LOKA

NestJS

Solution for clean node.js architecture

Skopje, November **2022**

Who are we ?

Loka.com

```
const engineer = 'Nikola Kusibojovski'
const company = 'Loka'

const companyDetails = {
  id: 1,
  name: 'Loka',
  website: 'https://loka.com/',
  headquarters: 'USA',
  founded: 2004,
  location: ['USA', 'Macedonia', 'Portugal', 'Columbia' ],
  specialities: [ 'Web Development', 'Mobile Development', 'Data Science', 'Machine
                  Learning','Artificial Intelligence']
}
```

Backend Development

What is most important when creating a web application from scratch looking from the backend perspective?

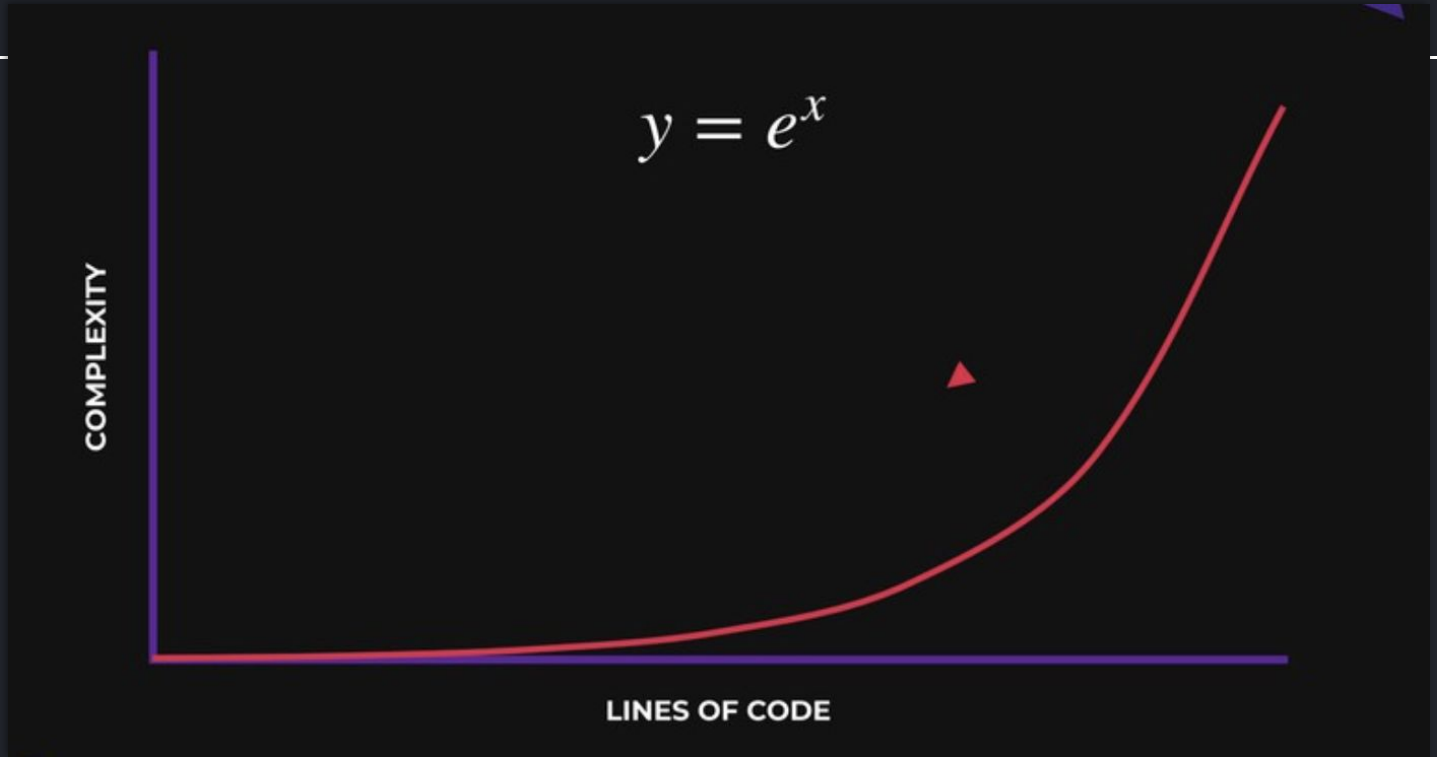
LOKA

Software Architecture

maintainability? performance? simplicity? scalability? productivity?

Why?

Because of **SIMPLICITY**



Most of the Node.js frameworks are

- Unopinionated
- Without predefined “software architecture”
- Complex for writing unit tests
- Don’t support Typescript by default
- “Everything” is a middleware

Express koa 

NestJS is a possible solution

- Opinionated
- Domain Driven Design
- Easy Unit-Testing
- Follows Angular way
- MVC ready
- Easy integration with OpenApi
- SOLID principles
- GraphQl
- Websockets
- Written in Typescript
- Microservices
- Command-line interface tool

We need to choose between

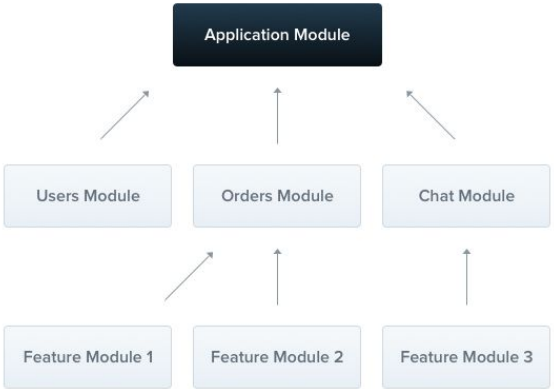
VS

Nest JS Architecture

Nest provides an out-of-the-box application architecture which allows developers and teams to create highly testable, scalable, loosely coupled, and easily maintainable applications.

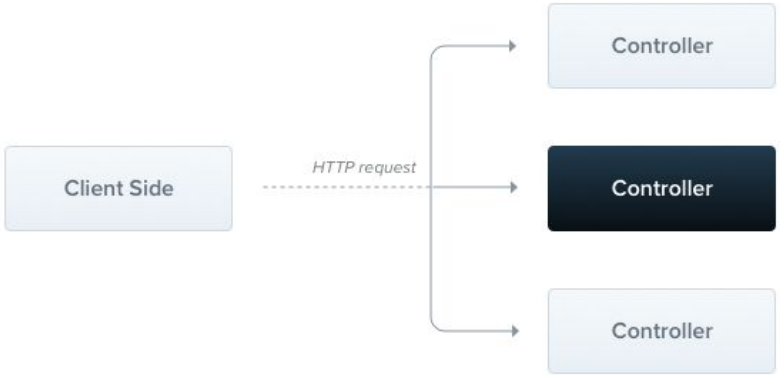
Modules

A module is a class annotated with a @Module() decorator. The @Module() decorator provides metadata that Nest makes use of to organize the application structure.



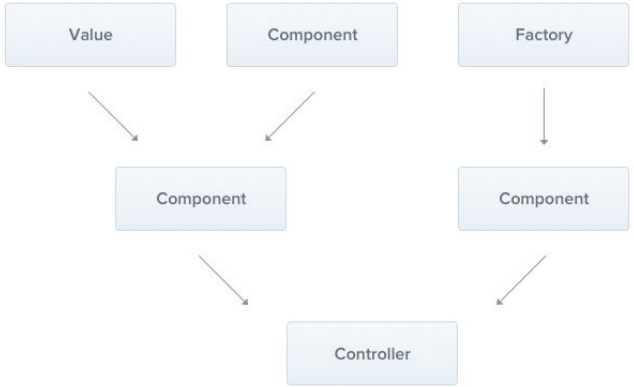
Controllers

Controllers are responsible for handling incoming requests and returning responses to the client.



Providers

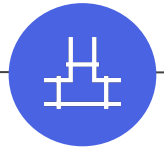
Providers are a fundamental concept in Nest. Many of the basic Nest classes may be treated as a provider – services, repositories, factories, helpers, and so on. The main idea of a provider is that it can be injected as a dependency; this means objects can create various relationships with each other, and the function of "wiring up" instances of objects can largely be delegated to the Nest runtime system.



What else ?

Middlewares

Exception filters



Pipes

Custom decorators

Guards

Interceptors

QA

Thank you!

Let’s talk.

Follow us on instagram!
@lifeatloka