

## 로그에 대해서

애플리케이션을 운영할 때 보면 에러가 날 때가 많이 있습니다. 그럴 때 어디 부분이 문제인지 빠르고 정확하게 파악하기 위해서는 어떠한 곳에서 에러가 어떻게 나고 있는지 보기 위해서 로그를 보는게 중요합니다. 그래서 이번 시간에는 로그에 대해서 알아보겠습니다.

### 로그의 종류

Log - 중요한 정보의 범용 로깅

Warning - 치명적이거나 파괴적이지 않은 처리되지 않은 문제

Error - 치명적이거나 파괴적인 처리되지 않은 문제

Debug - 오류 발생시 로직을 디버그하는 데 도움이되는 유용한 정보입니다. 개발자 용

Verbose - 응용 프로그램의 동작에 대한 통찰력을 제공하는 정보입니다. 운영자 용

### 로그 레벨

원하는 대로 환경에 따라서 로그의 레벨을 정의해서 넣어줄 수 있습니다.

	Log	Error	Warning	Debug	Verbose
Development	o	o	o	o	o
Staging	o	o	o	x	x
Production	o	o	x	x	x

## 실제로 애플리케이션에 로그 적용하기

로그에 대해서 알아보았으니 이제는 만들고 있는 애플리케이션에 로그를 넣어주겠습니다. 모든 로그를 넣는 것보다 몇가지 케이스를 넣어서 어떠한 식으로 넣는지 전체적인 감을 잡아보겠습니다.

### 원래 로그 넣는 것은...

하나하나 개발하면서 넣는게 맞지만 현재는 개발을 다하고 난 후에 로그를 넣고 있습니다. 그 이유는 한번에 많은 개념이 섞여서 배우면 힘들기에 다른 것들 먼저 배우고 Logging 부분을 배우기 위해서 입니다

### 로그를 처리 하기 위해서 사용하는 모듈은...

expressjs를 사용할 때는 Winston이란 모듈을 주로 사용합니다. 하지만 nestjs에는 이미 built-in된 logger 클래스가 있기에 그걸 사용하겠습니다.

main.ts

```
import { NestFactory } from '@nestjs/core';
import { Logger } from '@nestjs/common';
import { AppModule } from './app.module';

async function bootstrap() {
  const logger = new Logger();
  const app = await NestFactory.create(AppModule);

  const port = 3000;
  await app.listen(3000);
  logger.log(`Application running on port ${port}`)
}
bootstrap();
```

## boards.controller.ts

```
export class BoardsController {
  private logger = new Logger('BoardsController');

  constructor(private boardsService: BoardsService) { }

  @Get()
  getAllBoards(
    @GetUser() user: User,
  ): Promise<Board[]> {
    this.logger.verbose(`User "${user.username}" trying to get all boards`);
    return this.boardsService.getAllBoards(user);
  }
}
```

GET



localhost:3000/boards

```
[Nest] 35723 - 2021-02-27 11:26:00 PM [RouterExplorer] Mapped {/auth/authTest, POST} route +2ms
[Nest] 35723 - 2021-02-27 11:26:00 PM [NestApplication] Nest application successfully started +3ms
[Nest] 35723 - 2021-02-27 11:26:00 PM Application running on port 3000
[Nest] 35723 - 2021-02-27 11:26:58 PM [BoardsController] User "john213" trying to get all boards
```

```
@Post()
@UsePipes(ValidationPipe)
createBoard(
  @Body() createBoardDto: CreateBoardDto,
  @GetUser() user: User,
): Promise<Board> {
  this.logger.verbose(`User "${user.username}" creating a new board.
  Payload: ${JSON.stringify(createBoardDto)}`)
  return this.boardsService.createBoard(createBoardDto, user);
}
```

```
[Nest] 36269 - 2021-02-27 11:31:33 PM [NestApplication] Nest application successfully started +3ms
[Nest] 36269 - 2021-02-27 11:31:33 PM Application running on port 3000
[Nest] 36269 - 2021-02-27 11:31:55 PM [BoardsController] User "john213" creating a new board.
  Payload: {"title":"board title","description":"board description2222"}
```