**Hướng Dẫn Sử Dụng DatabaseHandler và Makefile**

Tài liệu này hướng dẫn cách sử dụng DatabaseHandler để quản lý dữ liệu trong file CSV cũng như các lệnh make để biên dịch và chạy các phần của dự án.

## **Mục Lục**

1. Sử Dụng DatabaseHandler
   * Hàm getData
   * Hàm setData
2. Sử Dụng Makefile
   * Các Lệnh Trong Makefile
   * Hướng Dẫn Chi Tiết Cho Các Lệnh

## **Sử Dụng DatabaseHandler**

DatabaseHandler là thư viện quản lý dữ liệu CSV với các hàm getData để lấy dữ liệu và setData để cập nhật dữ liệu. Dưới đây là cách sử dụng chi tiết từng hàm:

### **Hàm getData - Lấy Dữ Liệu**

Hàm getData được sử dụng để lấy dữ liệu từ file CSV theo các kiểu int, std::string, hoặc enum. Dữ liệu được lấy dựa trên key là một SystemAttribute.

#### **Cú pháp:**

| template <typename ReturnType> ReturnType getData(SystemAttribute key) const; |
| --- |

#### **Ví dụ:**

| #include "DatabaseHandler.hpp"  int main() {  DatabaseHandler dbHandler;   // Lấy chế độ lái dưới dạng enum  DriveMode driveMode = dbHandler.getData<DriveMode>(SystemAttribute::DRIVE\_MODE);    // Lấy mức pin dưới dạng số nguyên  int batteryLevel = dbHandler.getData<int>(SystemAttribute::BATTERY\_LEVEL);   // Lấy tốc độ xe dưới dạng chuỗi  std::string vehicleSpeed = dbHandler.getData<std::string>(SystemAttribute::VEHICLE\_SPEED);   return 0; } |
| --- |

### **Hàm setData - Cập Nhật Dữ Liệu**

Hàm setData cho phép cập nhật dữ liệu trong file CSV. Có ba phiên bản của hàm setData cho các kiểu giá trị khác nhau: std::string, int, và enum.

#### **1. Cập nhật dữ liệu với chuỗi**

| void setData(SystemAttribute key, const std::string &value); |
| --- |

**Ví dụ:**

| dbHandler.setData(SystemAttribute::VEHICLE\_SPEED, "120"); |
| --- |

#### **2. Cập nhật dữ liệu với số nguyên**

| void setData(SystemAttribute key, int value); |
| --- |

**Ví dụ:**

| dbHandler.setData(SystemAttribute::BATTERY\_LEVEL, 85); |
| --- |

#### **3. Cập nhật dữ liệu với enum**

| template <typename EnumType> void setData(SystemAttribute key, EnumType value); |
| --- |

**Ví dụ:**

| dbHandler.setData(SystemAttribute::DRIVE\_MODE, DriveMode::SPORT); |
| --- |

## **Sử Dụng Makefile**

Makefile hỗ trợ các lệnh tự động hóa việc biên dịch, chạy, và làm sạch dự án. Các lệnh chính bao gồm make, make run, make server, make all, và make clean.

### **Các Lệnh Trong Makefile**

* make hoặc make main: Chỉ biên dịch main mà không biên dịch server.
* make run: Clean main, biên dịch main, và chạy main trong terminal hiện tại.
* make server: Clean server, biên dịch server, và chạy server trong một terminal mới.
* make all: Clean tất cả các file, biên dịch cả main và server, chạy server trong terminal mới và main trong terminal hiện tại.
* make clean: Xóa tất cả các file .o và các file thực thi.

### **Hướng Dẫn Chi Tiết Cho Các Lệnh**

#### **1. make hoặc make main**

* Mục đích: Chỉ biên dịch main mà không biên dịch server.

Cách sử dụng:

| make |
| --- |

* Chi tiết hoạt động:
  + Tạo các thư mục cần thiết nếu chưa tồn tại.
  + Biên dịch tất cả các file nguồn (source files) liên quan đến main thành các file .o.
  + Liên kết các file .o để tạo file thực thi Car trong thư mục ./bin.

#### **2. make run**

* Mục đích: Xóa (clean) main, biên dịch main, và chạy main trong terminal hiện tại.

Cách sử dụng:

| make run |
| --- |

* Chi tiết hoạt động:
  + Xóa tất cả các file .o và file thực thi của main.
  + Biên dịch tất cả các file nguồn liên quan đến main.
  + Chạy main từ file thực thi ./bin/Car trong terminal hiện tại.

#### **3. make server**

* Mục đích: Clean các file của server, biên dịch server, và chạy server trong một terminal mới (trên macOS).

Cách sử dụng:

| make server |
| --- |

* Chi tiết hoạt động:
  + Xóa tất cả các file .o và file thực thi của server.
  + Biên dịch tất cả các file nguồn của server.
  + Mở một terminal mới trên macOS và chạy server từ file thực thi ./bin/Server.
* Lưu ý: osascript được dùng để mở terminal mới trên macOS. Nếu bạn dùng hệ điều hành khác, có thể cần thay đổi lệnh này.

#### **4. make all**

* Mục đích: Clean tất cả các file, biên dịch server và main, chạy server trong một terminal mới và main trong terminal hiện tại.

Cách sử dụng:

| make all |
| --- |

* Chi tiết hoạt động:
  + Xóa tất cả các file .o và file thực thi cho cả main và server.
  + Biên dịch tất cả các file nguồn của main và server.
  + Mở một terminal mới và chạy server từ file thực thi ./bin/Server.
  + Chạy main từ file thực thi ./bin/Car trong terminal hiện tại.

#### **5. make clean**

* Mục đích: Xóa tất cả các file .o và các file thực thi cho cả main và server.

Cách sử dụng:

| make clean |
| --- |

* Chi tiết hoạt động:
  + Xóa toàn bộ các file .o trong thư mục ./bin.
  + Xóa các file thực thi Car và Server trong thư mục ./bin, làm sạch toàn bộ các file biên dịch cũ.