

1 第一次实验接口文档

- 1.1 五种方法结果展示接口
 - 1.1.1 请求URL
 - 1.1.2 请求方式
 - 1.1.3 请求示例
 - 1.1.4 请求参数说明
 - 1.1.5 返回示例
 - 1.1.6 返回参数说明
 - 1.1.7 备注
- 1.2 迭代法求系统环境最大斐波那契值接口
 - 1.2.1 请求URL
 - 1.2.2 请求方式
 - 1.2.3 请求示例
 - 1.2.4 请求参数说明
 - 1.2.5 返回示例
 - 1.2.6 返回参数说明
 - 1.2.7 备注
- 1.3 递归法求系统环境最大斐波那契值接口
 - 1.3.1 请求URL
 - 1.3.2 请求方式
 - 1.3.3 请求示例
 - 1.3.4 请求参数说明
 - 1.3.5 返回示例
 - 1.3.6 返回参数说明
 - 1.3.7 备注
- 1.4 1分钟递归求系统最大支持值接口
 - 1.4.1 请求URL
 - 1.4.2 请求方式
 - 1.4.3 请求示例
 - 1.4.4 请求参数说明
 - 1.4.5 返回示例
 - 1.4.6 返回参数说明
 - 1.4.7 备注
- 1.5 递归法和迭代法30秒斐波那契值接口
 - 1.5.1 请求URL
 - 1.5.2 请求方式
 - 1.5.3 请求示例
 - 1.5.4 请求参数说明
 - 1.5.5 返回示例
 - 1.5.6 返回参数说明
 - 1.5.7 备注
- 1.6 近似公式结果展示接口
 - 1.6.1 请求URL

- 1.6.2 请求方式
- 1.6.3 请求示例
- 1.6.4 请求参数说明
- 1.6.5 返回示例
- 1.6.6 返回参数说明
- 1.6.7 备注

2 第二次实验接口文档

2.1 生成随机数组接口

- 2.1.1 请求URL
- 2.1.2 请求方式
- 2.1.3 请求示例
- 2.1.4 请求参数说明
- 2.1.5 返回示例
- 2.1.6 返回参数说明
- 2.1.7 备注

2.2 生成无序数组接口

- 2.2.1 请求URL
- 2.2.2 请求方式
- 2.2.3 请求示例
- 2.2.4 请求参数说明
- 2.2.5 返回示例
- 2.2.6 返回参数说明
- 2.2.7 备注

2.3 先升后降or先降后升数组创建接口

- 2.3.1 请求URL
- 2.3.2 请求方式
- 2.3.3 请求示例
- 2.3.4 请求参数说明
- 2.3.5 返回示例
- 2.3.6 返回参数说明
- 2.3.7 备注

2.4 判断数组状态接口

- 2.4.1 请求URL
- 2.4.2 请求方式
- 2.4.3 请求示例
- 2.4.4 请求参数说明
- 2.4.5 返回示例
- 2.4.6 返回参数说明
- 2.4.7 备注

2.5 数组检索元素接口

- 2.5.1 请求URL
- 2.5.2 请求方式
- 2.5.3 请求示例
- 2.5.4 请求参数说明

- 2.5.5 返回示例
- 2.5.6 返回参数说明
- 2.5.7 备注
- 2.6 四种方法检索元素接口
 - 2.6.1 请求URL
 - 2.6.2 请求方式
 - 2.6.3 请求示例
 - 2.6.4 请求参数说明
 - 2.6.5 返回示例
 - 2.6.6 返回参数说明
 - 2.6.7 备注
- 2.7 对先升后降or先降后升数组找最大最小值接口
 - 2.7.1 请求URL
 - 2.7.2 请求方式
 - 2.7.3 请求示例
 - 2.7.4 请求参数说明
 - 2.7.5 返回示例
 - 2.7.6 返回参数说明
 - 2.7.7 备注
- 2.8 数组第k个最小元素接口
 - 2.8.1 请求URL
 - 2.8.2 请求方式
 - 2.8.3 请求示例
 - 2.8.4 请求参数说明
 - 2.8.5 返回示例
 - 2.8.6 返回参数说明
 - 2.8.7 备注

3 第三次实验接口文档

- 3.1 贪心法求解分数背包问题接口
 - 3.1.1 请求URL
 - 3.1.2 请求方式
 - 3.1.3 请求示例
 - 3.1.4 请求参数说明
 - 3.1.5 返回示例
 - 3.1.6 返回参数说明
 - 3.1.7 备注
- 3.2 贪心法求0-1背包问题接口
 - 3.2.1 请求URL
 - 3.2.2 请求方式
 - 3.2.3 请求示例
 - 3.2.4 请求参数说明
 - 3.2.5 返回示例
 - 3.2.6 返回参数说明
 - 3.2.7 备注

- 3.3 蛮力法求解01背包问题接口
 - 3.3.1 请求URL
 - 3.3.2 请求方式
 - 3.3.3 请求示例
 - 3.3.4 请求参数说明
 - 3.3.5 返回示例
 - 3.3.6 返回参数说明
 - 3.3.7 备注
- 3.4 动态规划求解0-1背包问题接口
 - 3.4.1 请求URL
 - 3.4.2 请求方式
 - 3.4.3 请求示例
 - 3.4.4 请求参数说明
 - 3.4.5 返回示例
 - 3.4.6 返回参数说明
 - 3.4.7 备注
- 3.5 记忆功能改进动态规划接口
 - 3.5.1 请求URL
 - 3.5.2 请求方式
 - 3.5.3 请求示例
 - 3.5.4 请求参数说明
 - 3.5.5 返回示例
 - 3.5.6 返回参数说明
 - 3.5.7 备注

1 第一次实验接口文档

1.1 五种方法结果展示接口

1.1.1 请求URL

- `http://localhost:5000/api/fiveMethodsShow`

1.1.2 请求方式

- POST

1.1.3 请求示例

```
1 http://localhost:5000/api/fiveMethodsShow
2 {
3   "n":0,
4 }
```

1.1.4 请求参数说明

参数名	是否必选	类型	说明
n	是	intger	第n个斐波那契数列

1.1.5 返回示例

```
1  {
2    "status_code": 0,
3    "status_msg": "xxx",
4    "data": {
5      nameArr:[
6        {
7          "name":0,
8          "resVal":1.0,
9          "optNum":0,
10         "time":10.0,
11        },
12        {
13          "name":0,
14          "resVal":1.0,
15          "optNum":0,
16          "time":10.0,
17        },
18        {
19          "name":0,
20          "resVal":1.0,
21          "optNum":0,
22          "time":10.0,
23        },
24        {
25          "name":0,
26          "resVal":1.0,
27          "optNum":0,
28          "time":10.0,
29        },
30        {
31          "name":0,
32          "resVal":1.0,
33          "optNum":0,
34          "time":10.0,
35        }
36      ]
37    }
38  }
```

1.1.6 返回参数说明

参数名	类型	说明
status_code	integer	状态码
status_msg	string	状态描述
nameArr	object arr	五种方法返回结果的对象数组
resVal	float	斐波那契数列返回结果
optNum	intger	操作次数
time	float	操作的时间
name	intger	操作码（另有约定文档）

1.1.7 备注

无

1.2 迭代法求系统环境最大斐波那契值接口

1.2.1 请求URL

- `http://localhost:5000/api/iterativeEnvMax`

1.2.2 请求方式

- GET

1.2.3 请求示例

```
1 http://localhost:5000/api/iterativeEnvMax
```

1.2.4 请求参数说明

参数名	是否必选	类型	说明
无	无	无	无

1.2.5 返回示例

```
1 {
2   "status_code": 0,
3   "status_msg": "xxx",
4   "data": {
5     "index":10,
6     "time":12.541546,
7   }
8 }
```

1.2.6 返回参数说明

参数名	类型	说明
status_code	integer	状态码
status_msg	string	状态描述
index	intger	系统环境最大斐波那契数的位置
time	float	程序运行时间

1.2.7 备注

无

1.3 递归法求系统环境最大斐波那契值接口

1.3.1 请求URL

- `http://localhost:5000/api/recursiveEnvMax`

1.3.2 请求方式

- GET

1.3.3 请求示例

```
1 http://localhost:5000/api/recursiveEnvMax
```

1.3.4 请求参数说明

参数名	是否必选	类型	说明
无	无	无	无

1.3.5 返回示例

```
1 {
2   "status_code": 0,
3   "status_msg": "xxx",
4   "data": {
5     "index":10,
6     "time":12.541546,
7   }
8 }
```

1.3.6 返回参数说明

参数名	类型	说明
status_code	integer	状态码
status_msg	string	状态描述
index	intger	系统环境最大斐波那契数的位置
time	float	程序运行时间

1.3.7 备注

无

1.4 1分钟递归求系统最大支持值接口

1.4.1 请求URL

- http://localhost:5000/api/oneMinuteRecursive

1.4.2 请求方式

- POST

1.4.3 请求示例

```
1 http://localhost:5000/api/oneMinuteRecursive
2 {
3   "n":0,
4 }
```

1.4.4 请求参数说明

参数名	是否必选	类型	说明
n	是	integer	第n个斐波那契数列

1.4.5 返回示例

```
1  {
2    "status_code": 0,
3    "status_msg": "xxx",
4    "data": {
5      "finish":0,
6      "time":12.541546,
7    }
8  }
```

1.4.6 返回参数说明

参数名	类型	说明
status_code	integer	状态码
status_msg	string	状态描述
finish	integer	是否完成（0是，1否）
time	float	程序运行时间

1.4.7 备注

无

1.5 递归法和迭代法30妙斐波那契值接口

1.5.1 请求URL

- http://localhost:5000/api/recursiveIterationThirty

1.5.2 请求方式

- GET

1.5.3 请求示例

```
1 http://localhost:5000/api/recursiveEnvMax
```

1.5.4 请求参数说明

参数名	是否必选	类型	说明
无	无	无	无

1.5.5 返回示例


```
1 {
2   "status_code": 0,
3   "status_msg": "xxx",
4   "data": {
5     "nameArr":[
6       {
7         "index":0,
8         "time":12.156,
9         "name":1,
10      }
11    ]
12  }
13 }
```

1.5.6 返回参数说明

参数名	类型	说明
status_code	integer	状态码
status_msg	string	状态描述
index	intger	斐波那契值的位置
time	float	程序运行时间
name	intger	操作码（0是递归法，1是迭代法）

1.5.7 备注

无

1.6 近似公式结果展示接口

1.6.1 请求URL

- `http://localhost:5000/api/formula`

1.6.2 请求方式

- GET

1.6.3 请求示例

```
1 http://localhost:5000/api/formula
```

1.6.4 请求参数说明

参数名	是否必选	类型	说明
无	无	无	无

1.6.5 返回示例

```
1 {
2   "status_code": 0,
3   "status_msg": "xxx",
4   "data": {
5     "num":10,
6   }
7 }
```

1.6.6 返回参数说明

参数名	类型	说明
status_code	integer	状态码
status_msg	string	状态描述
num	intger	斐波那契数的位置

1.6.7 备注

无

2 第二次实验接口文档

2.1 生成随机数组接口

2.1.1 请求URL

- http://localhost:5000/api2/searchMenu

2.1.2 请求方式

- POST

2.1.3 请求示例

```
1 http://localhost:5000/api2/searchMenu
2 {
3   "arrLen": 0
4 }
```

2.1.4 请求参数说明

名称	位置	类型	必选	中文名	说明
body	body	object	否		none
» arrLen	body	integer	是		如果用户是自动生成才发请求

2.1.5 返回示例

```
1 {
2   "status_code": 200,
3   "status_msg": "success",
4   "data": {
5     "arr": "1,2,3"
6   }
7 }
```

2.1.6 返回参数说明

名称	类型	必选	约束	中文名	说明
» status_code	integer	true	none	错误码	none
» status_msg	string	true	none	错误信息	none
» data	object	true	none	正式数据	none
»» arr	string	true	none	生成的数组	格式为“数字,数字,……”

2.1.7 备注

无

2.2 生成无序数组接口

2.2.1 请求URL

- http://localhost:5000/api2/disorderArr

2.2.2 请求方式

- POST

2.2.3 请求示例

```
1 http://localhost:5000/api2/disorderArr
2 {
3   "arrLen": 71
4 }
```

2.2.4 请求参数说明

名称	位置	类型	必选	中文名	说明
body	body	object	否		none
» arrLen	body	integer	是	所需生成数组长度	none

2.2.5 返回示例

```
1 {
2   "status_code": 200,
3   "status_msg": "succeed",
4   "data": {
5     "arr": "1,2,3"
6   }
7 }
```

2.2.6 返回参数说明

名称	类型	必选	约束	中文名	说明
» status_code	integer	true	none	错误码	none
» status_msg	string	true	none	错误信息	none
» data	object	true	none	正式数据	none
»» arr	string	true	none	生成的数组	格式为“数字,数字,……”

2.2.7 备注

无

2.3 先升后降or先降后升数组创建接口

2.3.1 请求URL

- `http://localhost:5000/api2/createMountainArr`

2.3.2 请求方式

- POST

2.3.3 请求示例

```
1 http://localhost:5000/api2/createMountainArr
2 {
3   "pattern": 0,
4   "arrLen": 13
5 }
```

2.3.4 请求参数说明

名称	位置	类型	必选	中文名	说明
body	body	object	否		none
» pattern	body	integer	是	生成模式	0 先升后降 1 先降后升
» arrLen	body	integer	是	数组长度	none

2.3.5 返回示例

```
1 {
2   "status_code": 200,
3   "status_msg": "success",
4   "data": {
5     "arr": "1,2,3"
6   }
7 }
```

2.3.6 返回参数说明

名称	类型	必选	约束	中文名	说明
» status_code	integer	true	none	错误码	none
» status_msg	string	true	none	错误信息	none
» data	object	true	none	正式数据	none
»» arr	string	true	none	生成数组	格式“1,2,3”

2.3.7 备注

无

2.4 判断数组状态接口

2.4.1 请求URL

- `http://localhost:5000/api2/judgementStatus`

2.4.2 请求方式

- POST

2.4.3 请求示例

```
1 http://localhost:5000/api2/judgementStatus
2 {
3   "arr": "string"
4 }
```

2.4.4 请求参数说明

名称	位置	类型	必选	中文名	说明
body	body	object	否		none
» arr	body	string	是	数组	格式为“数字,数字,……”

2.4.5 返回示例

```
1 {
2   "status_code": 200,
3   "status_msg": "success",
4   "data": {
5     "status": 0
6   }
7 }
```

2.4.6 返回参数说明

名称	类型	必选	约束	中文名	说明
» status_code	integer	true	none	错误码	none
» status_msg	string	true	none	错误信息	none
» data	object	true	none	正式数据	none
0 未排序					
1 升序					
2 降序					
3 先升后降					
4 先降后升					
»» status	integer	true	none	数组状态	

2.4.7 备注

无

2.5 数组检索元素接口

2.5.1 请求URL

- http://localhost:5000/api2/seqRet

2.5.2 请求方式

- POST

2.5.3 请求示例

```
1 http://localhost:5000/api2/seqRet
2 {
3   "arr": "string",
4   "item": 0
5 }
```

2.5.4 请求参数说明

名称	位置	类型	必选	中文名	说明
body	body	object	否		none
» arr	body	string	是	被检索数组	格式为“1,2,3”
» item	body	integer	是	目标元素	none

2.5.5 返回示例

```
1 {
2   "status_code": 200,
3   "status_msg": "success",
4   "data": {
5     "haveOrNot": 1,
6     "compareTime": 151293968696
7   }
8 }
```

2.5.6 返回参数说明

名称	类型	必选	约束	中文名	说明
» status_code	integer	true	none	错误码	none
» status_msg	string	true	none	错误信息	none
» data	object	true	none	正式数据	none
»» haveOrNot	integer	true	none	是否存在	0 存在 1 不存在
»» compareTime	integer	true	none	比较次数	none

2.5.7 备注

无

2.6 四种方法检索元素接口

2.6.1 请求URL

- http://localhost:5000/api2/fourMethod

2.6.2 请求方式

- POST

2.6.3 请求示例

```

1 http://localhost:5000/api2/fourMethod
2 {
3   "upDown": 59,
4   "arr": "1,2,3",
5   "item": 33
6 }

```

2.6.4 请求参数说明

名称	位置	类型	必选	中文名	说明
body	body	object	否		none
» upDown	body	integer	是	升序还是降序	0 升序
» arr	body	string	是	数组	格式为“数字,数字,.....”
» item	body	integer	是	需检查元素	none

2.6.5 返回示例

```

1 {
2   "status_code": 200,
3   "status_msg": "success",
4   "data": {
5     "method1": {
6       "haveOrNot": 0,
7       "compareTime": 24795171261,
8       "position": 73
9     },
10    "method2": {
11      "haveOrNot": 1,
12      "compareTime": 366730494703,
13      "position": 8
14    },
15    "method3": {
16      "haveOrNot": 1,
17      "compareTime": 1267813293219,
18      "position": 68
19    },
20    "method4": {
21      "haveOrNot": 0,
22      "compareTime": 310118843137,
23      "position": 44
24    }
25  }
26 }

```

2.6.6 返回参数说明

名称	类型	必选	约束	中文名	说明
» status_code	integer	true	none	错误码	none
» status_msg	string	true	none	错误信息	none
» data	object	true	none	正式数据	none
»» method1	object	true	none	顺序查找	none
»»» haveOrNot	integer	true	none	是否出现	0 出现 1 未出现
»»» compareTime	integer	true	none	比较次数	none
»»» position	integer	true	none	出现位置	-1 升序出现问题 -2 降序出现问题
»» method2	object	true	none	二分查找	none
»»» haveOrNot	integer	true	none	是否出现	0 出现 1 未出现
»»» compareTime	integer	true	none	比较次数	none

名称	类型	必选	约束	中文名	说明
»»» position	integer	true	none	出现位置	-1 升序出现问题 -2 降序出现问题
»» method3	object	true	none	三分查找	none
»»» haveOrNot	integer	true	none	是否出现	0 出现 1 未出现
»»» compareTime	integer	true	none	比较次数	none
»»» position	integer	true	none	出现位置	-1 升序出现问题 -2 降序出现问题
»» method4	object	true	none	插值查找	none
»»» haveOrNot	integer	true	none	是否出现	0 出现 1 未出现
»»» compareTime	integer	true	none	比较次数	none
»»» position	integer	true	none	出现位置	-1 升序出现问题 -2 降序出现问题

2.6.7 备注

无

2.7 对先升后降or先降后升数组找最大最小值接口

2.7.1 请求URL

- `http://localhost:5000/api2/searchMenuUpDown`

2.7.2 请求方式

- POST

2.7.3 请求示例

```

1 http://localhost:5000/api2/fourMethod
2 {
3   "arr": "1,2,3"
4 }
```

2.7.4 请求参数说明

名称	位置	类型	必选	中文名	说明
body	body	object	否		none
» arr	body	string	是	数组	none

2.7.5 返回示例

```

1 {
2   "status_code": 200,
3   "status_msg": "success",
4   "data": {
5     "method1": {
6       "position": 51,
7       "compareTime": 291836455872
8     },
9     "method2": {
10      "position": 4,
11      "compareTime": 1254367139033
12    }
13  }
14 }
```


2.7.6 返回参数说明

名称	类型	必选	约束	中文名	说明
» status_code	integer	true	none	错误码	none
» status_msg	string	true	none	错误信息	none
» data	object	true	none	正式数据	none
»» method1	object	true	none	二分搜索	none
»»» position	integer	true	none	最大或最小值位置	none
»»» compareTime	integer	true	none	比较次数	none
»» method2	object	true	none	三分搜索	none
»»» position	integer	true	none	最大或最小值位置	none
»»» compareTime	integer	true	none	比较次数	none

2.7.7 备注

无

2.8 数组第k个最小元素接口

2.8.1 请求URL

- http://localhost:5000/api2/ThreeMethods

2.8.2 请求方式

- POST

2.8.3 请求示例

```
1 http://localhost:5000/api2/ThreeMethods
2 {
3   "arr": "1,2,34,4,3,5",
4   "k": 2
5 }
```

2.8.4 请求参数说明

名称	位置	类型	必选	中文名	说明
body	body	object	否		none
» arr	body	string	是	数组	格式"1,2,3"
» k	body	integer	是	k值	none

2.8.5 返回示例

```
1 {
2   "status_code": 200,
3   "status_msg": "success",
4   "data": {
5     "method1": {
6       "kNum": 25,
7       "compareTime": 326374230455
8     },
9     "method2": {
10      "kNum": 96,
11      "compareTime": 1597484094156
12    },
13    "method3": {
```

```

14     "kNum": 3,
15     "compareTime": 66094065517
16   }
17 }
18 }

```

2.8.6 返回参数说明

名称	类型	必选	约束	中文名	说明
» status_code	integer	true	none	错误码	none
» status_msg	string	true	none	错误信息	none
» data	object	true	none	正式数据	none
»» method1	object	true	none	蛮力法	none
»»» kNum	integer	true	none	第k个最小元素	none
»»» compareTime	integer	true	none	比较次数	none
»» method2	object	true	none	预排序	none
»»» kNum	integer	true	none	第k个最小元素	none
»»» compareTime	integer	true	none	比较次数	none
»» method3	object	true	none	减可变规模	none
»»» kNum	integer	true	none	第k个最小元素	none
»»» compareTime	integer	true	none	比较次数	none

2.8.7 备注

无

3 第三次实验接口文档

3.1 贪心法求解分数背包问题接口

3.1.1 请求URL

- <http://localhost:5000/api3/bagMenu>

3.1.2 请求方式

- POST

3.1.3 请求示例

```

1 http://localhost:5000/api3/bagMenu
2 {
3   "bagCap": 65,
4   "bagNum": 34,
5   "eachVal": [
6     1,
7     2,
8     3
9   ],
10  "eachWeight": [
11    1,
12    2,
13    3
14  ]
15 }

```

3.1.4 请求参数说明

名称	位置	类型	必选	中文名	说明
body	body	object	否		none
» bagCap	body	integer	是	背包容量	none
» bagNum	body	integer	是	物品个数	none
» eachWeight	body	number	是	每个物品的重量	空格隔开
» eachVal	body	number	是	每个物品的价值	空格隔开

3.1.5 返回示例

```
1 {
2   "status_code": 35,
3   "status_msg": "enim eiusmod amet",
4   "data": {
5     "weight": [
6       0,
7       1,
8       2,
9       3
10    ],
11    "value": [
12      0,
13      1,
14      2,
15      3
16    ]
17  }
18 }
```

3.1.6 返回参数说明

名称	类型	必选	约束	中文名	说明
» status_code	integer	true	none	错误码	none
» status_msg	string	true	none	错误信息	none
» data	object	true	none	正式数据	none
»» weight	number	true	none	重量数组	空格隔开
»» value	number	true	none	价值数组	空格隔开

3.1.7 备注

无

3.2 贪心法求0-1背包问题接口

3.2.1 请求URL

- http://localhost:5000/api3/Greedy01

3.2.2 请求方式

- POST

3.2.3 请求示例

```
1 http://localhost:5000/api3/Greedy01
2 {
3   "bagCap": 84,
4   "bagNum": 53,
5   "eachVal": [
6     0,
7     1,
8     2,
9     3
10  ],
11  "eachWeight": [
12    0,
13    1,
14    2,
15    3
16  ]
17 }
```

3.2.4 请求参数说明

名称	位置	类型	必选	中文名	说明
token	query	string	是		none
body	body	object	否		none
» bagCap	body	integer	是		none
» bagNum	body	integer	是		none
» eachWeight	body	number	是	空格隔开	none
» eachVal	body	number	是	空格隔开	none

3.2.5 返回示例

```
1 {
2   "status_code": 200,
3   "status_msg": "success",
4   "data": {
5     "weight": [
6       0,
7       1,
8       2,
9       3
10    ],
11    "value": [
12      0,
13      1,
14      2,
15      3
16    ]
17  }
18 }
```

3.2.6 返回参数说明

名称	类型	必选	约束	中文名	说明
» status_code	integer	true	none		none
» status_msg	string	true	none		none
» data	object	true	none		none
»» weight	number	true	none		none

名称	类型	必选	约束	中文名	说明
»» value	number	true	none		none

3.2.7 备注

无

3.3 蛮力法求解01背包问题接口

3.3.1 请求URL

- http://localhost:5000/api3/Brute01

3.3.2 请求方式

- POST

3.3.3 请求示例

```
1 http://localhost:5000/api3/Brute01
2 {
3   "bagCap": 32,
4   "bagNum": 40,
5   "eachWeight": [
6     0,
7     1,
8     2,
9     3
10  ],
11  "eachVal": [
12    0,
13    1,
14    2,
15    3
16  ]
17 }
```

3.3.4 请求参数说明

名称	位置	类型	必选	中文名	说明
body	body	object	否		none
» bagCap	body	integer	是		none
» bagNum	body	integer	是		none
» eachWeight	body	number	是	空格隔开	none
» eachVal	body	number	是	空格隔开	none

3.3.5 返回示例

```
1 {
2   "status_code": 0,
3   "status_msg": "string",
4   "data": 0
5 }
```

3.3.6 返回参数说明

名称	类型	必选	约束	中文名	说明
» status_code	integer	true	none		none
» status_msg	string	true	none		none
» data	integer	true	none		none

3.3.7 备注

无

3.4 动态规划求解0-1背包问题接口

3.4.1 请求URL

- `http://localhost:5000/api3/Dynamic01`

3.4.2 请求方式

- POST

3.4.3 请求示例

```
1 http://localhost:5000/api3/Dynamic01
2 {
3   "bagCap": 7,
4   "bagNum": 55,
5   "eachWeight": [
6     0,
7     1,
8     2,
9     3
10  ],
11   "eachVal": [
12     0,
13     1,
14     2,
15     3
16   ]
17 }
```

3.4.4 请求参数说明

名称	位置	类型	必选	中文名	说明
body	body	object	否		none
» bagCap	body	integer	是		none
» bagNum	body	integer	是		none
» eachWeight	body	number	是		空格隔开
» eachVal	body	number	是		空格隔开

3.4.5 返回示例

```
1 {
2   "status_code": 0,
3   "status_msg": "string",
4   "data": 0
5 }
```

3.4.6 返回参数说明

名称	类型	必选	约束	中文名	说明
» status_code	integer	true	none		none
» status_msg	string	true	none		none
» data	integer	true	none		none

3.4.7 备注

无

3.5 记忆功能改进动态规划接口

3.5.1 请求URL

- `http://localhost:5000/api3/ImproveDy`

3.5.2 请求方式

- POST

3.5.3 请求示例

```
1 http://localhost:5000/api3/ImproveDy
2 {
3   "bagCap": 24,
4   "bagNum": 44,
5   "eachWeight": [
6     0,
7     1,
8     2,
9     3
10  ],
11   "eachVal": [
12     0,
13     1,
14     2,
15     3
16   ]
17 }
```

3.5.4 请求参数说明

名称	位置	类型	必选	中文名	说明
body	body	object	否		none
» bagCap	body	integer	是		none
» bagNum	body	integer	是		none
» eachWeight	body	number	是		空格隔开
» eachVal	body	number	是		空格隔开

3.5.5 返回示例

```
1 {
2   "status_code": 0,
3   "status_msg": "string",
4   "data": 0
5 }
```

3.5.6 返回参数说明

名称	类型	必选	约束	中文名	说明
» status_code	integer	true	none		none
» status_msg	string	true	none		none
» data	integer	true	none		none

3.5.7 备注

无