

Introduction to microservices with NestJS

Node.js Accelerator – Jan'23

Agenda

1. What is a microservice?

2. REST and Microservices

3. Api Gateway and REST

4. Internal communication

5. NestJS Microservices



Node.js Accelerator

Jan 2023

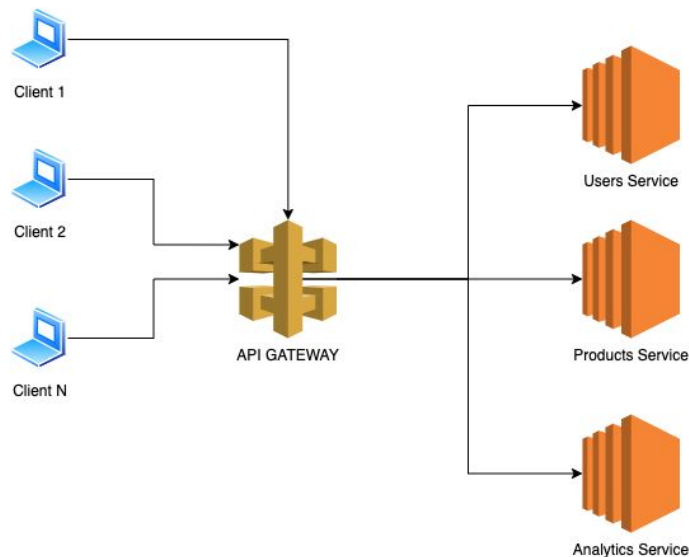
Microservices arranges a collection of loosely-coupled, fine-grained services, communicating through lightweight protocols.

Some of benefits includes:

- Improved Scalability
- Programming Language and Technology Agnostic
- Business Agility

More on benefits of [microservices](#)

Microservices are an architectural approach on how you build isolated applications meanwhile REST APIs can be divided into single modules (microservices) and to be orchestrated via API Gateway.



For REST API using microservices, is common to have the API Gateway on the top of it. One the simplest ways to manage your own API Gateway with Node.JS (apart from cloud providers) is to use one of the following npm packages:

- [http-proxy-middleware](#)
- [express-gateway](#)

For other approaches such as GraphQL you can use [Apollo Federation](#)

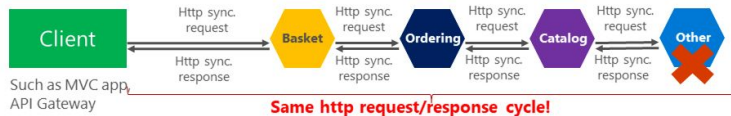
A good practice when having several microservices is to **not** make synchronous requests but to use asynchronous requests

Synchronous vs. async communication across microservices

Anti-pattern



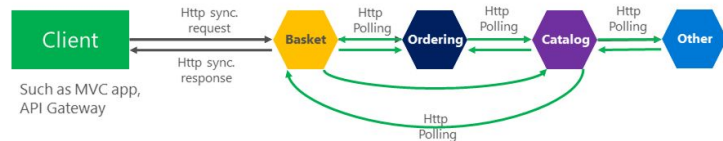
Synchronous
all request/response
cycle



Asynchronous
Comm. across internal
microservices
(EventBus: like **AMQP**)



"Asynchronous"
Comm. across
internal microservices
(Polling: **Http**)



Extracted from this [Microsoft Article](#)

Let's see one way to build microservices with [NestJS](#)



ACTION ITEMS

Week 5

1. We strongly recommend you to explore below module through Udemy this week and learn about Node.js essentials

Module	Topic	Description	Udemy Link	
Self-paced learning on Udemy	Session, cookies, Authentication, Sending Emails	Understand error handling strategy, catch erroring in async functions, use environment variables effectively, handle mongoose validation errors etc.	Click here	3 hrs 18 min
Self-paced learning on Udemy	Best Practices	Learn best practices in implementing validations, error handling, file uploads etc.	Click here	9 Hrs 23 Min

