

Todo list note接口文档以及单元测试

· 表结构设计

SQL

```
1  -- auto-generated definition
2  create table Notes
3  (
4      id            int auto_increment
5          primary key,
6      msg           varchar(255) not null,
7      finish_status int default 0 not null,
8      time          varchar(255) not null
9  );
```

新建提醒事项

接口url:

[http://127.0.0.1:7001/POST/notes?msg="明天早上6点送孩子上学"&finish_status=0](http://127.0.0.1:7001/POST/notes?msg=)

传入参数:

Perl

```
1  { msg="今天6点要去看牙医", finish_status="0"}
```

- finish_status 是0，没有完成，1:代表完成状态

返回数据:

JSON

```
1  {"success":true,"message":"OK","id":6,"code":200}
```

我的列表（展示所有，包括已完成和没完成，按时间排序，最新在前面）

接口url:

<http://127.0.0.1:7001/GET/notes>

传入参数:

Perl

```
1 { }
```

返回数据:

Ada

```
1 { "success":true,"message":"OK","data":[{"id":6,"msg":"完成监测预警项目优化","finish_status":0,"time":"1640496728562"}, {"id":5,"msg":"今天6点要去看牙医","finish_status":0,"time":"1640496707078"}, {"id":4,"msg":"今天6点要去看牙医","finish_status":0,"time":"1640496704990"}, {"id":3,"msg":"今天6点要去看牙医","finish_status":0,"time":"1640496612224"}, {"id":2,"msg":"kkkk","finish_status":1,"time":"1640489417758"}],"code":200}
```

展示我的todo list （没完成/或者没完成，按时间排序）

接口url:

<http://127.0.0.1:7001/GET/notes/finished?finish=1>

- finish_status 是0，没有完成，1:代表完成状态

传入参数:

Plain Text

```
1 {
2     finish = 1//      显示完成的todo。 如果finish=0 ，那么显示没有完成的
3 }
```

返回参数：

Ada

```
1 {"success":true,"message":"OK","data":[{"id":2,"msg":"kkkk","finish_status":1,"time":"1640489417758"}],"code":200}
```

更新修改todo list（更新标记完成状态，更新文本内容）

接口url：

[http://172.20.10.9:7001/PUT/notes?id=3&msg=明天早上7点出门看牙医"&finish=0](http://172.20.10.9:7001/PUT/notes?id=3&msg=明天早上7点出门看牙医)

<http://172.20.10.9:7001/PUT/notes?id=6&finish=1>

传入参数：

Plain Text

```
1 {
2     id = 3// note的id,
3     msg="明天早上7点出门看牙医",
4     finish = 0
5
6 }
```

返回参数(成功)：

JSON

```
1 {"success":true,"message":"OK","id":3,"code":200}
```

返回参数(失败)：

Ada

```
1 {"data":{"success":false,"message":"erro","id":"67","code":400}}
```

删除todo

接口url：

<http://127.0.0.1:7001/DELETE/notes?id=2>

传入参数：

Plain Text

```
1  {
2      id = 3
3  }
```

返回参数：

JSON

```
1  {"data":{"success":true,"message":"OK","id":"7","code":200}}
```

失败：

JSON

```
1  {"data":{"success":false,"message":"erro","id":"23","code":400}}
```

测试

controller层的单元测试

分别对创建，删除，修改，展示的内容做了单元测试，其中展示（包括已完成和未完成状态的区分）

所以测试的案例有5个，用断言的方式对，返回码以及返回的消息进行判断。以下是测试的报告

Apache

```
1  Test Suites: 1 passed, 1 total
2  Tests:      5 passed, 5 total
3  Snapshots:  0 total
4  Time:       2.712 s, estimated 3 s
5  Ran all test suites matching /\test\/.*\.test\.ts$/test/i.
```