Propel Cutover Procedures

**Contents**

[Cutover Procedures 2](#_Toc161670497)

[How to export collections data from a Propel Database 2](#_Toc161670498)

[How to drop Propel Database 2](#_Toc161670499)

[How to create Propel Database 3](#_Toc161670500)

[How to Import collections to a Propel Database 3](#_Toc161670501)

# Cutover Procedures

## How to export collections data from a Propel Database

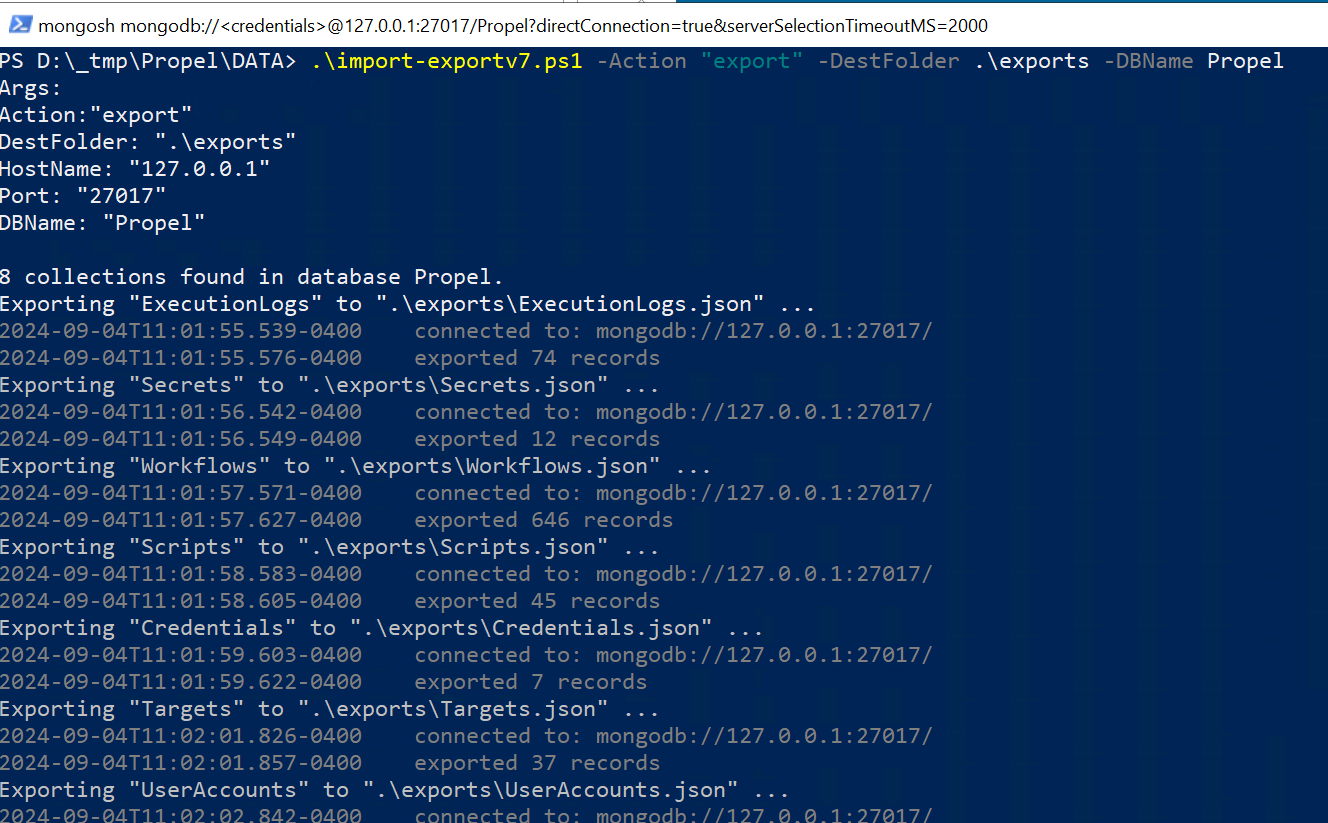
To export the collections from a Propel Database you must run the script “distrib\cutover\import-exportv{Mongo version}.ps1”, you must choose the one that match the Mongo DB version currently installed.

Open a PowerShell console and run the following:

.\import-exportv7.ps1 -Action "export" -DestFolder .\exports -DBName Propel

Just as example we are using MongoDB Version 7.0 script.

The script is going to create the “exports” folder if not exists and locate there the Json files with all the exported data.



## How to drop Propel Database

To do this we need to open a PowerShell console change the working directory to “\distrib\cutover” and run the following command:

mongosh --eval "var adu='DBA'; var adp='Admin password'; var apu='PropelUser'" **drop-database.js**

That script is going to drop the user “PropelUser” as also the Propel database.

## How to create Propel Database

Follow these steps to create the Propel database:

Start a PowerShell console, change the working directory to the place were the scripts for the current version of Propel are located and run this command:

mongosh --eval "var adu='DBA'; var adp='Admin password'; var apu='PropelUser'; var app='Propel user password';" **0000-01-create-db-and-user.js**

This is going to create the propel DB and the Propel user with the provided password, (Recall that password will be changed every time you run the Propel installer).  
Next step is to create the propel Collections and indexes. So run the below command:

mongosh --eval "var adu='DBA'; var adp='Admin password'; var apu='PropelUser'; var app='Temporal Propel user password';" **0000-02-db-script.js**

## How to Import collections to a Propel Database

To import collections data to a Propel Database you must **first ensure the collections are created and they are empty** or at least they are not having the data you try to import. Otherwise, the import process will fail returning a duplicated ID error.

You must run the script “distrib\cutover\import-exportv{Mongo version}.ps1”, you must choose the one that match the Mongo DB version currently installed.

Open an **elevated PowerShell console** and run the following:

.\import-exportv7.ps1 -Action "import" -DestFolder .\exports -DBName Propel

**NOTE: For this step, be sure the console is elevated. Otherwise, the script is going to fail when requesting the credentials.**

Just as example we are using MongoDB Version 7.0 script.

The script is going to import all the collections data from the folder indicated in the “DestFolder” parameter.

A screenshot of a computer screen

Description automatically generated

## How to Change user role and lock status using Mongosh

This is particularly useful when running Propel in a machine without Compass or other DB admin software.

To do This you must start a PowerShell session and type “mongosh”, once in the Mongo Shell you must execute the following commands:

use admin

db.auth("DBA", "HERE the DB admin password")

use Propel

db.UserAccounts.updateOne({name: "HERE the account name to change"}, {$set:{ lockedSince: null, role: "Administrator"}})

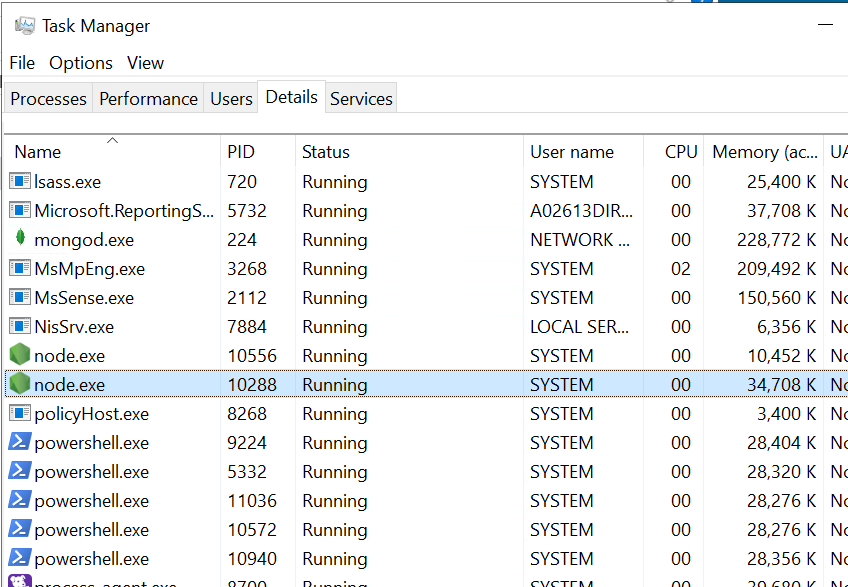
Doing this your user will be automatically unlocked and changed to the Administrator role. You can also set the value “User” for the role otherwise.

You can verify the changes by running:

db.UserAccounts.find({name: "HERE the account name"})

## Migration Known Issues

In some weird cases after install Propel will display some error toast indicating a data error and you will not be able to see any data in the app. If you check the “propel.err.log” in “C:\Propel\propel-api\daemon”, You will see a message similar to this: “Error: listen EADDRINUSE: address already in use :::3000”. This is caused by the previous instance of the Propel API service already running in the server. You can confirm this by checking in the task manager:



Also, if you check the Propel service you can see it fail to start after the installation:



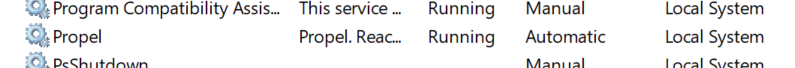
So, what is actually running is the previous instance of the Propel service that for some reason the installer failed to stop and uninstall properly.

How to proceed:

1st – Kill both Node.JS processes in the Task Manager.

2nd – Try to start the Propel service again.

If all is good and after a few seconds Propel service must remain running:



Also, you can check the check the “propel.out.log” in “C:\Propel\propel-api\daemon” where if all is fine you will see something like this:  
