

SI При запуске кода приведенного ниже появляется ошибка "expected User {

```
    id: 113,
    username: 'test-user-create',
    password: 'test-password-create'
} response body, got {
    id: 113,
    username: 'test-user-create',
    password: 'test-password-create'
}"
в строке " return await
request(app.getHttpServer()).post('/users').send(userCreate).expect(201).expect(userService.create(userCreateReque
```

Как это можно исправить?

Код:

```
import { Test, TestingModule } from '@nestjs/testing';
import { INestApplication } from '@nestjs/common';
import * as request from 'supertest';
import { AppModule } from './src/app.module';
import { UserController } from './src/user/user.controller';
import { UserService } from './src/user/user.service';
import { User } from './src/entities/user.entity';
import ormconfig from './orm.config';
import { TypeOrmModule } from '@nestjs/typeorm';
import { UserModule } from './src/user/user.module';
import { UserRepository } from './src/repository/user.repository';

// С Использованием реальных данных в бд и подключением реальных данных
describe('UserController', () => {
  let app: INestApplication;
  // В бд предварительно должен быть такой юзер:
  const user = { id: 89, username: 'test-user', password: 'test-password' };
  //
  const userCreateRequest = new User();
  userCreateRequest.id = 113;
  userCreateRequest.username = 'test-user-create';
  userCreateRequest.password = 'test-password-create';
  // const userCreateRequest = { id: 105, username: 'test-user-create', password: 'test-password-create' };
  // Можно так вариант 1
  // const userService = {
  //   findAll() {
  //     return [ { id: 89, username: 'test-user', password: 'test-password' }];
  //   },
  // };

  // Можно так вариант 2
  const userService = {
    findAll: () => [user],
    // findByd: (id) => { id: 89, username: 'test-user', password: 'test-password' };
    findByd(id: number) {
      return user;
    },
    create(user: User) {
      return user;
    },
    update(id: number, user: User) {},
    delete(id: number) {},
  };

  beforeAll(async () => {
    const moduleRef = await Test.createTestingModule({
      imports: [UserModule, TypeOrmModule.forRoot(ormconfig), TypeOrmModule.forFeature([User, UserRepository])],
      // providers: [UserService],
    })
      .overrideProvider(userService)
      .useValue(userService)
      .compile();

    app = moduleRef.createNestApplication();
    await app.init();
  });

  it('/GET users', async () => {
    const usersRequest = await request(app.getHttpServer()).get('/users').expect(200).expect(userService.findAll());
    // следующий вариант из офи источника не работает
    // https://docs.nestjs.com/fundamentals/testing
    // const usersRequest = request(app.getHttpServer()).get('/users').expect(200).expect({
    //   data: userService.findAll(),
    // });

    console.log('userService.findAll()= ', userService.findAll());
    return await usersRequest;
  });

  it('/GET user by id', async () => {
    if (userService) {
      return await request(app.getHttpServer()).get('/users/89').expect(200).expect(userService.findByd(89));
    }
  });

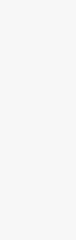
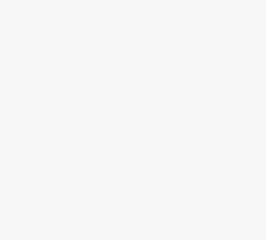
  it('/POST user', async () => {
    // const userUpdate = { password: 'test-password-update' };
    // const userCreate = new User();
    // // user.id = 89;
    // // user.username = 'test-user-create';
    // // user.password = 'test-password-create';
    const userCreate = { id: 1, username: 'test-user-create', password: 'test-password-create' };

    // const userCreateRequest = new User();
    // user.id = 104;
    // user.username = 'test-user-create';
    // user.password = 'test-password-create';

    if (userService) {
      // try {
      //   return await
      request(app.getHttpServer()).post('/users').send(userCreate).expect(201).expect(userService.create(userCreateRequest));
      // } catch (e) {
      //   // console.log('/POST user error= ', e);
      //   // return e;
      // }
    }
  });

  afterAll(async () => {
    await app.close();
  });
});
```

Ошибка "expected User response body, got User" говорит о том, что в результате запроса был ожидаемый тип данных, но полученный тип данных отличается от ожидаемого.



Чтобы исправить эту ошибку, нужно убедиться, что ожидаемый тип данных совпадает с тем, что был получен в результате запроса. В данном случае это ожидается тип User, поэтому нужно убедиться, что userService.create(userCreateRequest) возвращает экземпляр User.

Если userService.create(userCreateRequest) возвращает экземпляр User, то можно попробовать изменить строку с ошибкой таким образом:

```
return await request(app.getHttpServer())
  .post('/users')
  .send(userCreate)
  .expect(201)
  .expect(userService.create(userCreateRequest));
```

Это должно решить проблему с ошибкой "expected User response body, got User". Однако, нужно обратить внимание, что в коде уже используется тестовый модуль NestJS, который позволяет заменить зависимости компонента на моки (загл