Example1

Essential Packages on Ubuntu

1. apt-get update

```
apt-get update
```

2. apt-get install -y sudo vim wget unzip g++ cmake curl pkg-config libssl-dev libsasl2-dev git python3 nano

```
apt-get install -y sudo vim wget unzip g++ cmake curl pkg-config libssl-dev
libsasl2-dev git python3 nano
```

3. mkdir simpleapp

```
mkdir simpleapp
```

4. cd simpleapp

```
cd simpleapp
```

Extract Crow Framework

5. wget https://github.com/CrowCpp/Crow/releases/download/v1.0%2B5/crow-v1.0+5.tar.gz

```
wget https://github.com/CrowCpp/Crow/releases/download/v1.0%2B5/crow-v1.0+5.tar.gz
```

6. mkdir crow

```
mkdir crow
```

7. tar xvfz crow-v1.0+5.tar.gz -C crow --strip-components=1

```
tar xvfz crow-v1.0+5.tar.gz -C crow --strip-components=1
```

Extract Boost Libraries

8. wget https://boostorg.jfrog.io/artifactory/main/release/1.83.0/source/boost_1_83_0.tar.gz

```
wget
https://boostorg.jfrog.io/artifactory/main/release/1.83.0/source/boost_1_83_0.tar.
gz
```

9. tar -xzvf boost_1_83_0.tar.gz

```
tar -xzvf boost_1_83_0.tar.gz
```

10. nano main.cpp

```
nano main.cpp
```

```
#include "crow.h"
//#include "crow_all.h"

int main()
{
    crow::SimpleApp app; //define your crow application

    //define your endpoint at the root directory
    CROW_ROUTE(app, "/")([](){
        return "Hello world";
    });

    //set the port, set the app to run on multiple threads, and run the app
    app.port(8080).multithreaded().run();
}
```

11. nano CMakeLists.txt

```
nano CMakeLists.txt
```

```
cmake_minimum_required(VERSION 3.15)
project(simplecpp)

# Define the include directories
```

```
set(INCLUDE_PATHS ./boost_1_83_0 ./crow/include)

# Add the executable target
add_executable(simplecpp main.cpp)

# Include the defined paths
target_include_directories(simplecpp PUBLIC ${INCLUDE_PATHS})

# Specify the C++ standard
set_target_properties(simplecpp PROPERTIES
    CXX_STANDARD 17
    CXX_STANDARD_REQUIRED TRUE
)
```

Build the Application

12. mkdir build

```
mkdir build
```

13. cd build

```
cd build
```

14. cmake ..

```
cmake ..
```

15. make

```
make
```

Run the Application

16. ./simplecpp

```
./simplecpp &
```

Once executed, it will start a web server on a specified port 8080

17. curl http://localhost:8080

```
curl http://localhost:8080
```

Excpected Output

```
HelloWorld
```