TypeORM and CRUD opreations

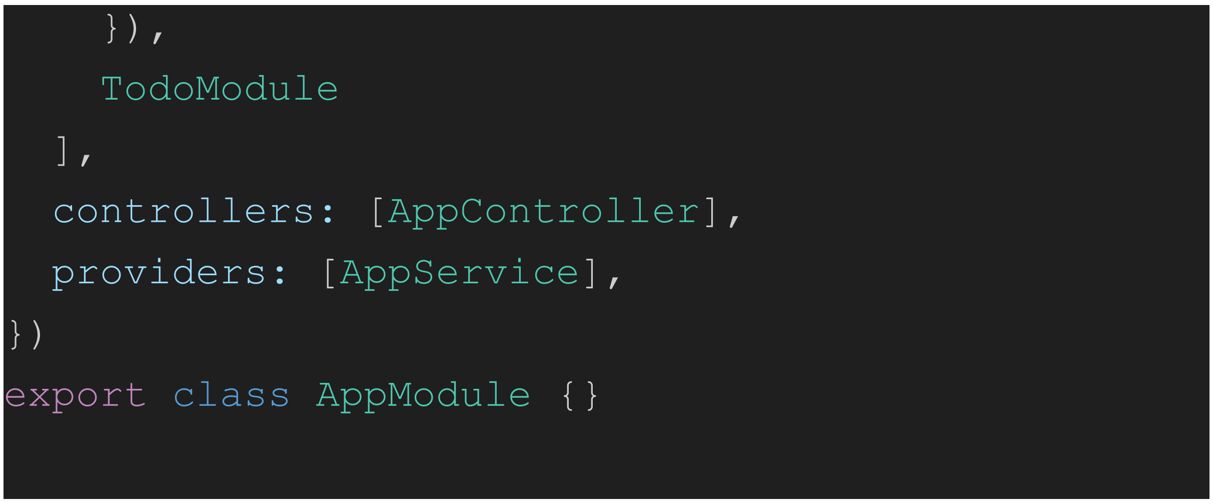
```
import { todo } from "node:test";
import { Column, Entity, PrimaryGeneratedColumn }
from "typeorm";

@Entity({name:'todo'})
export class TodoEntity {

    @PrimaryGeneratedColumn()
    id:number

    @Column()
    title:string
}
```

```
import { Injectable, NotFoundException } from
'@nestjs/common';
import { InjectRepository } from
'@nestjs/typeorm';
import { TodoEntity } from
'src/todo/todo/todo/todo.entity';
import { Repository } from 'typeorm';

@Injectable()
export class TodoService {

    constructor(
        @InjectRepository(TodoEntity) private
readonly todoRepository:Repository<TodoEntity>,
    ){}

    async create(dto: {title:string}){
        const todo
=this.todoRepository.create(dto)

        return await
this.todoRepository.save(todo)
    }

    async findOne(id: number): Promise<TodoEntity
| null> {

        return this.todoRepository.findOneBy({ id
});
      }

      async updateFull(
        id: number,
        dto: { title: string },
      ): Promise<TodoEntity> {
        const todo = await
this.todoRepository.preload({ id, ...dto });
        if (!todo) {
          throw new NotFoundException(`Todo #${id}
not found`);
        }
        return this.todoRepository.save(todo);
      }


      async updatePartial(
        id: number,
        dto: Partial<{ title: string }>,
      ): Promise<TodoEntity> {
        const result = await
this.todoRepository.update(id, dto);
        if (result.affected === 0) {
          throw new NotFoundException(`Todo #${id}
not found`);
        }

        const todo = await this.findOne(id);
        if (!todo) {
        throw new NotFoundException(`Todo #${id}
not found after update`);
        }
        return todo;
      }
}
```

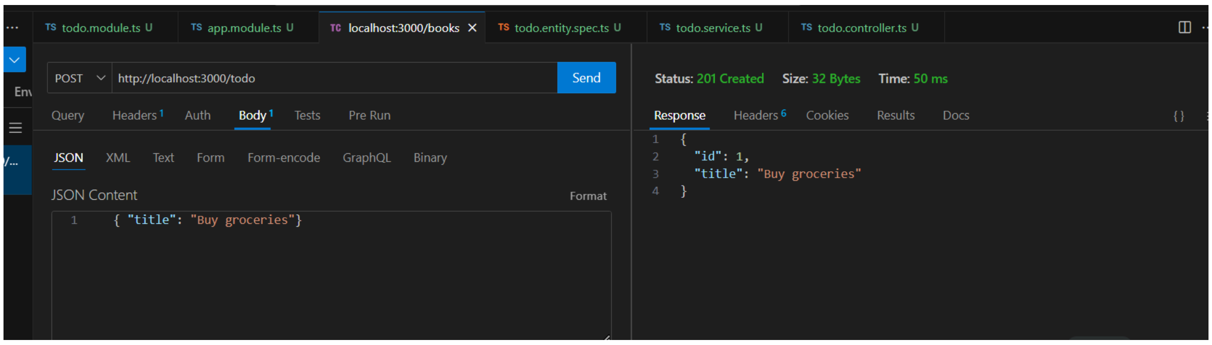
```
import { Body, Controller, Get, Param,
ParseIntPipe, Patch, Post, Put } from
'@nestjs/common';
import { TodoService } from './todo.service';
import { TodoEntity } from
'src/todo/todo/todo/todo.entity';
```

```
@Controller('todo')
export class TodoController {

    constructor(private readonly
todoService:TodoService){}

    @Post()
    create(@Body() dto){
        return this.todoService.create(dto)
    }

    @Get(':id')
    findOne(
      @Param('id', ParseIntPipe) id: number,
    ): Promise<TodoEntity | null> {
      return this.todoService.findOne(id);
    }


    @Put(':id')
    updateFull(
      @Param('id', ParseIntPipe) id: number,
      @Body()dto: {title:string},
    ): Promise<TodoEntity> {
      return this.todoService.updateFull(id, dto);
    }


    @Patch(':id')
    updatePartial(
      @Param('id', ParseIntPipe) id: number,
      @Body() dto:{title:string},
    ): Promise<TodoEntity> {
      return this.todoService.updatePartial(id,
dto);
    }
}
```

```
import { Module } from '@nestjs/common';
import { AppController } from './app.controller';
import { AppService } from './app.service';
import { TypeOrmModule } from '@nestjs/typeorm';

import { TodoModule } from
'./todo/todo/todo/todo.module';


@Module({
  imports: [
    TypeOrmModule.forRoot({
      type: 'mysql',
      host: 'localhost',
      port: 3306,
      username: 'root',
      password: 'root123',
      database: 'test',
      autoLoadEntities: true,
      synchronize: true,
```



```
    }),
    TodoModule
  ],
  controllers: [AppController],
  providers: [AppService],
})
export class AppModule {}
```



POST http://localhost:3000/todo

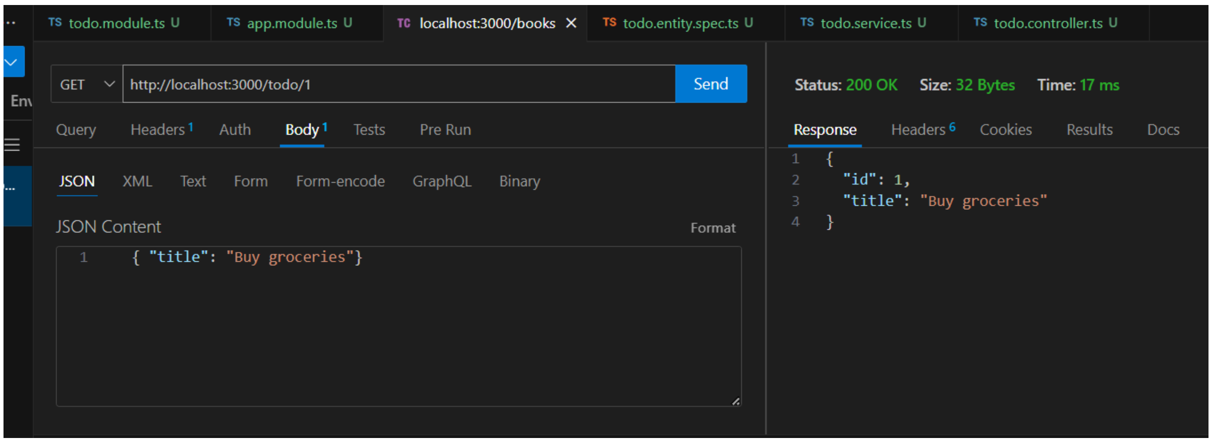
JSON Content

```
{ "title": "Buy groceries"}
```

Status: 201 Created Size: 32 Bytes Time: 50 ms

```
{
  "id": 1,
  "title": "Buy groceries"
}
```

GET



GET http://localhost:3000/todo/1

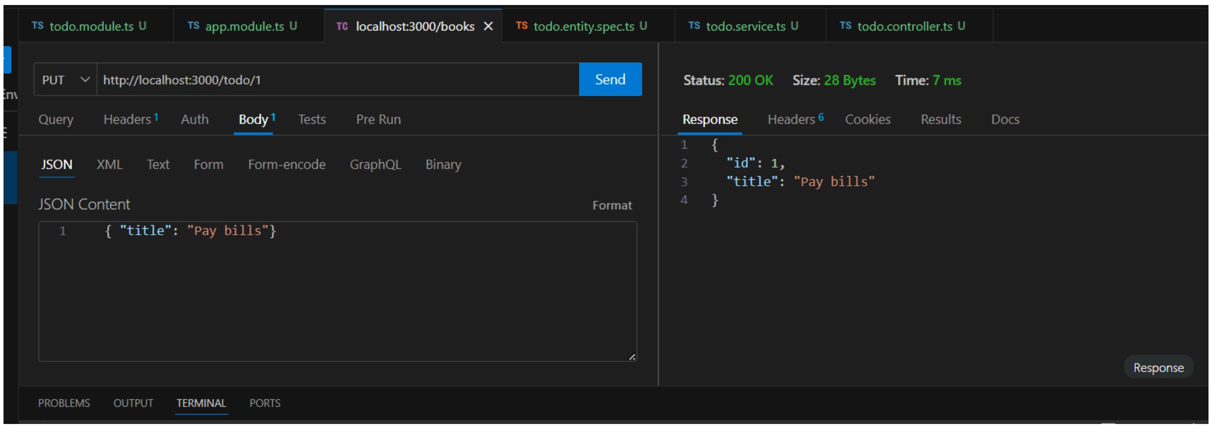
JSON Content

```
{ "title": "Buy groceries"}
```

Status: 200 OK Size: 32 Bytes Time: 17 ms

```
{
  "id": 1,
  "title": "Buy groceries"
}
```

PUT



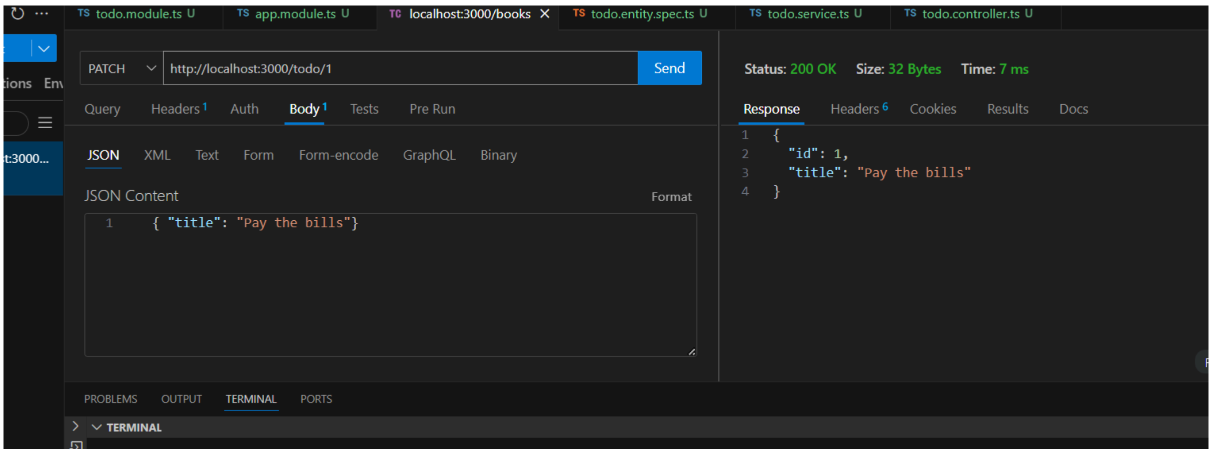
PUT http://localhost:3000/todo/1

```
{ "title": "Pay bills"}
```

Status: 200 OK Size: 28 Bytes Time: 7 ms

```
{
  "id": 1,
  "title": "Pay bills"
}
```

Patch

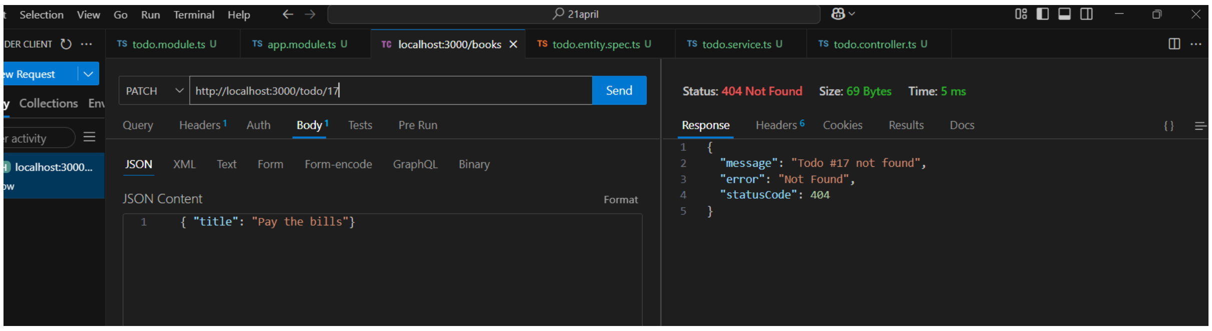


PATCH http://localhost:3000/todo/1

```
{ "title": "Pay the bills"}
```

Status: 200 OK Size: 32 Bytes Time: 7 ms

```
{
  "id": 1,
  "title": "Pay the bills"
}
```



PATCH http://localhost:3000/todo/17

```
{ "title": "Pay the bills"}
```

Status: 404 Not Found Size: 69 Bytes Time: 5 ms

```
{
  "message": "Todo #17 not found",
  "error": "Not Found",
  "statusCode": 404
}
```