Go

ODDS | Thaibev

Day 5

Coverage

- Echo With Pagination
- Tour project sx expo

Sort Direction

To specify the order of your results, pass an interface specifying the sort fields and direction to the ***SetSort()*** method of a read operation's options.

Ascending

An ascending sort orders your results from smallest to largest. To specify this sort, pass the field you want to sort by and 1 to the SetSort() method.

```
filter := bson.D{}

opts := options.Find().SetSort(bson.D{{"enrollment", 1}})

cursor, err := coll.Find(context.TODO(), filter, opts)
```

Descending

A descending sort orders your results from largest to smallest. To specify this sort, pass the field you want to sort by and -1 to the SetSort() method.

```
filter := bson.D{}

opts := options.Find().SetSort(bson.D{{"enrollment", -1}})

cursor, err := coll.Find(context.TODO(), filter, opts)
```

Aggregation

You can also include the $sort stage to specify a sort in an aggregation pipeline.

```
sortStage :=
       bson.D{{"$sort", bson.D{{"enrollment", -1}, {"title", 1}}}}
cursor, err := coll.Aggregate(context.TODO(),
mongo.Pipeline{sortStage})
if err != nil {
    panic(err)
}
```

Regex query with the Go driver

```
bson.E{
        Key: key, Value: primitive.Regex{
            Pattern: searchKeyword,
            Options: "im",
        },
}
```

Excelize

Excelize is a library written in pure Go providing a set of functions that allow you to write to and read from XLAM / XLSM / XLSX / XLTM / XLTX files.

```
$ go get github.com/xuri/excelize/v2
```