Dumpdata exporta los datos de toda la base de datos, tambien puedes escoger una aplicacion en especifico o algun modelo de alguna aplicacion.

* Retorna todos los datos de la base de datos

Python manage.py dumdata

* Retorna los datos con identacion

Python manage.py dumdata –indent <numero>

* Vuelca los datos en un archive

Python manage.py dumdata > archivo.json/txt,etc

* Vuelca los datos de los modelos de la app erp

Python manage.py dumdata –indent 2 [appname] > archive.json

Python manage.py dumdata –indent 2 [appname].[modelname] > archive.json

* Volcar dos aplicaciones

Python manage.py dumdata –indent 2 [appname] [appname] > archive.json

* Excluir apps o modelos del volcado

Python manage.py dumdata –indent 2 [appname] –exclude=[contentype]> archive.json

# LOADDATA

Cuando se encuentra alguna clave primaria ya creada, este loaddata actualiza los datos.

* Importar datos

Python manage.py loaddata database.json

# Importando de SQLITE a POSTGRES

Ya después de haber cambiado la configuración y haber volcado la base de datos en sqlite, vamos a crear la base de datos en POSTGRES y haremos la migración.

Si subimos los datos asi como asi esto nos dara un error, ya que django crea registros de forma automática como el de los permisos. Lo idóneo es borrar esa data y dejar la base de dato en 0 y reiniciar los incrementadores.

* Postgresql
* delete from public.auth\_permission;
* ALTER SEQUENCE public.auth\_permission\_id\_seq RESTART WITH 1;
* delete from public.django\_content\_type;
* ALTER SEQUENCE public.django\_content\_type\_id\_seq RESTART WITH 1;