

الجمهورية العربية السورية
جامعة دمشق
كلية الهندسة المعلوماتية
السنة الخامسة قسم الشبكات والنظم

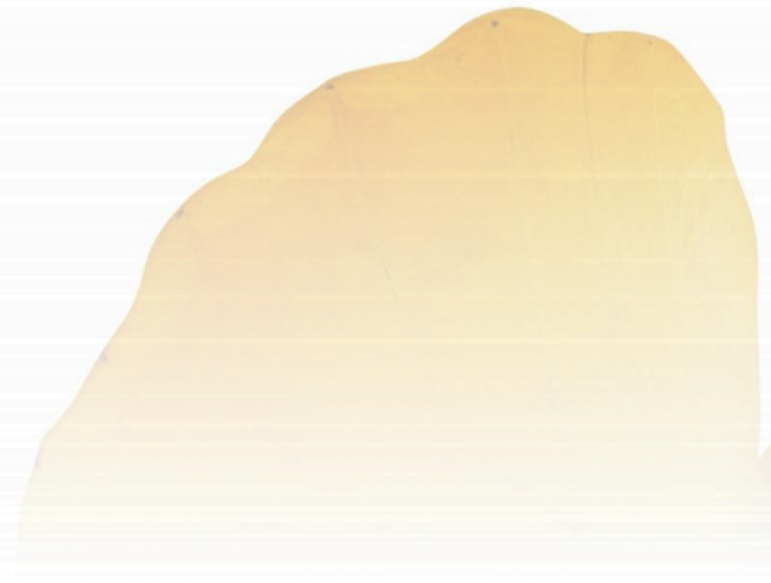


Distributed systems

Google Maps API

إشراف المهندسة: بشرى داوود مرح منصور

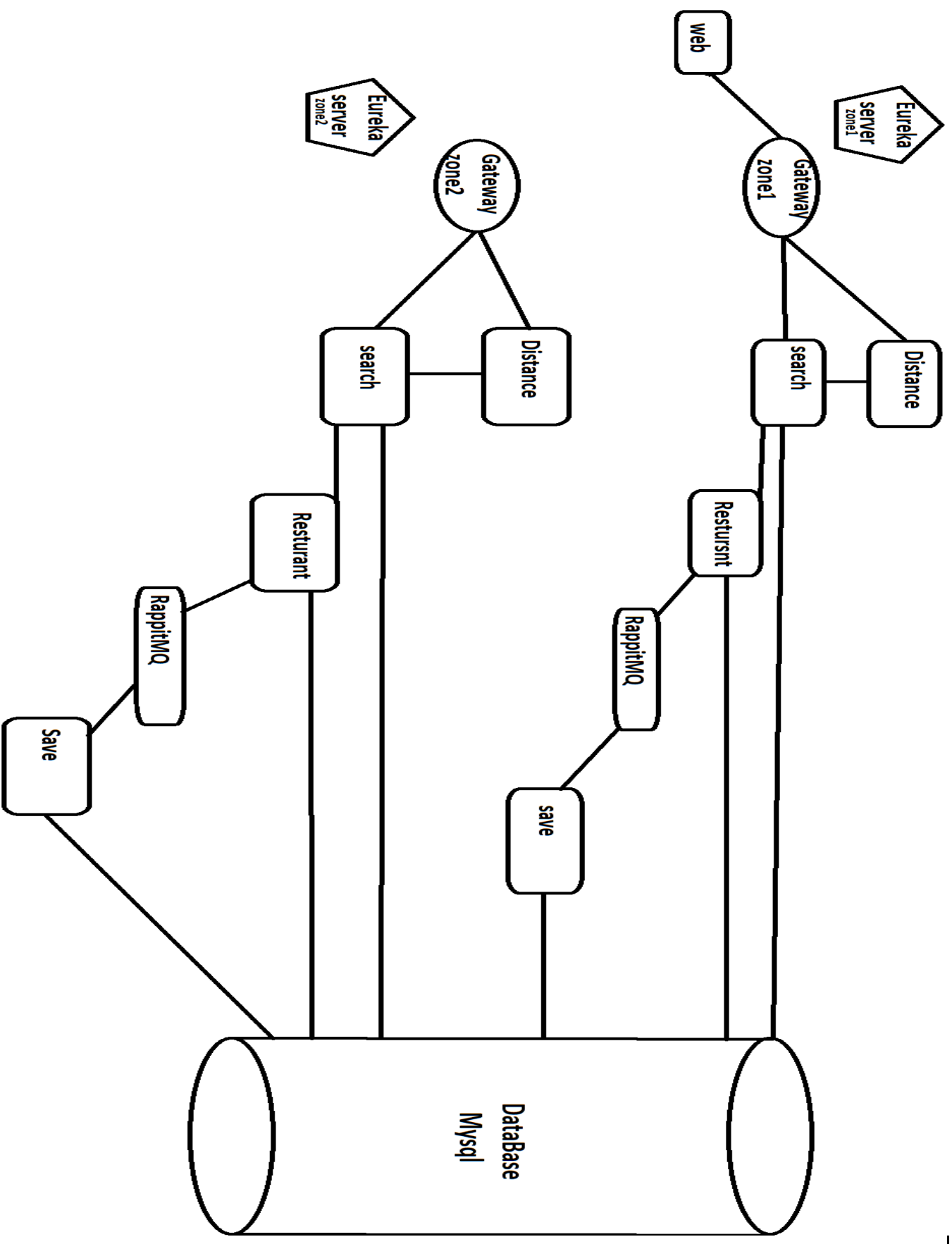
الطلاب : شعيب حاج حسين سندس درزي
أحمد قصار راما خزنة
شادي مصلح





Spring Cloud Gateway





الخدمة المقترحة : خرائط غوغل

الخدمات التي قمنا بإختيارها

لتنفيذ خرائط غوغل

Search service

Distance service

Resturant service

Save service

شرح الخدمات ووظائفها

Gatway service

✓ تقوم هذه الخدمة بعمل بوابة واحدة

لجميع الخدمات .

✓ أي طلب من الزبون يمر عبرها ثم

للخدمة المطلوبة

✓ تقوم بتوزيع الحمل بين نسخ

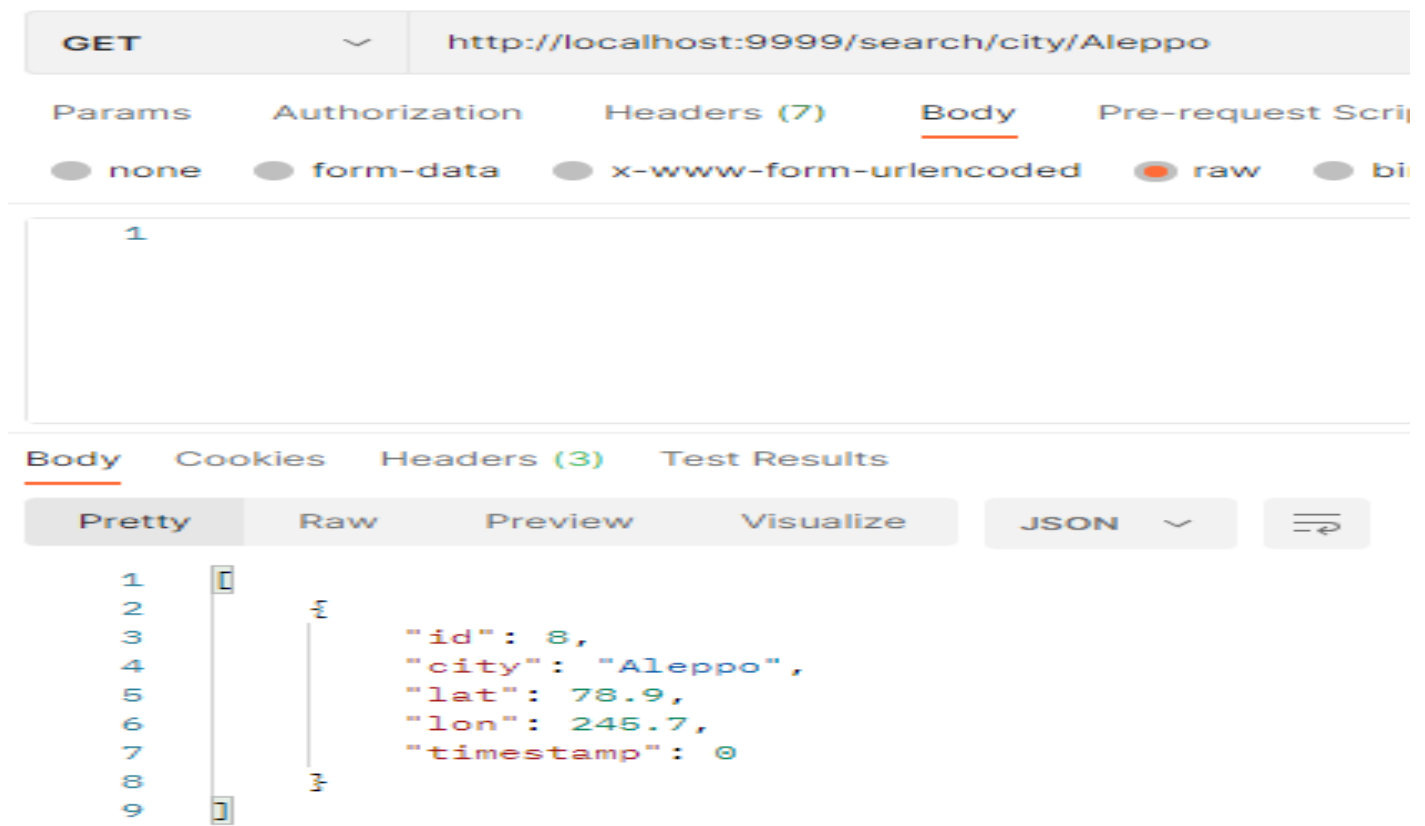
الخدمات في حال وجود أكثر من

نسخة.

✓ تعمل على البورت 9999

Search service

- ✓ تقوم هذه الخدمة بالبحث عن إحداثيات المدينة المختارة [1].
- ✓ تطلب قائمة المطاعم من خدمة المطاعم بتواصل متزامن [2].
- ✓ تطلبها خدمة المسافة عندما يطلب الزبون المسافة بينه وبين المدينة بتواصل متزامن.
- ✓ هي خدمة Statefull لأن لديها تخزين وجلب من DB



[1]

GET ⌵ http://localhost:9999/search/rest/14

Params Authorization Headers (7) Body Pre-request Script Test Results

Query Params

KEY	VALUE
Key	Value

Body Cookies Headers (3) Test Results

Pretty Raw Preview Visualize Text ⌵ ≡

```

1 Halap
2 ALSalam
3 Big5

```

[2]

Distance service

- ✓ تقوم هذه بحساب المسافة بين احداثيات الزبون واحداثيات مدينة يبحث عنها الزبون.
- ✓ وبذلك يتطلب استعاء خدمة البحث لجلب الاحداثيات بتواصل متزامن.
- ✓ هي خدمة Stateless لأنها لا تتعامل مع قاعدة المعطيات ابدا وانما تتعامل مع خدمة أخرى.

GET http://localhost:9999/distance/get/AI_nabek/14

Params Authorization Headers (7) Body Pre-request Script Tests Settings

Query Params

KEY	VALUE
Key	Value

Body Cookies Headers (3) Test Results Stat

Pretty Raw Preview Visualize Text

1 The Distance Between U And this City is :1175.3999999999999 Km

Resturant service

- ✓ تقوم هذه بعدة مهام
- ✓ أولها البحث عن المطاعم في مدينة معينة وهذه الخدمة تطلبها خدمة البحث بعد البحث عن مدينة .
- ✓ تتواصل مع خدمة البحث بشكل متزامن .
- ✓ ثانيها ارسال طلب لحفظ قائمة المطاعم مع المدينة المتواجدة فيها
- ✓ يتم ارسال الطلب الى رتل حيث تكون خدمة الحفظ متنصة على نفس الرتل لحفظ المعلومات في ال DB

✓ وبذلك يتم التواصل بشكل غير
متزامن بين الخدمتين .
✓ تم استخدام RappitMQ لعمل الرتل

POST ⌵ http://localhost:9999/resturant/publish

Params Authorization Headers (9) Body ● Pre-request S

Query Params

KEY	VALUE
Key	Value

Body Cookies Headers (3) Test Results

Pretty Raw Preview Visualize Text ⌵ ⌵

```
1 Message Published
```

http://localhost:15672 ✓

✓ بعد ارسال الطلب يظهر في الرتل

Save service

✓ تقوم هذه الخدمة بالتنصت على الرتل وجلب الرسائل الجديدة لحفظها

✓ فتقوم بعمل ربط بين Id الزبون والمطاعم التي طلب حفظها
✓ تتواصل مع خدمة المطعم بشكل غير متزامن .

✓ بعد تشغيلها مباشرة تأخذ طلب الحفظ من الرتل

✓ والفائدة من جعل هاتين الخدمتين يتوصلا بشكل غير متزامن هو أن

RabbitMQ™ RabbitMQ 3.10.2 Erlang 25.0

Refreshed 2022-06-04 20:21:31 Refresh every 5 seconds

Virtual host All

Cluster rabbit@WIN-8ECVD3QR9DO

Overview Connections Channels Exchanges **Queues** Admin

Consumers (1)

Bindings (2)

Publish message

Get messages

Warning: getting messages from a queue is a destructive action. ?

Ack Mode: Nack message requeue true

Encoding: Auto string / base64 ?

Messages: 1

Get Message(s)

Queue is empty

Close

Encoding: string

معلومات الحفظ مهمة بالنسبة
للزبون وبالتالي مهمة لنا والرتل

GET http://localhost:9999/save/14

Params Authorization Headers (7) Body Pre-request Script Test Results

Query Params

KEY	VALUE
Key	Value

Body Cookies Headers (3) Test Results

Pretty Raw Preview Visualize JSON

```
1 {  
2   "id": 14,  
3   "rest1": "Halap,Aleppo",  
4   "rest2": "Alsalam,Aleppo",  
5   "rest3": "Big5,Aleppo"  
6 }
```

يسمح لنا بفحظها مؤقتا في حال
فشل خدمة الحفظ
✓ تقوم أيضا بالبحث عن المطاعم
المحفوظة بالنسبة لكل زبون .

Eureka server

✓ يقوم بتسجيل جميع الخدمات التي
تحتوي Eureka client .

- ✓ يظهر حالة الخدمة Up/Down
- ✓ يظهر ال Availability zones لكل خدمة.
- ✓ يفيد في تحقيق العنونة الديناميكية حيث تتعرف الخدمات على بعضها من اسمها وبذلك نتفادى الفشل في حال تغير البورت
- ✓ <http://localhost:8761>
- ✓ <http://localhost:8762>

DS Replicas

127.0.0.1

Instances currently registered with Eureka

Application	AMIs	Availability Zones	Status
DISTANCESSERVICE	n/a ()	()	UP () - host.docker.internal:DistancesService:8084
GATEWAY	n/a ()	()	UP () - host.docker.internal:Gateway:9999
RESTURANTSERVICE	n/a ()	()	UP () - host.docker.internal:ResturantService:9991
SAVESERVICE	n/a ()	()	UP () - host.docker.internal:SaveService:8088
SEARCHSERVICE	n/a ()	()	UP () - host.docker.internal:SearchService:8181

General Info

✓ بعد تشغيل الخدمة على أكثر من zone

DS Replicas

127.0.0.1

Instances currently registered with Eureka

Application	AMIs	Availability Zones	Status
DISTANCESSERVICE	n/a (l)	(l)	UP (l) - host.docker.internal:DistancesService:8084
GATEWAY	n/a (l)	(l)	UP (l) - host.docker.internal:Gateway:9999
RESTURANTSERVICE	n/a (l)	(l)	UP (l) - host.docker.internal:ResturantService:9991
SAVESERVICE	n/a (l)	(l)	UP (l) - host.docker.internal:SaveService:8088
SEARCHSERVICE	n/a (r)	(r)	UP (r) - host.docker.internal:SearchService:8182 , host.docker.internal:SearchService:8181

Hystrix

✓ تقوم بكسر الإعتماذية على الخدمة في حال فشلها.
✓ أي لتجنب فشل خدمة تعتمد على خدمة أخرى تتواصل معها وهذه الخدمة متوقفة لسبب ما.

```

    @HystrixCommand(fallbackMethod = "alter",commandProperties = {})
    @GetMapping("/rest/{id1}")
    public String GetRest(@PathVariable int id1){
        int id=id1;
        String city =country.get(0).getCity();
        System.out.println(country.get(0).getCity());
        return restTemplate.getForObject( url: "http://ResturantService/resturant/{id}/{city}", String.class,id,city)
    }

```

- ✓ هنا خدمة البحث تعتمد على خدمة المطعم .
- ✓ وبالتالي في حال فشل خدمة المطعم او توقفها يؤدي لفشل خدمة البحث
- ✓ هنا يأتي دور ال Hystrix لتجنب الفشل وذلك بعمل fallbackMethod أي عمل طريق بديل .

GET
http://localhost:9999/search/rest/14

Params
Authorization
Headers (7)
Body
Pre-request Script
Tests
Settings

Query Params

KEY	VALUE
Key	Value

Body
Cookies
Headers (3)
Test Results
Status: 200

Pretty
Raw
Preview
Visualize
Text

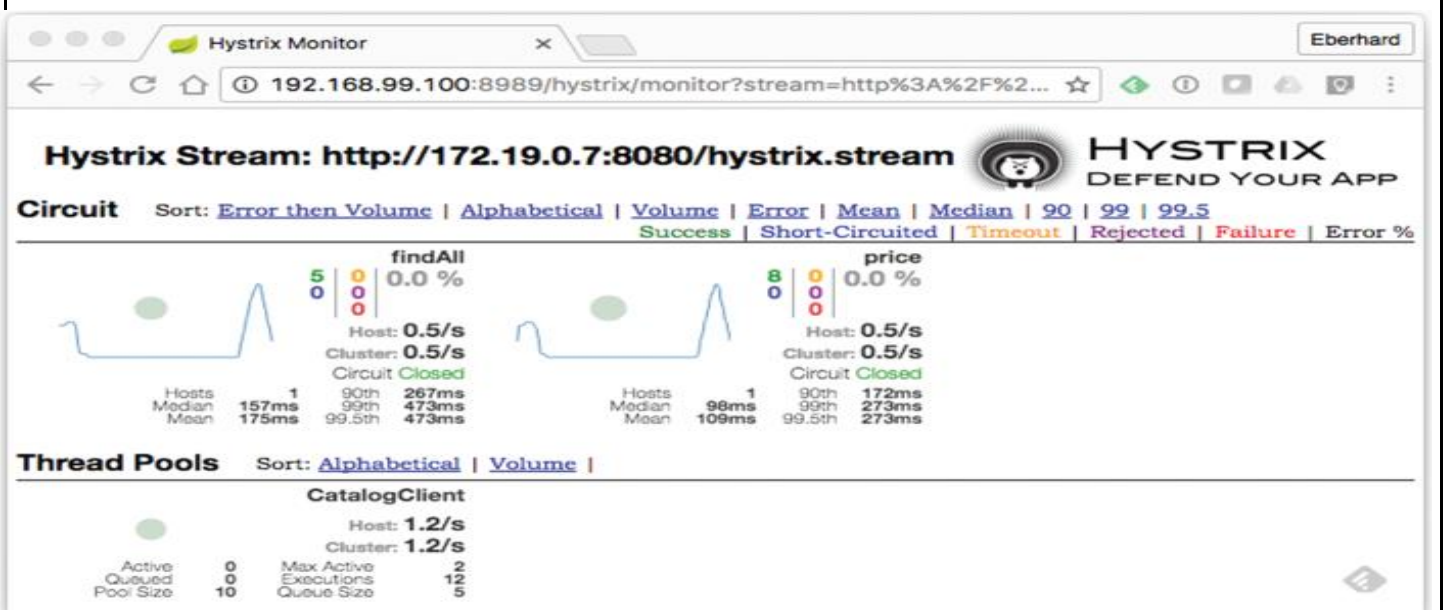
```

1 We are sorry but this service is busy now , try again after one minute

```

Hystrix dashboard

✓ تتخاطب مع ال Actueator لجلب
حالة الخدمة حالتها ومن ثم عرضها
بطريقة تفاعلية انية.



LoadBalancer

- ✓ تقوم بتوزيع الحمل على النسخ المختلفة للخدمة المطلوبة
- ✓ لدينا موضعين فيها توزيع حمل
- ✓ في ال Gateway
- ✓ عند التواصل المتزامن وغير المتزامن
- ✓ نقول بتوزيع الحمل عند فتح قناة الاتصال

```
@Bean
@LoadBalanced
public RestTemplate rest() { return new RestTemplate(); }
```

ZipKin

- ✓ منظومة تقفي موزع
- ✓ مهمته تقفي الطلبات لمراجعتها والبحث عن موضع الطلب وفي أي خدمة توقف.
- ✓ تم تنزيل ال zipkin server على Docker
- ✓ [/http://localhost:9411/zipkin](http://localhost:9411/zipkin)



exciting_tu [openzipkin/zipkin](#)
RUNNING PORT: 9411

✓ بعد تشغيل السيرفر وازافة ال client dependences
بتقفي الطلبات.

GET



<http://localhost:9999/search/city/Aleppo>



Zipkin

Find a trace

Dependencies

ENGLISH



Search by trace ID

GATEWAY: get

Duration: 27.960ms Services: 2 Depth: 2 Total Spans: 2 Trace ID: 0bce35ebc7f5a319

DOWNLOAD JSON



GATEWAY

get

Span ID: 0bce35ebc7f5a319 Parent ID: None

Annotations



SHOW ALL ANNOTATIONS

Tags

http.method

GET

http.path

/search/city/Aleppo

GET



http://localhost:9999/distance/get/AI_nabek/14



Zipkin



Find a trace



Dependencies



ENGLISH



Search by trace ID

GATEWAY: get

Duration: 788.745ms Services: 3 Depth: 4 Total Spans: 8 Trace ID: 647ee03d11ffb3df

DOWNLOAD JSON

