



New chat



Write me a simple crud app in Go language.

Use in memory store as database and use only standard library.

The code should contain a very simple web interface with results represented in a table and buttons to interact.



Sure, here's a simple CRUD app in Go that uses an in-memory store as a database and a very simple web interface. This app provides a RESTful API that allows you to perform CRUD operations on a collection of items, where each item has an ID and a Name.

go

Copy code

```
package main

import (
    "encoding/json"
    "fmt"
    "net/http"
)

type Item struct {
    ID    int    `json:"id"`
    Name  string `json:"name"`
}

var items = []Item{}

func main() {
    http.HandleFunc("/", home)
    http.HandleFunc("/items", listItems)
    http.HandleFunc("/items/add", addItem)
    http.HandleFunc("/items/delete", deleteItem)
    http.HandleFunc("/items/update", updateItem)
    http.ListenAndServe(":8080", nil)
}

func home(w http.ResponseWriter, r *http.Request) {
    fmt.Fprintf(w, "Welcome to the simple CRUD app!")
}
```

```
func listItems(w http.ResponseWriter, r *http.Request) {
    if r.Method != http.MethodGet {
        http.Error(w, "Invalid request method", http.StatusMethodNotAllowed)
        return
    }

    json.NewEncoder(w).Encode(items)
}

func addItem(w http.ResponseWriter, r *http.Request) {
    if r.Method != http.MethodPost {
        http.Error(w, "Invalid request method", http.StatusMethodNotAllowed)
        return
    }

    var item Item
    err := json.NewDecoder(r.Body).Decode(&item)
    if err != nil {
        http.Error(w, err.Error(), http.StatusBadRequest)
        return
    }

    item.ID = len(items) + 1
    items = append(items, item)

    json.NewEncoder(w).Encode(item)
}

func deleteItem(w http.ResponseWriter, r *http.Request) {
    if r.Method != http.MethodDelete {
        http.Error(w, "Invalid request method", http.StatusMethodNotAllowed)
        return
    }

    var item Item
    err := json.NewDecoder(r.Body).Decode(&item)
    if err != nil {
        http.Error(w, err.Error(), http.StatusBadRequest)
        return
    }

    for i, v := range items {
```

```
        if v.ID == item.ID {
            items = append(items[:i], items[i+1:]...)
            break
        }
    }

    json.NewEncoder(w).Encode(items)
}

func updateItem(w http.ResponseWriter, r *http.Request) {
    if r.Method != http.MethodPut {
        http.Error(w, "Invalid request method", http.StatusMethodNotAllowed)
        return
    }

    var item Item
    err := json.NewDecoder(r.Body).Decode(&item)
    if err != nil {
        http.Error(w, err.Error(), http.StatusBadRequest)
        return
    }

    for i, v := range items {
        if v.ID == item.ID {
            items[i] = item
            break
        }
    }

    json.NewEncoder(w).Encode(items)
}
```

To test this app, you can use a tool like curl or Postman to send HTTP requests to the app's endpoints. For example, you can use the following commands to add, list



Add comments to the code





err

There was an error generating a response

 Regenerate response

[ChatGPT Feb 13 Version](#). Free Research Preview. Our goal is to make AI systems more natural and safe to interact with. Your feedback will help us improve.