

C++ REST SDK (cpprestsdk) Cheat Sheet

1. Include Headers

```
#include <cpprest/http_listener.h>
#include <cpprest/json.h>
using namespace web;
using namespace http;
using namespace http::experimental::listener;
```

2. Create an HTTP Listener

```
http_listener listener("http://localhost:8080");
```

3. Support HTTP Methods

```
listener.support(methods::GET, handle_get);
listener.support(methods::POST, handle_post);
```

4. Start Listener

```
listener.open().wait();
```

5. Handle GET Request

```
void handle_get(http_request request) {
    json::value response_data;
    response_data["message"] = json::value::string("Hello from GET");
    request.reply(status_codes::OK, response_data);
}
```

6. Handle POST Request

```
void handle_post(http_request request) {
    request.extract_json().then([=](json::value body) {
        auto name = body.at("name").as_string();
        json::value res;
        res["received"] = json::value::string(name);
        request.reply(status_codes::OK, res);
    }).wait();
}
```

```
}
```

7. Extract Query Parameters

```
uri::split_query(request.request_uri().query())["param"];
```

8. Construct JSON

```
json::value obj;  
obj["id"] = json::value::number(1);  
obj["name"] = json::value::string("Siva");
```

9. Extract JSON Fields

```
auto val = body.at("key").as_integer();
```

10. JSON Serialization

```
obj.serialize(); // returns std::wstring
```

11. Error Handling

```
request.reply(status_codes::BadRequest, "Invalid request");
```

12. Close Listener

```
listener.close().wait();
```

13. Compile with g++ (example)

```
g++ -std=c++17 server.cpp -lcpprest -lboost_system -lssl -lcrypto -o server
```