C:\Users\12408\Desktop\BoltQ\cmd\api\main.go:

package main

import (

    "fmt"

    "net/http"

)

func main() {

    fmt.Println("Starting Job API Service...")

    http.HandleFunc("/", func(*w* http.ResponseWriter, *r* \*http.Request) {

        w.Write([]*byte*("Job API Running!"))

    })

    http.ListenAndServe(":8080", nil)

}

C:\Users\12408\Desktop\BoltQ\cmd\worker\main.go:

package main

import "fmt"

func main() {

    fmt.Println("Worker service started...")

*// TODO: Worker logic will go here*

}

C:\Users\12408\Desktop\BoltQ\internal\api\handler.go:

package api

import "net/http"

func SubmitJobHandler(*w* http.ResponseWriter, *r* \*http.Request) {

    w.Write([]*byte*("Job submitted successfully!"))

}

C:\Users\12408\Desktop\BoltQ\internal\queue\redis\_queue.go:

package queue

import (

    "context"

    "fmt"

    "github.com/go-redis/redis/v8"

)

var ctx = context.Background()

type RedisQueue struct {

    client \*redis.Client

}

func NewRedisQueue() \*RedisQueue {

    client := redis.NewClient(&redis.Options{

        Addr: "localhost:6379",

    })

    return &RedisQueue{client: client}

}

func (*q* \*RedisQueue) Publish(*task* *string*) *error* {

    err := q.client.LPush(ctx, "task\_queue", task).Err()

    if err != nil {

        return err

    }

    fmt.Println("Task added:", task)

    return nil

}

func (*q* \*RedisQueue) Consume() (*string*, *error*) {

    task, err := q.client.RPop(ctx, "task\_queue").Result()

    if err != nil {

        return "", err

    }

    return task, nil

}

C:\Users\12408\Desktop\BoltQ\internal\worker\worker.go:

package worker

import (

    "fmt"

    "time"

)

func ProcessTask(*task* *string*) {

    fmt.Println("Processing task:", task)

    time.Sleep(2 \* time.Second) *// Simulate work*

    fmt.Println("Task completed:", task)

}

C:\Users\12408\Desktop\BoltQ\pkg\config\config.go:

package config

import "os"

func GetEnv(*key*, *fallback* *string*) *string* {

    if value, exists := os.LookupEnv(key); exists {

        return value

    }

    return fallback

}

C:\Users\12408\Desktop\BoltQ\pkg\logger\logger.go:

package logger

import "log"

func Info(*msg* *string*) {

    log.Println("[INFO]:", msg)

}

func Error(*msg* *string*) {

    log.Println("[ERROR]:", msg)

}

C:\Users\12408\Desktop\BoltQ\Dockerfile:

FROM golang:1.20

WORKDIR /app

COPY . .

RUN go mod tidy

CMD ["go", "run", "cmd/api/main.go"]