Evaluación técnica

JAVA Developer (Sr)

Para la realización de las consignas, tener en cuenta:

- El candidato tendrá un plazo **MÁXIMO** de **7(siete)** días para entregar las resoluciones. Dichos días comienzan a correr desde el momento en que se envió el examen.
- Una vez finalizada la evaluación, se deberá entregar los resultados al **selector** con el que mantuvo contacto por mail y deberá tener el asunto "Evaluación técnica - (nombre)". Esto puede ser a través de un archivo comprimido o subiéndolo a Github.
- La corrección del respectivo examen será dentro de las 48 hs posteriores a la entrega.

Dashboard Hoteles . (AppName: hdash)

Despegar necesita saber que cantidad de hoteles hay por ciudad, por país, y por continente. También necesita saber que disponibilidad, es decir, si el hotel tiene o no una habitación libre para ser reservada en una fecha, por ciudad, por país y por continente.

Tu sistema debe:

1. Exponer por REST un recurso GET /dashboards/hotels?scope=(cities,countries,continents) poder sumarizar por ciudad, por país y por continente la cantidad de hoteles ordenado de mayor a menor por cantidad de hoteles.
a. Ejemplo:
http://localhost:8080/dashboards/hotels?scope=cities
[{ "982": 20 }, {"4442": 10 }]
Ver 'Recursos REST' (/hotels, /cities, /countries, /continents).
Tener en cuenta que el campo id representa el id de cada uno de los recursos. En el caso de /hotels se debe limitar a mil hoteles el set de datos, utilizar limit=200 y hacer los request secuenciales para obtener el listado de hoteles y sus ciudades. El hotel tiene un atributo location.city.id donde figura el ID de la ciudad a la que pertenece.

2. Exponer por REST un recurso GET /dashboards/availabilities?destinations={ids}&scope={cities,countries,continents} donde se debe sumarizar los que tuvieron disponibilidad y los que no tuvieron. La llamada a los destinos se debe poder realizar en paralelo y poder manejar reintento si hay una falla en la conexión.
a. Ejemplo:
http://localhost:8080/dashboards/availabilities?destinations=982,4844,6381,4971,7848,3696,6624,4088,1569,31684
[{ "availables": [{"982": 20 },{"4844": 19}, {"6381": 18},{"4971": 10 }, { "7848": 5 }, { "3696": 4 }, { "6624": 2 }, {"4088": 2 }, { "1569": 1 }], "unavailables": ["31684"] }
Ver 'Recursos REST' (/hotels, /cities, /countries, /continents, /hotels/prices).
El recurso /hotels/prices les va a poder indicar si el hotel tiene o no disponibilidad. Pueden pedir hasta 10 hoteles por request, y si en la respuesta no viene el hotel, eso significa que no tiene disponibilidad.

https://api.despegar.com/v3/hotels/prices?hotels=20000,300400&country=ar&distribution=2&include=cheapest va a devolver 1 solo hotel (el 300400)

items:
[
{
id: "site|AR|4844|2|0|300400|2017-05-15|1|BRB|prepaid_installments,prepaid_one_payment|2017-05-09T13:56:35",
hotel_id: "300400",

```
...
price_detail:
 {
 currency: "USD",
 subtotal: 42.58,
 taxes: 0,
 taxes_detail:
 [],
 fee: 2,
 discounts: 0,
 total: 44.71, ...}
```


Recursos REST:
Para pegarle a los recursos REST de despegar ver “Cómo usar la API v3 de Despegar”, incluye ejemplo de recursos que pueden ser consumidos en el ejercicio.

Consideraciones generales: Se evaluará el diseño de la solución, el manejo de errores, el uso de logs, claridad de código, y que el sistema haga lo que tiene que hacer. El código fuente tiene que estar en el archivo entregable, ver “Condiciones de entrega”. El mismo debe compilar y los test deben correr sin errores. Si hacen test contra los recursos expuestos en https://api.despegar.com tener la prevención de que si los recursos no están disponibles sus test no deben fallar. El uso de librerías, frameworks, está permitido y tienen que estar disponible en https://mvnrepository.com/ . El código puede hacerse en Scala o Java.

Condiciones de entrega Se debe entregar un archivo tar zipeado con el siguiente formato de nombre apellido-nombre-appName.evaluacion.tar.gz, debe incluir un README para explicar cómo levantar la aplicación . El mismo debe contar con un startup.sh que inicie la aplicación, El source code debe estar incluido en el zip, en una carpeta llamada src El log se debe escribir en el standard output.