

# Crow C++ Web Framework - Important Functions Cheat Sheet

## 1. Define a Crow App

```
crow::SimpleApp app;
```

## 2. Define a Route (GET)

```
CROW_ROUTE(app, "/hello")([](){  
    return "Hello, world!";  
});
```

## 3. Route with Parameters

```
CROW_ROUTE(app, "/user/<int>")([](int id){  
    return "User ID: " + std::to_string(id);  
});
```

## 4. Route with Multiple HTTP Methods

```
CROW_ROUTE(app, "/api/data").methods(crow::HTTPMethod::GET, crow::HTTPMethod::POST)  
([](const crow::request& req){  
    return crow::response("Handled GET or POST");  
});
```

## 5. Parsing JSON from Request

```
auto body = crow::json::load(req.body);  
if (!body) return crow::response(400);
```

## 6. Returning JSON Response

```
crow::json::wvalue x;  
x["status"] = "ok";  
return x;
```

## 7. Set HTTP Status and Headers

```
crow::response res(200);  
res.set_header("Content-Type", "application/json");
```

```
res.write("{\"success\": true}");  
return res;
```

## 8. Route Prefix (Grouping)

```
CROW_ROUTE(app, "/api/v1/hello")([]){  
    return "Hello from API v1";  
};
```

## 9. Start the Server

```
app.port(8080).multithreaded().run();
```

## 10. Logging Control

```
#define CROW_ENABLE_DEBUG before including crow_all.h
```

## 11. Middleware Usage (Custom authentication, logging, etc.)

Define a struct implementing `before_handle` and `after_handle`.

## 12. Handling Query Parameters

```
std::string token = req.url_params.get("token");
```

## 13. Route with Method Suffix (`_method`)

```
CROW_ROUTE(app, "/upload").methods("POST"_method)(...);
```

## 14. Handling JSON in Response using `nlohmann`

```
return crow::response(nlohmann::json(u).dump());
```

## 15. Dynamic Routes with `Crow::black_magic`

```
CROW_ROUTE(app, "/api/<string>/<int>")(...);
```