

Write a small js script called `loanapps.js` to run in mongosh so that it continually inserts records that look like the example below, you can use the fixed values. Have the script insert documents into the database with a majority write and when it's safely stored in MongoDB, print out the message "Recorded application " + `_id`. Have your script drop the collection it's using before it starts. Let the shell set the `_id` field for you.

```
{
  _id: ObjectId(),
  name: "A. Customer",
  address : "1600 Hoover Avenue",
  type: "Loan Application",
  value: 50000
}
```

Put your script in the box below.

```
record = {
  name: "A. Customer",
  address : "1600 Hoover Avenue",
  type: "Loan Application",
  value: 50000
}

db.loans.drop()
while(true) {
  var v = db.loans.insertOne(record,{writeConcern: { w: "majority" }})
  print ("Recorded application " + v.insertedId)
}
```

Save your script on your application server as `loanapps.js`, now run it using the command

```
mongosh
"mongodb://mongoadmin:passwordone@mongodb-emea,mongodb-us,mongodb-apac/?replicaSet=TestRS" loanapps.js | tee loanapps.txt
```

Using an editor or the provided IDE (your choice), write a second script called `mailer.js` to look for 'New' loan applications and print out their `_id` field. It should run in a loop querying for any document where `_id` is greater than the last one seen and print out "Sending an mail to A.Customer acknowledging application" + `_id`.

Save `mailer.js` and put it in the box below

```
lastidseen=MinKey

while(true) {
  var cursor = db.loans.find({_id:{$gt:lastidseen}})
  while (cursor.hasNext()) {
    var v = cursor.next()
    var id = v._id
    print("Sending a mail to A.Customer about application " +id)
    lastidseen = id;
  }
}
```

Once saved, run it a new VSC terminal as shown below - you should, in two terminal tabs have an output showing 'recording' and 'mail sending'

```
mongosh
"mongodb://mongoadmin:passwordone@mongodb-emea,mongodb-us,mongodb-apa
c/?replicaSet=TestRS" mailer.js | tee mailssent.txt
```

Now kill the MongoDB process running on the primary. You can use `rs.status()` in `mongosh` to identify which server is the primary - although in this lab's configuration it should always be the EMEA server..

```
kill -9 <mongod_pid>
```

Note that this will kill the `mongosh` shells as well.