

Если вы видите что-то необычное, просто сообщите мне.

# Neo4j + triplet

```
package main

import (
    "context"
    "github.com/neo4j/neo4j-go-driver/v5/neo4j"
    "log"
    "os"
    "fmt"
)

type Triplet struct {
    Object    string
    Predicate string
    Subject   string
}

func main() {
    ctx := context.TODO()
    uri := "bolt://localhost:7687"
    username := "neo4j"
    password := "neo4jneo4j"
    var driver neo4j.DriverWithContext
    var session neo4j.SessionWithContext
    var triplet Triplet
    triplet.Object = os.Args[1]
    triplet.Predicate = os.Args[2]
    triplet.Subject = os.Args[3]
    driver, err := createDriver(uri, username, password)
    if err != nil {
        log.Fatal(err)
    }
    session, err = createSession(ctx, driver)
    if err != nil {
```

```

    []log.Fatal(err)
    []}
    []message, err := createTriplet(ctx, triplet, session)
    []if err != nil {
    [][]log.Fatal(err)
    []}
    []log.Println("message: ", message)
    []defer driver.Close(ctx)
    []defer session.Close(ctx)

    }

    func createDriver(uri, username, password string) (neo4j.DriverWithContext, error) {
    []driver, err := neo4j.NewDriverWithContext(uri, neo4j.BasicAuth(username, password, ""))
    []if err != nil {
    [][]return nil, err
    []}
    []return driver, nil
    }

    func createSession(ctx context.Context, driver neo4j.DriverWithContext)
    (neo4j.SessionWithContext, error) {
    []session := driver.NewSession(ctx, neo4j.SessionConfig{AccessMode: neo4j.AccessModeWrite})
    []return session, nil
    }

    func createTriplet(ctx context.Context, t Triplet, session neo4j.SessionWithContext) (string,
    error) {
    []var template = "MERGE (o:Node {name: \"%s\"}) " +
    [][]"MERGE (s:Node {name: \"%s\"}) " +
    [][]"MERGE (o)-[:%s]-(s)"

    []request := fmt.Sprintf(template, t.Object, t.Subject, t.Predicate)

    [], err := session.ExecuteWrite(
    [][]ctx,
    [][]func(transaction neo4j.ManagedTransaction) (any, error) {
    [][][]result, err := transaction.Run(ctx, request, map[string]any{})
    [][][]if err != nil {
    [][][]return nil, err

```

```
    }

    if result.Next(ctx) {
        return result.Record().Values[0], nil
    }

    return nil, result.Err()
})

if err != nil {
    return "", err
}

return "", nil
}
```

---

Revision #3

Created 2023-10-19 16:52:28 UTC by gasick

Updated 2023-10-19 19:03:18 UTC by gasick