

MongoDB Restore from Zip File

. zMongoRestore.sh CRM CRM.zip

```
#!/bin/sh

Escape="\033";
Reset="${Escape}[0m";
BlackF="${Escape}[30m";
RedF="${Escape}[31m";
GreenF="${Escape}[32m";
YellowF="${Escape}[33m";

zConfirmYesNo () {
    zCommand=$*

    printf "\n"
    printf "${GreenF} '$zCommand' will be executed. Are you sure? [Yes|No]: ${BlueF}Yes ${RedF}No"
    read zComfirm

    if [[ "$zComfirm" =~ ^(y|Y|Yes|YES|e|E|Evet|EVET)$ ]]
    then
        printf "${RedF} $zCommand ${Reset}\n"
        $zCommand
    elif [[ "$zComfirm" =~ ^(n|N|No|NO|h|H|Hay?r|HAYIR)$ ]]
    then
        printf "${RedF} $zCommand ${Reset}\n"
    else
        printf "${RedF} $zCommand ${Reset}\n"
        $zCommand
    fi
}

mongoRestore () {
    zDatabase=$1
    zDatabaseZipFile=$2

    #####

    if [[ "$zDatabase" = "" ]]
    then
        echo 'show dbs' | mongo
        printf "${RedF} Database name is empty! Please enter database name: ${BlueF}"
        read zDatabase
        printf "${Reset}\n"
    else
        printf "${RedF} Pre Entered zDatabase:$zDatabase ${Reset}\n"
    fi

    if [[ "$zDatabaseZipFile" = "" ]]
    then
        ls -ahltr
        printf "${RedF} Database file name is empty! Please enter database file name: ${BlueF}"
        read zDatabaseZipFile
        printf "${Reset}\n"
    else
        printf "${RedF} Pre Entered zDatabaseZipFile:$zDatabaseZipFile ${Reset}\n"
    fi

    zToday=`date +%Y_%m_%d`
    zHomeDir=`echo ~`
}
```

```
zBackUpFolder="$zHomeDir/$zDatabase-$zToday"
```

```
printf "zDatabase:${GreenF} $zDatabase ${Reset}\n"
printf "zDatabaseZipFile:${GreenF} $zDatabaseZipFile ${Reset}\n"
printf "zBackUpFolder:${GreenF} $zBackUpFolder ${Reset}\n"
```

```
zCommand="unzip $zDatabaseZipFile -d $zDatabase-$zToday"
zConfirmYesNo $zCommand
```

```
zCommand="ls -lhSr --color=auto $zBackUpFolder/"
zConfirmYesNo $zCommand
```

```
#####
# Drop Database
```

```
printf "\n ${GreenF} Do you want drop all collections from: $zDatabase [Yes|No]: ${BlueF}"
read zDropDatabase
printf "${Reset}\n"
```

```
if [[ "$zDropDatabase" =~ ^(y|Y|Yes|YES)$ ]]
then
    printf "${RedF} Database all collections will be dropped! ${Reset}\n"
    mongo $zDatabase --eval 'printjson(db.dropDatabase())'
else
    printf "${GreenF} Database will not be dropped. ${Reset}\n"
fi
```

```
#####
# Drop Collections
```

```
printf "${GreenF} Do you want to first drop then insert collections? [Yes|No]: ${BlueF}"
read zDropCollection
printf "${Reset}\n"
zDropDatabaseCommand=""
if [[ "$zDropCollection" =~ ^(y|Y|Yes|YES)$ ]]
then
    printf "${RedF} Database will be dropped! ${Reset}\n"
    zDropDatabaseCommand="--drop"
elif [[ "$zDropCollection" =~ ^(n|N|No|NO)$ ]]
then
    printf "${GreenF} Database will NOT be dropped. All records will be inserted. ${Reset}\n"
    zDropDatabaseCommand=""
else
    printf "${RedF} Database will be dropped! ${Reset}\n"
    zDropDatabaseCommand="--drop"
fi

zCommand="mongorestore $zDropDatabaseCommand --db $zDatabase $zBackUpFolder/"
zConfirmYesNo $zCommand
```

```
#####
```

```
printf "\n ${GreenF} Do you want to repair DB $zDatabase [Yes|No]: ${BlueF}Yes"
read zRepairDatabase
printf "${Reset}\n"

if [[ "$zRepairDatabase" =~ ^(n|N|No|NO)$ ]]
then
    printf "${RedF} Database will NOT be repaired! ${Reset}\n"
else
    printf "${GreenF} Database will be repaired! ${Reset}\n"
    mongo $zDatabase --eval 'printjson(db.repairDatabase())'
fi
```

```
#####
```

```
zCommand="ls -l --color=auto $zHomeDir"
zConfirmYesNo $zCommand

zCommand="rm -r $zBackUpFolder/"
zConfirmYesNo $zCommand

zCommand="ls -l --color=auto $zHomeDir"
$zCommand

echo "show dbs" | mongo
```

```
}
```

```
zDatabase=$1
zDatabaseZipFile=$2
```

```
echo "usage: '. zMongoRestore.sh CRM CRM.zip'"
printf "\n\n"

mongoRestore $zDatabase $zDatabaseZipFile
```

Created Wed, Jun 21, 2017 10:40 AM by Ahmet Faruk Bişkinler
Last Updated Tue, Jul 4, 2017 1:38 PM by Ahmet Faruk Bişkinler