# API GATEWAY

This service is used as a central enrty point to route requests (**coming from the clients – POSTMAN -** ) to customer-service and address-service (later on hope to logging-service). API Gateway is essential in a microservices architecture. In this project, API Gateway is a Spring Cloud Gateway, which is a part of Spring Cloud and acts as a reverse proxy server. It sits between the clients(in this casei it’s POSTMAN – since there is no frontend) and the backend microservices (address-service, customer-service, logging-service) to handle and route requests.

## 1.What is an API Gateway, Its Importance?

An API Gateway is a central point to receives client requests and forwards them to the appropriate microservice. It is an entry point (giriş noktası) for all client requests in a microservice architecture. Instead of exposing multiservices directly, it exposes only the gateway, which:

1. Routes requests to the correct microservice
2. Handles authentication and security (later when adding Keycloak)
3. Provides centralized logging and monitoring (for the future logging-service integration)
4. Allows load balancing and failover

API Gateway is a Spring Cloud Server designed for:

- Routing(sending request to the right microservice)

- Load Balancing (distributing requests efficiently)

- Security (authentication, authorization)

- Logging, Monitoring (tracking requests/responses)

- Rate Limiting (preventig overload with request limits)

## 2.Which technologies to use?

**Spring Cloud Gateway**: Recommended in Spring Boot microservices arhitecture

1. **Lightweight and fast** (built on Spring WebFlux & Reactor)
2. **Easily integrates with Eureka** for service discovery
3. **Supports filters** (useful for authentication, logging, etc.)
4. **Can handle rate-limiting, circuit breaking, and resilience patterns**

Since Spring Cloud Gateway dependency

## 3.Required Dependencies:

1. **spring-cloud-starter-gateway** – for API Gateway functionality
2. **spring-cloud-starter-netflix-eureka-client** – for service discovery
3. **spring-boot-starter-actuator** – for monitoring
4. **spring-boot-starter-webflux** – because Spring Cloud Gateway is **reactive\***

### 3.1 Reactive Programming:

Spring Cloud Gateway operates based on **reactive programming** principles.This means that is specifically has **non-blocking** and **asynchronous** processing capabilities.

**Reactive Programming Paradigm** models data as streans that can be processed over time. The idea is to handle changes or events as they happen, rather than in a step-by-step or blocking way. The goal isto make systems more efficeint and capable of handling high loads with lower resource consumption.

Reactive systems don’t block threads while waiting for tasks to complete. For example when an http request is made, the system doesn’t need to wait for it to finish before performing other operations. This helps achieve higher performance with reduced latency.

Reactive systems are typically asynchronous, meaning that tasks are executed independently of each other. This allows the system to handle multiple tasks in parallel, which improves scalibility and resource utilization.

Since Spring Cloud Gateway is built on Spring Webflux, a framework designed to support reactive programming, SCG uses non-blocking I/O operations to process incoming HTTP requests. This allows it to handle a larger number of concurrent connections more efficiently. Each incoming request is processed asynchronously. This means that the system doesn’t wait for one request to complete before starting the next, leading to faster response times. Reactive programming ensures that resources (like threads and CPU) are used efficiently, as tasks do not block each other. This results in lower resource consumption and quicker processing.

## 4.Implementation: Step-by-Step Guide:

### 4.1. Creating the API Gateway Service:

Create a new Spring Boot Project for api-gateway with dependencies of:

1. Spring Cloud Gateway (Spring Cloud Routing – Gateway option in intellij idea dependencies section)
2. Spring Boot Actuator
3. Spring Cloud Netflix Eureka Client (Spring Cloud Gateway already includes WebFlux internally – no need to add as a seperate dependency)
4. Spring Boot Starter WebFlux

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

### 4.2.MANIFEST.MF:

|  |
| --- |
| Manifest-Version: 1.0  Created-By: Maven JAR Plugin 3.3.0  Build-Jdk-Spec: 21  Implementation-Title: api-gateway  Implementation-Version: 0.0.1-SNAPSHOT  Main-Class: org.springframework.boot.loader.launch.JarLauncher  Start-Class: com.microservices.apigateway.ApiGatewayApplication  Spring-Boot-Version: 3.2.2  Spring-Boot-Classes: BOOT-INF/classes/  Spring-Boot-Lib: BOOT-INF/lib/  Spring-Boot-Classpath-Index: BOOT-INF/classpath.idx  Spring-Boot-Layers-Index: BOOT-INF/layers.idx |

## 5. Extra Implementations:

### 5.1.CORS Filter:

**CORS** (Cross-Origin Resource Sharing) controls how browsers allow or block requests from different origins (domains). Its advantages:

* **Centralized CORS Handling:** Instead of adding CORS to every other microservices (customer-service, address-service) you can define it once in api-gateway. This reduces duplication and simplifies management.
* **Better Security:** You can restrict which domains (origins) are allowed to cal you APIs. Example: allow only “htto://someURL.com” instead of \* (open to all).
* **Frontend Compatiblity:** Without CORS enabled, browser may block API requests made from frontend applications hosted on different domains.

Its disadvanteges:

* **If misconfigured, it can weaken security:** Setting “AllowedOrigins=\*” allows any website to send requests to your API, which may be risky.
* **Potential Performance Impact:** Some browser make preflight requests (OPTIONS requests) before the actual request, adding extra overhead.

Current controls in CORSConfig:

|  |
| --- |
| TODO: //ADD LATER AFTER CONTROLLING  Hiç anlamadım, daha sonra CORS’da pre-flight OPTIONS seçeneği neymiş bak anla. Ayrıca WebFluxConfigurer’ı de incele.  ReactiveWebServerApplicationContext |

Test cases (since there is no frontend in current project structure):

TODO://Add test cases later

## 6.API-gateway-based Communication between Microservices:

### 6.1.The Purpose:

Since logging-service is not working for now, and we didn’t update the project for a authorization service yet, the only communication between the microservices is from customer-service to address-service (for now) via restTemplate. So it’s better to provide this communication over api-gateway. Intercommunication without api-gateway has some pros and cons:

Pros:

* Direct restTemplate communication is faster and direct (current situation)
* It causes less load on API gateway, reduces bottlenecks
* It provide independent microservices, if API gateway failures somehow, it won’t block internal calls.

But on the other hand, it has some cons:

* It is harder to manage security: Each microservices must handle its own authentication, authorization.
* Circuit breaking and rate limiting must be implemented manually.
* It is harder to centralize logs: calls bypass the gateway logging mechanism.

So it’s better to provide communication of microservices instead of calling directly, each microservices calls API gateway in a required position. (customer-service to address-service)

### 6.2. Update api-gateway to Route Internal Microservice Calls:

Before updating the project for an API-GW-based communication, the customer-service calls address-service directly like this (CustomerController):

|  |
| --- |
| private List<AddressDTO> fetchCustomerAddresses(Long customerId) {  //logger.info("fetchCustomerAddress method starts in CustomerController");  String tempServiceUrl = **addressServiceUrl** + "/customer/" + customerId;  //logger.info("The address-service url related with customerId is: " + tempServiceUrl);  return restTemplate.exchange(  tempServiceUrl, HttpMethod.GET, null, new ParameterizedTypeReference<List<AddressDTO>>() {}  ).getBody();  } |

To provide an API-GW-based communication between microservices (it is only from customer-service to address-service for now):

1. Update RestTemplate URLs in customer-service
2. Ensure customer-service uses @LoadBalanced RestTemplate
3. Configure API Gateway to route requests

#### 6.2.1.Updating RestTemplate URLs:

Update any place calling “address-service URL” with api-gateway URL:

|  |
| --- |
| Current addressServiceUrl: **“http://address-service:8082/addresses”** |
| Updated to: **“http://api-service:8082/address-service/addresses”** |

Directly update the addressServiceUrl parameter (used in customer-service):

|  |  |
| --- | --- |
| @Value(**"${address-service.service.url}"**) **private** String **addressServiceUrl**; | @Value(**"${api-service.service.url}"**) **private** String **apiGatewayServiceUrl**; |
| apiGateway url in application.properties:  **api-service.service.url**=**http://api-gateway** | |

#### 6.2.2.@LoadBalanced RestTemplate:

A RestTemplate bean with @LoadBalanced annotation added to AppConfig of both customer-service and address-service:

|  |
| --- |
| @Configuration **public class** AppConfig {  @Bean  @LoadBalanced *// Enable load balancing with Eureka* **public** RestTemplate restTemplate() {  **return new** RestTemplate();  } } |

So that Spring Cloud Load Balancer allows RestTemplate to resolve logical service names (lb://address-service) using Eureka.

#### 6.2.3.Configuring API Gateway to route requests:

|  |  |
| --- | --- |
| **spring.application.name**=**api-gateway** | The name of the application. This name will be used when registering the service with Eureka. Other services can find and interact with the gateway with this name definition |
| **server.port**=**8082** | It specifies the port of the api-gateway application. The api-gateway will run on port 8082 |
| **eureka.client.service-url.defaultZone**= **http://localhost:8761/eureka** | It sets the Eureka server URL where the API Gateway will register itself. It points to a locally running Eureka server. Api-gateway will uses this URL to discover other services and to register itself in the Eureka registry. |
| **eureka.instance.prefer-ip-address**= **true** | It ensures that Eureka registers the service with its IP Address rather than its hostname. This is useful in environments where DNS resoloution might not be reliable or necessary. TODO://How to understand if it’s working |
| **management.endpoints.web.exposure.include**=  **\***  TODO://Will be explained in details | This enables all management endpoints (from Spring Boot Actuator) to be exposed over HTTP. This is useful for monitoring and managing the application. |
| **management.endpoint.gateway.enabled**= **true** | This enables the Spring Cloud Gateway Actuator endpoint, which exposes metrics and information about the routes configured in the gateway, making it easier to manage the gateway through Actuator. |
| **eureka.client.initial-instance-info-replication-interval-seconds**= **5** | This sets the interval in seconds for the replication of initial instance information from the Eureka client to the Eureka server. It is configured to 5 seconds here. |
| **spring.cloud.gateway.discovery.locator.enabled**= **false** | This disables the automatic discovery of services by the gateway through Spring Cloud's DiscoveryLocator. By default, this would allow the gateway to automatically pick up services registered in Eureka based on specific route patterns, but it’s set to false here, meaning manual route configuration will be used instead. |
| **spring.cloud.gateway.discovery.locator.lower-case-service-id**= **true** | This configures the gateway to convert service IDs to lowercase when discovering them. This can be useful if service names are case-sensitive or inconsistent |
| **spring.cloud.gateway.routes[0].id**= **customer-service** | The route ID for the customer-service route. Each route must have a unique identifier. |
| **spring.cloud.gateway.routes[0].uri**= **lb://CUSTOMER-SERVICE** | It specifies the URI for the customer-service service. The lb:// prefix indicates that the API Gateway should use load balancing to resolve the service through Eureka. It refers to the CUSTOMER-SERVICE service registered in Eureka. |
| **spring.cloud.gateway.routes[0].predicates[0]**= **Path=/customers/\*\*** | This defines the predicate (a condition for matching requests). In this case, it matches requests whose paths start with /customers/ (e.g., /customers/123). These requests will be routed to the customer-service. |
| **spring.cloud.gateway.routes[1].id**=  **address-service** | The route ID for the address-service route. Each route must have a unique identifier. |
| **spring.cloud.gateway.routes[1].uri**=  **lb://ADDRESS-SERVICE** | It specifies the URI for the customer-service service. The lb:// prefix indicates that the API Gateway should use load balancing to resolve the service through Eureka. It refers to the ADDRESS-SERVICE service registered in Eureka. |
| **spring.cloud.gateway.routes[1].predicates[0]**=  **Path=/addresses/\*\*** | This defines the predicate (a condition for matching requests). In this case, it matches requests whose paths start with /addresses/ (e.g., /addresses/123). These requests will be routed to the address-service. |
| **logging.level.org.springframework.cloud.gateway**=  **DEBUG** | This sets the logging level for the Spring Cloud Gateway to DEBUG. It provides detailed logs for the routing and handling of HTTP requests, making it easier to debug and trace what’s happening in the gateway. |
| **logging.level.org.springframework.boot.context.properties.bind**=  **DEBUG** | This enables debugging for the property binding mechanism in Spring Boot, which is helpful if you need to troubleshoot issues with loading or parsing properties. |
| **logging.level.org.springframework.cloud.netflix.eureka**=  **DEBUG** | This sets the logging level for Eureka-related logs to DEBUG. It will log detailed information about the communication with the Eureka server (e.g., registration of services, service discovery). |

### 6.3.Spring Boot Actuator:

#### 6.3.1.Spring Boot Actuator Management Endpoints:

|  |
| --- |
| **GET** [http://localhost:8082/**actuator/gateway/routes**](http://localhost:8082/actuator/gateway/routes)  **Response:**  [      {          "predicate": "Paths: [/customers/\*\*], match trailing slash: true",          "route\_id": "customer-service",          "filters": [],          "uri": "lb://CUSTOMER-SERVICE",          "order": 0      },      {          "predicate": "Paths: [/addresses/\*\*], match trailing slash: true",          "route\_id": "address-service",          "filters": [],          "uri": "lb://ADDRESS-SERVICE",          "order": 0      }  ] |
| **GET** <http://localhost:8082/actuator/health>  **Response:**  {      "status": "UP"  }  Shows the health status of the application |
| GET <http://localhost:8082/actuator/metrics> **Response:**  {      "names": [          "application.ready.time",          "application.started.time",          "disk.free",          "disk.total",          "executor.active",          "executor.completed",          "executor.pool.core",          "executor.pool.max",          "executor.pool.size",          "executor.queue.remaining",          "executor.queued",          "http.server.requests",          "http.server.requests.active",          "jvm.buffer.count",          "jvm.buffer.memory.used",          "jvm.buffer.total.capacity",          "jvm.classes.loaded",          "jvm.classes.unloaded",          "jvm.compilation.time",          "jvm.gc.live.data.size",          "jvm.gc.max.data.size",          "jvm.gc.memory.allocated",          "jvm.gc.memory.promoted",          "jvm.gc.overhead",          "jvm.gc.pause",          "jvm.info",          "jvm.memory.committed",          "jvm.memory.max",          "jvm.memory.usage.after.gc",          "jvm.memory.used",          "jvm.threads.daemon",          "jvm.threads.live",          "jvm.threads.peak",          "jvm.threads.started",          "jvm.threads.states",          "logback.events",          "process.cpu.usage",          "process.start.time",          "process.uptime",          "spring.cloud.gateway.routes.count",          "system.cpu.count",          "system.cpu.usage"      ]  }  Shows the performans data of the application |
| **GET** <http://localhost:8082/actuator/info>  **Response:** {}  Shows the information about application (version, build information, etc.) |
| **GET** <http://localhost:8082/actuator/env>  **Response:** {      "activeProfiles": [],      "propertySources": [          {              "name": "server.ports",              "properties": {                  "local.server.port": {                      "value": "\*\*\*\*\*\*"                  }              }          },          {              "name": "gateway-properties",              "properties": {                  "spring.webflux.hiddenmethod.filter.enabled": {                      "value": "\*\*\*\*\*\*"                  }              }          },          {              "name": "systemProperties",              "properties": {                  "java.specification.version": {                      "value": "\*\*\*\*\*\*"                  },                  "sun.cpu.isalist": {                      "value": "\*\*\*\*\*\*"                  },                  "sun.jnu.encoding": {                      "value": "\*\*\*\*\*\*"                  },                  "java.class.path": {                      "value": "\*\*\*\*\*\*"                  },                  "java.vm.vendor": {                      "value": "\*\*\*\*\*\*"                  },                  "sun.arch.data.model": {                      "value": "\*\*\*\*\*\*"                  },                  "user.variant": {                      "value": "\*\*\*\*\*\*"                  },                  "java.vendor.url": {                      "value": "\*\*\*\*\*\*"                  },                  "user.timezone": {                      "value": "\*\*\*\*\*\*"                  },                  "org.jboss.logging.provider": {                      "value": "\*\*\*\*\*\*"                  },                  "java.vm.specification.version": {                      "value": "\*\*\*\*\*\*"                  },                  "os.name": {                      "value": "\*\*\*\*\*\*"                  },                  "sun.java.launcher": {                      "value": "\*\*\*\*\*\*"                  },                  "user.country": {                      "value": "\*\*\*\*\*\*"                  },                  "sun.boot.library.path": {                      "value": "\*\*\*\*\*\*"                  },                  "spring.application.admin.enabled": {                      "value": "\*\*\*\*\*\*"                  },                  "sun.java.command": {                      "value": "\*\*\*\*\*\*"                  },                  "com.sun.management.jmxremote": {                      "value": "\*\*\*\*\*\*"                  },                  "jdk.debug": {                      "value": "\*\*\*\*\*\*"                  },                  "spring.liveBeansView.mbeanDomain": {                      "value": "\*\*\*\*\*\*"                  },                  "sun.cpu.endian": {                      "value": "\*\*\*\*\*\*"                  },                  "user.home": {                      "value": "\*\*\*\*\*\*"                  },                  "user.language": {                      "value": "\*\*\*\*\*\*"                  },                  "sun.stderr.encoding": {                      "value": "\*\*\*\*\*\*"                  },                  "java.specification.vendor": {                      "value": "\*\*\*\*\*\*"                  },                  "java.version.date": {                      "value": "\*\*\*\*\*\*"                  },                  "java.home": {                      "value": "\*\*\*\*\*\*"                  },                  "spring.output.ansi.enabled": {                      "value": "\*\*\*\*\*\*"                  },                  "file.separator": {                      "value": "\*\*\*\*\*\*"                  },                  "java.vm.compressedOopsMode": {                      "value": "\*\*\*\*\*\*"                  },                  "sun.stdout.encoding": {                      "value": "\*\*\*\*\*\*"                  },                  "line.separator": {                      "value": "\*\*\*\*\*\*"                  },                  "java.vm.specification.vendor": {                      "value": "\*\*\*\*\*\*"                  },                  "java.specification.name": {                      "value": "\*\*\*\*\*\*"                  },                  "FILE\_LOG\_CHARSET": {                      "value": "\*\*\*\*\*\*"                  },                  "java.awt.headless": {                      "value": "\*\*\*\*\*\*"                  },                  "user.script": {                      "value": "\*\*\*\*\*\*"                  },                  "sun.management.compiler": {                      "value": "\*\*\*\*\*\*"                  },                  "java.runtime.version": {                      "value": "\*\*\*\*\*\*"                  },                  "user.name": {                      "value": "\*\*\*\*\*\*"                  },                  "spring.jmx.enabled": {                      "value": "\*\*\*\*\*\*"                  },                  "stdout.encoding": {                      "value": "\*\*\*\*\*\*"                  },                  "path.separator": {                      "value": "\*\*\*\*\*\*"                  },                  "management.endpoints.jmx.exposure.include": {                      "value": "\*\*\*\*\*\*"                  },                  "os.version": {                      "value": "\*\*\*\*\*\*"                  },                  "java.runtime.name": {                      "value": "\*\*\*\*\*\*"                  },                  "file.encoding": {                      "value": "\*\*\*\*\*\*"                  },                  "java.vm.name": {                      "value": "\*\*\*\*\*\*"                  },                  "java.vendor.url.bug": {                      "value": "\*\*\*\*\*\*"                  },                  "java.io.tmpdir": {                      "value": "\*\*\*\*\*\*"                  },                  "java.version": {                      "value": "\*\*\*\*\*\*"                  },                  "user.dir": {                      "value": "\*\*\*\*\*\*"                  },                  "os.arch": {                      "value": "\*\*\*\*\*\*"                  },                  "java.vm.specification.name": {                      "value": "\*\*\*\*\*\*"                  },                  "PID": {                      "value": "\*\*\*\*\*\*"                  },                  "sun.os.patch.level": {                      "value": "\*\*\*\*\*\*"                  },                  "CONSOLE\_LOG\_CHARSET": {                      "value": "\*\*\*\*\*\*"                  },                  "native.encoding": {                      "value": "\*\*\*\*\*\*"                  },                  "java.library.path": {                      "value": "\*\*\*\*\*\*"                  },                  "java.vm.info": {                      "value": "\*\*\*\*\*\*"                  },                  "stderr.encoding": {                      "value": "\*\*\*\*\*\*"                  },                  "java.vendor": {                      "value": "\*\*\*\*\*\*"                  },                  "java.vm.version": {                      "value": "\*\*\*\*\*\*"                  },                  "java.rmi.server.randomIDs": {                      "value": "\*\*\*\*\*\*"                  },                  "sun.io.unicode.encoding": {                      "value": "\*\*\*\*\*\*"                  },                  "java.class.version": {                      "value": "\*\*\*\*\*\*"                  },                  "LOGGED\_APPLICATION\_NAME": {                      "value": "\*\*\*\*\*\*"                  }              }          },          {              "name": "systemEnvironment",              "properties": {                  "USERDOMAIN\_ROAMINGPROFILE": {                      "value": "\*\*\*\*\*\*",                      "origin": "System Environment Property \"USERDOMAIN\_ROAMINGPROFILE\""                  },                  "LOCALAPPDATA": {                      "value": "\*\*\*\*\*\*",                      "origin": "System Environment Property \"LOCALAPPDATA\""                  },                  "PROCESSOR\_LEVEL": {                      "value": "\*\*\*\*\*\*",                      "origin": "System Environment Property \"PROCESSOR\_LEVEL\""                  },                  "USERDOMAIN": {                      "value": "\*\*\*\*\*\*",                      "origin": "System Environment Property \"USERDOMAIN\""                  },                  "FPS\_BROWSER\_APP\_PROFILE\_STRING": {                      "value": "\*\*\*\*\*\*",                      "origin": "System Environment Property \"FPS\_BROWSER\_APP\_PROFILE\_STRING\""                  },                  "LOGONSERVER": {                      "value": "\*\*\*\*\*\*",                      "origin": "System Environment Property \"LOGONSERVER\""                  },                  "JAVA\_HOME": {                      "value": "\*\*\*\*\*\*",                      "origin": "System Environment Property \"JAVA\_HOME\""                  },                  "SESSIONNAME": {                      "value": "\*\*\*\*\*\*",                      "origin": "System Environment Property \"SESSIONNAME\""                  },                  "ALLUSERSPROFILE": {                      "value": "\*\*\*\*\*\*",                      "origin": "System Environment Property \"ALLUSERSPROFILE\""                  },                  "PROCESSOR\_ARCHITECTURE": {                      "value": "\*\*\*\*\*\*",                      "origin": "System Environment Property \"PROCESSOR\_ARCHITECTURE\""                  },                  "PSModulePath": {                      "value": "\*\*\*\*\*\*",                      "origin": "System Environment Property \"PSModulePath\""                  },                  "SystemDrive": {                      "value": "\*\*\*\*\*\*",                      "origin": "System Environment Property \"SystemDrive\""                  },                  "MAVEN\_HOME": {                      "value": "\*\*\*\*\*\*",                      "origin": "System Environment Property \"MAVEN\_HOME\""                  },                  "OneDrive": {                      "value": "\*\*\*\*\*\*",                      "origin": "System Environment Property \"OneDrive\""                  },                  "APPDATA": {                      "value": "\*\*\*\*\*\*",                      "origin": "System Environment Property \"APPDATA\""                  },                  "USERNAME": {                      "value": "\*\*\*\*\*\*",                      "origin": "System Environment Property \"USERNAME\""                  },                  "ProgramFiles(x86)": {                      "value": "\*\*\*\*\*\*",                      "origin": "System Environment Property \"ProgramFiles(x86)\""                  },                  "CommonProgramFiles": {                      "value": "\*\*\*\*\*\*",                      "origin": "System Environment Property \"CommonProgramFiles\""                  },                  "Path": {                      "value": "\*\*\*\*\*\*",                      "origin": "System Environment Property \"Path\""                  },                  "FPS\_BROWSER\_USER\_PROFILE\_STRING": {                      "value": "\*\*\*\*\*\*",                      "origin": "System Environment Property \"FPS\_BROWSER\_USER\_PROFILE\_STRING\""                  },                  "PATHEXT": {                      "value": "\*\*\*\*\*\*",                      "origin": "System Environment Property \"PATHEXT\""                  },                  "DriverData": {                      "value": "\*\*\*\*\*\*",                      "origin": "System Environment Property \"DriverData\""                  },                  "OS": {                      "value": "\*\*\*\*\*\*",                      "origin": "System Environment Property \"OS\""                  },                  "COMPUTERNAME": {                      "value": "\*\*\*\*\*\*",                      "origin": "System Environment Property \"COMPUTERNAME\""                  },                  "PROCESSOR\_REVISION": {                      "value": "\*\*\*\*\*\*",                      "origin": "System Environment Property \"PROCESSOR\_REVISION\""                  },                  "CommonProgramW6432": {                      "value": "\*\*\*\*\*\*",                      "origin": "System Environment Property \"CommonProgramW6432\""                  },                  "ComSpec": {                      "value": "\*\*\*\*\*\*",                      "origin": "System Environment Property \"ComSpec\""                  },                  "ProgramData": {                      "value": "\*\*\*\*\*\*",                      "origin": "System Environment Property \"ProgramData\""                  },                  "ProgramW6432": {                      "value": "\*\*\*\*\*\*",                      "origin": "System Environment Property \"ProgramW6432\""                  },                  "HOMEPATH": {                      "value": "\*\*\*\*\*\*",                      "origin": "System Environment Property \"HOMEPATH\""                  },                  "SystemRoot": {                      "value": "\*\*\*\*\*\*",                      "origin": "System Environment Property \"SystemRoot\""                  },                  "TEMP": {                      "value": "\*\*\*\*\*\*",                      "origin": "System Environment Property \"TEMP\""                  },                  "ESET\_OPTIONS": {                      "value": "\*\*\*\*\*\*",                      "origin": "System Environment Property \"ESET\_OPTIONS\""                  },                  "HOMEDRIVE": {                      "value": "\*\*\*\*\*\*",                      "origin": "System Environment Property \"HOMEDRIVE\""                  },                  "PROCESSOR\_IDENTIFIER": {                      "value": "\*\*\*\*\*\*",                      "origin": "System Environment Property \"PROCESSOR\_IDENTIFIER\""                  },                  "USERPROFILE": {                      "value": "\*\*\*\*\*\*",                      "origin": "System Environment Property \"USERPROFILE\""                  },                  "TMP": {                      "value": "\*\*\*\*\*\*",                      "origin": "System Environment Property \"TMP\""                  },                  "CommonProgramFiles(x86)": {                      "value": "\*\*\*\*\*\*",                      "origin": "System Environment Property \"CommonProgramFiles(x86)\""                  },                  "ProgramFiles": {                      "value": "\*\*\*\*\*\*",                      "origin": "System Environment Property \"ProgramFiles\""                  },                  "PUBLIC": {                      "value": "\*\*\*\*\*\*",                      "origin": "System Environment Property \"PUBLIC\""                  },                  "NUMBER\_OF\_PROCESSORS": {                      "value": "\*\*\*\*\*\*",                      "origin": "System Environment Property \"NUMBER\_OF\_PROCESSORS\""                  },                  "windir": {                      "value": "\*\*\*\*\*\*",                      "origin": "System Environment Property \"windir\""                  },                  "=::": {                      "value": "\*\*\*\*\*\*",                      "origin": "System Environment Property \"=::\""                  },                  "IDEA\_INITIAL\_DIRECTORY": {                      "value": "\*\*\*\*\*\*",                      "origin": "System Environment Property \"IDEA\_INITIAL\_DIRECTORY\""                  },                  "ZES\_ENABLE\_SYSMAN": {                      "value": "\*\*\*\*\*\*",                      "origin": "System Environment Property \"ZES\_ENABLE\_SYSMAN\""                  }              }          },          {              "name": "Config resource 'class path resource [application.properties]' via location 'optional:classpath:/'",              "properties": {                  "spring.application.name": {                      "value": "\*\*\*\*\*\*",                      "origin": "class path resource [application.properties] - 2:25"                  },                  "server.port": {                      "value": "\*\*\*\*\*\*",                      "origin": "class path resource [application.properties] - 3:13"                  },                  "eureka.client.service-url.defaultZone": {                      "value": "\*\*\*\*\*\*",                      "origin": "class path resource [application.properties] - 6:39"                  },                  "eureka.instance.prefer-ip-address": {                      "value": "\*\*\*\*\*\*",                      "origin": "class path resource [application.properties] - 7:35"                  },                  "management.endpoints.web.exposure.include": {                      "value": "\*\*\*\*\*\*",                      "origin": "class path resource [application.properties] - 9:43"                  },                  "management.endpoint.gateway.enabled": {                      "value": "\*\*\*\*\*\*",                      "origin": "class path resource [application.properties] - 10:37"                  },                  "eureka.client.initial-instance-info-replication-interval-seconds": {                      "value": "\*\*\*\*\*\*",                      "origin": "class path resource [application.properties] - 12:66"                  },                  "spring.cloud.gateway.discovery.locator.enabled": {                      "value": "\*\*\*\*\*\*",                      "origin": "class path resource [application.properties] - 13:48"                  },                  "spring.cloud.gateway.discovery.locator.lower-case-service-id": {                      "value": "\*\*\*\*\*\*",                      "origin": "class path resource [application.properties] - 14:62"                  },                  "spring.cloud.gateway.routes[0].id": {                      "value": "\*\*\*\*\*\*",                      "origin": "class path resource [application.properties] - 17:35"                  },                  "spring.cloud.gateway.routes[0].uri": {                      "value": "\*\*\*\*\*\*",                      "origin": "class path resource [application.properties] - 18:36"                  },                  "spring.cloud.gateway.routes[0].predicates[0]": {                      "value": "\*\*\*\*\*\*",                      "origin": "class path resource [application.properties] - 20:46"                  },                  "spring.cloud.gateway.routes[1].id": {                      "value": "\*\*\*\*\*\*",                      "origin": "class path resource [application.properties] - 23:35"                  },                  "spring.cloud.gateway.routes[1].uri": {                      "value": "\*\*\*\*\*\*",                      "origin": "class path resource [application.properties] - 24:36"                  },                  "spring.cloud.gateway.routes[1].predicates[0]": {                      "value": "\*\*\*\*\*\*",                      "origin": "class path resource [application.properties] - 26:46"                  },                  "logging.level.org.springframework.cloud.gateway": {                      "value": "\*\*\*\*\*\*",                      "origin": "class path resource [application.properties] - 30:49"                  },                  "logging.level.org.springframework.boot.context.properties.bind": {                      "value": "\*\*\*\*\*\*",                      "origin": "class path resource [application.properties] - 31:64"                  },                  "logging.level.org.springframework.cloud.netflix.eureka": {                      "value": "\*\*\*\*\*\*",                      "origin": "class path resource [application.properties] - 32:56"                  }              }          },          {              "name": "springCloudClientHostInfo",              "properties": {                  "spring.cloud.client.hostname": {                      "value": "\*\*\*\*\*\*"                  },                  "spring.cloud.client.ip-address": {                      "value": "\*\*\*\*\*\*"                  }              }          }      ]  }  Shoes the environment information of the application |
| **GET** <http://localhost:8082/actuator/loggers>  **Response:** asdsa  {  "levels": [  "OFF",  "ERROR",  "WARN",  "INFO",  "DEBUG",  "TRACE"  ],  "loggers": {  "ROOT": {  "configuredLevel": "DEBUG",  "effectiveLevel": "DEBUG"  },  "\_org": {  "effectiveLevel": "DEBUG"  },  "\_org.springframework": {  "effectiveLevel": "DEBUG"  },  "\_org.springframework.web": {  "effectiveLevel": "DEBUG"  }, ..........  ]  Shows log levels in the application. Check also logLevels in properties:  **logging.level.org.springframework.cloud.gateway.handler.RoutePredicateHandlerMapping**=**DEBUG logging.level.com.microservices**=**DEBUG logging.level.org.springframework.cloud.netflix.eureka**=**DEBUG** |

## 7. General Operation Strategy of API-GW:

|  |  |
| --- | --- |
| <**dependency**>  *<!-- Spring Cloud Gateway -->* <**groupId**>org.springframework.cloud</**groupId**>  <**artifactId**>spring-cloud-starter-gateway</**artifactId**> </**dependency**> | This automatically tells Spring Boot that this service should act as a Gateway. |
| <**dependency**>  *<!-- Eureka Client -->* <**groupId**>org.springframework.cloud</**groupId**>  <**artifactId**>spring-cloud-starter-netflix-eureka-client</**artifactId**> </**dependency**> | This provides API Gateway to work as a Eureka Client. After the configuration in application.properties, other services recognize it as a service that routes requests. |
| Example of other services configuration:  spring.cloud.gateway.routes[0].id=customer-service spring.cloud.gateway.routes[0].uri=lb://CUSTOMER-SERVICE spring.cloud.gateway.routes[0].predicates[0]=Path=/customer-service/\*\* | When a client sends a request like <http://localhost:8080/customer-service/customers>, the gateway intercepts the request and routes it based on predefined routes. |
| spring.cloud.gateway.routes[0].uri=lb://CUSTOMER-SERVICE | This config tells the gateway to use Eureka load balancing. |
|  |  |
|  |  |

# 8.Updated Architecture of the Customer Web Service:

Now that with spring-cloud-starter-gateway dependency, the clients (POSTMAN Requests) will send their requests to the API Gateway instead of directly calling each microservices (customer-service, address-service).

It acts like a reverse proxy that sits between the client and backend microservices.

Microservices: logging-service, address-service, customer-service

Spring Cloud Server: api-gateway

Service Registry Server: Eureka Server