**Homework Week3**: <https://github.com/gin-gonic/gin/blob/master/README.md>

1. สร้าง folder “todos”
2. สร้าง main.go

package main

import (

    "net/http"

    "github.com/gin-gonic/gin"

)

func getTodosHandler(c \*gin.Context) {

    response := gin.H{

        "message": "This is REST function : getTodosHandler",

    }

    c.JSON(http.StatusOK, response)

}

func main() {

    r := gin.Default()

    api := r.Group("/api")

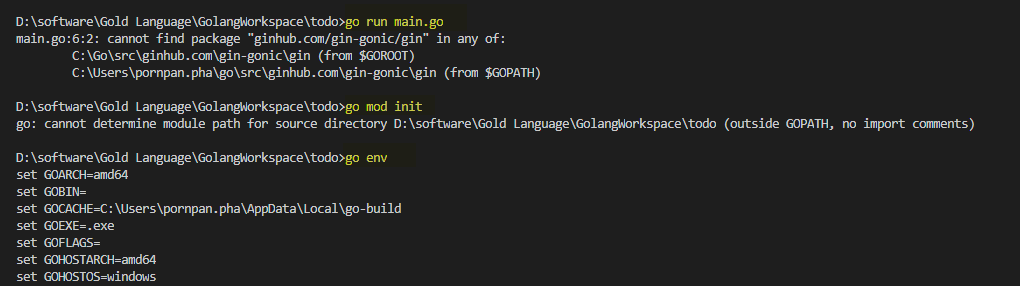
    api.GET("/todos", getTodosHandler)

    r.Run(":1234") //Go run listen and serve on 0.0.0.0:1234

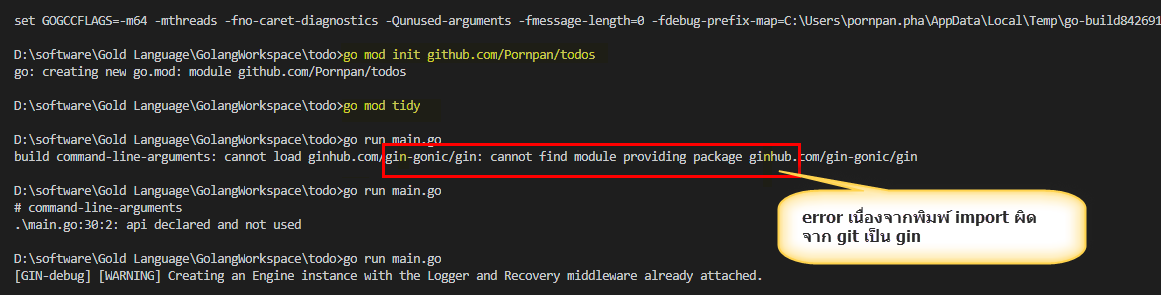
}

1. Run script ตามอาจารย์สอน ตามภาพ

* go run main.go
* go mod init
* go env



1. Run script ตามอาจารย์สอน ตามภาพ



1. แก้ไข main.go

package main

import (

    "net/http"

    "strconv"

    "github.com/gin-gonic/gin"

)

type Todo struct {

    ID string `json:"id"`

    Title string `json:"title`

    Status string `json:"status`

}

var todos = map[string]\*Todo{}

func getTodosHandler(c \*gin.Context) {

    todoVal := []\*Todo{}

    for \_, td := range todos {

        todoVal = append(todoVal, td)

    }

    c.JSON(http.StatusOK, todoVal)

}

func main() {

    r := gin.Default()

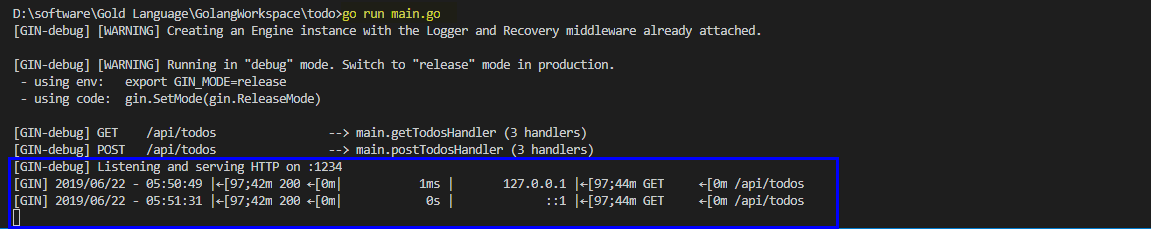
    api := r.Group("/api")

    api.GET("/todos", getTodosHandler)

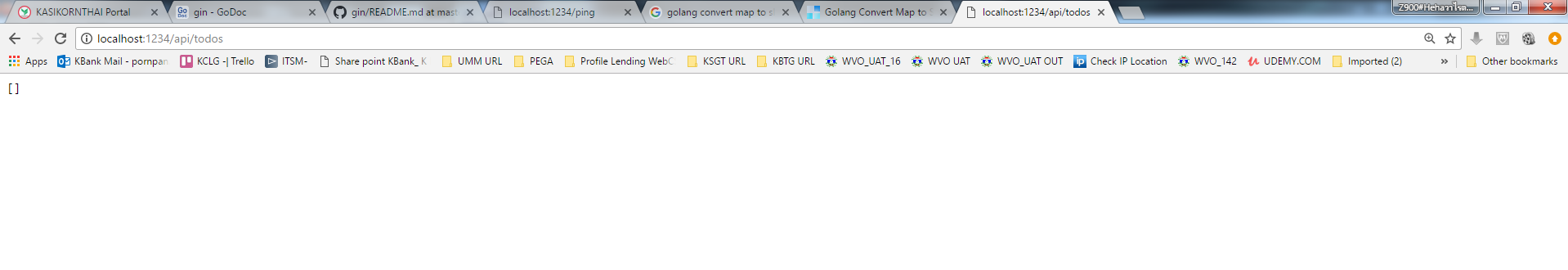
    r.Run(":1234") //Go run listen and serve on 0.0.0.0:1234

}

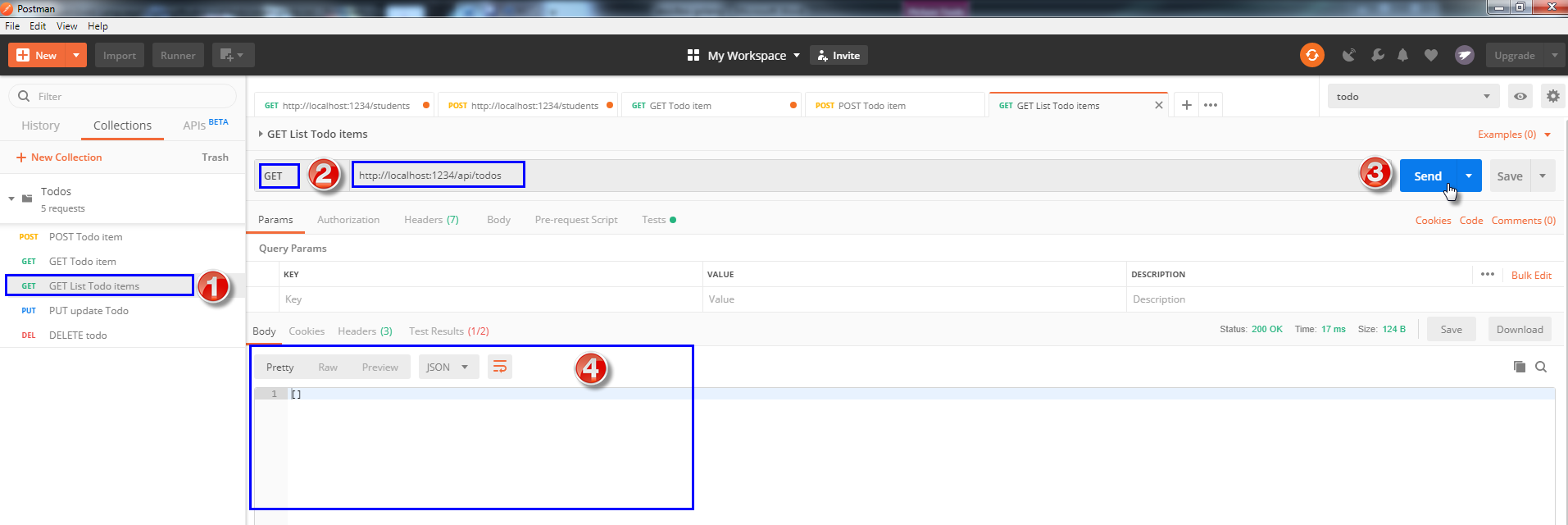
1. Run Go Lang application server



1. ทดสอบเข้า URL และ Postman



1. ทดสอบ postman :GET /api/todos โดย call function **getTodoHandler**



1. เพิ่ม function **postTodosHandler** ใน main.go

func postTodosHandler(c \*gin.Context) {

    todoVal := Todo{}

    if err := c.ShouldBindJSON(&todoVal); err != nil {

        c.JSON(http.StatusBadRequest, err)

        return

    }

    genId := len(todos)

    genId++

    id := strconv.Itoa(genId)

    todoVal.ID = id

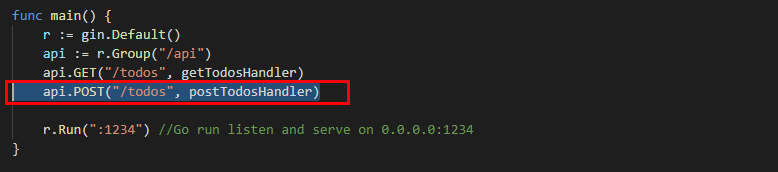
    todos[id] = &todoVal

    c.JSON(http.StatusCreated, todoVal)

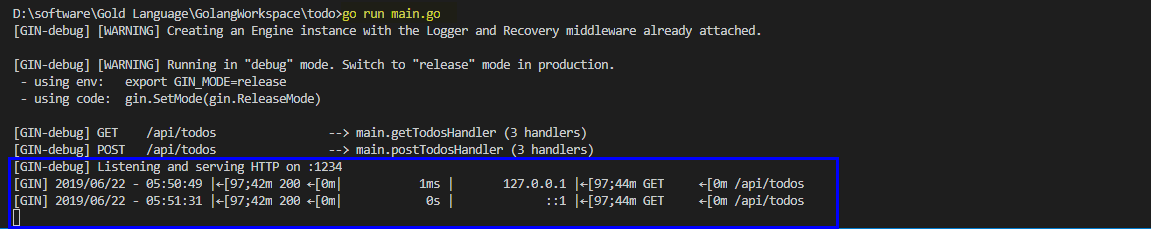
}

1. เพิ่ม code ใน func main เพื่อเรียก api url ของ function **postTodosHandler**

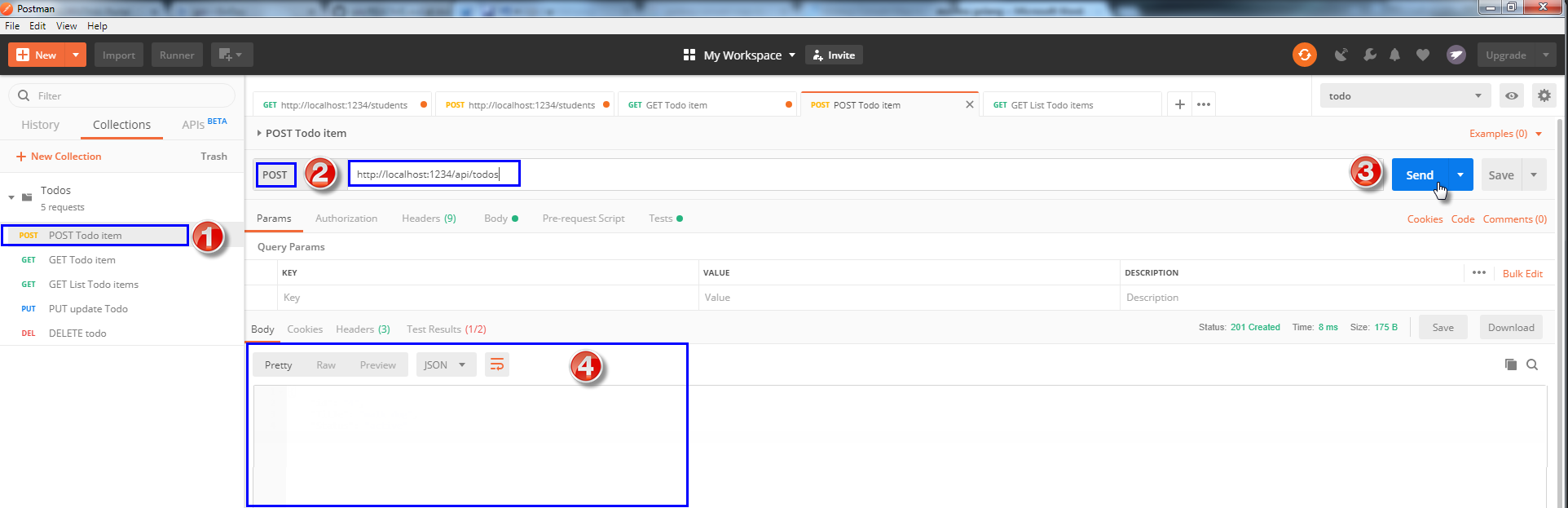
    api.POST("/todos", postTodosHandler)

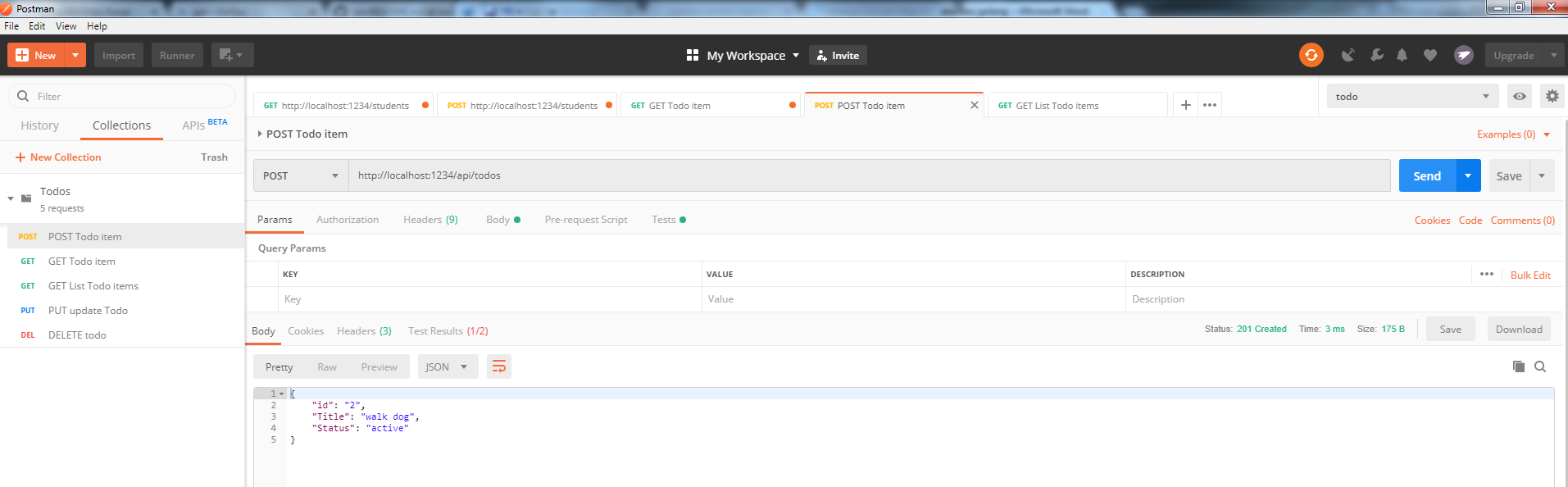


1. Run Go Lang application server



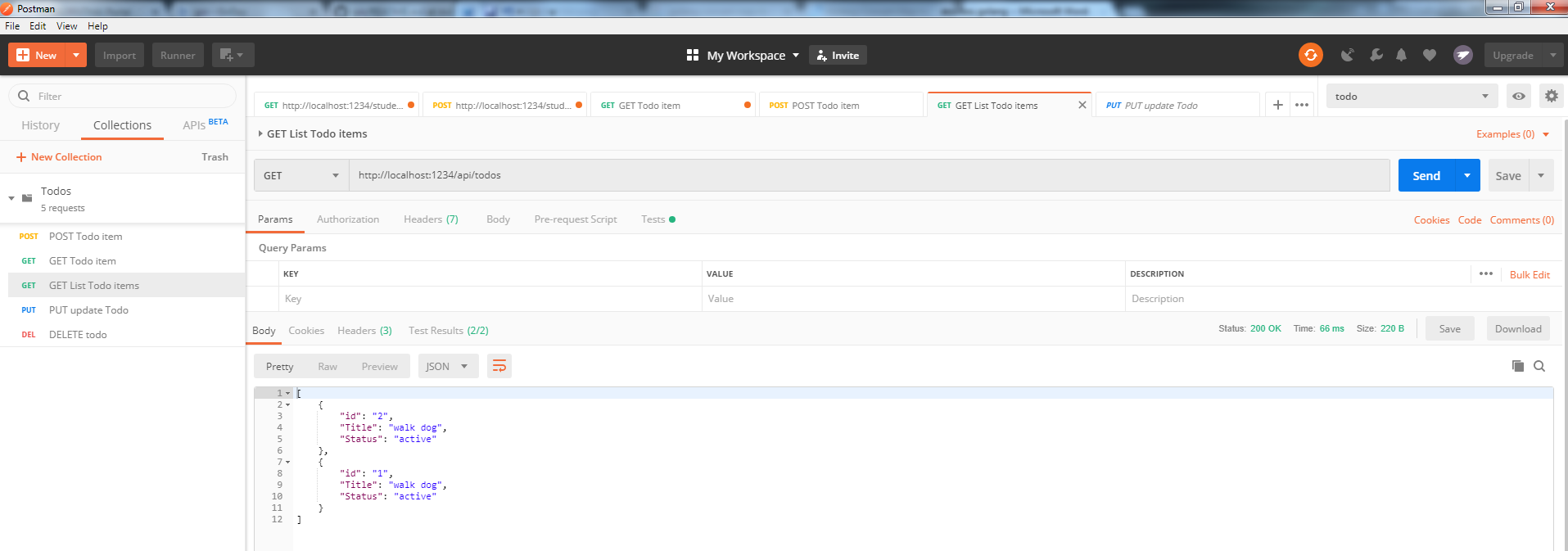
1. ทดสอบ postman : POST /api/todos โดย call function **postTodoHandler**





1. ทดสอบ postman :GET /api/todos โดย call function **getTodoHandler**

พบว่ามีการดึงข้อมูลที่ทำการทดสอบ postman : POST /api/todos มาแสดงเป็น list



1. เพิ่ม function **getTodosByIDHandler** ใน main.go

func getTodosByIDHandler(c \*gin.Context) {

    id := c.Param("id")

    todoVal, ok := todos[id]

    if !ok {

        c.JSON(http.StatusOK, gin.H{})

        return

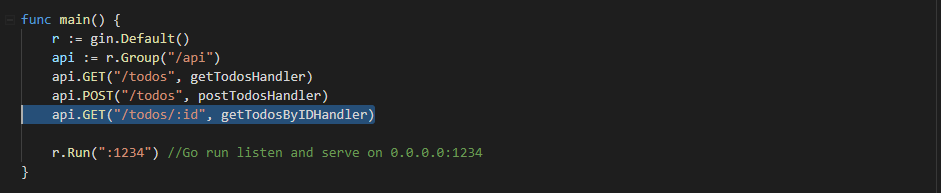
    }

    c.JSON(http.StatusOK, todoVal)

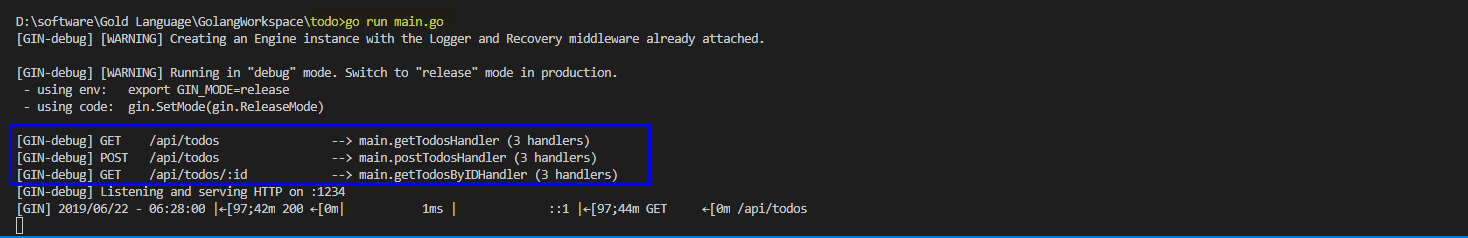
}

1. เพิ่ม code ใน func main เพื่อเรียก api url ของ function **getTodosByIDHandler**

    api.GET("/todos/:id", getTodosByIDHandler)

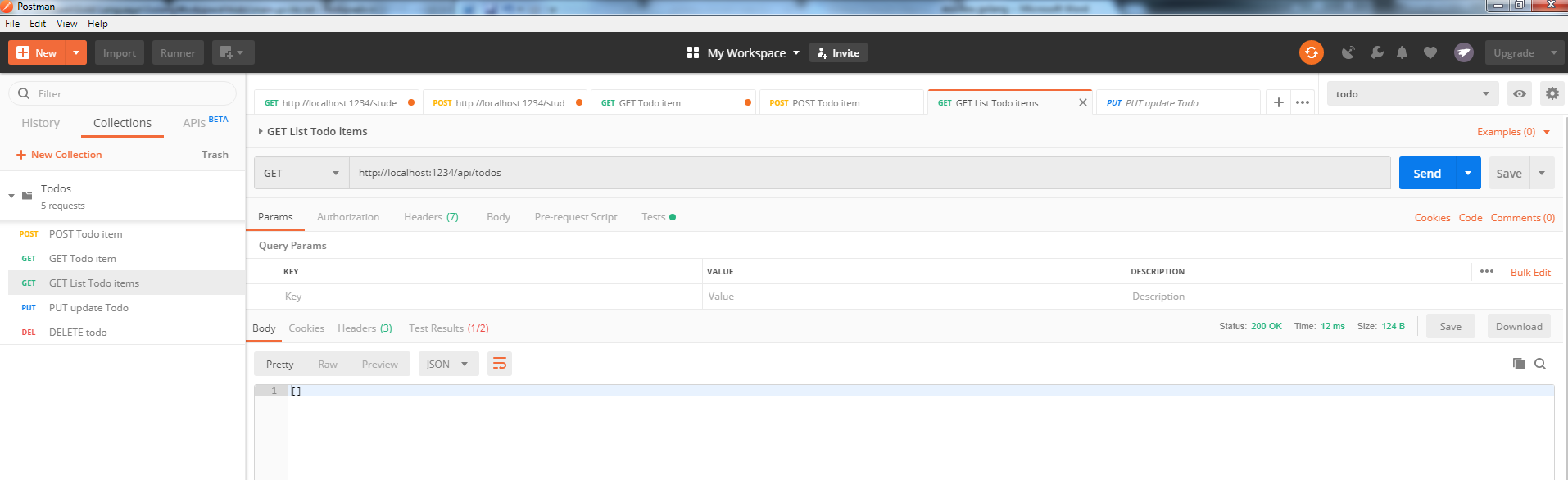


1. Run Go Lang application server

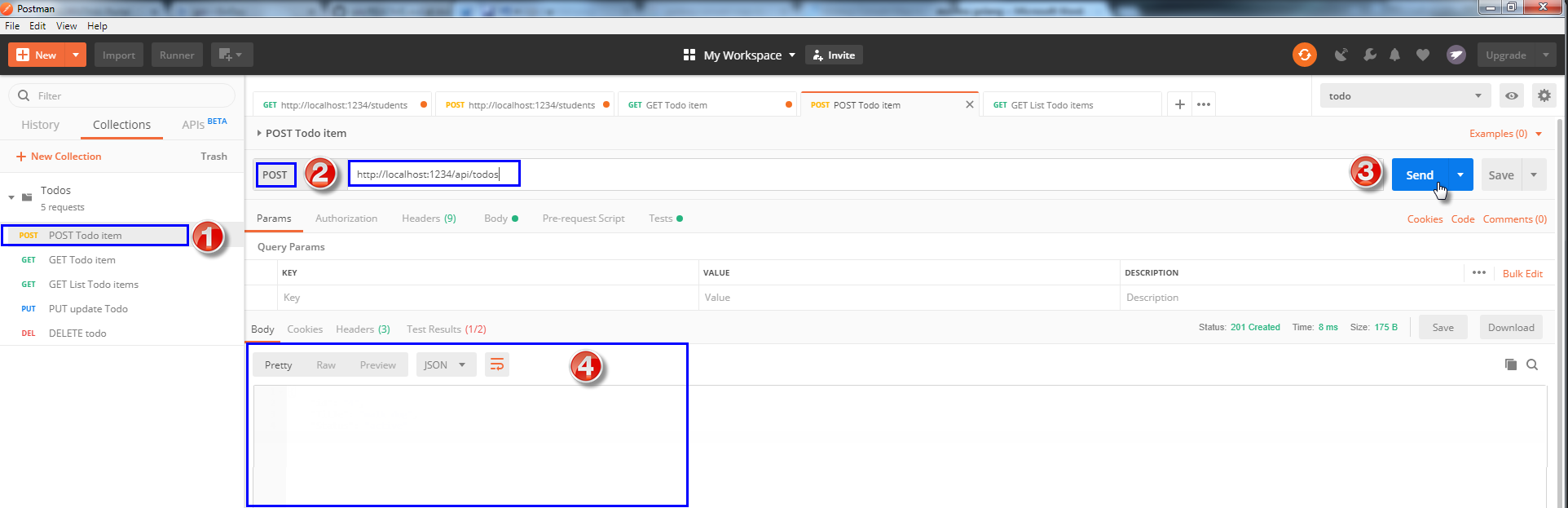


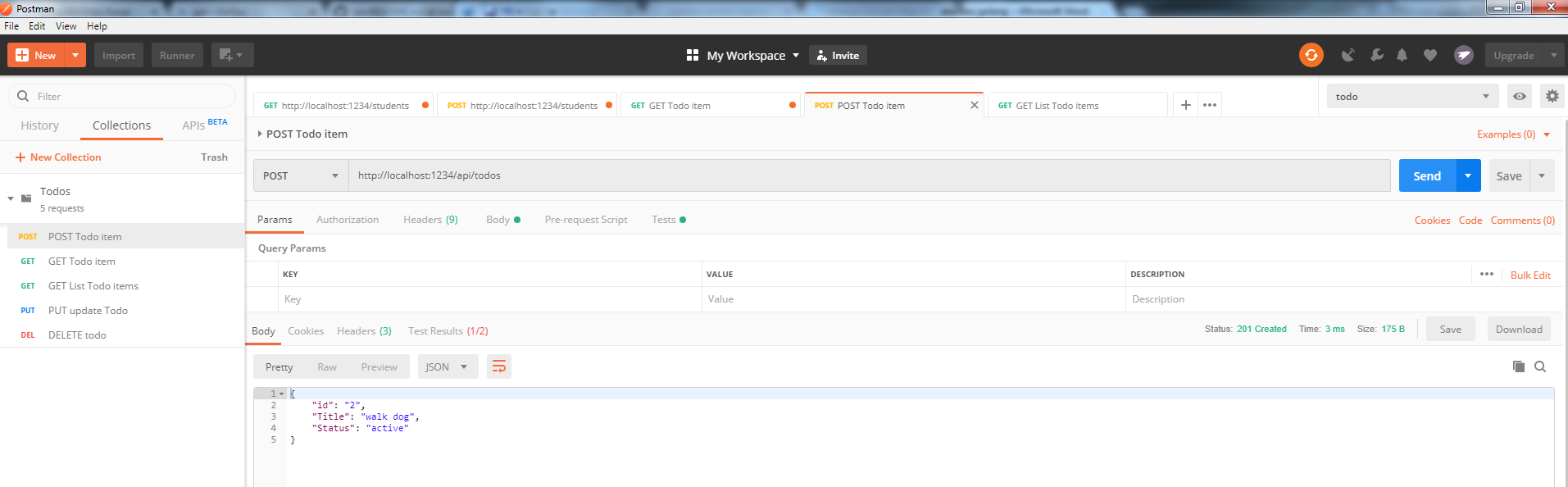
1. ทดสอบ postman :GET /api/todos โดย call function **getTodoHandler**

พบว่ามีการดึงข้อมูลที่ทำการทดสอบ postman : POST /api/todos มาแสดงเป็น list



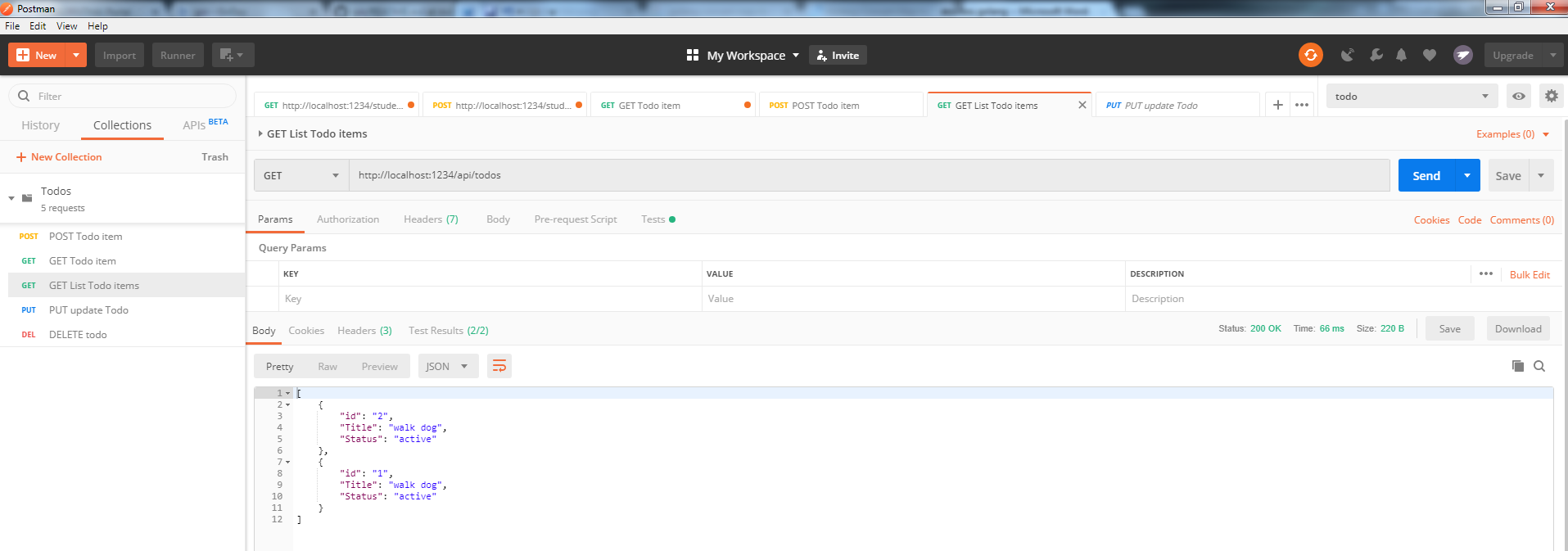
1. ทดสอบ postman : POST /api/todos โดย call function **postTodoHandler**



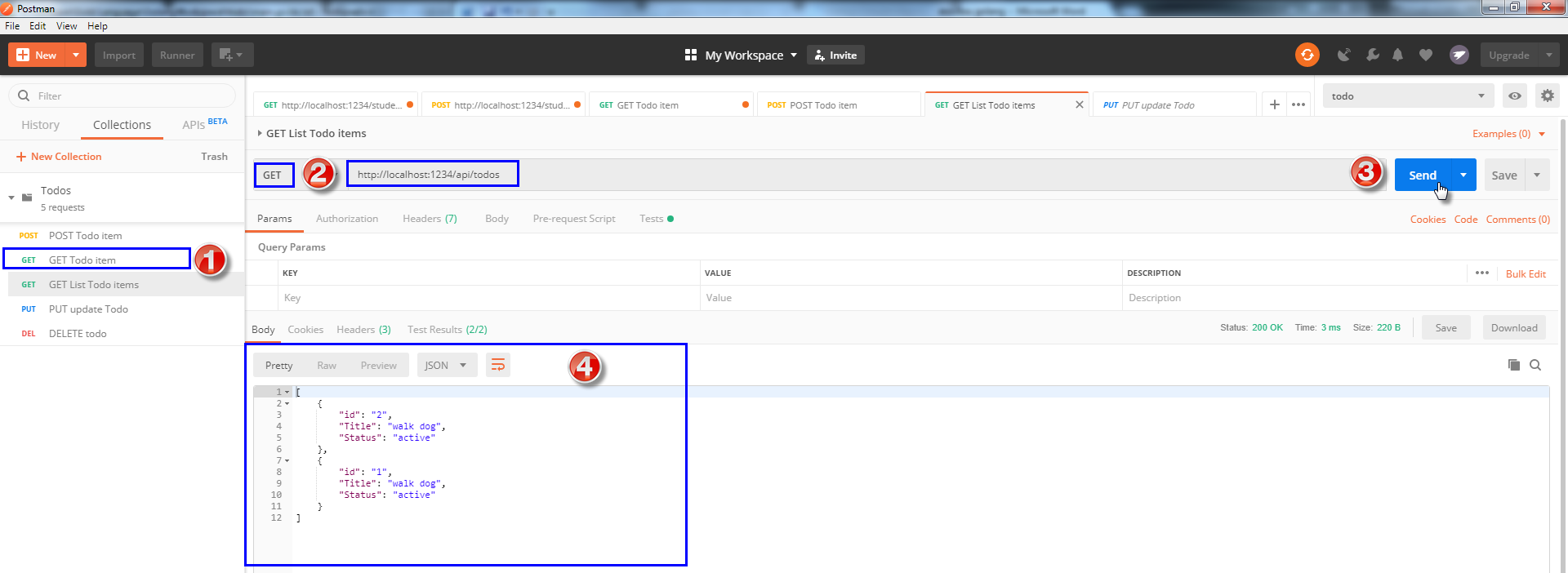


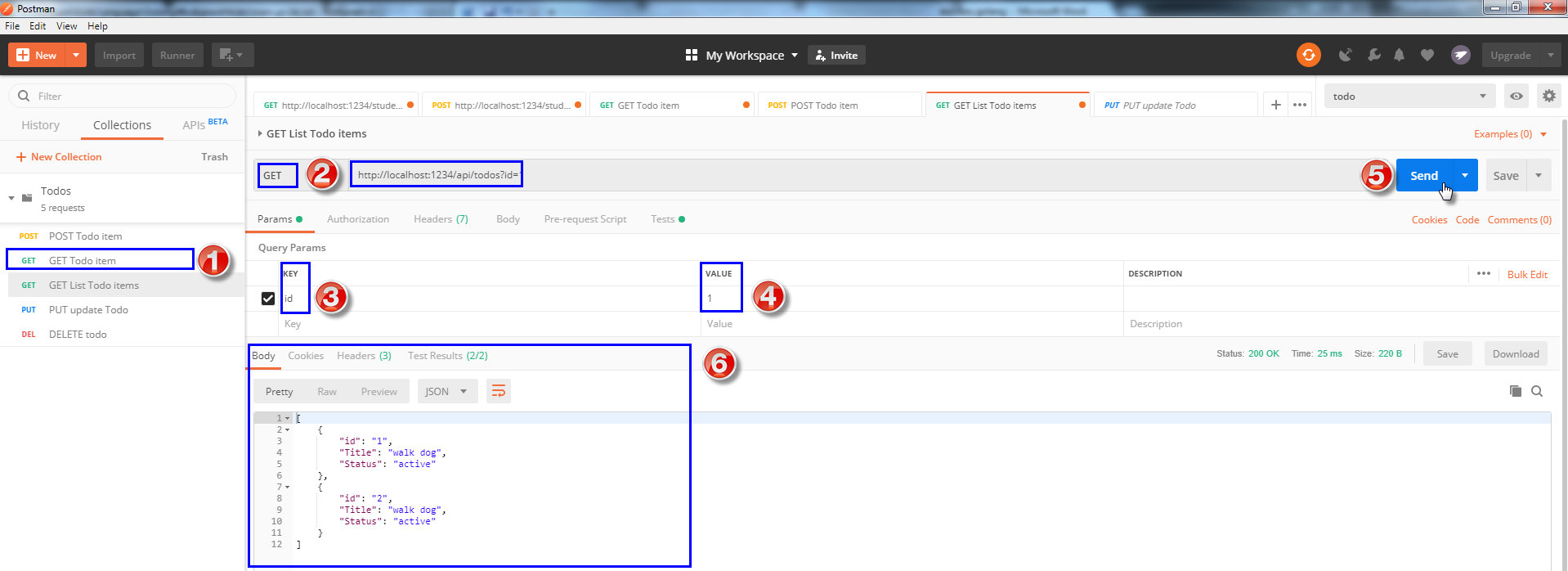
1. ทดสอบ postman :GET /api/todos โดย call function **getTodoHandler**

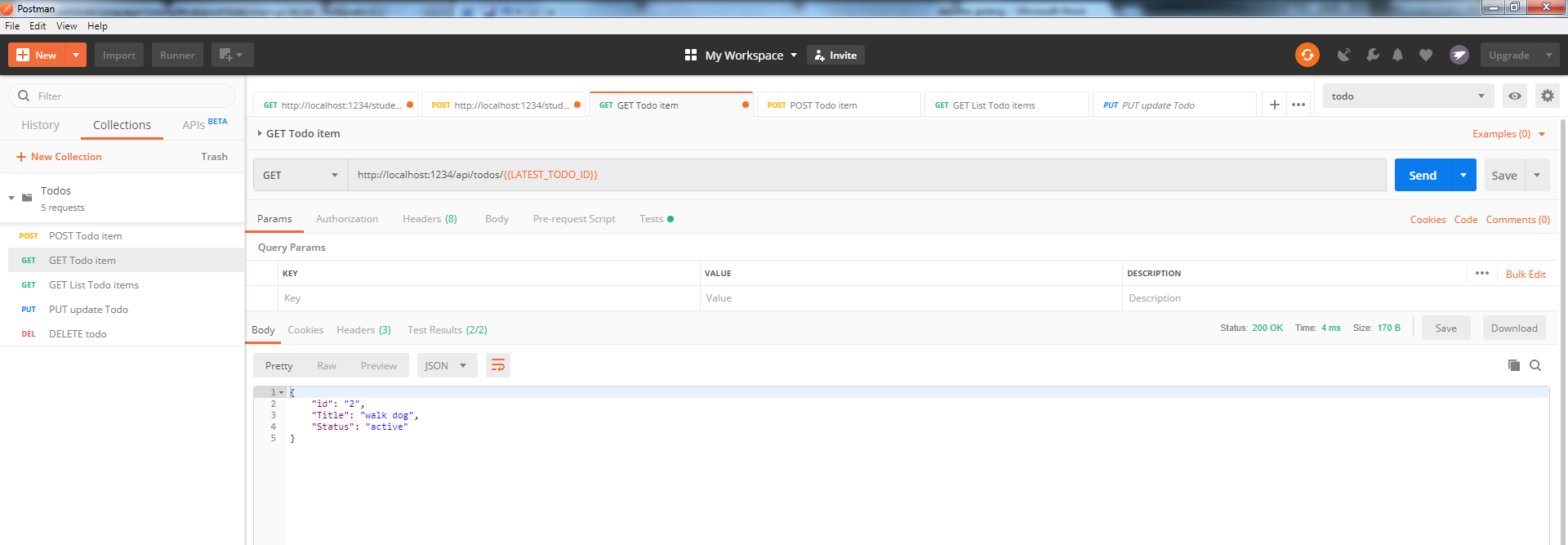
พบว่ามีการดึงข้อมูลที่ทำการทดสอบ postman : POST /api/todos มาแสดงเป็น list



1. ทดสอบ postman :GET /api/todos/:id โดย call function **getTodoByIDHandler**







1. เพิ่ม function **putTodosHandler** ใน main.go

func putTodosHandler(c \*gin.Context) {

    id := c.Param("id")

    todoVal := todos[id]

    if err := c.ShouldBindJSON(todoVal); err != nil {

        c.JSON(http.StatusBadRequest, gin.H{"error": err.Error()})

        return

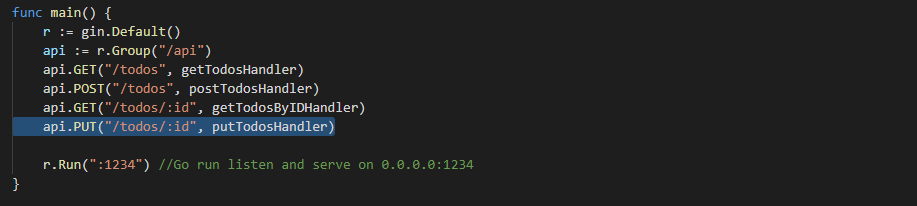
    }

    c.JSON(http.StatusOK, todoVal)

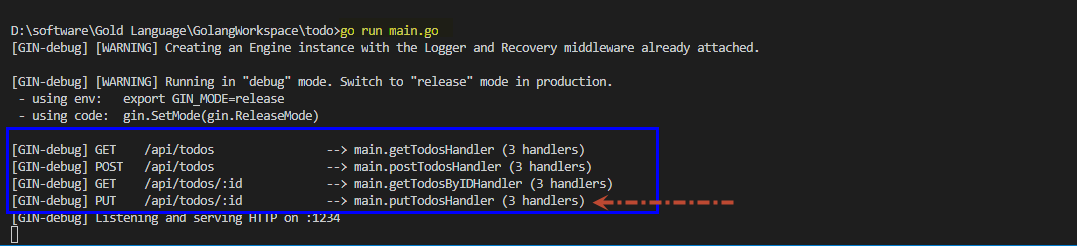
}

1. เพิ่ม code ใน func main เพื่อเรียก api url ของ function **putTodosHandler**

    api.PUT("/todos/:id", putTodosHandler)

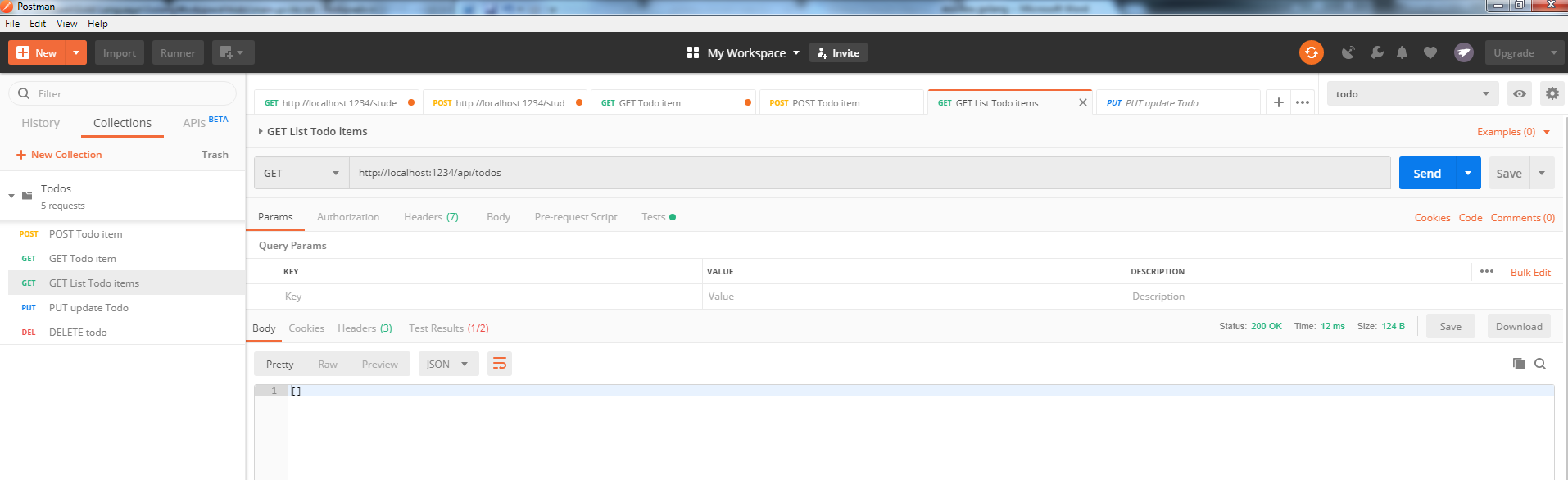


1. Run Go Lang application server

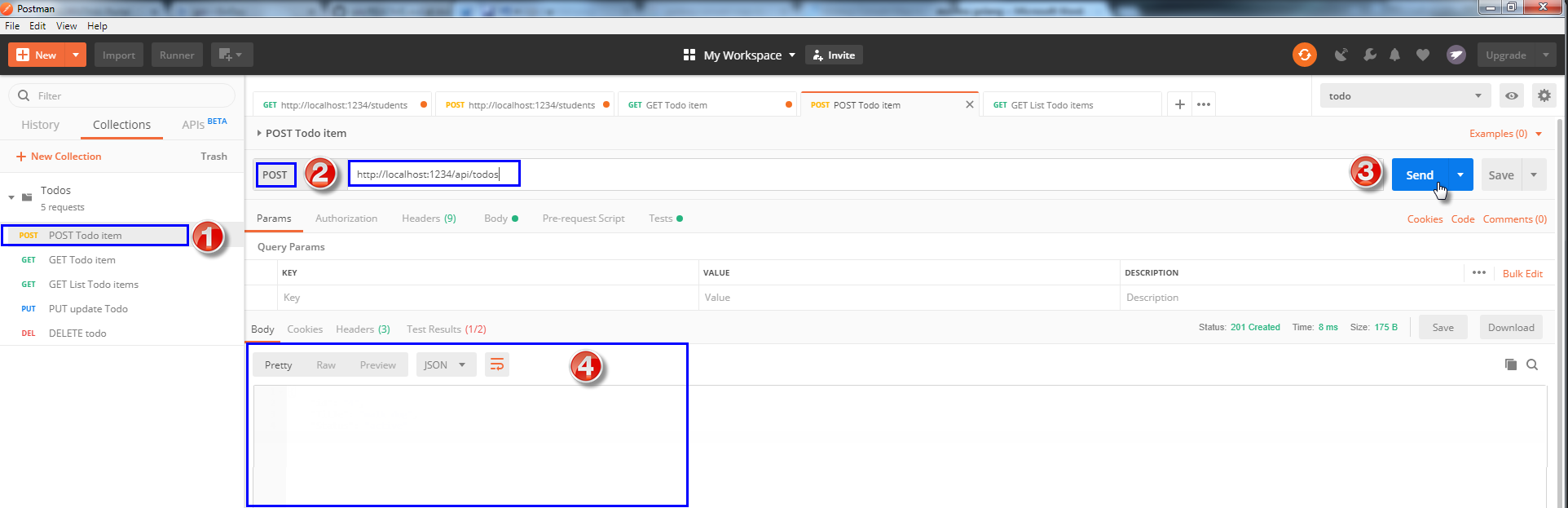


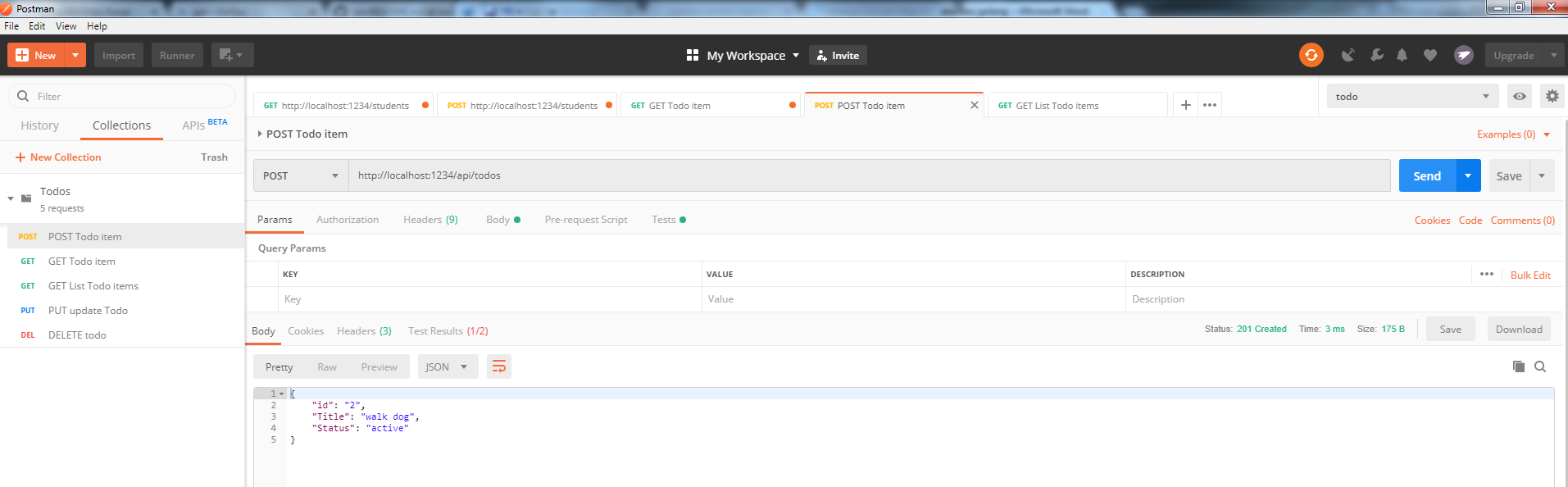
1. ทดสอบ postman :GET /api/todos โดย call function **getTodoHandler**

พบว่ามีการดึงข้อมูลที่ทำการทดสอบ postman : POST /api/todos มาแสดงเป็น list



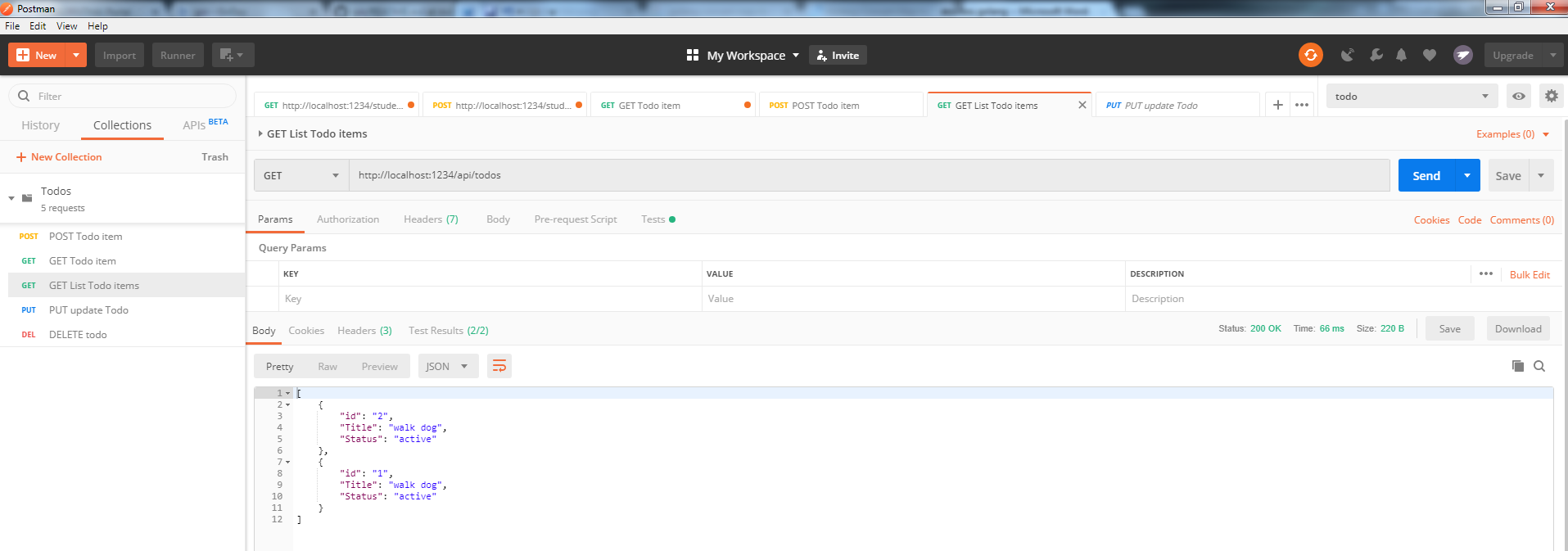
1. ทดสอบ postman : POST /api/todos โดย call function **postTodoHandler**





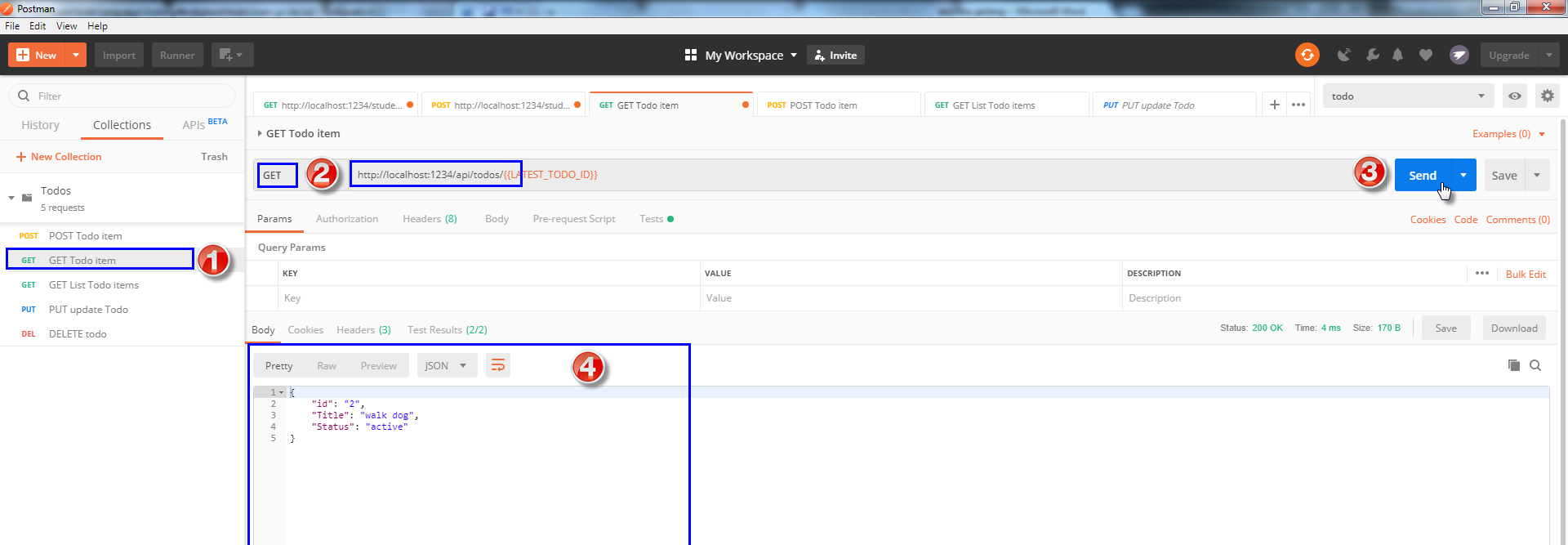
1. ทดสอบ postman :GET /api/todos โดย call function **getTodoHandler**

พบว่ามีการดึงข้อมูลที่ทำการทดสอบ postman : POST /api/todos มาแสดงเป็น list

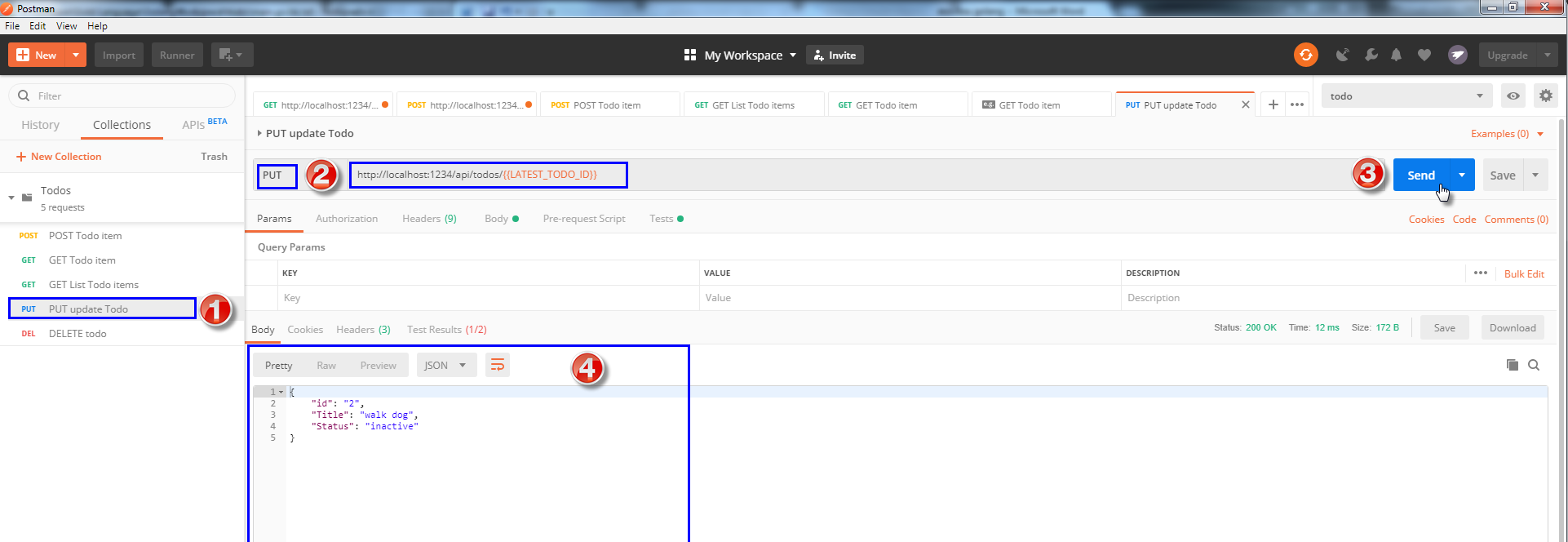


1. ทดสอบ postman :GET /api/todos โดย call function **getByIDTodoHandler**

พบว่ามีการดึงข้อมูลที่ทำการทดสอบ postman : POST /api/todos มาแสดงเป็น ค่าล่าสุด



1. ทดสอบ postman :GET /api/todos/:id โดย call function **putTodoByIDHandler**



1. เพิ่ม function **deleteTodosHandler** ใน main.go

func deleteTodosHandler(c \*gin.Context) {

    id := c.Param("id")

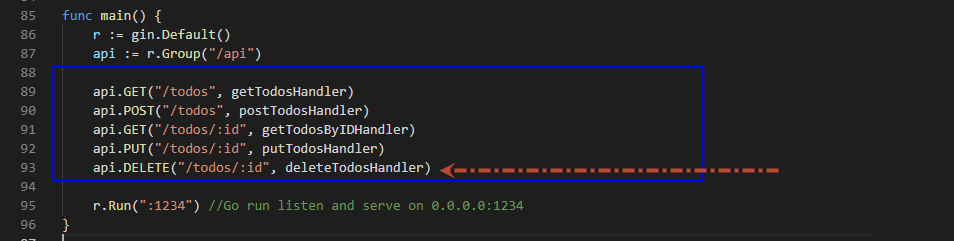
    delete(todos, id)

    c.JSON(http.StatusOK, gin.H{"status": "deleted success"})

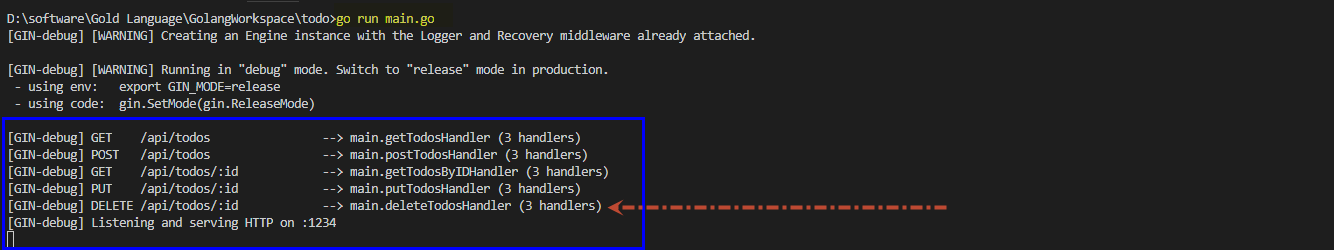
}

1. เพิ่ม code ใน func main เพื่อเรียก api url ของ function **deleteTodosHandler**

    api.DELETE("/todos/:id", deleteTodosHandler)

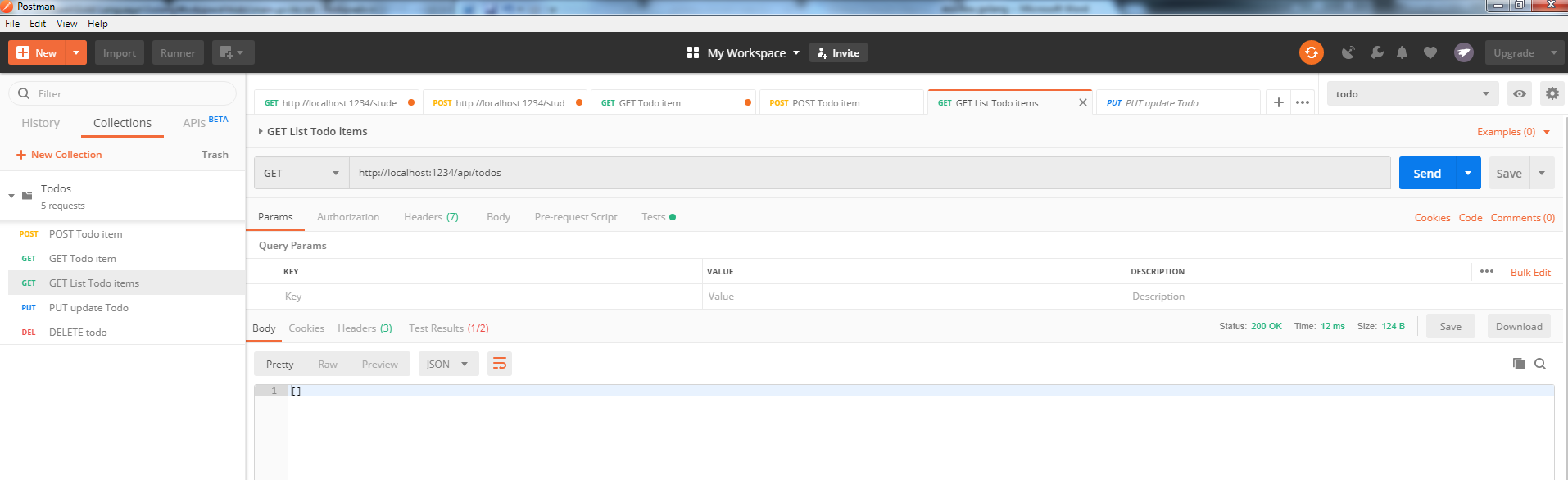


1. Run Go Lang application server

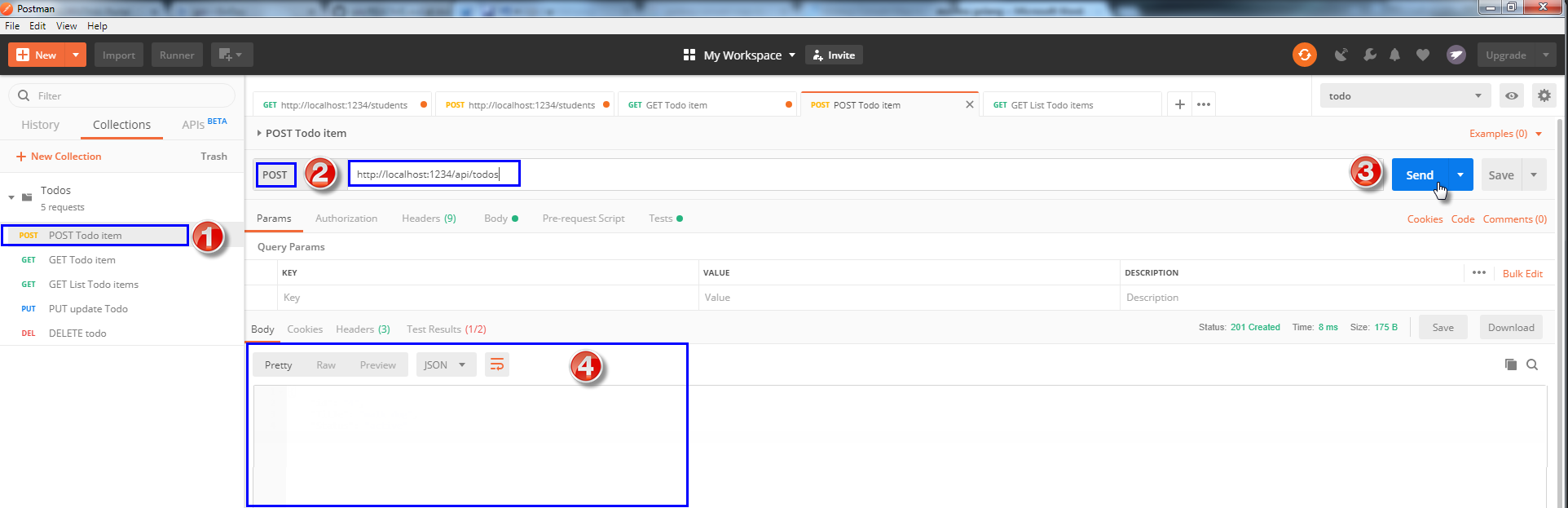


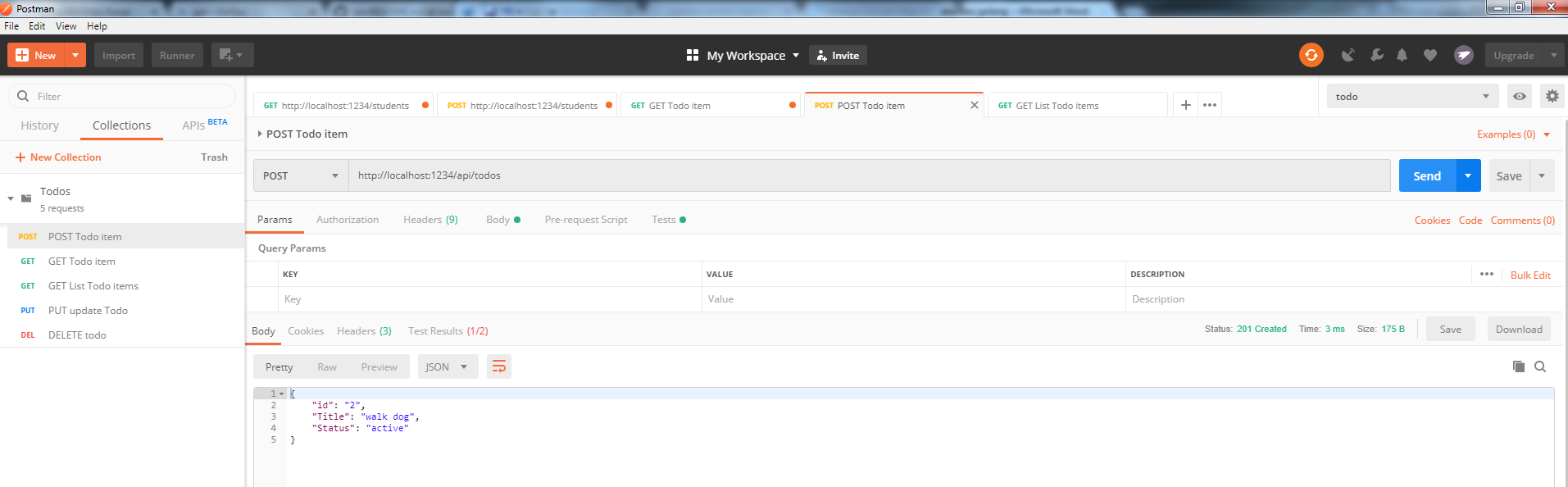
1. ทดสอบ postman :GET /api/todos โดย call function **getTodoHandler**

พบว่ามีการดึงข้อมูลที่ทำการทดสอบ postman : POST /api/todos มาแสดงเป็น list



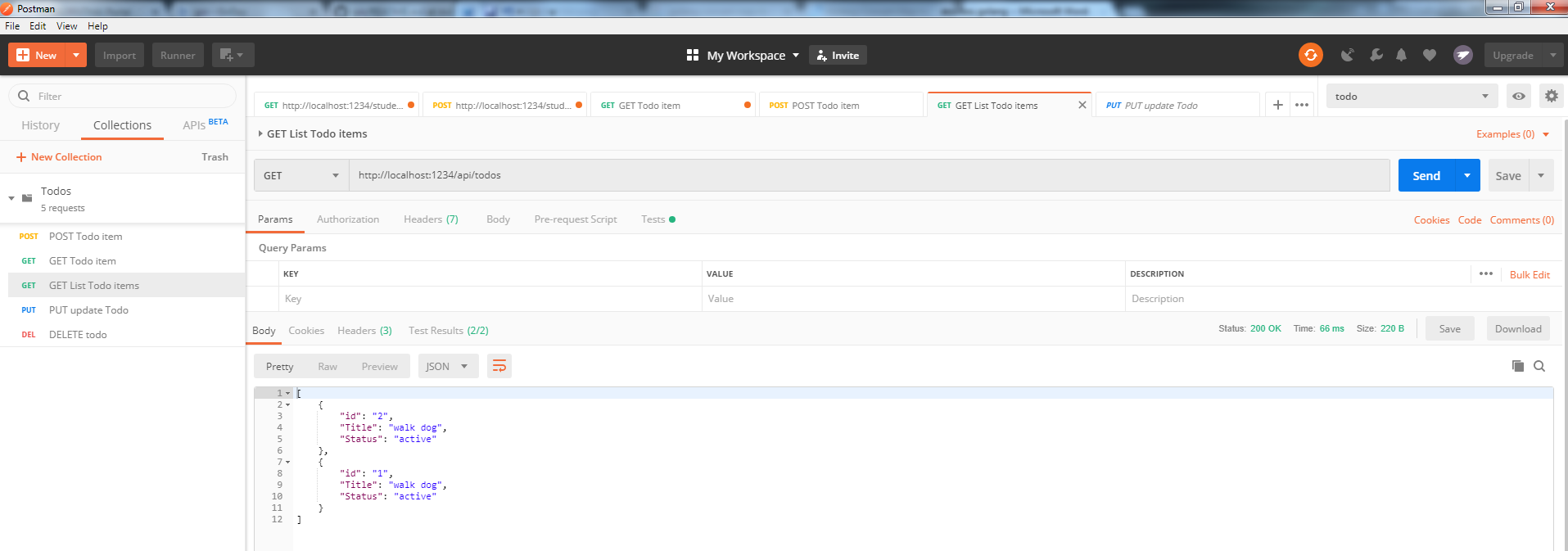
1. ทดสอบ postman : POST /api/todos โดย call function **postTodoHandler**





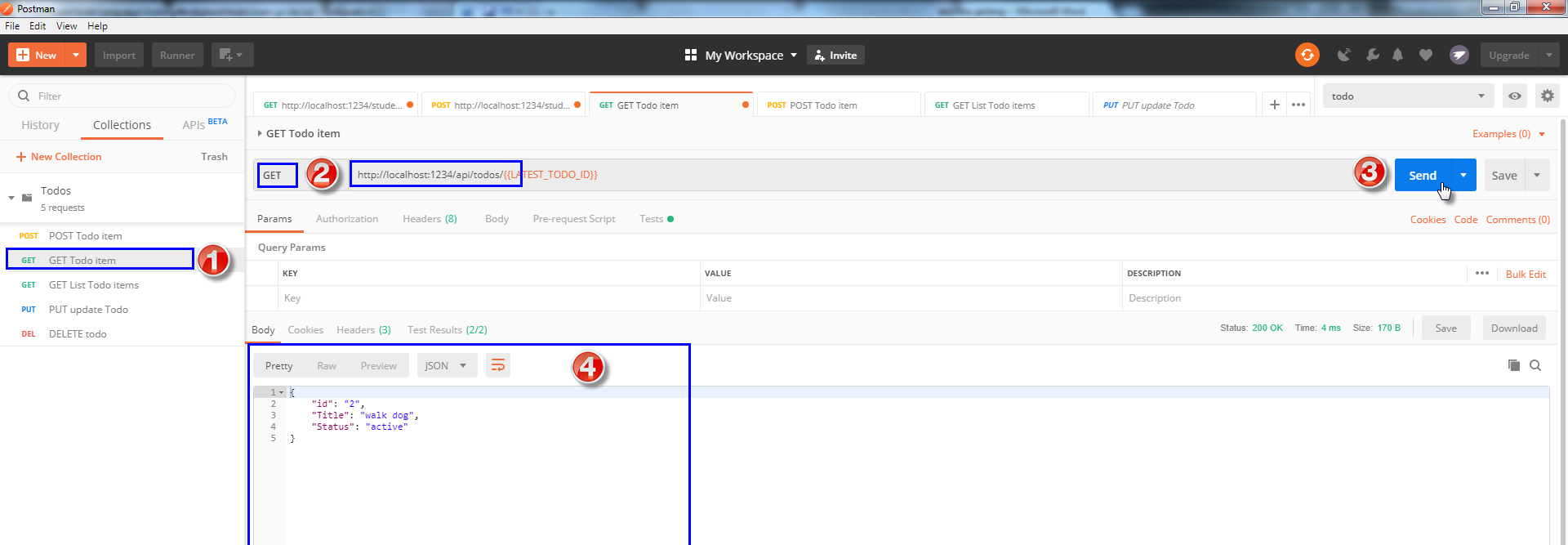
1. ทดสอบ postman :GET /api/todos โดย call function **getTodoHandler**

พบว่ามีการดึงข้อมูลที่ทำการทดสอบ postman : POST /api/todos มาแสดงเป็น list

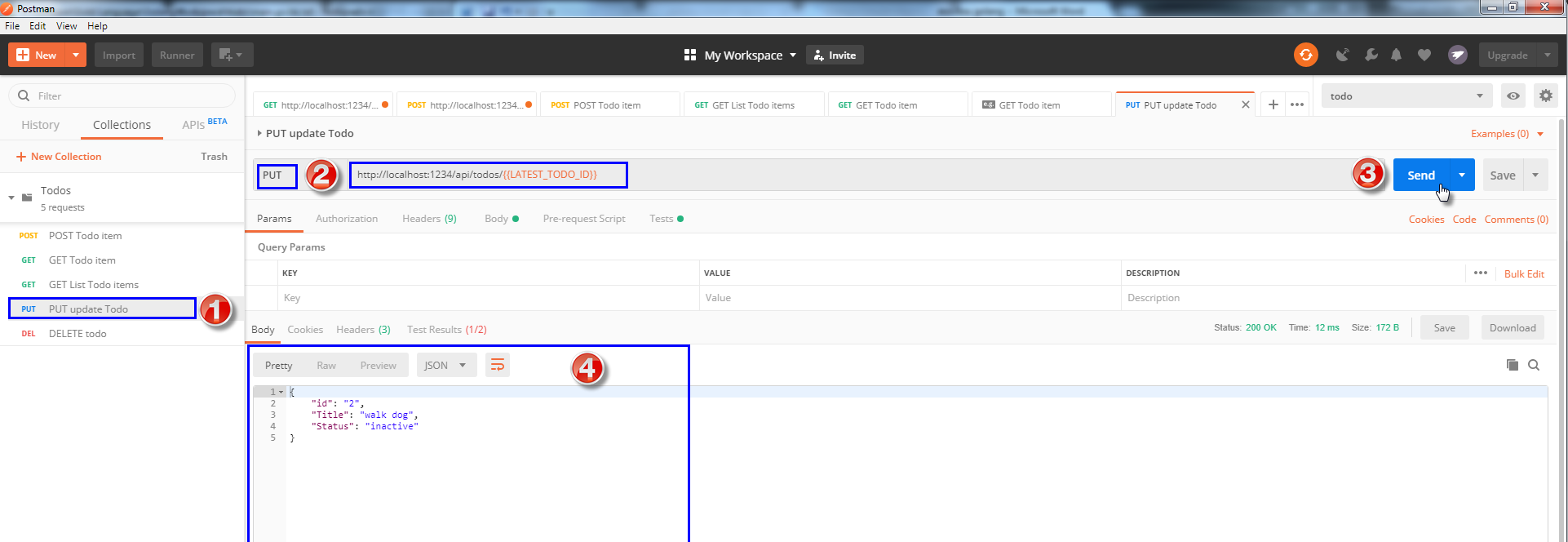


1. ทดสอบ postman :GET /api/todos โดย call function **getByIDTodoHandler**

พบว่ามีการดึงข้อมูลที่ทำการทดสอบ postman : POST /api/todos มาแสดงเป็น ค่าล่าสุด

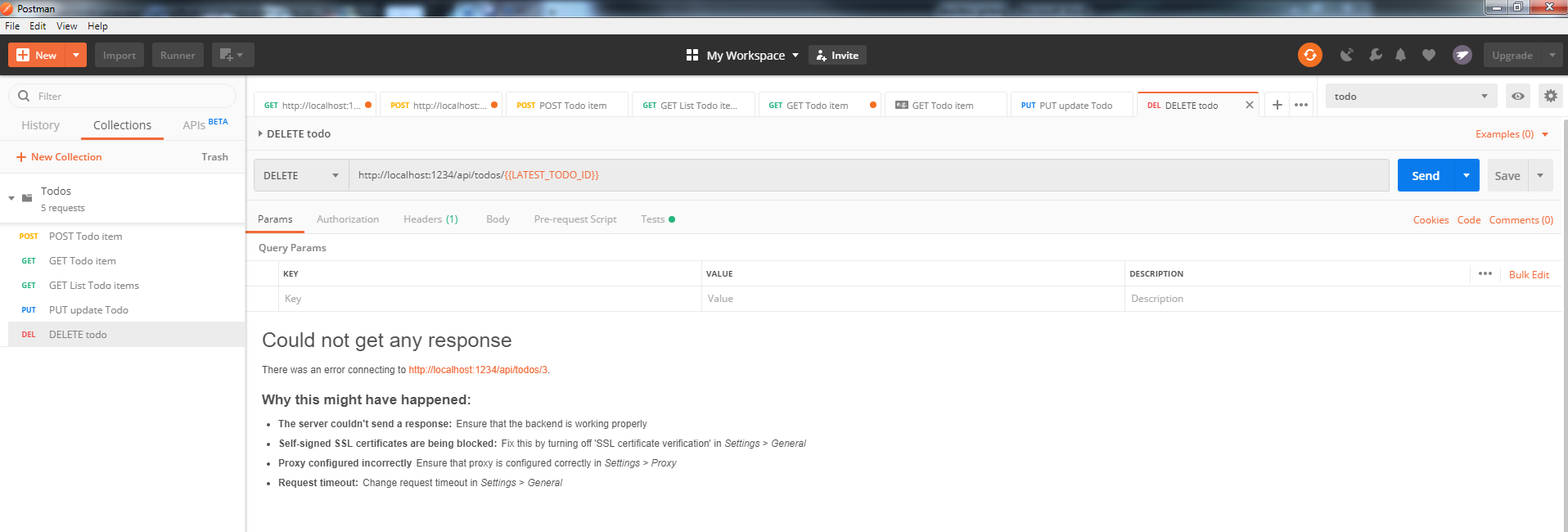


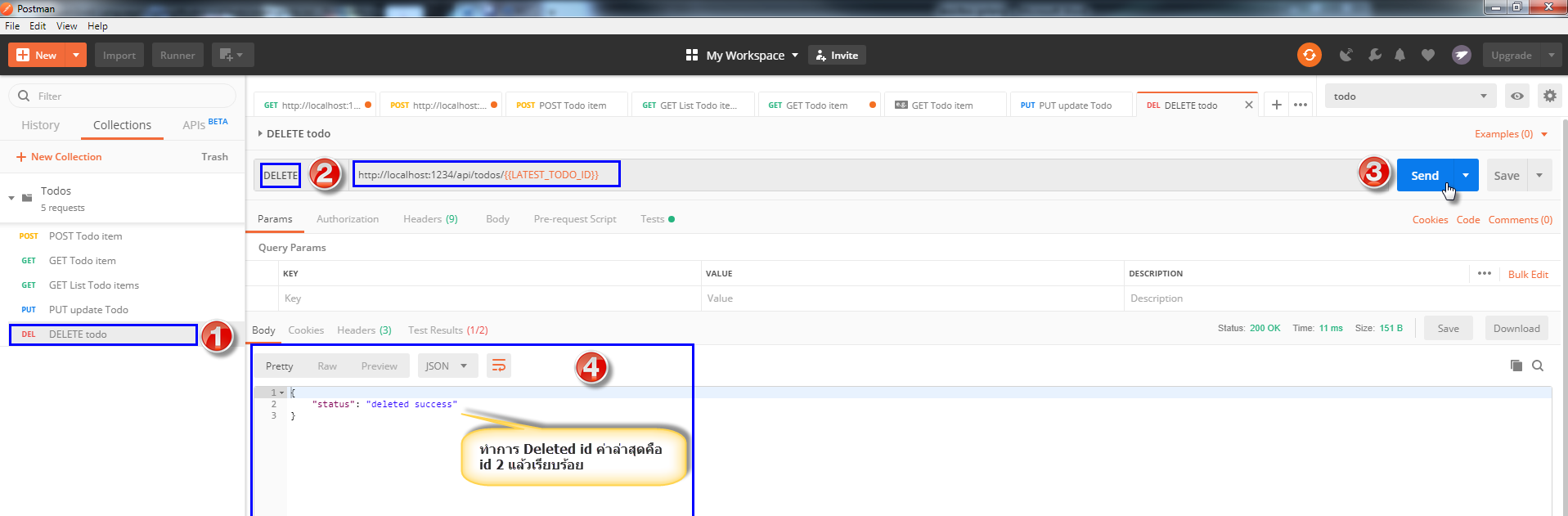
1. ทดสอบ postman :GET /api/todos/:id โดย call function **putTodoByIDHandler**



1. ทดสอบ postman :GET /api/todos/:id โดย call function **deleteTodoByIDHandler**

ระบบทำการ delete id ล่าสุด คือ id = 2 สำเร็จ





1. ทดสอบ postman :GET /api/todos โดย call function **getTodoHandler**

พบว่ามีการดึงข้อมูลที่ทำการทดสอบ postman : POST /api/todos มาแสดงเป็น list แต่จะมีเพียง id = 1 เท่านั้น เพราะ id = 2 ถูก deleted ไปแล้ว

