

# RocksDB - Snapshot data

Snapshot là một tính năng trong RocksDB cho phép bạn tạo bản sao dữ liệu hiện tại để đọc theo một thời điểm cụ thể. Khi bạn tạo một snapshot, RocksDB đảm bảo rằng dữ liệu trong snapshot không bị ảnh hưởng bởi các thay đổi sau đó. Điều này rất hữu ích khi bạn muốn đọc dữ liệu trong một trạng thái nhất định và tránh bị ảnh hưởng bởi các giao dịch hoặc thay đổi dữ liệu khác.

Dưới đây là một ví dụ về việc sử dụng Snapshot trong RocksDB bằng Java:

```
import org.rocksdb.*;

public class RocksDBSnapshotExample {

    public static void main(String[] args) {
        RocksDB.loadLibrary();
        try (final Options options = new Options().setCreateIfMissing(true);
            final RocksDB db = RocksDB.open(options, "/path/to/database")) {
            try (final Snapshot snapshot = db.getSnapshot()) {
                // Đọc dữ liệu từ snapshot
                try (final ReadOptions readOptions = new ReadOptions().setSnapshot(snapshot))
                {
                    byte[] key = "myKey".getBytes();
                    byte[] value = db.get(readOptions, key);
                    if (value != null) {
                        System.out.println("Value for key 'myKey': " + new String(value));
                    } else {
                        System.out.println("Value not found for key 'myKey'");
                    }
                }
            }
        } catch (RocksDBException e) {
            // Xử lý lỗi khi mở database không thành công
        }
    }
}
```

Trong ví dụ trên, chúng ta sử dụng `getSnapshot()` để tạo một snapshot của RocksDB. Sau đó, chúng ta sử dụng `setSnapshot()` để thiết lập snapshot cho các thao tác đọc dữ liệu. Bằng cách này, RocksDB đảm bảo rằng các thay đổi sau snapshot không ảnh hưởng đến dữ liệu được đọc từ snapshot. Trong ví dụ, chúng ta đọc giá trị của key "myKey" từ snapshot.

Đó là một ví dụ về việc sử dụng Snapshot trong RocksDB bằng Java.

## C++:

```
#include "rocksdb/db.h"
#include "rocksdb/utilities/transaction_db.h"

int main() {
    rocksdb::DB* db;
    rocksdb::Options options;
    options.create_if_missing = true;
    rocksdb::Status status = rocksdb::DB::Open(options, "/path/to/database", &db);
    if (status.ok()) {
        rocksdb::TransactionDBOptions txdbOptions;
        txdbOptions.create_if_missing = true;
        rocksdb::TransactionDB* transactionDb;
        rocksdb::TransactionDB::Open(txdbOptions, "/path/to/transaction_database",
&transactionDb);

        // Bắt đầu một giao dịch
        rocksdb::WriteOptions writeOptions;
        rocksdb::TransactionOptions txnOptions;
        rocksdb::Transaction* txn = transactionDb->BeginTransaction(writeOptions, txnOptions);

        // Tạo một snapshot
        const rocksdb::Snapshot* snapshot = txn->GetSnapshot();

        // Đọc dữ liệu từ snapshot
        rocksdb::ReadOptions readOptions;
        readOptions.snapshot = snapshot;
        std::string value;
        txn->Get(readOptions, "myKey",
```