

TalentedWebServer

An HTTP Web Server in C++ that supports traditional features used across. Talented Web Server is compatible with windows & linux platforms. The backbone of talented web server is socket programming / network programming available in built-in libraries such as `arpa/inet.h` & `sys/socket.h`.

Features

- Built-in encoding/decoding
- Supports static & dynamic mime type adding for any file format.
- Supports GET/POST type requests.
- HTTP/1.1 (version 1.1) support.

Test Case

- Reference : /appOne

```
//Bobby [The user]
//Assume that the below code is being written by the server user
int main()
{
    try
    {
        Bro bro;                //Creating object of Bro class
        bro.setStaticResourcesFolder("whatever"); //whatever folder will contain all the
        necessary files to be used
        bro.get("/save_test1_data",[])(Request &request,Response &response) void {
            const char *html=R""""(
                <!Doctype html>
                <html lang='en'>
                <head>
                <meta charset='utf-8' />
                <title>Bro Test Cases</title>
                </head>
                <body>
                <h1>Test Case 1 - GET with Query String</h1>
                <h3>Response from Server Side</h3>
                <b>Data Saved</b>
            )""";
        }
```



```

        <br/><br/>
        <a href='/index.html'>Home</a>
    </body>
</html>
)"""";
response.setContentType("text/html");    //setting mime type
response<<html;
});

bro.post("/save_test2_data",[(Request &request,Response &response) void {
    const char *html=R""""(
        <!Doctype html>
        <html lang='en'>
        <head>
        <meta charset='utf-8'/>
        <title>Bro Test Cases</title>
        </head>
        <body>
        <h1>Test Case 2 - POST with Form Data</h1>
        <h3>Response from Server Side</h3>
        <b>Data Saved</b>
        <br/><br/>
        <a href='/index.html'>Home</a>
        </body>
        </html>
    )"""";
    response.setContentType("text/html");    //setting mime type
    response<<html;
});

    bro.listen(6060,[(Error &error) void {        //putting server on listening mode
for specific port numbers
    if(error.hasError())        //hasError() can be used to tell whether anything
went wrong
    {
        cout<<error.getError()<<endl;
        return;
    }
    cout<<"Bro HTTP server is ready to listen on port 6060"<<endl;
    });
}catch(string exception)
{
    //in case if something is not going as per the rule
    cout<<"Exception : "<<exception<<endl;
}

```

```
return 0;  
}
```