

Qualité, conception, modélisation

Accéder aux données - C

JY Martin

Novembre 2023

Plan

1 Mécanisme général

2 LibPQxx

API spécifiques

Tres majoritairement, les connexions à des bases de données en C / C++ se font via des API spécifiques.

- mysql : Connector/C++ (<https://dev.mysql.com/doc/connector-cpp/8.0/en/connector-cpp-introduction.html>)
- postgresQL : libpqxx (<http://pqxx.org/>)
- Oracle : OTL (<http://otl.sourceforge.net/>)

ODBC - La solution Microsoft

Une autre solution, fonctionnant uniquement sous Windows, est de passer par un driver ODBC.

- Le driver ODBC est défini dans le système.
- La connexion se fait depuis une bibliothèque de fonction, qui se connecte au driver ODBC

Plan

1 Mécanisme général

2 LibPQxx

Informations

PQxx est une bibliothèque C++ pour se connecter à PostgreSQL.

- Le projet : <http://pqxx.org>
- Site de référence : <http://pqxx.org/development/libpqxx/>
- Documentation : <https://libpqxx.readthedocs.io/en/stable/>

Exemple PQXX

```
#include <iostream >
#include <pqxx/pqxx >
using namespace std;
using namespace pqxx;

int main(int argc, char* argv[]) {
    try {
        connection conn("dbname=mabase user=prweb password=prweb"
                        + " hostaddr=127.0.0.1 port=5432");
        if (! conn.is_open()) {
            cout << "Ouverture de la base impossible" << endl;
            return 1;
        }
        char *sql = "SELECT * FROM Personne";
        nontransaction N(conn);
        result R( N.exec( sql ));
        for (result::const_iterator iter = R.begin(); iter != R.end(); iter++) {
            cout << "ID = " << iter[0].as<int>() << endl;
            cout << "Nom = " << iter[1].as<string>() << endl;
            cout << "Prenom = " << iter[2].as<string>() << endl << endl;
        }
        conn.disconnect ();
    } catch (const std::exception &e) {
        cerr << e.what() << std::endl;
        return 1;
    }
}
```

