# API Gateway: Nestjs API Gateway Implementation in a Microservices Architecture

Fei Gu

May 28, 2025

Github-Repository: feix0033/easv.pbsw.dbd-sys.synopsis

## 1 Introduction

The Nestjs framework is a progressive Node.js framework for building efficient, reliable, and scalable server-side applications. It is built with TypeScript and heavily inspired by Angular, making it a great choice for developers familiar with Angular's architecture. Nestjs provides a powerful module system, dependency injection, and a rich set of decorators that make it easy to build complex applications. It also has built-in support for microservices, making it an ideal choice for implementing an API Gateway in a microservices architecture.

The API Gateway is a crucial component in a microservices architecture, serving as a single entry point for all client requests. It handles request routing, composition, and protocol translation, allowing clients to interact with multiple microservices seamlessly. Nestjs provides the tools and features necessary to implement an efficient and scalable API Gateway.

Because of the Nestjs framwork was inspired by Angular, it is easy to integrate with Angular applications. This makes it a suitable choice for building an API Gateway that can serve as a backend for Angular applications, providing a consistent and efficient way to manage API requests and responses.

## 2 Problem Statement

In a microservices architecture, managing multiple services can become complex and cumbersome. Each service may have its own API, requiring clients to make multiple requests to different endpoints. This can lead to increased latency, higher network traffic, and a more complicated client-side implementation. An API Gateway addresses these challenges by providing a unified interface for clients to interact with multiple microservices. It simplifies the client-side implementation, reduces latency by aggregating responses from multiple services, and can handle cross-cutting concerns such as authentication, logging, and rate limiting.

The goal of this project is to implement an API Gateway using the Nestjs framework that can efficiently route requests to various microservices, handle protocol translation, and provide a seamless experience for clients. The API Gateway will also be designed to integrate easily with Angular applications, leveraging Nestjs's capabilities to create a robust and scalable solution.

## 3 Methodology

## 4 Analysis & Results

## 5 Conclusion