Instalacion

Ejecutar install para w32 o w64

Heroku requiere establecer en variables de entorno, la dirección y puerto del proxy.

Las variables de entorno son:

set HTTPS\_PROXY=https://proxy.indra.es:8080

set HTTP\_PROXY=http://proxy.indra.es:8080

Praparación de un app para el deploy en Heroku.

* + En el fichero application.properties, establecer el port en heroku: server.port=${PORT:8080}
  + Heroku login
  + Git init (si no tenemos git en el proyecto)
  + Heroku git:remote -a micv-back (asociamos al proyecto nuestro repositorio git)
  + Heroky plugins:install java
  + Heroku addons:créate jawsdb (creamos y asociamos db mysql proyecto)

Al finalizar, muestra una variable de entorno (JAWSDB\_BLACK\_URL) que utilizaremos en siguiente comando

y con él nos mostrará como hay que configurar la db.

* + Heroku config:get JAWSDB\_BLACK\_URL

Y nos muestra:

mysql://t50lj98k9zdmitw7:ad9vuiiev70ow1zj@w1kr9ijlozl9l79i.chr7pe7iynqr.eu-west-1.rds.amazonaws.com:3306/gxnrw86sgg5m0kvq

ahí indica el Usuario: password, url de la db y el nombre de la base de datos (gxnrw86sgg5m0kvq) que debemos indicar en application.properties

* + Dos posibilidades para deploy:
* Git push heroky master (Heroku compila y genera el jar), de este modo no hacer falta el fichero jar para deploy
* Generamos el jar debido a los cambios del properties, lo subimos y desplegamos en heroku-> heroku jar:deploy .\target\micv-back-0.0.1-SNAPSHOT.jar
  + Probar despliegue en heroku con Postman: <https://micv-back.herokuapp.com/api/cursos>
  + Para ver la configuración: heroku config -a micv-front (proyecto)
  + heroku run bash -a micv-back (para entra en el path home de la app)

Si por ejemplo tenemos la app en github, podemos:

git clone https://github.com/heroku/node-js-getting-started.git

$ cd node-js-getting-started

De esta forma tenemos un entorno lista para utilizar con heroku. Con git y fichero package.json

También podemos entrar en la carpeta del proyecto y

heroku git:remote to <https://git.heroku.com/nombre-de-la-app-heroku.git>

la secuencia siguiente sería:

git commit –am “id-commit”

git push heroku master

Deploy de una app

heroku create

En las primeras pruebas, se ha visto que es necesario hacer heroky créate, para hecer el login. Sin ejecutarse este comando obtenemos error , que avisa de incompatibilidad con cwing.

Con esto se crea un git remote llamado heroku:

$ heroku create

Creating app... done, floating-plains-41571

https://floating-plains-41571.herokuapp.com/ | https://git.heroku.com/floating-plains-41571.git

aflucena@AFLUCENAPW7 MINGW32 /d/ProyectosAngular/angular-4-data-table-bootstrap-4-demo (master)

$ git remote –v

heroku https://git.heroku.com/floating-plains-41571.git (fetch)

heroku https://git.heroku.com/floating-plains-41571.git (push)

origin https://github.com/afermon/angular-4-data-table-bootstrap-4-demo.git (fetch)

origin https://github.com/afermon/angular-4-data-table-bootstrap-4-demo.git (push)

$ git branch –a

\* master

remotes/heroku/master

remotes/origin/HEAD -> origin/master

remotes/origin/gh-pages

remotes/origin/master

Para subir el proyecto:

git push heroku master

Para asegurarnos que hay al menos una app levantada:

heroku ps:scale web=1

Para visitar la página:

heroku open

Ver logs

heroku logs --tail

Fichero Procfile

El fichero Procfile es un fichero de texto que se pone en la raíz de la aplicación y que indica a heroku el fichero que ha de ejecutar para arrancar la app.

Un ejemplo para este fichero podría ser:

web: node index.js

Con web: indicamos que la aplicación se asociará a una pila de HTTP routing y el protocolo de mensajes será HTTP.