# Target Output

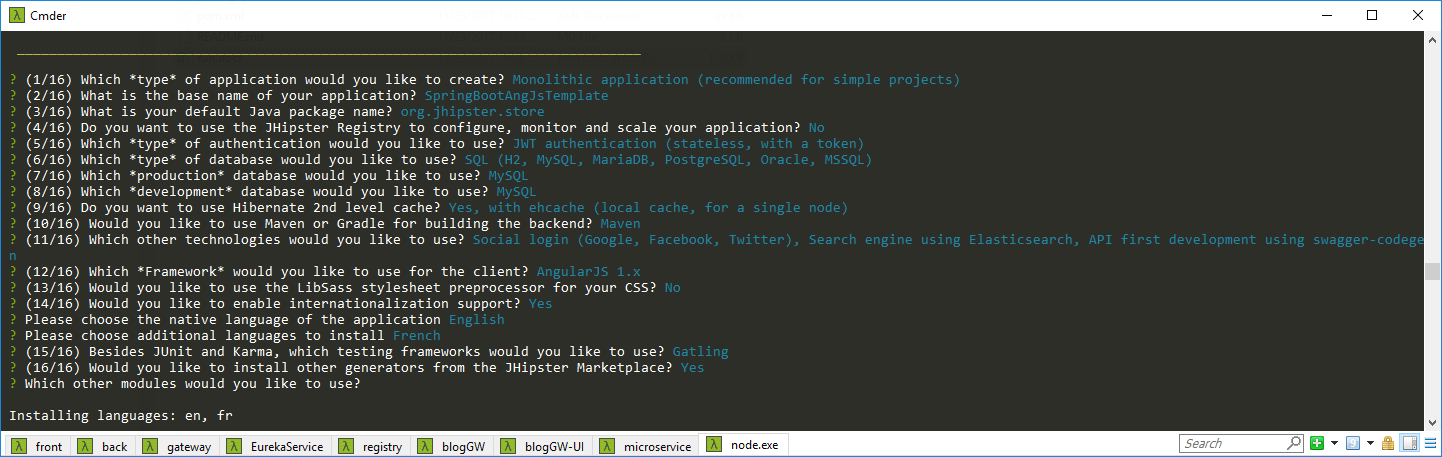
A source template with frontend is AngularJS 1, Backend is Java Spring Boot.

# Setup

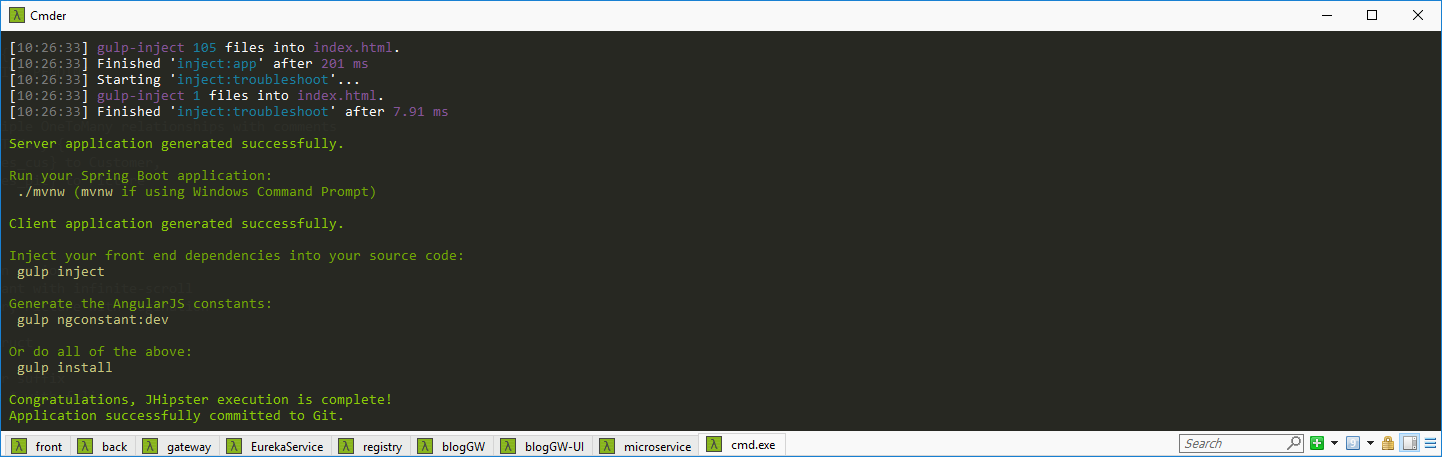
## Create template project

Run: *jhipster*

(monolithic, MySQL separated, Social Login, Search Engine Elasticsearch)



Finish:



Run: *gulp install*

## Generate entity

entity Restaurant {

id Long,

name String,

status StatusEnum

}

entity Customer {

id Long,

name String,

phone String,

address String,

buyDate ZonedDateTime,

avatar ImageBlob,

male Boolean

}

entity Picture {

id Long,

img ImageBlob

}

enum StatusEnum {

ACTIVE,INACTIVE,PENDING

}

// defining multiple OneToMany relationships with comments

relationship OneToMany {

Restaurant{res\_cus} to Customer,

Restaurant{res\_pic} to Picture

}

// Set pagination options

paginate Restaurant with infinite-scroll

paginate Customer, Picture with pagination

dto \* with mapstruct

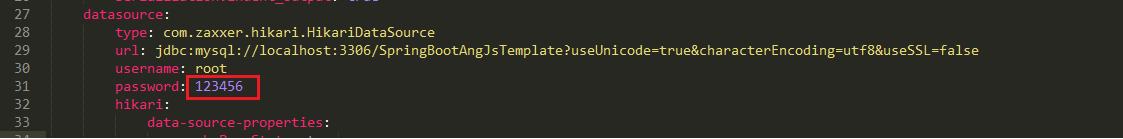
// Set an angular suffix

// angularSuffix \* with felix

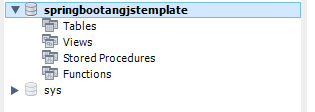
Run import file (Only generate entity, not generate table & column): jhipster import-jdl jhipster-jdl.jh

## Config MySQL

*src\main\resources\config\application-dev.yml*



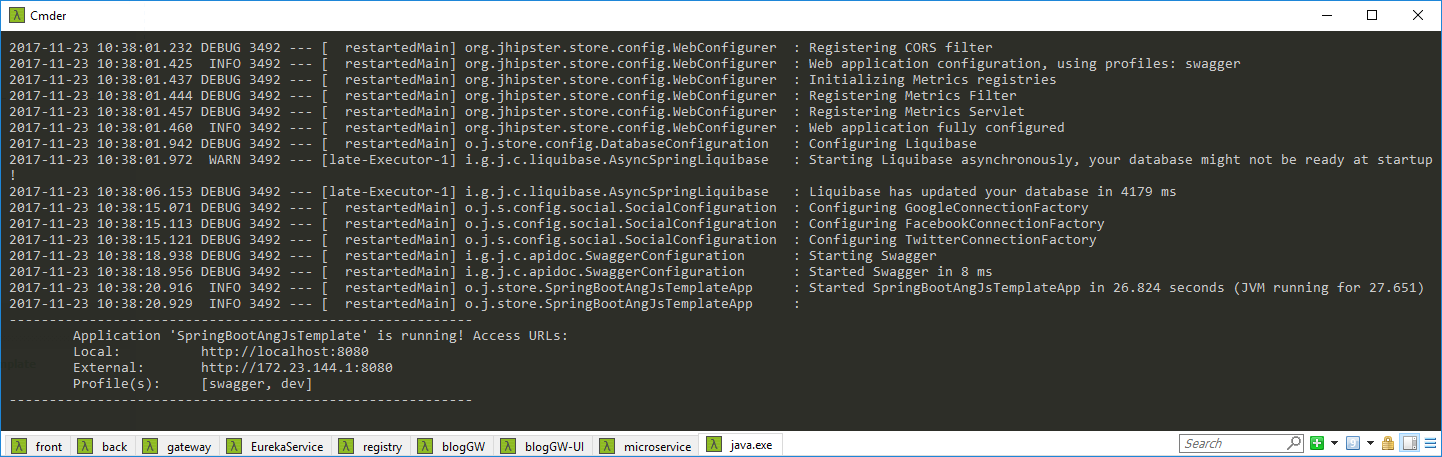
Tự tạo DB manual: *SpringBootAngJsTemplate*



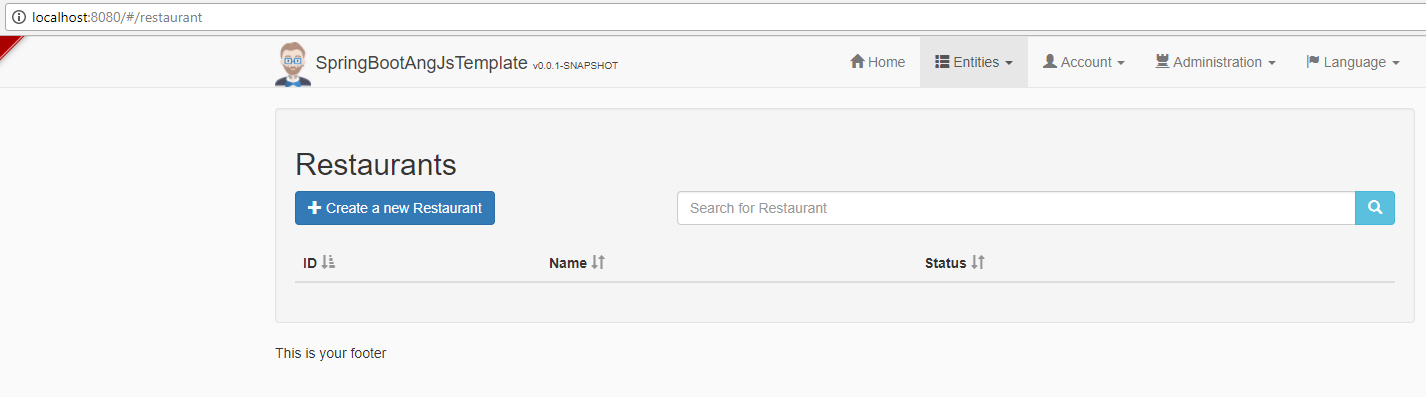
## Generate table, column & start server

mvnw

Result



Access: <http://localhost:8080/>



# Add/remove field

Delete all table in DB to re-generate (if not it will be conflicted)

2 ways:

* Run cmd: *jhispter entity* *[entity\_name]*
* Edit file jh, run import again: jhipster import-jdl jhipster-jdl.jh

# Final output

* Backend + Admin Frontend (frontend for customer may be different)
* Upload image
* Responsive UI
* Support Enum constant, Datetime, Image, True/False
* Auto link field 1-n ( when creating customer, have to choose a Restaurant)
* Validation
* Search ElasticSearch
* Support i18n
* List all pictures of restaurant ( added code )