

```

1 require('babel-register')
2 const express = require('express')
3 const morgan = require('morgan') //middleware permettant le debug des requete http
4 const app = express(); //création d'une instance de express.
5
6 // Array qui contient nos membres
7 const members = [
8   {
9     id: 1,
10    name: 'John'
11  },
12  {
13    id: 2,
14    name: 'Julie'
15  },
16  {
17    id: 3,
18    name: 'Mathieu'
19  }
20 ]
21
22 //middleware permettant de debug les requete http pendant la phase de développement&
23 app.use(morgan('dev'))
24
25 //requête qui nous permet de récupérer les membre en fonction de leur id
26 app.get('/api/v1/members/:id', (req, res)=>{
27   res.json(success(members[(req.params.id)-1])) //par convention on met res.json pour envoyer un fichier json
28 })
29
30 // requête permettant de récupérer tous les membres
31 // app.get('/api/v1/members', (req, res)=>{
32 //   res.json(members)
33 // })
34
35 //requête avec req.query qui nous permet de récupérer au choix le max le membres que l'on veut.
36 app.get('/api/v1/members', (req, res)=>{
37   console.log(req)
38   if(req.query.max != undefined && req.query.max > 0){
39     res.json(success(members.slice(0, req.query.max)))
40   } else if(req.query.max != undefined){ // gestion de l'erreur
41     res.json(error('Wrong max value'))
42   } else{
43     res.json(success(members))
44   }
45 })
46
47 //on donne un port a notre application avec une fonction de call back pour confirmer la conection
48 app.listen(8080, () => console.log('Started on port 8080'))
49
50 //la convention veut qu'on renvoie toujours dans notre json
51 //une variable qui est "status" qui a pour valeur soit success soit error.
52 //success renvoi le la variable success et result qui est se que l'on veut envoyer
53 function success(result){
54   return {
55     status: 'success',
56     result: result
57   }
58 }
59 //error renvoi le message d'erreur
60 function error(message){
61   return {
62     status: 'error',
63     message: message
64   }
65 }

```

GET

http://localhost:8080/api/v1/members?max=2

Authorization

Headers

Body

Pre-request Script

Tests

TYPE

Inherit auth from parent

This request is not inhe

The authorization header will be automatically generated when you send the request. [Learn more about authorization](#)

Body

Cookies

Headers (6)

Test Results

Pretty

Raw

Preview

JSON

```

1 {
2   "status": "success",
3   "result": [
4     {
5       "id": 1,
6       "name": "John"
7     },
8     {
9       "id": 2,
10      "name": "Julie"
11     }
12   ]
13 }

```

GET

http://localhost:8080/api/v1/members?max=2

Authorization

Headers

Body

Pre-request Script

Tests

TYPE

Inherit auth from parent

This request

The authorization header will be automatically generated when you send the request. [Learn more about authorization](#)

Body

Cookies

Headers (6)

Test Results

Connection → keep-alive

Content-Length → 78

Content-Type → application/json; charset=utf-8

Date → Sun, 05 Aug 2018 11:12:10 GMT

Etag → W/"4e-j8BchHYXVAfoNXNn7CUUQMqOfoQ"

X-Powered-By → Express

http://localhost:3000/

http://localhost:3000/

http://localhost:3000/

GET

http://localhost:8080/api/v1/members?max=

Authorization

Headers

Body

Pre-request Script

Tests

TYPE

Inherit auth from parent

This request is not in

The authorization header will be automatically generated when you send the request. [Learn more about authorization](#)

Body

Cookies

Headers (6)

Test Results

Pretty

Raw

Preview

JSON

```

1 {
2   "status": "error",
3   "message": "Wrong max value"
4 }

```