

# 孙广明 - Coding Test

---

Hello davy and tony,

I've prepared two demos, they have the same function.

The difference of them is that filter different users / assignees by name or email.

Because there are the same names with different emails records in the CSV files, so I can't distinguish them by name.

But the email address can be determined.

Demo1 filters user by name. ([Demo1](#))

Demo2 filters user by email. ([Demo2](#))

Please note that, two demo data storages are independent of each other.

Open the page as the default administrator.

When the administrator login, you can filter other people's tasks and drag and drop them.

However, you can not sort when filtering users.

You can switch the current user by clicking on the user name in the upper left corner.

Not the administrator login, can only see their own tasks, and cannot drag and drop sorting.

I modified part of the CSV file data, send it to you in the ZIP file.

The project supports mobile responsive.

If you encounter problems using it, please feel free to contact me.

Looking forward to receiving your valuable suggestions and feedback.

Best Regards,

Ming

## Backend API Documentation

---

open dir `demo1\server` or `demo2\server`

```
1 npm install
2
3 npm run start
```

BASE\_URL: `http://127.0.0.1:3000` or `http://127.0.0.1:3001`

## getUserList

---

Method	Url
GET	<a href="http://127.0.0.1:3000/api/userList">http://127.0.0.1:3000/api/userList</a>

#### Response:

```

1  {
2      "code":0,
3      "data": [
4          {
5              "Email": "Admin@waterloo.com",
6              "Name": "Admin",
7              "Title": "Admin",
8              "userId": "S000",
9              "_id": "62c0f3c9105d8014ce74e7e7"
10         }
11         ...
12     ],
13     "msg": "success"
14 }
```

## setTaskSort

Method	Url
POST	<a href="http://127.0.0.1:3000/api/setTaskSort">http://127.0.0.1:3000/api/setTaskSort</a>

#### Params:

Params	Type	Required	Remarks
_id	number	True	Task ID
Priority	number	True	New Priority

#### Example:

```

1  [
2      {
3          "_id": "62c0f3c9105d8014ce74e7e7",
4          "Priority":1
5      },
6      {
7          "_id": "62c0f3c9105d8014ce74e737",
8          "Priority":2
9      }
10 ]
```

## Response:

```
1 | { "code": 0, "msg": "success" }
```

## getTaskList

Method	Url
POST	<a href="http://127.0.0.1:3000/api/taskList">http://127.0.0.1:3000/api/taskList</a>

## Params:

Params	Type	Required	Remarks
Email	String	false	Name
Name	String	false	Email
type	String	false	filter type

## Example:

```
1 | {  
2 |   "Email": "",  
3 |   "Name": "",  
4 |   "type": "Email"  
5 | }
```

## Response:

```
1 | [  
2 |   "code": 0,  
3 |   "data": [  
4 |     {  
5 |       "By_who": "Perry",  
6 |       "Description": "Frontend creation<br/>expected to due on  
Thursday",  
7 |       "Email": "PerryP@waterloo.com",  
8 |       "Priority": 1,  
9 |       "Stage": 1,  
10 |      "Task": "S1_3",  
11 |      "_id": "62c0f3b8105d8014ce74e7d7",  
12 |    }  
13 |    ...  
14 |  ]  
15 |  ...,  
16 |  "msg": "success"
```

