# Recos-Injector

Recos-Injector is a streaming event processor used to build input streams for GraphJet-based services. It is a general-purpose tool that consumes arbitrary incoming event streams (e.g., Fav, RT, Follow, client\_events, etc.), applies filtering, and combines and publishes cleaned up events to corresponding GraphJet services. Each GraphJet-based service subscribes to a dedicated Kafka topic, and Recos-Injector enables GraphJet-based services to consume any event they want.

## How to run Recos-Injector server tests

You can run tests by using the following command from your project's root directory:

$ bazel build recos-injector/...

$ bazel test recos-injector/...

## How to run recos-injector-server in development on a local machine

The simplest way to stand up a service is to run it locally. To run

recos-injector-server in development mode, compile the project and then

execute it with `bazel run`:

$ bazel build recos-injector/server:bin

$ bazel run recos-injector/server:bin

A tunnel can be set up in order for downstream queries to work properly.

Upon successful server startup, try to `curl` its admin endpoint in another

terminal:

$ curl -s localhost:9990/admin/ping

pong

Run `curl -s localhost:9990/admin` to see a list of all available admin endpoints.

## Querying Recos-Injector server from a Scala console

Recos-Injector does not have a Thrift endpoint. Instead, it reads Event Bus and Kafka queues and writes to the Recos-Injector Kafka.

## Generating a package for deployment

To package your service into a zip file for deployment, run:

$ bazel bundle recos-injector/server:bin --bundle-jvm-archive=zip

If the command is successful, a file named `dist/recos-injector-server.zip` will be created.