

Setting Up Geotemporal API V3

(Work in Progress)

refer to [Geostreams-api-v3 - Local](#)

1. Install Java 1.8
2. Install git
3. Clone repository from: <https://opensource.ncsa.illinois.edu/bitbucket/projects/GEOD/repos/geostreams-api-v3/browse> (git clone repositoryUrl)
4. In IntelliJ
 - a. Import project from external model
 - b. select sbt
 - c. Leave default options
5. Install Scala and Play plugins on IntelliJ
6. Download postgres from your operating system: <https://www.postgresql.org/download/> or <https://postgresapp.com/> (Mac)
7. Clone a database from one of the DEV machines:

Ingest Database

```
# from https://gist.github.com/brock/63830f11c0945f82f9ea
curl -o pg_extract.sh https://gist.github.com/brock/63830f11c0945f82f9ea/raw/ff164b14311de62ee475d1612f1e80e4df6edc15/pg_extract.sh
chmod 755 pg_extract.sh

# get the database and unzip
scp isda:/home/shared/clowder/seagrant-dev/postgresql/dumpall.sql.gz seagrant-dev.sql.gz
gunzip seagrant-dev.sql.gz

# split the database (may take a while)
./pg_extract.sh seagrant-dev.sql
# should see names of databases:
# Evaluating geostream...

# create and load database
## In the line below, 'clowder' can be updated for your username on your machine
## 'geostream' will be the name of the database.
## It can be the name that you prefer, just use the same name throughout these instructions
## Note: This may not work as expected if you name with a "dash"
## - you may have to manually copy the contents of 'geostream' to your chosen database name
createdb geostream -O clowder
psql -d geostream < seagrant-dev.sql
```

8. Start your postgres db
9. In your geostreams-api-v3 directory you need to create and edit a file `conf/application.conf`
 - a. Copy the entirety of `conf/reference.conf` to `conf/application.conf` ensure proper functionality
 - b. Update the `play.filters.host` in `conf/application.conf` to contain the proper allowed host for Geodashboard
 - c. the `application.conf` on `seagrant-dev` has a copy under `yanzhao3/`
 - d. If you changed the name of the database
 - i. i.e.: Instead of `geostreams` you use something like `seagrant-dev`
 - ii. Add: `db.default.url="jdbc:postgresql://localhost/seagrant-dev"`
10. In a terminal, `cd` to the `geo-temporal-api-v2` repository and execute `sbt run` to start Clowder
 - a. If you use Geodashboard, you may need to update links in the Geodashboard code
 - b. Locations will vary based upon the version of Geodashboard
11. to setup `geo-temporal-api-v2` on server with your local build (`application.conf` & `messages.en` is under your local folder)

```
sudo rm -rf geo-temporal-api-v2/*
sudo cp -r ~/geostreams-3.0.0-alpha.1/* geo-temporal-api-v2/
sudo cp ~/application.conf geo-temporal-api-v2/conf/
sudo cp ~/messages.en geo-temporal-api-v2/conf/

sudo chown clowder: geo-temporal-api-v2/bin/
sudo chown clowder: geo-temporal-api-v2/lib/
sudo chown clowder: geo-temporal-api-v2/conf/
sudo chmod +r+w geo-temporal-api-v2/lib/*
sudo chmod +r+w geo-temporal-api-v2/conf/*
sudo chmod +r+x geo-temporal-api-v2/bin/geostreams
sudo systemctl restart geostreams.service
sudo journalctl -xe
sudo lsuf -i -P -n | grep LISTEN
```

