

GoLang: Casting Interface to Type

Interface

```
package dao

import "credential-management/src/internal/model"

type Client interface {
    Connect() error
    Disconnect() error
    InsertLookup(tr *Transaction, credentialLookup *model.CredentialLookup) error
    UpdateLookup(tr *Transaction, credentialLookup *model.CredentialLookup) error
    GetLookupById(tr *Transaction, id string) (*model.CredentialLookup, error)
    GetLookupByField(tr *Transaction, field string, value string) ([]*model.CredentialLookup, error)
    DeleteLookup(tr *Transaction, id string) error
}
```

Type Implementing Interface

```
type MongoClient struct {
    connectURI string
    dbName     string
    client     *mongo.Client
    collection *mongo.Collection
}

func NewMongoClient(host string) Client {

    ...
    return client
}

func (m *MongoClient) Connect() error {
    ....
}
```

Convert from Interface to Type

```
client := NewMongoClient("host")

//cast to MongoClient to set mt.Coll
var mongoClient = client.(*MongoClient)

//perform actions only available in mongoClient type
mongoClient.collection = mt.Coll
```