# API GATEWAY

This service is used as a central enrty point to route requests to customer-service and address-service (later on hope to logging-service). API Gateway is essential in a microservices architecture.

## 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. 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

## 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**

## 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**

## 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