¿Que son los modulos en app Engine?

Los modulos es una funcionalidad de App Engine que permite dividir una aplicación grande en componentes mas pequeños, siendo posible tener en cada uno una versión independiente y una ruta independiente.

Cada modulo puede tener un tipo de instancia independiente, esto se declara en el archivo yaml,

por ejemplo. El modulo A puede tener una instancia F1 (de 128 Megas de Ram y 600 Mhz de CPU) y la instancia B tener una instancia F4\_1G (de 1024 MB de Ram y 2.4 Ghz de CPU) obviamente cada una tendrá un costo diferente.

En el archivo yaml se estaría definiendo en el parametro instance\_class donde se vería de esta forma:

Modulo A/app.yaml

instance\_class: F1

Modulo b/app.yaml

instance\_class: F4\_1G

Asimismo es util manejar modulos para tener diferentes tipos de proyectos, por ejemplo, el modulo 1 sería la parte web de una aplicación, el modulo 2 sería un endpoint que atendería peticiones de un cliente movil, y el modulo 3 sería un task queue que manejaría procesos largos de la misma aplicación.

Una regla es que debe existir un modulo default, este esta marcado en el archivo yaml en el parametro module: default, o bien no agregando este parametro en el archivo yaml correspondiente (pero en el resto de los modulos debe de existir siempre)

El nombre de los modulos puede ser cualquier identificador y se usará en el archivo dispatch.yaml para el enrutado de las peticiones.

Estructura de multi carpetas

Cada modulo es tratada como una aplicación independiente, con su estructura como se ve en la imagen de la documentación oficial:

En los ejemplos de la documentación oficial la estructura de los modulos se encuentra dentro de la misma carpeta, eso facilita el entendimiento, pero en la vida real, dificilmente se mantendrá dicha configuración, es mas probable usar multiples subcarpetas, una por cada modulo.

Si no esta habilitado el cobro, se esta limitado a tener 5 modulos, en caso de tenerlo habilitado se puede tener hasta 20.

En cada subdirectorio existe un archivo yaml y en la carpeta raíz de la aplicación existe un archivo dispatch.yaml que se encarga de tomar las rutas que se dirigirán a cada modulo. Un ejemplo de archivo dispatch sería el siguiente:

application: app-getwell

dispatch:

# enviar todo el trafico al modulo default

- url: '\*/\*'

module: default

# enviar todo el trafico al modulo mobile

- url: "mobile.getwell.care/\*"

module: mobile

- url: "\*/mobile/\*"

module: mobile

En este caso, las peticiones del dominio myapp.appspot.com serían atendidas por el modulo default, mientras que el subdominio mobile o cualquier url que posterior tenga /mobile/ dentro de su URL sería antendido por el modulo mobile.

Dentro de cada modulo se puede mantener una estructura similar a como se mantiene en cualquier aplicacion app engine, por ejemplo:

MyApp/Modulo A/

/js

/css

/imgs

/templates

app.yaml

CodeModA.py

MyApp/Modulo B/

/js

/css

/imgs

/templates

app.yaml

CodeModB.py

Dentro de la aplicación el enrutado sería el mismo, siendo la raíz (/) el punto declarado en el dispatch, por ejemplo mobile.getwell.care sería el nodo raiz de la ruta del modulo mobile.

Ejecución de los modulos

Para ejecutar los modulos, si se intenta configurar el launcher se tendría que añadir cada modulo como una aplicación, pero solo se podría ejecutar una aplicación a la vez, ya que si se intenta acceder a un segundo modulo se mandará el siguiente error  "OperationalError: database is locked"

Esto es porque si bien, son aplicaciones independientes, comparten una misma base de datos (por lo cual es posible compartir datos entre modulos a traves del datastore) y el primero modulo que se inicie obtiene el bloque exlusivo del archivo.

Para ejecutar todos los modulos al mismo tiempo se debe iniciar desde consola con el comando dev\_appserver capturando primero el modulo default (en este caso modulo 1)

python dev\_appserver**.**py Modulo1**\**app**.**yaml Modulo2**\**app**.**yaml Modulo3**\**app**.**yaml Modulo4**\**app**.**yaml

Con ello estarían todos los modulos corriendo, escuchando en los siguientes puertos (en el servidor de desarrollo)

Modulo 1 – Puerto 8080 (localhost:8080)

Modulo 2 – Puerto 8081 (localhost:8081)

Modulo 3 – Puerto 8082 (localhost:8082)

Modulo 4 – Puerto 8083 (localhost:8083)

Carga de Modulos

La carga de modulos tambien es por medio de la consola, utilizando el comando appcfg.py, siendo obligatorio que el modulo default se indique al inicio

python appcfg.py update "C:\Ruta\A\MyApp\Modulo1\app.yaml" "C:\Ruta\A\MyApp\Modulo2\app.yaml" "C:\Ruta\A\MyApp\Modulo3\app.yaml" "C:\Ruta\A\MyApp\Modulo4\app.yaml"

Posteriormente es necesario actualizar el enrutado a las peticiones que se definieron en el archivo dispatch.yaml, esto se realiza con el comando appcfg.py con el parametro update\_dispatch

python appcfg.py update\_dispatch "C:\Ruta\A\MyApp\"

Un ejemplo de código está en la dirección

<https://github.com/kristiandamian/EjemploModulosAppEngine>

Esto sería el uso básico de modulos con app engine, en proximas entregas se verá el uso de endpoints y task queue con un ejemplo mas complejo.