# General

## Project Setup and Structure

Download and extract in main project folder: <https://sourceforge.net/projects/cppcms/files/>

tar -xjf cppcms-1.0.4.tar.bz2

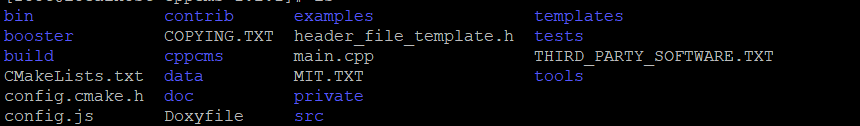
Run the following commands inside main project folder:

mkdir build  
cd build  
cmake ..  
make  
make test  
make install

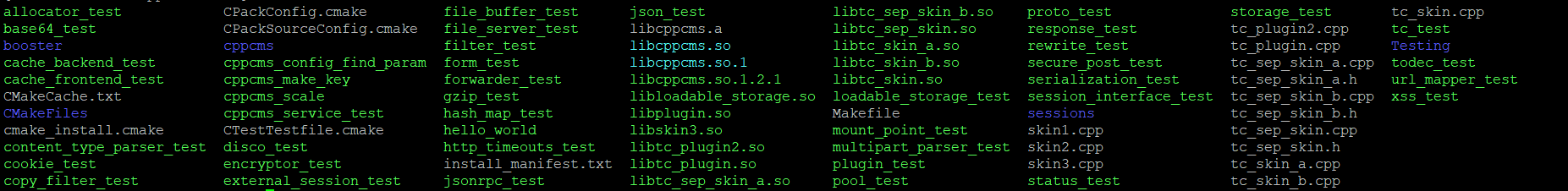
Make data and templates folders and main.cpp and config.js files in main folder

mkdir data  
mkdir tempaltes  
vi main.cpp  
vi config.js

Result Main Folder:



Result Build Folder:



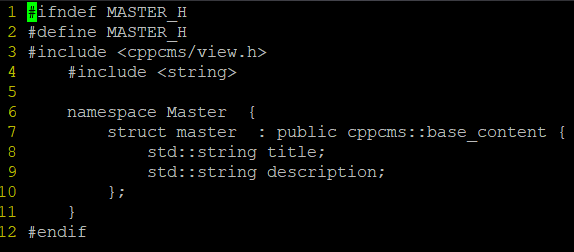
### config.js File



### Make Main Header File

cd data  
vi master.h

#### master.h File



### Make Main Template File

cd template  
vi master.tmpl

#### master.tmpl File

<% c++ #include "../data/master.h" %>  
<% skin my\_skin %>  
<% view master uses Master::master %>  
<% template title() %> <%= title %> <%end%>  
<% template description() %> <%= description %> <%end%>  
<% template page\_content() %>PAGE CONTENT<% end %>  
<% template render() %>  
<html>  
 <head>  
 <meta charset="utf-8" />  
 <title><% include title() %></title>  
 <meta name="description" content="<% include description() %>" />  
 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC" crossorigin="anonymous">  
 </head>  
 <body>  
 <div id="HEADER\_CONTENT" style="background-color: #181a1b">  
 </div>  
 <div id="PAGE\_CONTENT" style="background-color: #181a1b">  
 <% include page\_content() %>  
 </div>  
 </body>  
</html>  
<% end template %>  
<% end view %>  
<% end skin %>

### Make Main CPP File

#### main.cpp File

#include <cppcms/application.h>  
#include <cppcms/applications\_pool.h>  
#include <cppcms/service.h>  
#include <cppcms/http\_response.h>  
#include <cppcms/url\_dispatcher.h>  
#include <cppcms/url\_mapper.h>  
#include <iostream>  
#include "data/master.h"  
  
class Main : public cppcms::application {  
public:  
 Main(cppcms::service &srv) : cppcms::application(srv) {}  
 virtual void main(std::string url) {}  
}  
  
int main(int argc, char \*\* argv) {  
 try {  
 cppcms::service srv(argc, argv);  
 srv.applications\_pool().mount(cppcms::applications\_factory<Main>());  
 srv.run();  
 } catch (std::exception const &e) {  
 std::cerr << e.what() << std::endl;  
 }  
}

## Building

### Building Templates

Templates must always be built before compiling and running the project.

**IMPORTANT NOTE**: If multiple templates which inherit from the master template are being built, the master template MUST be put FIRST in the building command

cd templates  
cppcms\_tmpl\_cc master.tmpl [other templates after] -o my\_skin.cpp

## Compiling

The C++ must be compiled with the my\_skin.cpp file created when building the templates.

In addition, make sure to export the LD\_LIBRARY\_PATH before compiling

Run the following commands in the main project folder:

export LD\_LIBRARY\_PATH=/usr/local/lib/  
g++ main.cpp templates/my\_skin.cpp -o main -lcppcms -lbooster

### With Chrono

To measure time, use this compile command so chrono would work

export LD\_LIBRARY\_PATH=/usr/local/lib/  
g++ main.cpp templates/my\_skin.cpp -o main -std=c++0x -lcppcms -lbooster

## Starting Website

Run the following command in the main project folder:

./main -c config.js

## Curl

### Single File

curl -i -X POST -H "Content-Type: multipart/form-data" -F "data=@file\_0" http://192.168.0.120:8080/receive\_requests

### Multiple Files

curl -i -X POST -H "Content-Type: multipart/form-data" -F "data=@file\_0" -F "data=@file\_1" http://192.168.0.120:8080/receive\_requests

## Create File

truncate -s 5M ostechnix.txt

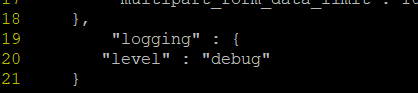
## View Uploaded Files with size in MB

ls uploads/ -l --block-size=M

## Debugging

### config.js

Add the following to the config file to activate debugging



### main.cpp

The equivalent of cout in CPPCMS is:

BOOSTER\_DEBUG("app\_name") << "message";