

# RocksDB - Put data

Trong RocksDB, `Put` được sử dụng để thêm một cặp key-value vào RocksDB. Dưới đây là các ví dụ về cách sử dụng `Put` trong RocksDB bằng Java và C++.

## Java:

```
import org.rocksdb.*;

public class RocksDBPutExample {
    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")) {
            byte[] key = "myKey".getBytes();
            byte[] value = "myValue".getBytes();
            db.put(key, value);
        } catch (RocksDBException e) {
            // X lý l i khi thêm key-value không thành công
        }
    }
}
```

## C++:

```
#include "rocksdb/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::WriteOptions writeOptions;
        rocksdb::Slice key = "myKey";
        rocksdb::Slice value = "myValue";
        db->Put(writeOptions, key, value);
    }
```

```
    } else {  
        // X lý l i khi thêm key-value không thành công  
    }  
    delete db;  
    return 0;  
}
```

Trong cả hai ví dụ trên, chúng ta tạo một RocksDB database và sau đó sử dụng `Put` để thêm một cặp key-value vào database. Đường dẫn đến database và giá trị key-value có thể được tùy chỉnh theo nhu cầu của bạn.

Lưu ý rằng trong RocksDB, `Put` là một thao tác ghi đồng bộ, điều này có nghĩa là thao tác `Put` sẽ chờ cho đến khi dữ liệu thực sự được ghi vào ổ đĩa trước khi trả về. Điều này đảm bảo tính toàn vẹn của dữ liệu.

Đó là một số thông tin cơ bản về cách sử dụng `Put` trong RocksDB. Bạn có thể tìm hiểu thêm về các tùy chọn và tính năng khác của RocksDB trong tài liệu chi tiết của nó.

---

Revision #1

Created 23 September 2023 09:11:29 by Laptrinh.vn

Updated 23 September 2023 10:56:18 by Laptrinh.vn