



how to write body for mux.HandleFunc in



All

Short videos

Videos

Forums

Shopping

Images

More ▾

Tools ▾

◆ AI Overview



The `mux.HandleFunc` in Go's Gorilla Mux package expects a function with the signature `func(w http.ResponseWriter, r *http.Request)`. This function is your "handler" and contains the logic for processing a specific HTTP request and generating a response.

Here's how to structure the body of such a handler function:

1. Accessing Request Data:

Request Body (for POST/PUT/PATCH).

Go



```
import (
    "encoding/json"
    "io/ioutil"
    "net/http"
)

func myHandler(w http.ResponseWriter, r *http.Request) {
    if r.Method == http.MethodPost {
        body, err := ioutil.ReadAll(r.Body)
        if err != nil {
            http.Error(w, "Error reading request body", http.StatusInternalServerError)
            return
        }
        defer r.Body.Close()

        var data MyStruct // Define your struct to unmarshal the JSON into
        err = json.Unmarshal(body, &data)
        if err != nil {
            http.Error(w, "Error unmarshaling JSON", http.StatusBadRequest)
            return
        }
        // Process the 'data'
    }
}
```

```
}  
}
```

URL Parameters.

Go



```
import (  
    "github.com/gorilla/mux"  
    "net/http"  
)  
  
func myHandler(w http.ResponseWriter, r *http.Request) {  
    vars := mux.Vars(r)  
    id := vars["id"] // Assuming your route is like "/items/{id}"  
    // Use the 'id'  
}
```

Query Parameters.

Go



```
func myHandler(w http.ResponseWriter, r *http.Request) {  
    param := r.URL.Query().Get("paramName")  
    // Use the 'param'  
}
```

2. Implementing Business Logic:

This is where you perform the core tasks of your application, such as: Interacting with a database, Calling other services, and Performing calculations or data manipulations.

3. Generating the Response:

Setting Status Code.

Go



```
w.WriteHeader(http.StatusOK) // Or http.StatusCreated, http.StatusBadRequest, etc
```

Setting Headers.

Go



```
w.Header().Set("Content-Type", "application/json")
```

Writing Response Body.

Go



```
json.NewEncoder(w).Encode(responseObject) // For JSON responses
// Or
w.Write([]byte("Hello, World!")) // For plain text
```

Oops, something went wrong.



Stack Overflow

1 answer · 11 years ago

how to define custom mux with golang

I wanna get value of "id" in the url like /url/{id}, setting the value in the req.Form and clean the path like **mux**.

[1 answer](#) · Top answer: We have used <http://www.gorillatoolkit.org/pkg/mux> for over a year in our produ...

Create **http.Response** instance with **sample body** string ... [4 answers](#) Nov 29, 2015

Using **HandleFunc** on **http** vs **mux** - Stack Overflow [2 answers](#) Feb 13, 2019

[More results from stackoverflow.com](#)



Alex Edwards

<https://www.alexedwards.net> › blog › an-introduction-t...

An introduction to Handlers and Servermuxes in Go

Jun 24, 2025 — The net/http package provides a couple of shortcuts for registering routes with the default **servemux**: **http.Handle()** and **http.HandleFunc()**.



Medium · Skye Geraghty

30+ likes · 5 years ago

Using Gorilla Mux and handlers in Go | by Skye Geraghty

To declare a route, you will need to use the **http.HandleFunc()** method which will first take in a route and then a function. The function called ...



Hackernoon

<https://hackernoon.com> › understanding-mux-and-hand...

Understanding Mux and Handler by Writing Your Own ...

Nov 9, 2022 — In this article, we will try to understand two crucial Go concepts - **mux** and Handler. But before we start **writing** any **code** let us first understand what we need ...



Reddit · r/golang

8 comments · 1 year ago

Please help me understand a middleware pattern. : r/golang

I cannot wrap my head around how the `commonHeaders()` middleware function is executed before the **mux** routes the request to a matching handler.



Better Stack

<https://betterstack.com> › community › guides › scaling-go

Routing in Go with Gorilla Mux

Apr 16, 2025 — Gorilla **Mux** extends Go's routing capabilities with URL patterns, path variables, and HTTP method filtering. Let's explore these concepts with ...



Go Forum

<https://forum.golangbridge.org> › ... › Code Review

How to handler segment uri? - Code Review - Go Forum

Jan 30, 2018 — You can easily get the user ID using the standard library like this:

mux.HandleFunc("/users/", GetUsers) // Notice the extra "/" in /users/.



DigitalOcean

<https://www.digitalocean.com> › community › tutorials

How To Make HTTP Requests in Go

Sep 24, 2025 — In this tutorial, you will create a program that makes several types of HTTP requests to an HTTP server.

Making A Get Request

Using Http. Get To Make A...

Sending A Post Request



Reddit · r/golang

4 comments · 2 years ago

http.ServeMux handler registration through Handle method ...

What happens is that the **mux.Handle(...)** registers only the last element of the handlers slice. The way I managed to fix it is by running a go-routine per slice ...





The Go Programming Language

<https://go.dev> > [blog](#) > [routing-enhancements](#)

Routing Enhancements for Go 1.22

Feb 13, 2024 — In Go 1.22, the existing **code** will continue to work, or you could instead **write** this:
`http.HandleFunc("GET /posts/{id}", handlePost2)`. This ...

People also search for

Golang **http** write **response** body Mux **vars** in golang Golang mux **example** **Http responsewriter** golang **example** **Http HandlerFunc** golang Golang **request** body Golang **HTTP handler example** Golang **read response** body 

Results are personalized - [Try without personalization](#)

● **Landsdowne, Cary, NC** - [Based on your places \(Home\)](#) - [Update location](#)

[Help](#)[Send feedback](#)[Privacy](#)[Terms](#)