

미래 노드  
떨감 스택  
NestJS 소개

백발백준(박인서)



# 강연자 소개

대학교 졸업과 동시에 공채로 N사 입사

Back-End 분야를 배정받고 팀에 가보니

팀원분들이 전부 Front-End 개발자?!

그렇게 노드 땀감으로 진화하게 되는데...

# — 사전 지식 점검

JavaScript?

Node.js?

# Node.js in B0J

node.js

- 언어 번호: 17
- 실행: `node --stack-size=65536 Main.js`
- 버전: `v16.13.1`
- 시간 제한:  $\times 3 + 2$  초
- 메모리 제한:  $\times 2$  MB

```
1 var fs = require('fs');
2 var input = fs.readFileSync('/dev/stdin').toString().split(' ');
3 var a = parseInt(input[0]);
4 var b = parseInt(input[1]);
5 console.log(a+b);
6
```

# Node.js



동적 웹 페이지를 만들기 위한 언어에 불과했던 JavaScript를  
다양한 환경 및 운영체제에서 활용 가능하도록 만들어 준  
JavaScript 런타임

# Node.js의 장/단점

## 장점

뛰어난 확장성

npm에 존재하는 다양한 라이브러리

## 단점

SW 품질 및 프로젝트 구조가 일정하지 않음

알맞은 라이브러리 찾기가 힘들음

# NestJS 소개



EXTENSIBLE



VERSATILE



PROGRESSIVE

효율적이고 신뢰성이 높으며  
확장성이 뛰어난 server-side application을  
구축하기 위한 진보적인 Node.js 프레임워크

# NestJS 소개

플랫폼에 구애받지 않는 프레임워크를 목표로 하고 있음.  
따라서 여러 HTTP 플랫폼과 연동할 수 있도록 구성됨.

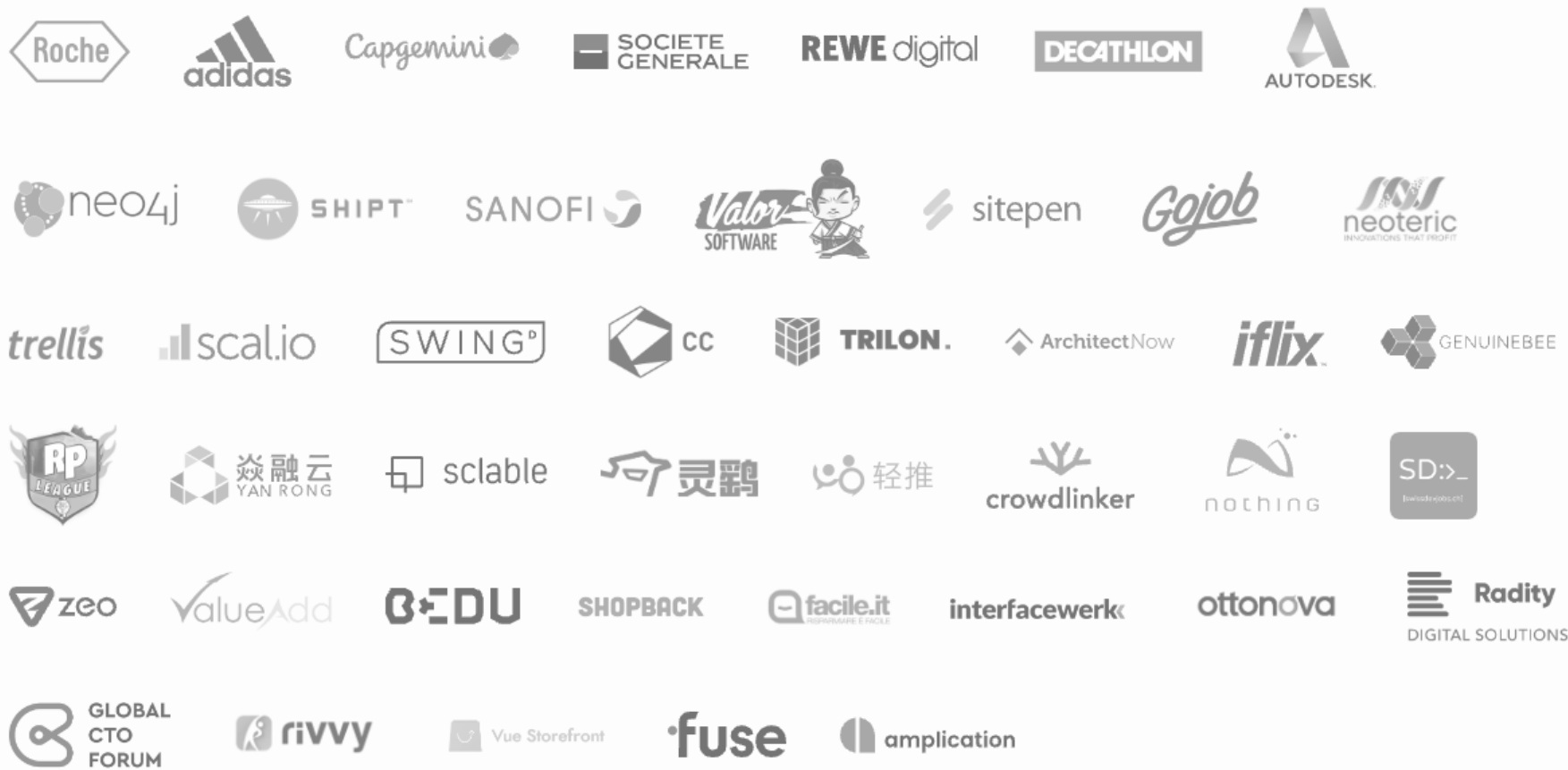
장기적으로 모든 Node.js의 HTTP 플랫폼과 연동이 목표지만,  
현재는 Express, Fastify라는 2가지 선택지가 존재함.



# Express vs NestJS

Express	NestJS
<ul style="list-style-type: none"><li>• 사용법이 단순하고 자유도가 높음</li><li>• 프로젝트 구조의 통일성을 갖기 힘들</li><li>• TypeScript 지원에 별도 설정이 필요함</li></ul>	<ul style="list-style-type: none"><li>• 디자인 패턴 적용을 통한 프로젝트 구조의 통일성을 가짐</li><li>• TypeScript 기본 지원</li></ul>

# NestJS 사용처 - 국외



# NestJS 사용처 - 국내

**NAVER Careers**

## G스마트플레이스서비스개발 ^

이런 일을 맡고 있어요

로컬 O2O 서비스를 개발합니다.  
예약/주문 서비스를 이용하는 고객뿐만 아니라 해당 업체를 관리하는 사업주 사이트의 서비스(스마트플레이스)를 함께 개발하고 있습니다.  
현재 일본 및 동남아, 미국 등 글로벌 진출 준비중에 있습니다.

입사하게 되면 담당할 업무입니다.

- 스마트플레이스 및 예약/주문 서비스의 글로벌 진출을 위한 국제화 적용 및 국가별 확장/특화 구조 설계를 담당하시게 됩니다.

업무를 수행하기 위해 필요한 자격 요건입니다.

- 기술 전파, 코드 리뷰 등 주변에 자극을 주시는 분
- 적극적으로 이슈 제기하고 능동적으로 해결하시는 분
- 동료를 배려하시는 분

추가로 보유하고 있다면 좋은 스킬셋이나 경험입니다.

- 다음의 기술 stack에 경험이 있으신분
- Javascript, Typescript, NodeJS, GraphQL, NestJS, TypeORM, Kafka, Elasticsearch, Redis, MongoDB, PostgreSQL
- 일본어, 영어

## [Tech] 기술고도화팀 NestJS 백엔드 개발자 모집

**[업무내용]**

- NestJS 기반의 백엔드 개발 표준 환경을 구축합니다.
  - 관련된 리서치와 개발을 진행하여 신뢰성있고 생산성 높은 개발 환경을 구축합니다.
  - 문서화를 포함하여 결과를 일부는 오픈소스로 공개될 수 있습니다.
- NestJS 기반의 백엔드 개발 표준 환경을 이용하여 다양한 프로젝트를 개발합니다.

**[지원자격]**

- 나이/성별/학력/전공 무관
- 웹/앱 서비스의 백엔드 실무 3년차 이상 또는 그에 준하는 역량을 보유하신 분
- Typescript에 익숙하고 OOP에 기반한 설계 및 개발 능력이 있으신 분
- Node.js 환경의 개발 경험이 있으신 분
- MySQL, PostgreSQL 등의 RDBMS 경험이 있으신 분

**[우대사항]**

- NestJS 또는 JVM 기반의 Spring 프레임워크(Spring Boot)를 이용한 개발 경험이 있으신 분
- TypeORM 또는 JPA 등 ORM 사용과 도메인 모델링 경험이 있으신 분
- 대용량 데이터/트래픽 서비스 개발 및 운영 경험이 있으신 분
- AWS와 같은 Cloud 환경에서 빌드, 배포, 모니터링 등의 운영 경험이 있으신 분
- Linux, Docker의 이해와 활용도가 높으신 분
- Unit/E2E 등 다양한 테스트 수행과 코드 커버리지 개선에 대한 경험이 있으신 분

## 직방 부동산 팀에서는 이런일을 해요.

- 직방의 메인서비스인 부동산 서비스 (아파트, 원룸, 오피스텔, 빌라등) 관련 개발업무를 해요.
- 서비스에 관련한 부수적인 기능들도 개발하고 관리해요. (배치, Admin, 검색 등)
- 온택트 파트너스와 같은 신규 비즈니스 서비스도 개발을 진행해요.
- 시스템의 안정적인 운영을 위한 노력을 해요. (모니터링, 트러블슈팅, 코드개선 등)
- AWS 를 통해 관련된 서비스를 운영해요.

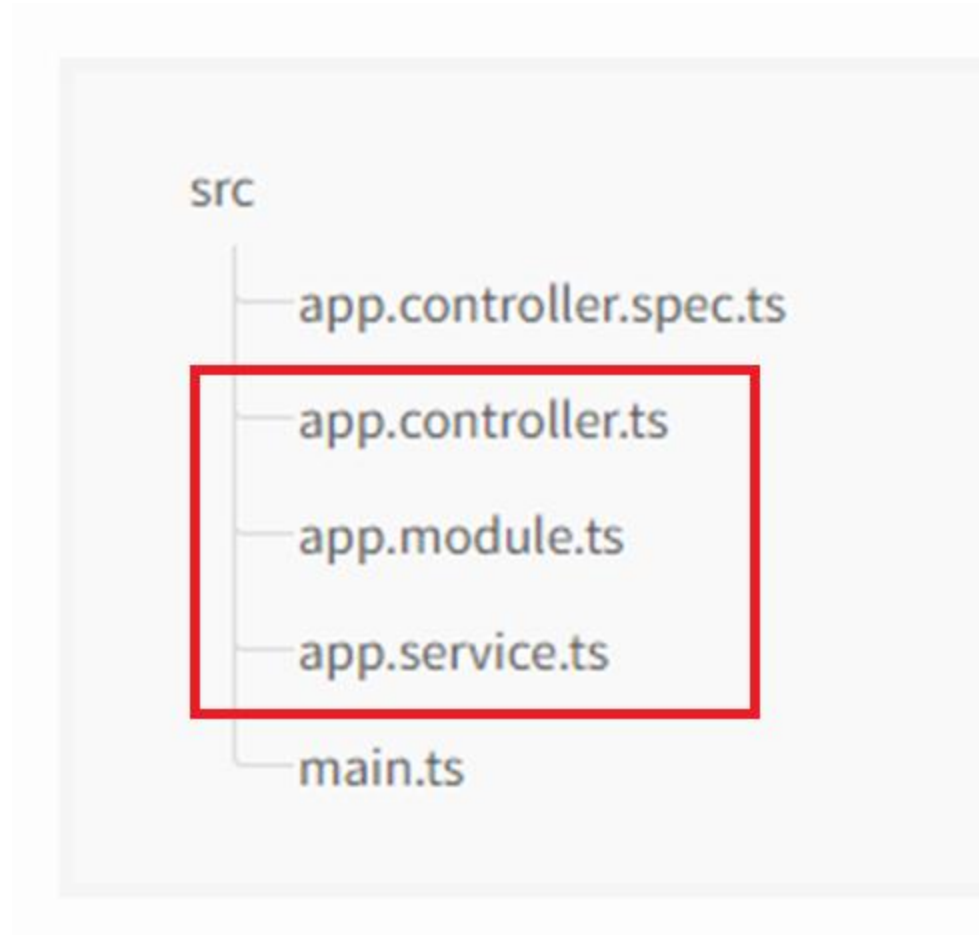
## 이런 기술을 사용해요.

- Node.js / TypeScript / C#
- Nest.js type-orm / Mongoose / Express / .net Core
- AWS / Azure DevOps / CodeBuild
- ECS / Fargate / Lambda / Elastic BeansTalk
- RDS / ElasticCache / ElasticSearch / Document DB / S3 / Athena
- GitHub / JIRA / Slack / Notion
- REST API

## 이런 분들 필요해요.

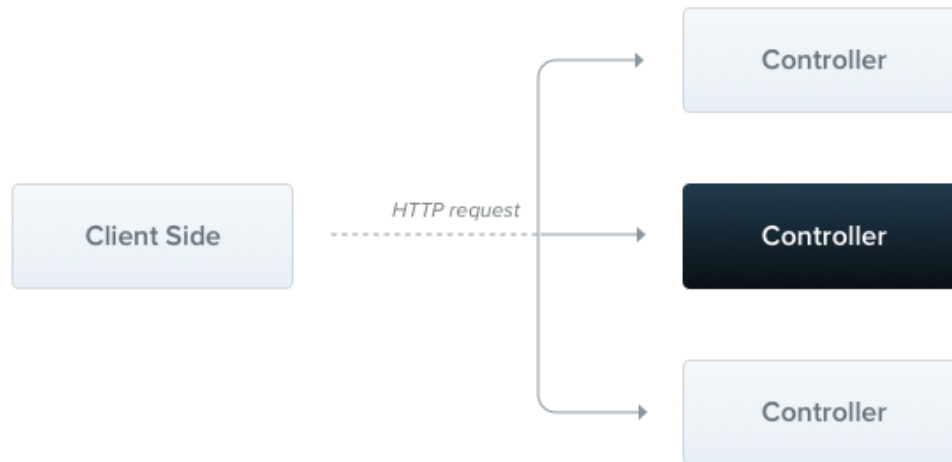
- 3년 이상의 개발경력이 있으신 분
- 최신 기술의 명목적 관심 보단, 비즈니스적 관점에서 개발을 하시는 분
- 말은 업무를 잘 파악하고, 일정을 잘 수립하고, 책임감 있게 수행하실 수 있는 분
- 인정적인 개발을 위해 테스트를 꼼꼼히 하실 수 있는 분
- 원격 근무 환경에 잘 적응하실 수 있는 분 (메타버스 활용, 화상 미팅시 비즈니스 매너등)
- 대화시 상대방을 배려하고, 친절한 언어를 사용하시는 분

# NestJS 구조



# Controller

클라이언트에서 들어오는 요청(request)과 응답(response)을  
가공하고 처리하는 역할



```
import { Request } from 'express';
import { Controller, Get, Query, Redirect, Req } from '@nestjs/common';
import { AppService } from './app.service';

@Controller()
export class AppController {
  constructor(private readonly appService: AppService) { }

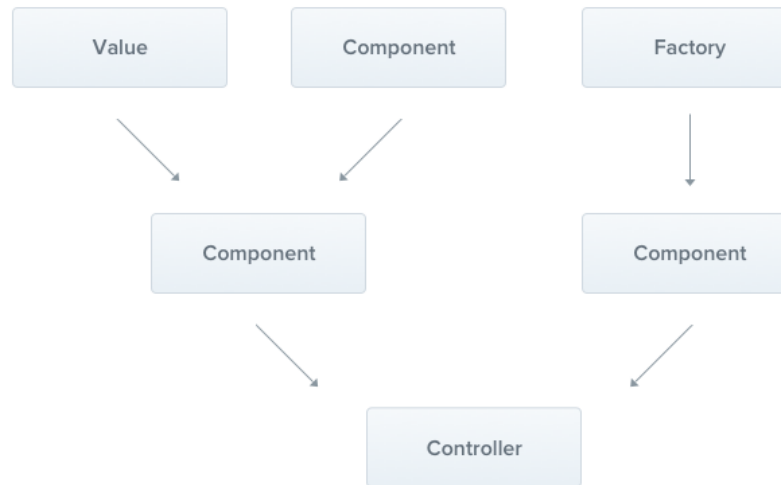
  @Get()
  getHello(@Req() req: Request): string {
    console.log(req);
    return this.appService.getHello();
  }

  @Get('/hello')
  getHello2(): string {
    return this.appService.getHello();
  }

  @Get('redirect/docs')
  @Redirect('https://docs.nestjs.com', 302)
  getDocs(@Query('version') version: string) {
    if (version && version === '5') {
      return { url: 'https://docs.nestjs.com/v5/' };
    }
  }
}
```

# Provider

제공하고자 하는 핵심 기능, 즉 비즈니스 로직을 수행하는 역할  
Service, Repository, Factory, Helper 등 여러가지 형태로 구현됨



# Provider



```
import { Injectable } from '@nestjs/common';

@Injectable()
export class AppService {
  getHello(): string {
    return 'Hello World!';
  }
}
```

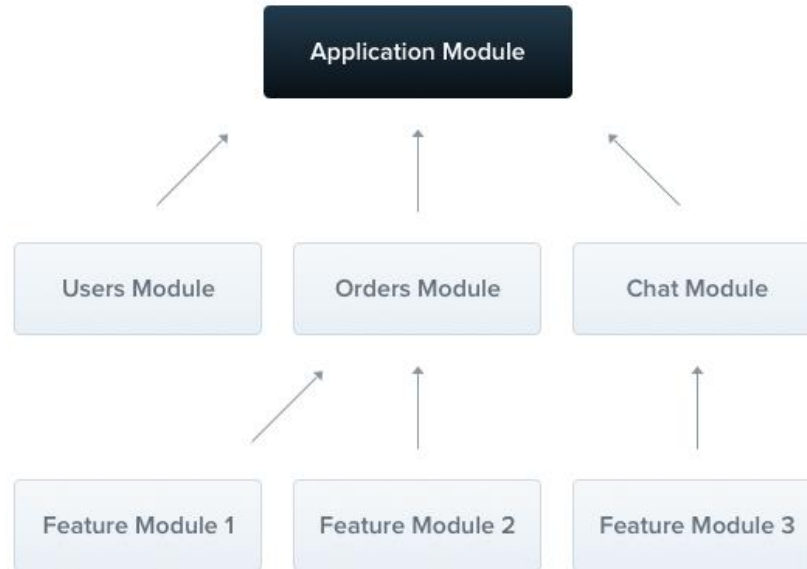


# 의존성 주입 (Dependency Injection, DI)

- 객체지향 5원칙(SOLID)에 해당함
- 추상성이 높고 안정적인 고수준의 클래스는 구체적이고 불안정한 저수준의 클래스에 의존해선 안된다는 원칙
- 이를 구현하기 위해서 제어의 역전(Inversion of Control, IoC)이라는 기술이 필요함
- NestJS에서는 IoC를 구현하고 있음

# Module

여러 컴포넌트를 조합하여 더 큰 작업을 수행할 수 있게 하는 단위



# Module

```
import { Module } from '@nestjs/common';
import { AppController } from './app.controller';
import { AppService } from './app.service';
import { CoreModule } from './core/core.module';

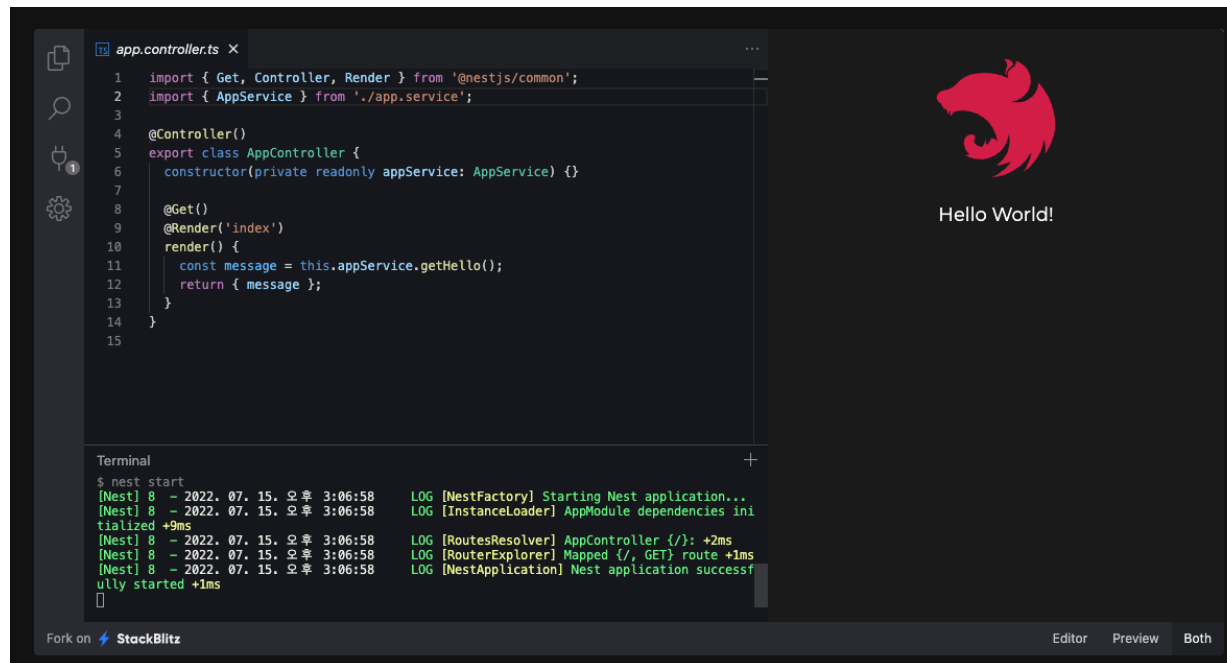
@Module({
  imports: [CoreModule],
  controllers: [AppController],
  providers: [AppService],
})
export class AppModule { }
```

# NestJS 주요 기능

- 미들웨어
- 파이프
- 가드
- 로깅
- 예외 처리
- ORM
- 테스트
- 인터셉터
- GraphQL
- 기타 등등...

# 데모

공식 홈페이지에서 Live Preview를 해볼 수 있습니다.



The screenshot displays a code editor with a file named `app.controller.ts`. The code defines a NestJS controller that uses the `@Controller()` decorator and the `@Render('index')` decorator. The `render()` method calls `this.appService.getHello()` and returns the result. The live preview on the right shows the NestJS logo and the text "Hello World!". The terminal at the bottom shows the output of the `nest start` command, indicating that the application has started successfully.

```
1 import { Get, Controller, Render } from '@nestjs/common';
2 import { AppService } from './app.service';
3
4 @Controller()
5 export class AppController {
6   constructor(private readonly appService: AppService) {}
7
8   @Get()
9   @Render('index')
10  render() {
11    const message = this.appService.getHello();
12    return { message };
13  }
14 }
15
```

Terminal

```
$ nest start
[Nest] 8 - 2022. 07. 15. 오후 3:06:58 LOG [NestFactory] Starting Nest application...
[Nest] 8 - 2022. 07. 15. 오후 3:06:58 LOG [InstanceLoader] AppModule dependencies initialized +9ms
[Nest] 8 - 2022. 07. 15. 오후 3:06:58 LOG [RoutesResolver] AppController {}: +2ms
[Nest] 8 - 2022. 07. 15. 오후 3:06:58 LOG [RouterExplorer] Mapped {/, GET} route +1ms
[Nest] 8 - 2022. 07. 15. 오후 3:06:58 LOG [NestApplication] Nest application successfully started +1ms
```

Fork on [StackBlitz](#) Editor Preview Both

# 참고자료

- NestJS 공식 문서


<https://docs.nestjs.com/>


- 도서) NestJS로 배우는 백엔드 프로그래밍(한용재, 제이펍)


<https://search.shopping.naver.com/book/catalog/36250090622>


# — 감사합니다.

## Q & A

 wb092311@naver.com

 dlstj0923 (Platinum 1)

 inclue (Specialist)

 inclue