

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Бек-энд разработка

Отчет

Лабораторная работа 6

Выполнил:

Прокопец Семен

К3439

Проверил:

Добряков Д. И.

Санкт-Петербург

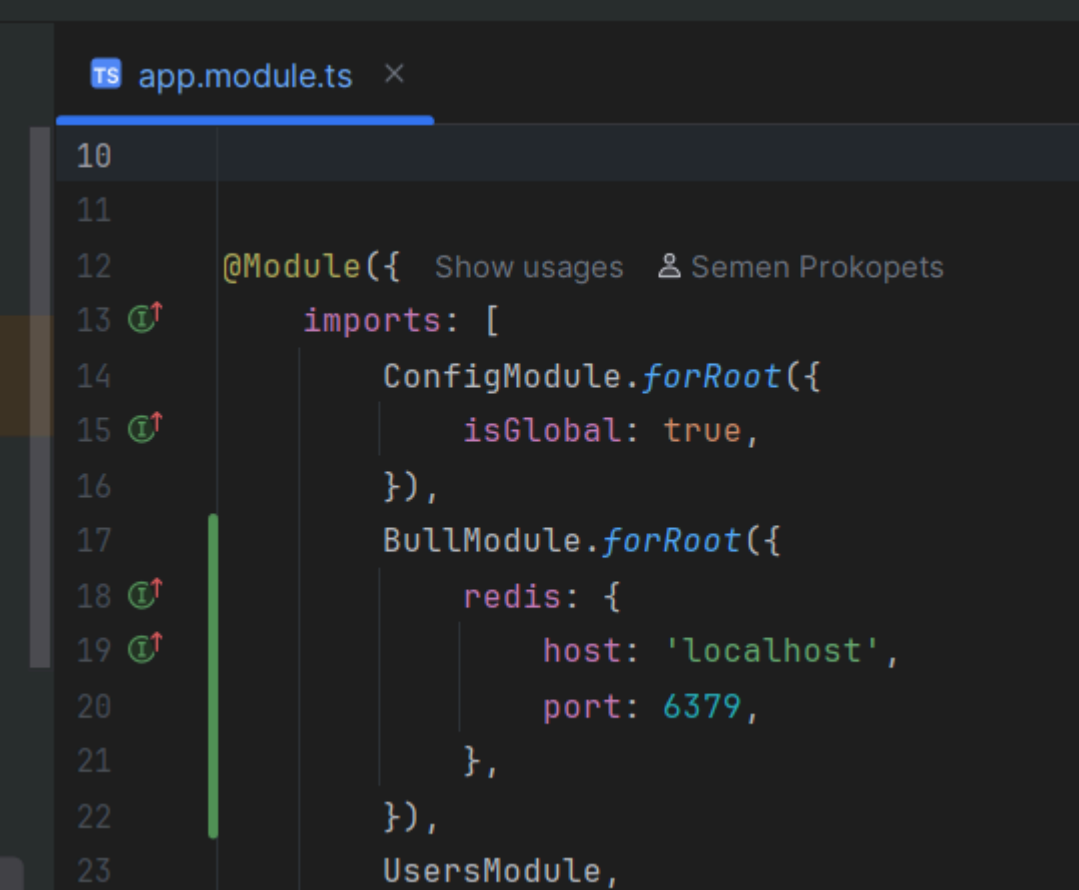
2025 г.

Задача

Добавить очередь

Ход работы

подключаем Bull в [app.module.ts](#)



```
10
11
12 @Module({ Show usages  Semen Prokopets
13   imports: [
14     ConfigModule.forRoot({
15       isGlobal: true,
16     }),
17     BullModule.forRoot({
18       redis: {
19         host: 'localhost',
20         port: 6379,
21       },
22     }),
23     UsersModule,
```

создаем

очередь

```
TS app.module.ts TS process.ts x
1 import { Process, Processor } from '@nestjs/bull';
2 import bull from 'bull';
3
4 @Processor({ queueName: 'quiz' }) no usages new *
5 export class QuizProcessor {
6   @Process({ name: 'processQuiz' }) no usages new *
7   async handleProcessQuiz(job: bull.Job<{ quizId: string }>) : Promise<{ status: string; quizId: string ... } {
8     console.log( message: 'Обработка квиза:', job.data.quizId);
9
10    await new Promise(resolve : (value: unknown) => void => setTimeout(resolve, delay: 3000));
11
12    return { status: 'completed', quizId: job.data.quizId };
13  }
14 }
```

регистраруем очередь в модуле

```
TS app.module.ts TS process.ts TS quiz.module.ts x
1 > import ...
7
8 @Module({ Show usages 👤 Semen Prokopets
9   imports: [
10     BullModule.registerQueue({
11       name: 'quiz',
12     }),
13   ],
```

добавляем логику в сервис

```
app.module.ts process.ts quiz.module.ts quiz.service.ts x
1 import { Injectable } from '@nestjs/common';
2 import { InjectQueue } from '@nestjs/bull';
3 import bull from 'bull';
4
5 @Injectable() no usages new *
6 export class QuizService {
7   constructor(@InjectQueue( name: 'quiz') private quizQueue: bull.Queue) {} no usages new *
8
9   async addQuizToQueue(quizId: string) : Promise<void> { no usages new *
10     await this.quizQueue.add( name: 'processQuiz', { quizId }, {
11       attempts: 3,
12       backoff: 1000,
13     });
14   }
15 }
```

и в контроллер

```
@Post('/:id/process')
async processQuiz(@Param('id') id: string) {
  await this.quizService.addQuizToQueue(id);
  return { message: 'Задача добавлена в очередь' };
}
```

Вывод: Научились создавать и работать с очередью Bull