d'eng insert*
- int `rows`  
*fin de traitement get records*
- int `cols`  
*initialise par les requetes*
- int `rown`  
*initialise par les requetes*
- int `coln`  
*variable de traitement*
- PGresult \* `res`  
*variable de traitement*



## Protected Member Functions

- unsigned `count_format_specifiers` (std::string const &format)  
*contrôle PGresult*

## Private Attributes

- PGconn \* **conn**
- std::stringstream **sqlstream**
- bool **first** = true
- std::string **fin\_stage**
- bool **clean**

## Static Private Attributes

- static constexpr char **FORMAT\_SPECIFIER** = '?'

## Friends

- `std::istream & operator>> (std::istream &is, char *t)`  
*Double to char.*

### 2.1.1 Detailed Description

## gestion parametre sql

##### prepare sql for-  
matage avec template variadique etc..... #####

Definition at line 95 of file libpqwrp.h.

### 2.1.2 Member Function Documentation

### 2.1.2.1 void libpqwrp::libPQwrp::begin ( )

close to the database

```
start transaction for commit
```

Definition at line 240 of file libpqwrp.cpp.

### 2.1.2.2 `const char * libpqwrp::libPQwrp::cfield ( int nfield )`

number column of the field

name of the field

Definition at line 184 of file libpqwrp.cpp.

**2.1.2.3 void libpqwrp::libPQwrp::clearRes ( )**

end transaction for init

clear PGresult

Definition at line 267 of file libpqwrp.cpp.

**2.1.2.4 void libpqwrp::libPQwrp::closeDB ( )**

if exist table of the database

close to the database

Definition at line 232 of file libpqwrp.cpp.

**2.1.2.5 void libpqwrp::libPQwrp::commit ( )**

start transction for commit

commit for transaction init

Definition at line 246 of file libpqwrp.cpp.

**2.1.2.6 void libpqwrp::libPQwrp::connectDB ( std::string *info* )**

connect to the database

init PGconn

init PGresult

Definition at line 74 of file libpqwrp.cpp.

**2.1.2.7 unsigned libpqwrp::libPQwrp::count\_format\_specifiers ( std::string const & *format* )** *[protected]*

contrôle PGresult

process

Definition at line 14 of file libpqwrp.cpp.

**2.1.2.8 int libpqwrp::libPQwrp::countfield ( )**

name of the field

number of columns

Definition at line 196 of file libpqwrp.cpp.

### 2.1.2.9 int libpqwrp::libPQwrp::countrow ( )

number of columns

number of rows

Definition at line 203 of file libpqwrp.cpp.

### 2.1.2.10 char \* libpqwrp::libPQwrp::DoubleToChar ( double \_X\_, unsigned \_precision\_ = 0 )

out buffer;

Double to char.

Definition at line 421 of file libpqwrp.cpp.

### 2.1.2.11 void libpqwrp::libPQwrp::end ( )

rollback for transaction init

end transaction for init

Definition at line 260 of file libpqwrp.cpp.

### 2.1.2.12 void libpqwrp::libPQwrp::fechsql ( std::string *cursor* = "mycursor" )

query for fetchsql record / record

fetch record use openSQL read a line record only

end of row or without row

close cursor

no end of row

Definition at line 392 of file libpqwrp.cpp.

### 2.1.2.13 char \* libpqwrp::libPQwrp::fetch ( int *row*, int *column* )

PQquery.

get the value of the row and the column

Definition at line 132 of file libpqwrp.cpp.

### 2.1.2.14 void libpqwrp::libPQwrp::fetchAll ( std::string *sql*, std::string *cursor* = "mycursor" )

delete savepoint

fetch ALL include requete

---

process complex

lecture full rc ??? memory

end of row or without row

close cursor

no end of row

Definition at line 300 of file libpqwrp.cpp.

#### 2.1.2.15 `bool libpqwrp::libPQwrp::is_Table ( const char * table )`

number of rows

if exist table of the database false for return value count

Definition at line 209 of file libpqwrp.cpp.

#### 2.1.2.16 `int libpqwrp::libPQwrp::nfield ( std::string field )`

get the value of the row and the column

number column of the field

Definition at line 171 of file libpqwrp.cpp.

#### 2.1.2.17 `void libpqwrp::libPQwrp::opensql ( std::string sql, std::string cursor = "mycursor" )`

fetch ALL include requete

query for fetchsql record / record read a line record only

end of row or without row

close cursor

no end of row

Definition at line 350 of file libpqwrp.cpp.

#### 2.1.2.18 `template<typename Head , typename... Args> std::string libpqwrp::libPQwrp::prepare ( std::string const & format, Head && head, Args &&... args ) [inline]`

format without parameters causes an error

formatting the query with parameters

Definition at line 201 of file libpqwrp.h.

#### 2.1.2.19 void libpqwrp::libPQwrp::qexec ( std::string *sql* )

connect to the database

PQexec.

Definition at line 90 of file libpqwrp.cpp.

#### 2.1.2.20 void libpqwrp::libPQwrp::query ( std::string *sql*, bool *binary* = true )

PQexec.

PQquery.

Definition at line 105 of file libpqwrp.cpp.

#### 2.1.2.21 void libpqwrp::libPQwrp::rollback ( )

commit for transaction init

rollbck for transaction init

Definition at line 253 of file libpqwrp.cpp.

#### 2.1.2.22 void libpqwrp::libPQwrp::savpoint ( )

clear PGresult

point de sauvegarde

Definition at line 273 of file libpqwrp.cpp.

#### 2.1.2.23 void libpqwrp::libPQwrp::savpointRelease ( )

roolback base savepoint

delete savepoint

Definition at line 287 of file libpqwrp.cpp.

#### 2.1.2.24 void libpqwrp::libPQwrp::savpointRollback ( )

point de sauvegarde

roolback base savepoint

Definition at line 280 of file libpqwrp.cpp.

### 2.1.3 Friends And Related Function Documentation

#### 2.1.3.1 std::istream& operator>> ( std::istream & *is*, char \* *t* ) [friend]

Double to char.

retrieve char\* for stringstream dummy

Definition at line 241 of file libpqwrp.h.

The documentation for this class was generated from the following files:

- src/libpqwrp.h
- src/libpqwrp.cpp



# Index

- begin
  - [libpqwrp::libPQwrp, 5](#)
- cfield
  - [libpqwrp::libPQwrp, 5](#)
- clearRes
  - [libpqwrp::libPQwrp, 5](#)
- closeDB
  - [libpqwrp::libPQwrp, 6](#)
- commit
  - [libpqwrp::libPQwrp, 6](#)
- connectDB
  - [libpqwrp::libPQwrp, 6](#)
- count\_format\_specifiers
  - [libpqwrp::libPQwrp, 6](#)
- countfield
  - [libpqwrp::libPQwrp, 6](#)
- countrow
  - [libpqwrp::libPQwrp, 6](#)
- DoubleToChar
  - [libpqwrp::libPQwrp, 7](#)
- end
  - [libpqwrp::libPQwrp, 7](#)
- fechsql
  - [libpqwrp::libPQwrp, 7](#)
- fetch
  - [libpqwrp::libPQwrp, 7](#)
- fetchAll
  - [libpqwrp::libPQwrp, 7](#)
- is\_Table
  - [libpqwrp::libPQwrp, 8](#)
- libpqwrp::libPQwrp, 3
  - [begin, 5](#)
  - [cfield, 5](#)
  - [clearRes, 5](#)
  - [closeDB, 6](#)
  - [commit, 6](#)
  - [connectDB, 6](#)
  - [count\\_format\\_specifiers, 6](#)
  - [countfield, 6](#)
  - [countrow, 6](#)
  - [DoubleToChar, 7](#)
  - [end, 7](#)
  - [fechsql, 7](#)
  - [fetch, 7](#)
  - [fetchAll, 7](#)
  - [is\\_Table, 8](#)
  - [nfield, 8](#)
  - [openssl, 8](#)
  - [operator>>, 9](#)
  - [prepare, 8](#)
  - [qexec, 8](#)
  - [query, 9](#)
  - [rollback, 9](#)
  - [savpoint, 9](#)
  - [savpointRelease, 9](#)
  - [savpointRollback, 9](#)
- nfield
  - [libpqwrp::libPQwrp, 8](#)
- openssl
  - [libpqwrp::libPQwrp, 8](#)
- operator>>
  - [libpqwrp::libPQwrp, 9](#)
- prepare
  - [libpqwrp::libPQwrp, 8](#)
- qexec
  - [libpqwrp::libPQwrp, 8](#)
- query
  - [libpqwrp::libPQwrp, 9](#)
- rollback
  - [libpqwrp::libPQwrp, 9](#)
- savpoint
  - [libpqwrp::libPQwrp, 9](#)
- savpointRelease
  - [libpqwrp::libPQwrp, 9](#)
- savpointRollback
  - [libpqwrp::libPQwrp, 9](#)