[Need Backup]

Microservices Architecture. and docker

https://www.youtube.com/watch?v=HFl2dzhVuUo

Monolothic design -> Everything is tightly coupled, everything is at one place, businesslogic, frontend and database. Creates problem once userbase starts growing. Its kind of bottleneck. We cannot expand it.

3 tier architecture -> Frontend at difference place, businesslogic at difference and database at difference. But it did not solve the issue.

Microservice -> Independent service which serves one particular task of the project. One microservice will work on its own. It will have its busness logic and it will have its db separate. All microservice talks to each other only via api.
Like employee service will talk to different department service only via api. Both service will have their own things whatever is required like db api service etc..

Lets jump to microservice architecutre :
===============================

if we install eclipse market place -> spring boot something helper.. it will give run as spring boot app. this has few additional feature.. not required but good to have.

Service Registry/Discover Service - Netflix/OSS service.
==

its main point of contact for all services. All services will be attached to it. It will have track of how many services are attached to it and how many instances of particular serivces are running. It will know everything about application architecture.

Service registry will also do load balancing... If there are 2 instances of

Department service and 5 instance of Employee service then, load balancing will be done by service registry..

if department service needs data from employee service then how department service will know which instance of employee service it should call. ?

if department service needs anything from employee service then it will ask service registry and service registry will contact employee service to get that data.

API Gateway
——————

All the public services will be exposed by API gateway itself. Microservice wont exponse api services directly to the public.

API gateway will decide that this request is for employee service then call employee sercie, this request for X service then call that service..

API gateway will also play role of security service… if that request is secure then only call that particular microservice.

Config server
——————

Its a cemtrlized configuration service.. All service will call config server to get configurations. Full architecurre configuration will be stored inside config server.

while spinning config server, it gives an option where to store configuration. Generally we store it in the git.

It can store common config as well specific config as well..

Zipkin
———

Its a distributed logging tool. It keeps track of all request and reponse from public and within microservices as well.

==========
Actual Practical
==============

We will create each and every service using spring initializer website..

Lets create Service Registry first. — Refer service-regsitry project.
———————————————————

Very simple.. just create a project and run it in eclipse after chaning yaml file.. Thats it..

Create a project from spring intitializer for this.

Dependency -
Eureka Server (spring cloud netflix eureka server)
Spring Web.

Import that project in IDE.

We are going to use yaml config instead of properties config… So change application.properties to application.yaml.

Add annotation @EnableEurekaServer above main file below @SpringBootApplication. This will help in treating this microservice as eureka server (discovery servive/service registry).

Eureka can be server as well as eureka can be client… need to check further

yaml config :

register-with-eureka : false.. means dont register for itself.. act as a server..
defaultZone: http://${eureka.instance.hostname}:${server.port}/eureka   … here url is refer to same yaml file.. eureka.instance.hostname means..in same file eureka -> instance -> name.

8761 is default port for eureka server.

once all things is done… just run the app and go to URL : http://localhost:8761

You can see that there is no instance and none of the instance is connected yet.

==========
Lets create a department-service .. which will have all departments of orgznization.
======

Again springboot project …

Dependencies :
Spring web
Eureka discover service - client —> make sure its a clint.. we are not setting up discovery server… this service is a client for disocvery service.
Spring Accurater — for matrix etc..

Import project -> change properties file to yaml file.

Add annotatin here for client below spring boot annotation -
@EnableDiscoveryClient
public

give some different port to this service.. 8081.
now ask this department service to connect to our eureka server.. we will enter url of eureka server here.

full yaml :

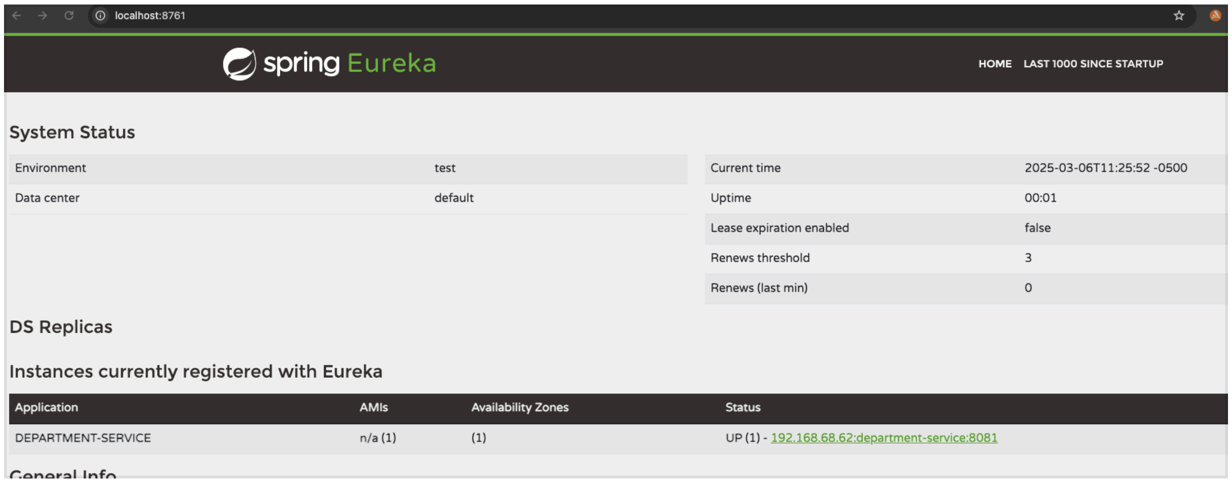
```
server:
  port: 8081

spring:
 application:
   name: department-service

eureka:
```

```
client:
  serviceUrl:
    defaultZone: http://localhost:8761/eureka
```

After doing this much, if I run both services, first eureka server and then department service, then I will be able to view department service listed under eureka service - http://localhost:8761/eureka



System Status

Environment	test
Data center	default

Current time	2025-03-06T11:25:52 -0500
Uptime	00:01
Lease expiration enabled	false
Renews threshold	3
Renews (last min)	0

DS Replicas

Instances currently registered with Eureka

Application	AMIs	Availability Zones	Status
DEPARTMENT-SERVICE	n/a (1)	(1)	UP (1) - 192.168.68.62:department-service:8081

General Info

============
Config Server
============

Lets create a config server.

This will host all configuration..

This all are non-functional required stuff.. then we will have actual business logic..

Again create a project…
only dependency would be

Config server (spring cloud config) — git svn hashicorp vault config server.

import project and change properties to yaml…

Now this is our config server… so all the service should connect to this service to get configuration… department service employee service etc.. should pick up all configuration from this service by connecting to ti.

Add annotation in main file — @EnableConfigServer

By default it expects all configuration to be stored on github… we can do that as well, but here we will just store yaml files for our projects and get from there (so look for the config natively and do not look up for github url)

By default profile is git, but we will add spring: profile: native in yaml file. native means it will look for all the configuration with the config-server project itself without going to git. It will publish configs to other project from native config-server project.

so my config-server yaml file will look like. :

```yaml
server:
  port: 8088

spring:
  profiles:
    active: native
```

inside resource folder, we will add one more folder — config and create a yaml file with the same name as service name…

Config -> deparment-service.yaml.

pick all the config from deparment-service project yaml file and put it here… except name …

we also have to now tell department service yaml file to go to config server yaml file and pickup configuration…

so now I have remvoed everything from my department-service — application.yaml file and inserted optinal config import from cofnig server… so application.yaml file within department-service will look like :

```yaml
spring:
  application:
    name: department-service
  config:
    import: "optional:configserver:http://localhost:8088"
```

I had removed optional keyword to debug why my department service was not getting connected to config server... i was able to view error and resolve it...

We have to also make department-service client of config-server as well.. so we need to add depenendency of config-client in department-service..

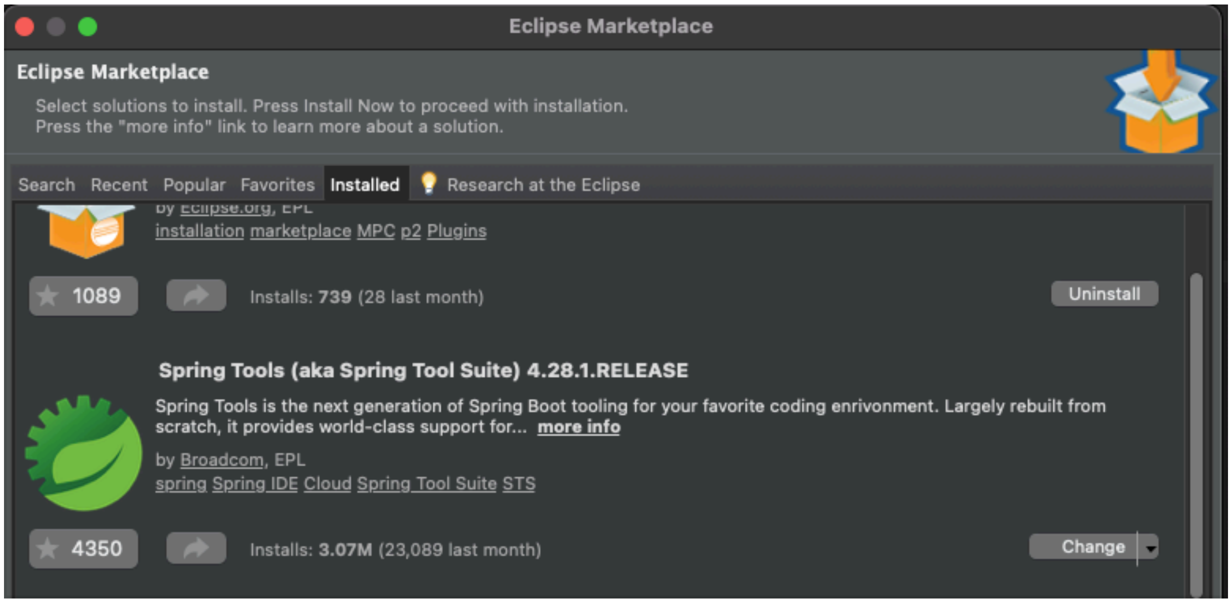
go to spring intitzlizer website... select just depenendency config client... and click on explore instead of generate to see dependency xml tags... copy those to department service pom.xml

Restart services so that department-service connect to config-server to pickup configuration.

now department service is connected to config server to pickup configuration and its also connected to eureka server..

check eureka server page again to reverify connectivity.

I have installed help -> eclipse microservices -> spring tools addon... this help in listing all services togeather under one local and it also provide start stop etc functionality togeather..


Eclipse Marketplace

Eclipse Marketplace

Select solutions to install. Press Install Now to proceed with installation.
Press the "more info" link to learn more about a solution.

by Eclipse.org, EPL
installation marketplace MPC p2 Plugins

Installs: 739 (28 last month)

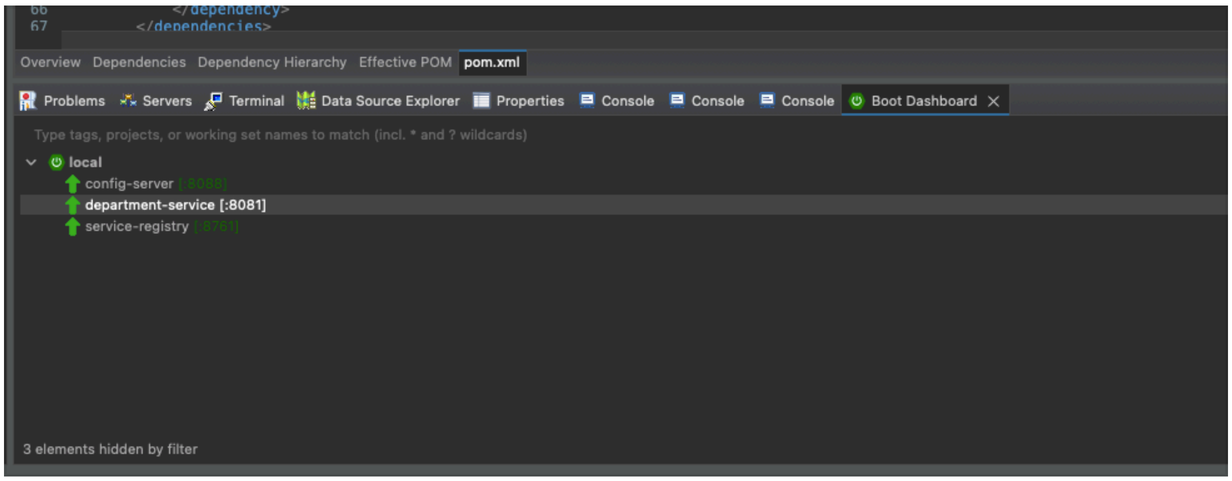
Spring Tools (aka Spring Tool Suite) 4.28.1.RELEASE

Spring Tools is the next generation of Spring Boot tooling for your favorite coding enrivonment. Largely rebuilt from scratch, it provides world-class support for... more info

by Broadcom, EPL
spring Spring IDE Cloud Spring Tool Suite STS

Installs: 3.07M (23,089 last month)

look for boot dashboard icon in tool bar... green switch off button. it will give this functionality..


```
        </dependency>
    </dependencies>
```

Type tags, projects, or working set names to match (incl. * and ? wildcards)

```
local
  config-server [:8088]
  department-service [:8081]
  service-registry [:8761]
```

3 elements hidden by filter

========
Zipkin
=======

lets start with zipkin now to get all the traces etc of all services...

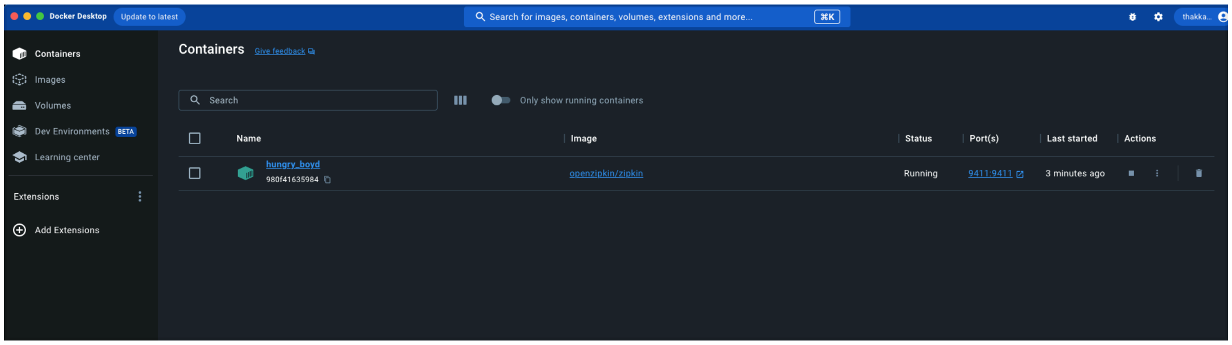
To install zipkin, either download zipkin jar and run using java or install docker and run docker command... go to the website of zipkin as a starting point.

https://zipkin.io/pages/quickstart.html

I used DOCKER… Installed docker for Mac Bigsur from somewhere… registered and started docker desktop… and then just went to the terminal and typed whatever was available on zipkin website…

docker run -d -p 9411:9411 openzipkin/zipkin

This started container on my docker for zipkin on 9411 port.



	Name	Image	Status	Port(s)	Last started	Actions
☐	hungry_boyd 980f41635984	openzipkin/zipkin	Running	9411:9411	3 minutes ago	

docker was up and running on localhost:9411 as said above… we can click on 9411 link as well to open browser.. otherwise type localhost:9411

I saw quick doker 7 mins video and it was good in youtube.

Lets connect our application to zipkin…

Add dependency of zipkin .. go to spring initializer…

Add dependnency zipkin.   if you explore you will see two dependency … io.micrometer and io.zipkin.reporter2.. add both.. it will also show you .brave… its a version and it dependends on which version of spring you are using.. it can change based on that.

```
<dependency>
            <groupId>io.micrometer</groupId>
         <artifactId>micrometer-tracing-bridge-brave</artifactId>
            </dependency>
      <dependency>
```

```
            <groupId>io.zipkin.reporter2</groupId>
            <artifactId>zipkin-reporter-brave</artifactId>
        </dependency>
```

now we have to ask department service to publish all traces to zipkin.. by default it publishes to same 9411 port without doing any port config... 9411 is the port we configured for zipkin...

we will just tell to publish all probability... as config is being managed by config server, we will add department service zipking config there...

```
  management:
    tracing:
      sampling:
        probability: 1.0
```

probability 1.0 means 100% publish.

now stop all service and start config server, service registry and departemnt service in that order..

==========

Lets start with department service business logic to expose some apis...

Lets create Model, Controller and Repository for Department... create folder for each in department service...

Model -> Actual department class (table)
Repository -> Business logic .. like fetching struff from db etc... our findbyAll method , delete method etc..
Controller -> API exposed using this..

Here, we are only creating list, not connecting to db but its easy as seen earlier...

We will add Logger as well in department controller for our Zipkin service.

I have created postman collection - First Microservice in my account.

Our applicaiton - Department service is running on 8081.

AFter adding getting etc request in postman.. go to zipkin… click on on run query.. blank… you will find all requests…

================

Now lets create an employee service.

go to spring initializer and create employee service

Dependency :

Spring web
config client
eureka client
zipkin
acuator

we will change application.properties to application yaml.. give name of the application and will tell it to connect to config server..

we will create a employee-service.yaml file in config server and tell there to connect to eureka , change port to 8082 and also tell it to do all tracing..

Remember that for record… you dont do Employee.getId()… we just do employee.id();

After doing all for employee service just like department service, you will be able to see service in eureka server with port 8082.

go to postman and try apis… port will be 8082 for employee service.

==============================

Communication between services
————————————————————

Now next task is to connect department service to employee service..

Communication between services.. we have already created a list of employee in Department model already.. but til now we are not getting any employees, we are just getting id and name…

I want to list down all the departments with employees as well..

So what we have to do is, whenever I call department service to get department, it should also call employee service internallly and get employees for supplied department.

We will not directly connect to employee service from department service, we will go through the service registry. As service registry will do internal load balancing as well for internal communication.

KEY CONCEPT OF SPRING BOOT 3..

We will use concent of HTTP exchanger of spring boot 3 and spring 6…

We will create a employee service client within deparment service, employee client from department will connect to employee service server…

Lets create employee client within department now… create a new package .client..

And create an INTERFACE for employee client.

Annotate interface with @HTTPExchange and then declare employee service method which you want to call…

We want method from employee controller which calls employees by department. We have already created that method in employee controller.

remember, for HTTPExchange, getmapping should be annotated as @GetExchange… !!

we already have employee pojo in department.. check import statement.. we have changed url to employee/departmentid/{id} … as we are within employee service..

We have just done declaration, here… in such communication, Whenever we want to connect to other service … we create a WEB CLIENT…

So we need a webclient to connect to employee service..

WEB CLIENT is a part of REACTIVE programming, so we need to add dependnecy fo REACTIVE programming.

Go to initiazlizer and add SPRING REACTIVE WEB dependency…

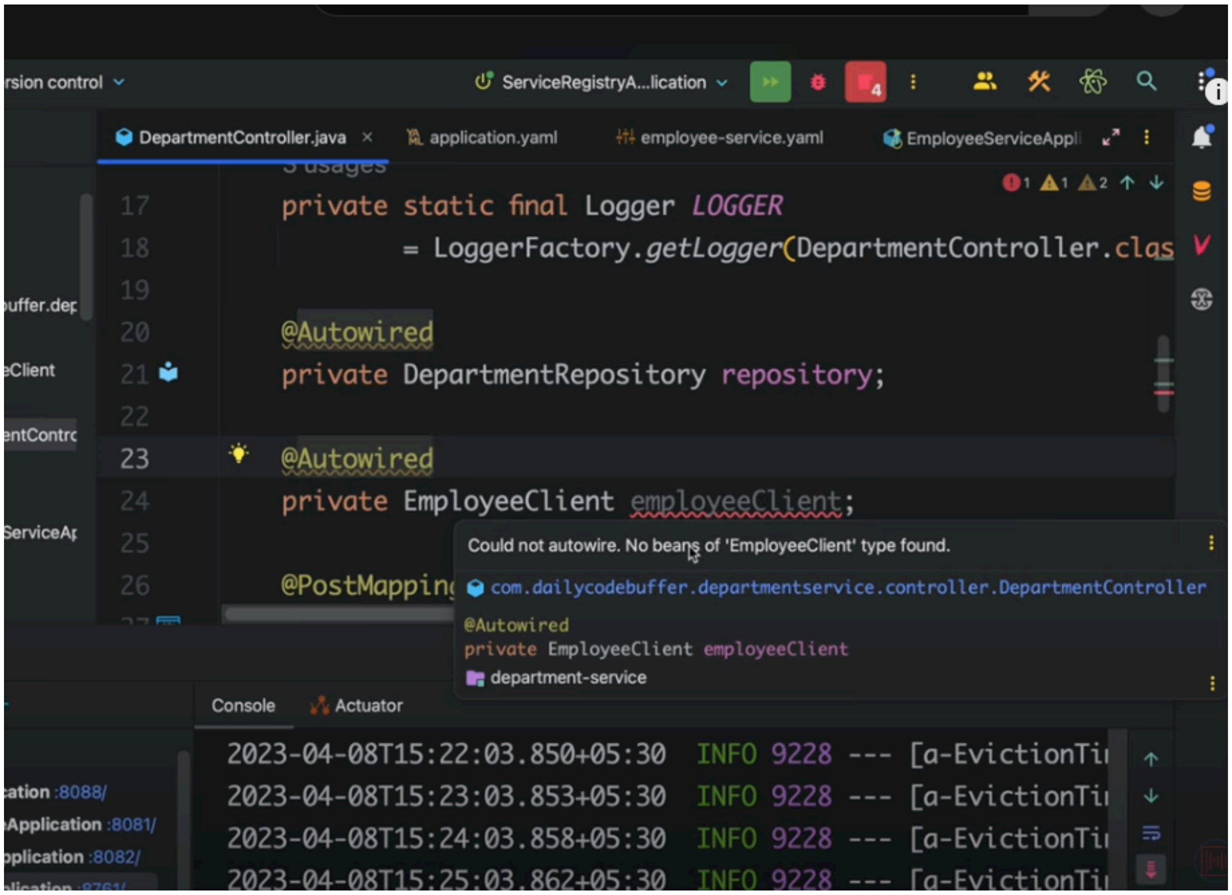
Copy, to deparment service :

```
<dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-webflux</artifactId>
    </dependency>
```

Now we want to call employee client so that in turn it calls actual employee service to get data of employees by department id…

But employee client is an interface and we cannot create an object of interface… If I do following in Department conroller then it will throw and error that bean for EmployeeClient is missing.. it needs been..

snapshot of video..



```
private static final Logger LOGGER
        = LoggerFactory.getLogger(DepartmentController.clas

@Autowired
private DepartmentRepository repository;

@Autowired
private EmployeeClient employeeClient;

@PostMappin
```

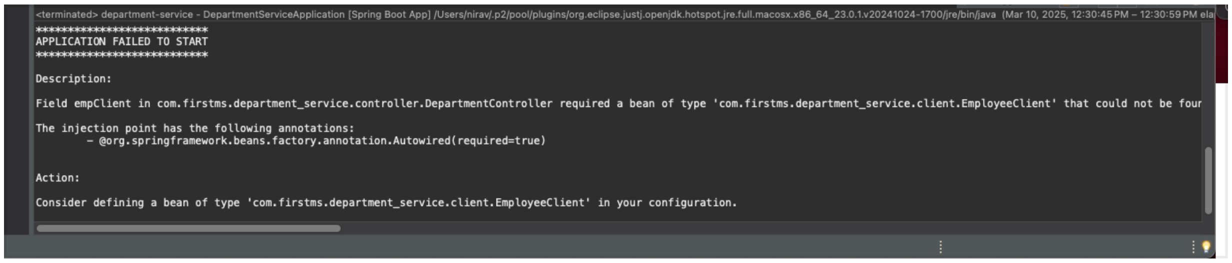
Could not autowire. No beans of 'EmployeeClient' type found.

```
com.dailycodebuffer.departmentservice.controller.DepartmentController
@Autowired
private EmployeeClient employeeClient
```

department-service

```
2023-04-08T15:22:03.850+05:30  INFO 9228 --- [a-EvictionTi
2023-04-08T15:23:03.853+05:30  INFO 9228 --- [a-EvictionTi
2023-04-08T15:24:03.858+05:30  INFO 9228 --- [a-EvictionTi
2023-04-08T15:25:03.862+05:30  INFO 9228 --- [a-EvictionTi
```

In Eclipse while starting department service it was throwing an error. :



```
<terminated> department-service - DepartmentServiceApplication [Spring Boot App] /Users/nirav/.p2/pool/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.macosx.x86_64_23.0.1.v20241024-1700/jre/bin/java  (Mar 10, 2025, 12:30:45 PM – 12:30:59 PM ela
***************************
APPLICATION FAILED TO START
***************************

Description:

Field empClient in com.firstms.department_service.controller.DepartmentController required a bean of type 'com.firstms.department_service.client.EmployeeClient' that could not be four

The injection point has the following annotations:
        - @org.springframework.beans.factory.annotation.Autowired(required=true)


Action:

Consider defining a bean of type 'com.firstms.department_service.client.EmployeeClient' in your configuration.
```

My notes.. need to reverify…

Its Employee service is kind of third party for us… so according to previous spring/spring bean lecture… @Bean we need to create to refer .. we cannot make @component as its not our class.

—

Now, we should create a configuration that whenever I refer to Employee Client, it should create a bean of webclient which connects to actual employee webservice.

So lets create a package … config package… and create a new class WebClientConfig

Now lets create a configiguation for webclient which will point to employee client

We have createn an object of webclinet… This webclinet should go and connect to Employee service… This Employee service is running in service registry as a name "Employee-Service". We do not know anything except its name, how many instances, on which port service is running etc… We will just give url to connect to employee-service which is running on service registry.

baseUrl("http://employee-service").

Employee-Service is being load balanced by service registry. if any service is connected to load balancer service registry then we have to tell webclient as well that this is load balanced service… Filter it accordingly..

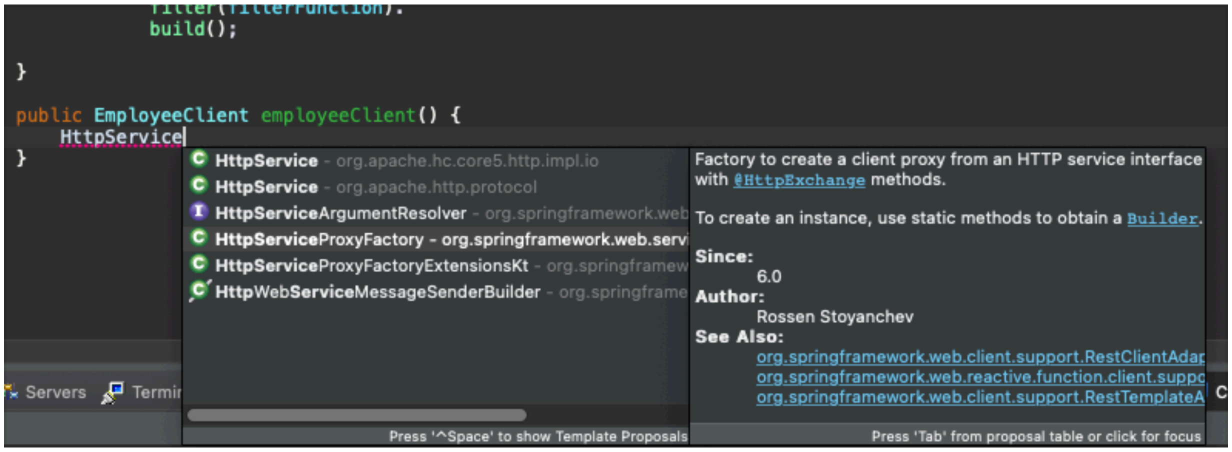
So adding filter of load balancer as well..

```
@Autowired
private LoadBalancedExchangeFilterFunction filterFunction;
```

At this point we have an object ready for webclient… Now using this webclient we have to tell that we want to connect to employee-service…

Creating an object of EmployeeClient now…

From where we will get EmployeeClient, we need to use httpproxyservice. according to doc , its a way to access @HttpExchange interface.



```
            filter(filterFunction).
            build();

}

public EmployeeClient employeeClient() {
    HttpService
}
```

HttpService - org.apache.hc.core5.http.impl.io
HttpService - org.apache.http.protocol
HttpServiceArgumentResolver - org.springframework.web
HttpServiceProxyFactory - org.springframework.web.serv
HttpServiceProxyFactoryExtensionsKt - org.springframew
HttpWebServiceMessageSenderBuilder - org.springframe

Factory to create a client proxy from an HTTP service interface with @HttpExchange methods.

To create an instance, use static methods to obtain a Builder.

Since:
6.0
Author:
Rossen Stoyanchev
See Also:
org.springframework.web.client.support.RestClientAdap
org.springframework.web.reactive.function.client.suppo
org.springframework.web.client.support.RestTemplateA

Servers
Termi
Press '^Space' to show Template Proposals
Press 'Tab' from proposal table or click for focus

using httpservice proxy filter, we are attaching our webclient employeeWebClient to the webclient adapter.

Basically we are creting object of webclient.. we are using this webclient to connect create an object of EmployeeClient class. We will use to access employee service api to get list of all department with employee.

At this point — below wont throw any error.

@Autowired
private EmployeeClient empClient;

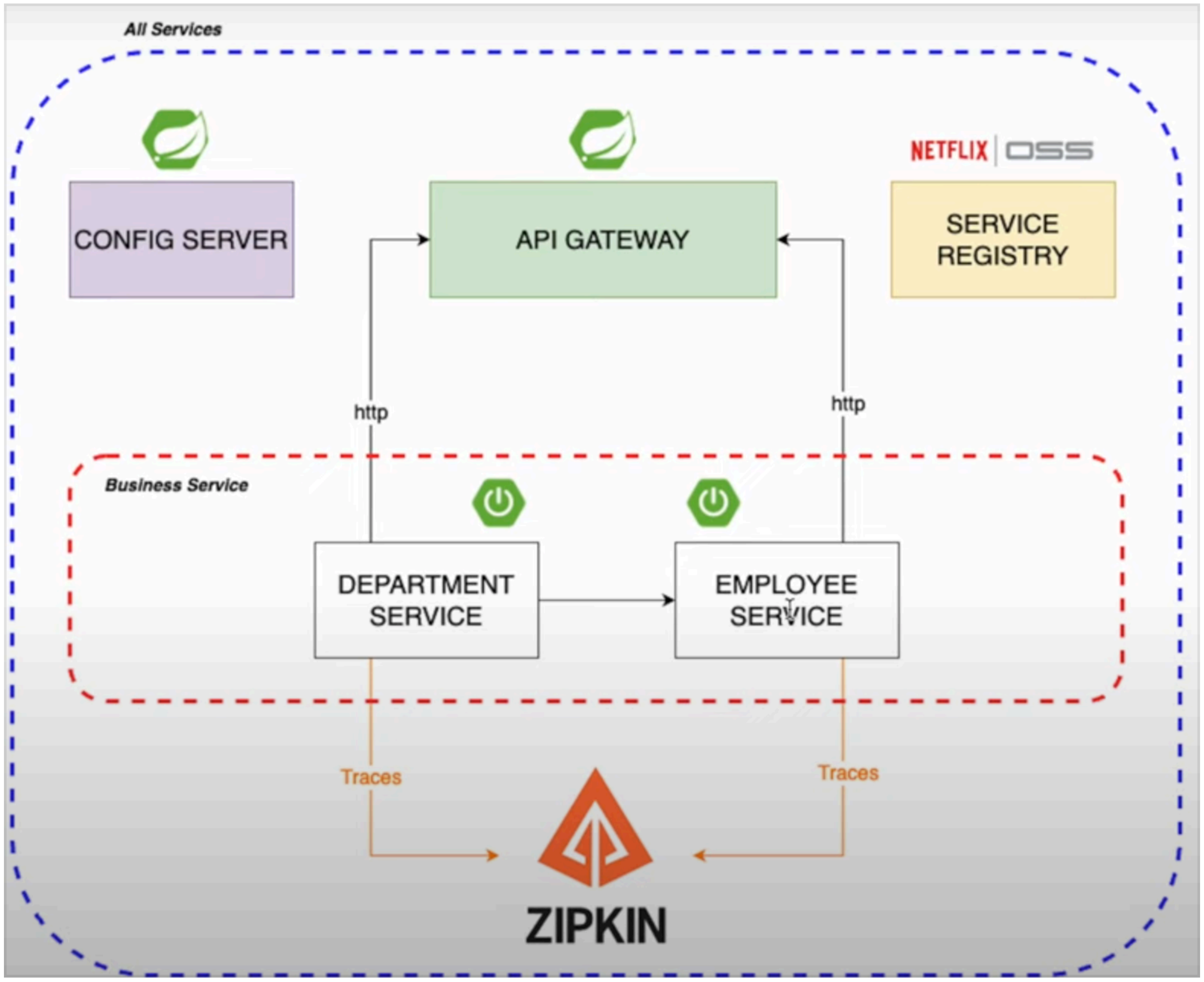
we got an object of EmployeeClient. We already told how to create an object of EmployeeClient to spring.

I have creted a new method in Departmetn Controller to get list of all departmetns along with employee..

Fetch all department… Then for each department, fetch all employees.

==============================

Everything is done now except API gateway…



As of now, we are directly calling services of employee and department… 8081 port and 8082 port… We do not have any centeralized public api gateway till now.

Lets create API gateway from spring initializer…

Dependencies :

gateway reactive (spring-cloud) — new gateway i guess.
euraka client
config client
zipkin
accurator

change properties to yaml again and add port and eureka clinet details.

add annotation on main class @EnableEurekaClient

after adding all general config like port, eureka clinet and zipking details… add details of services where it should redirect to.

We have to give same name of service as eureka..

```
spring:
  application:
    name: api-gateway
  cloud:
    gateway:
      routes:
        - id: employee-service
          uri: lb://employee-service
          predicates:
            - Path=/employee/**
```

Simple configuration to tell api gateway that whenever I hit 8060 port with path /employee/* redirect trafic to employee-service via service registry load balance… lb:// is load balancer.

We can add all this configuration to config-server… just tutor do not wanted to restart services so he added here as it… hahaha

one everything is done, if I execute same api with port 8060 (api gateway), everything should run as it is..

API gateway is giving some error. Did not get time to solve it. Many others are getting same error. api gateway is not getting registered on eureka servier somehow.