



**SWK Magdeburg - März 2020**

**@bstrobach**

**<https://github.com/bstrobach/nestjs-swk-md>**

# Wer ist der Typ da?

**Björn Strobach**

**@bstrobach**

**Software-Entwickler @MrAppWare**



# Worüber redet der da eigentlich?

## Nest.js - Was?

**„ Nest is a framework for building efficient, scalable Node.js server-side applications, which uses progressive JavaScript, is built with and fully supports TypeScript. “**

*–Nest.js documentation*



# Okay, erzähl mir mehr...

- **Kamil Myśliwiec - @kammysliwiec**
- **Open Source Software - MIT Lizenz**
- **Node.js Framework**
- **Setzt auf Express.js (Fastify kompatibel)**

# Und warum sollte ich das jetzt nutzen?

- **Moderne Technologien**
- **Dependency Injection & Modularität („Angular für's Backend“)**
- **Kann für MVC Apps, REST- oder GraphQL APIs verwendet werden**
- **Durch CLI-Tools wird eine klare Projektstruktur und „Clean Code“ forciert**
- **„Enterprise ready“**

# Cool, erzähl mir was über die Grundkonzepte

- **Controller**
- **Providers**
- **Module**
- **Middleware**
- **Exception Filter**
- **Pipes**
- **Guards**
- **Interceptors**
- **Custom Decorator**

# Controller?

Verantwortlich für Requests und Responses



```
1 import { Controller, Get } from '@nestjs/common';  
2  
3 @Controller('cats')  
4 export class CatsController {  
5   @Get()  
6   findAll(): string {  
7     return 'This action returns all cats';  
8   }  
9 }
```

# Provider?

Stellt die Abhängigkeiten bereit und enthält i.d.R. Business Logik



```
1 import { Injectable } from '@nestjs/common';
2 import { Cat } from '../interfaces/cat.interface';
3
4 @Injectable()
5 export class CatsService {
6   private readonly cats: Cat[] = [];
7
8   create(cat: Cat) {
9     this.cats.push(cat);
10  }
11
12  findAll(): Cat[] {
13    return this.cats;
14  }
15 }
```



# Module?

Enthält Metadaten, die zum Organisieren der App Struktur verwendet werden



```
1 import { Module } from '@nestjs/common';
2 import { CatsController } from './cats.controller';
3 import { CatsService } from './cats.service';
4
5 @Module({
6   controllers: [CatsController],
7   providers: [CatsService],
8 })
9 export class CatsModule {}
```

# Und wie kann ich jetzt loslegen?

„Nur“ drei Schritte



```
$ npm i -g @nestjs/cli
```

```
$ nest new project-name
```

```
$ npm run start
```

# Demo

# ICH WILL MEHR !!!

- Authentifizierung (Passport.js)
- Config (dotenv)
- Caching
- CORS
- GraphQL (Apollo)
- Websockets (socket.io)
- Microservices
- ORM (TypeORM, Sequelize, Mongoose)
- CQRS
- OpenAPI
- Prisma
- Health checks (Terminus)
- ...

Q & A



**Vielen Dank.**

# Nützliche Links

- <https://nestjs.com/>
- <https://docs.nestjs.com>
- <https://github.com/nestjs/nest>