

用户注册

接口地址: /user/regist

请求方式: POST

请求参数:

POST

http://localhost:8080/user/regist

Params

Authorization

Headers (9)

Body

Pre-request Script

Tests

Settings

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

1

{

2

"username": "wfwf",

3

"password": "123",

4

"phoneNumber": "110",

5

"idCard": "350121",

6

"email": "123@qq.com",

7

"sex": "1"

8

}

返回正确json示例:

1

2

"publicKey":

"5863193227057839280782184734470618254562455601655787972335116968918757251123443006188325101589816356008837359973928143995699065981154

94964099931791297931737867676982924778884166700435168989306743004483115432137377559966817391816731278194934452172272526829275398800305

9147400694143480023116966810425071095659044189713380801487921045214270929759430048265445566026050104413352388403979249842935178307307

1394615147715615569385014968547437543764016276836905185778206479210854075675914398543200870138892192962594746510489218700615082652521

24047550611165480306354334285638308678400564739999186526127915576660485529348159958342",

3

"message": "注册成功!",

4

"status": true

5

用户登录

接口地址: /user/login

请求方式: POST

请求参数:

POST

http://localhost:8080/user/login

Params

Authorization

Headers (9)

Body

Pre-request Script

Tests

Settings

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

1

{

2

"username": "aabb",

3

"password": "123"

4

}

返回正确json示例:

```
Body Cookies (1) Headers (8) Test Results
Pretty Raw Preview Visualize JSON ↕
1 {
2   "message": "登录成功!",
3   "userId": "6d2b52187b5e4da99824769d2f346fed",
4   "status": true
5 }
```

返回错误json示例:

```
POST http://localhost:8080/user/login
Params Authorization Headers (9) Body ● Pre-request Script Tests Settings
● none ● form-data ● x-www-form-urlencoded ● raw ● binary ● GraphQL JSON ▼
1 {
2   ... "username": "aabb",
3   ... "password": "12"
4 }
5
6
7

Body Cookies (1) Headers (8) Test Results
Pretty Raw Preview Visualize JSON ↕
1 {
2   "message": "登录失败!",
3   "status": false
4 }
```

用户完善信息

接口地址: /user/update

请求方式: POST

请求参数:

POST

http://localhost:8080/user/update

Params

Authorization

Headers (8)

Body

Pre-request Script

Tests

Settings

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

1

{

2

"userId": "3d817ab092c64930bb877538902a7cb6",

3

"newPassword": "123456",

4

"phoneNumber": "119",

5

"idCard": "350121",

6

"email": "110@163.com",

7

"sex": "1"

8

}

9

返回正确json示例：

Body

Cookies

Headers (8)

Test Results

Pretty

Raw

Preview

Visualize

JSON

1

{

2

"message": "更新成功!",

3

"status": true

4

}

生成公证机构

接口地址： /organizationGen

请求方式： POST

请求参数：

POST

http://localhost:8080/organizationGen

Params

Authorization

Headers (8)

Body

Pre-request Script

Tests

Settings

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

1

{

2

"organizationName": "上海公证处",

3

"address": "上海市",

4

"phoneNumber": "110",

5

"email": "123@qq.com",

6

"legalPeople": "上海"

7

}

返回正确json示例：

Body Cookies Headers (8) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "message": "注册成功!",
3   "status": true
4 }
```

公证员注册

接口地址: /notar/regist

请求方式: POST

请求参数:

POST http://localhost:8080/notar/regist ...

Params Authorization Headers (8) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   "notaryName": "PDD",
3   "password": "123",
4   "phoneNumber": "110",
5   "idCard": "350121",
6   "email": "110@qq.com",
7   "sex": "1",
8   "organizationId": "69c2be6d629a4148a4517f7bf4d7112b",
9   "notarizationType": "1",
10  "jobNumber": "123"
11 }
```

返回正确json示例:

Body Cookies (1) Headers (8) Test Results

Status: 200 OK Time: 489 ms Size: 925 B Save Response

Pretty Raw Preview Visualize JSON

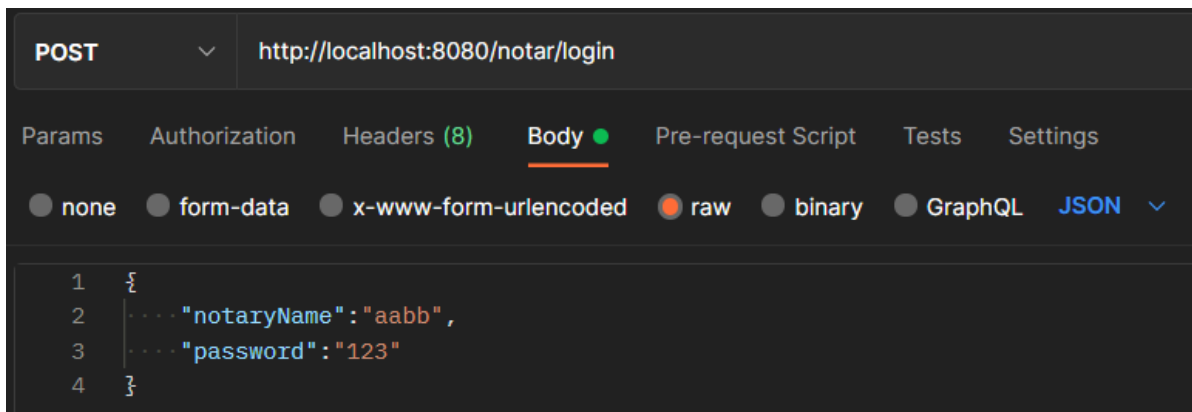
```
1 {
2   "publicKey":
3     "647123949933907683621296517455393128732505898539836327140537298816386054472953333666928292896766271218091400024742799429903104862335
4     8726784568263138810950047591734466057316621769042067454724627220054692939949123963139929983309289755862796453465619006959790781603979
5     4009615432138738596617012385925851540292233080477866915432412504999675590396336450198489700894961895401780505895460108557005118680671
6     1389375818738786899309114595770418863603331488206063810125278568617213173438580698212707721504480459666894164934956288498139420091395
7     5619001415047294511879040381176902127152732165688893032008716664592484404713619281556",
8   "message": "注册成功!",
9   "status": true
10 }
```

公证员登录

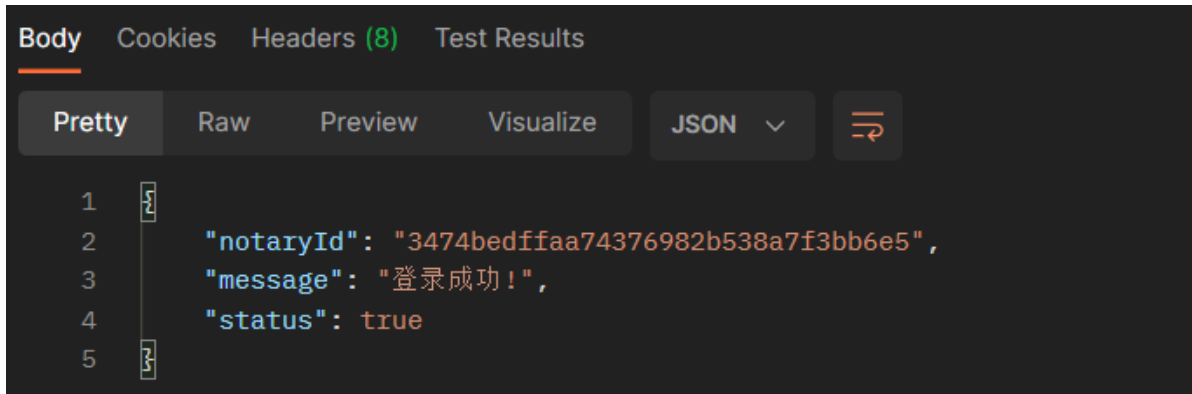
接口地址: /notar/login

请求方式: POST

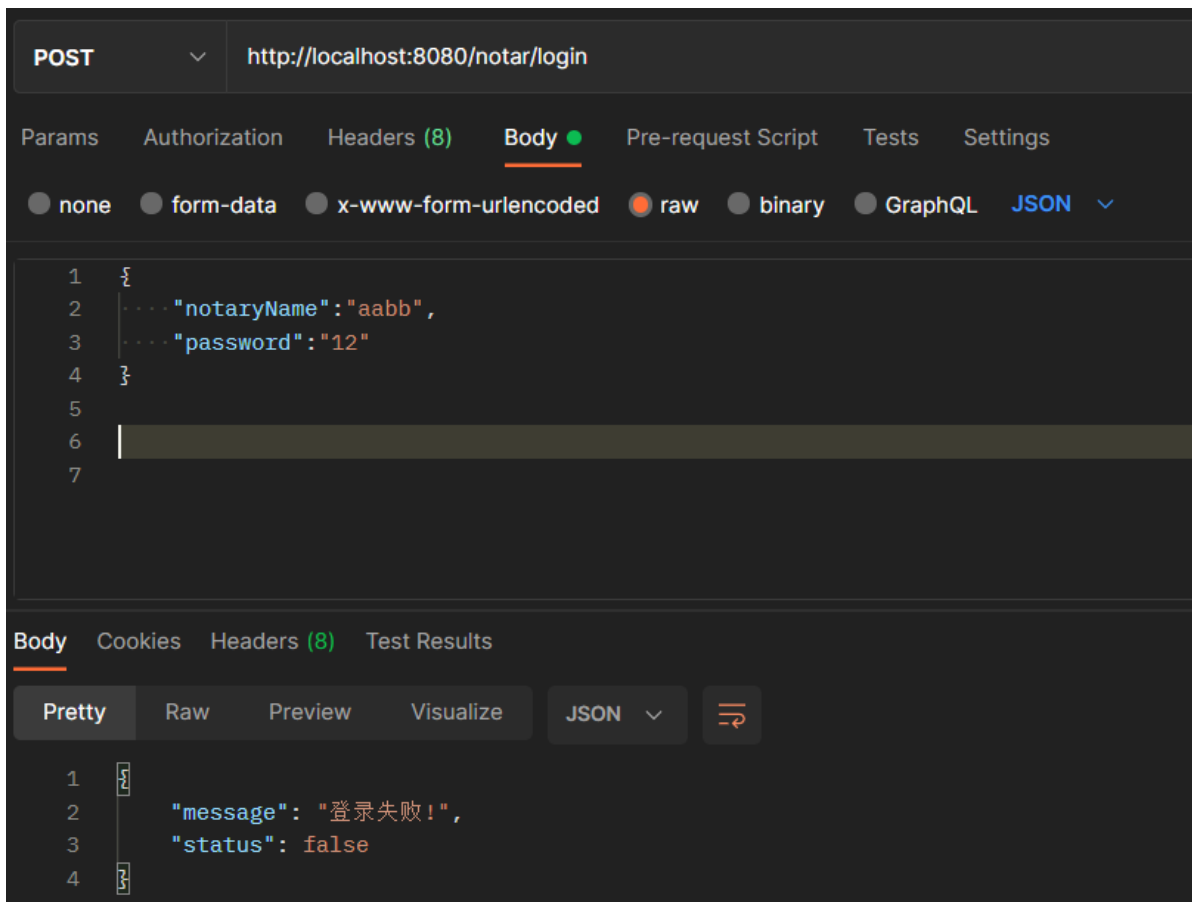
请求参数:



返回正确json示例：



返回错误json示例：

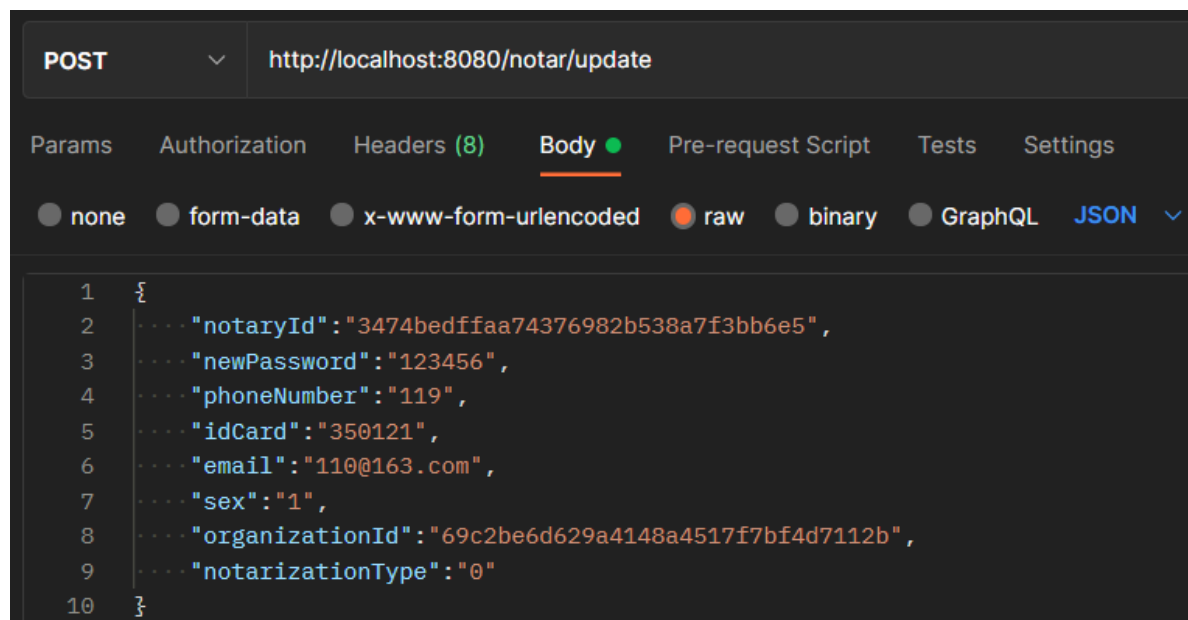


公证员完善信息

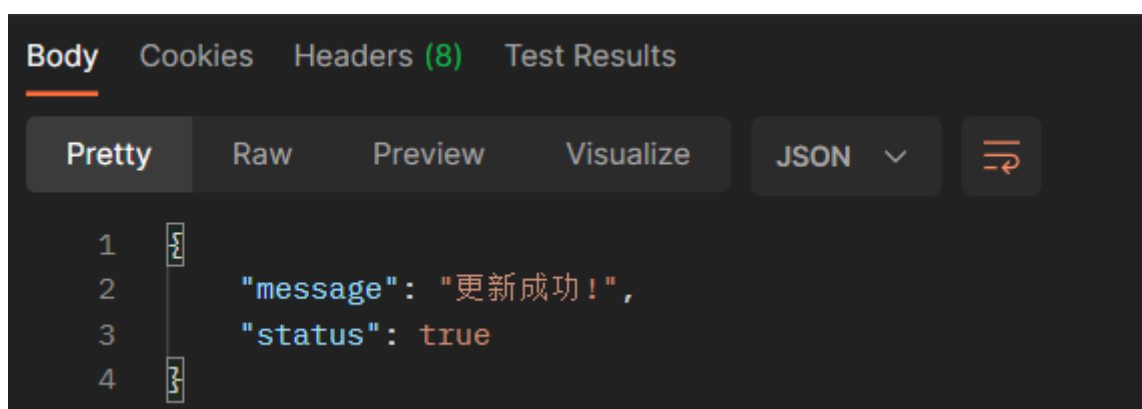
接口地址：/notar/update

请求方式：POST

请求参数:



返回正确json示例:

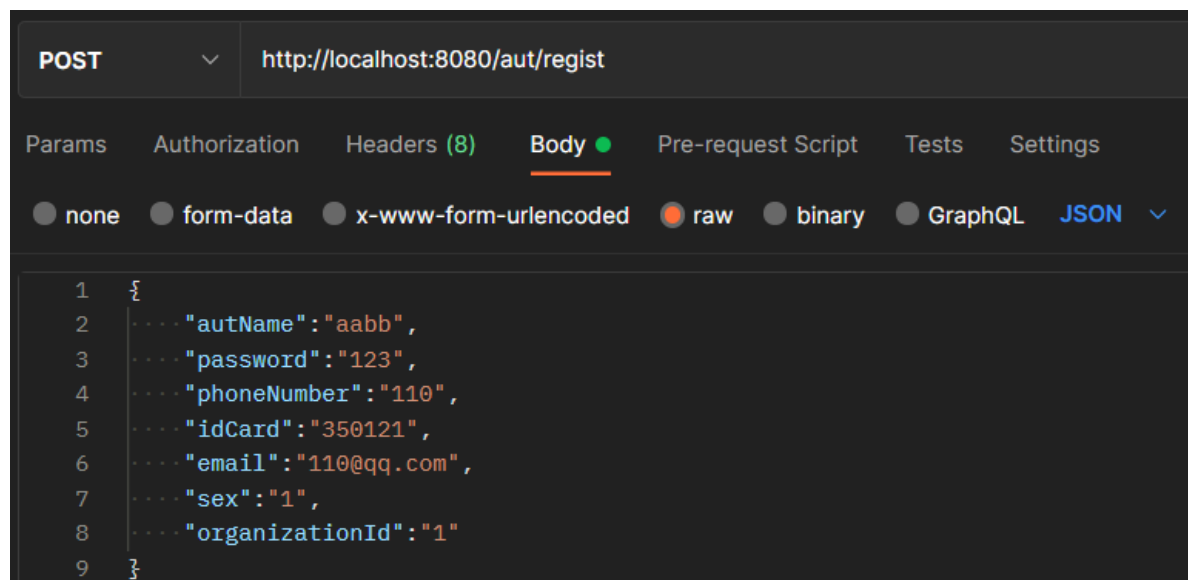


机构管理员注册

接口地址: /aut/regist

请求方式: POST

请求参数:



返回正确json示例:



The screenshot shows a REST client interface with tabs for Body, Cookies, Headers (8), and Test Results. The Body tab is selected, and the response is displayed in a 'Pretty' JSON format. The JSON object contains a success message, an authentication ID, and a status flag.

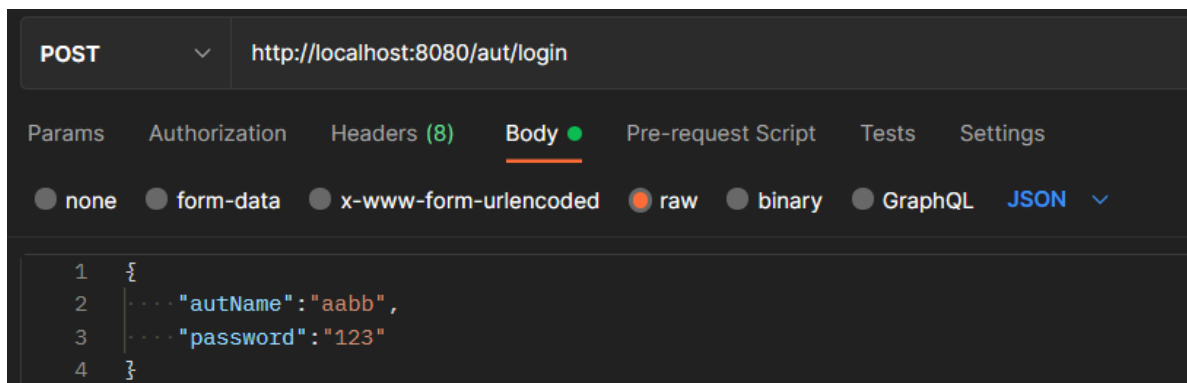
```
1 {  
2   "message": "注册成功!",  
3   "autManId": "bca3a7117e20432da46c182633d7b372",  
4   "status": true  
5 }
```

机构管理员登录

接口地址: /aut/login

请求方式: POST

请求参数:



The screenshot shows a REST client interface with a POST request to http://localhost:8080/aut/login. The 'Body' tab is selected, and the request body is shown in a 'JSON' format. The JSON object contains the username and password for login.

```
1 {  
2   "autName": "aabb",  
3   "password": "123"  
4 }
```

返回正确json示例:



The screenshot shows a REST client interface with tabs for Body, Cookies, Headers (8), and Test Results. The Body tab is selected, and the response is displayed in a 'Pretty' JSON format. The JSON object contains a success message, an authentication ID, and a status flag.

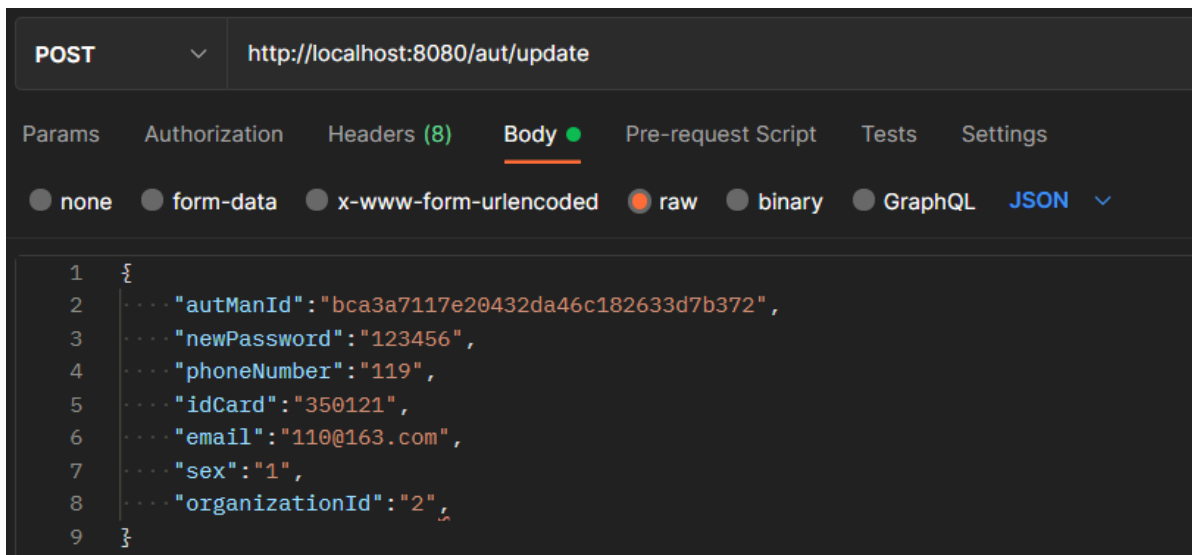
```
1 {  
2   "message": "登录成功!",  
3   "autManId": "bca3a7117e20432da46c182633d7b372",  
4   "status": true  
5 }
```

机构管理员完善信息

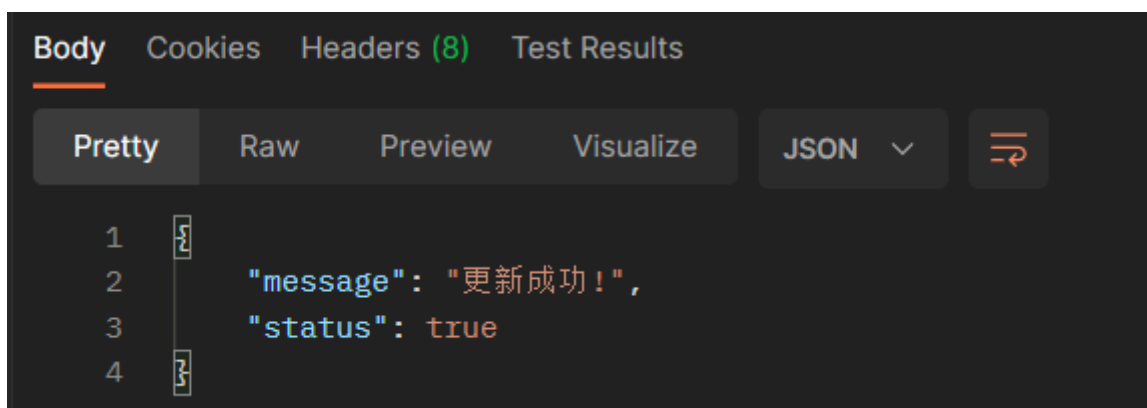
接口地址: /aut/update

请求方式: POST

请求参数:



返回正确json示例：

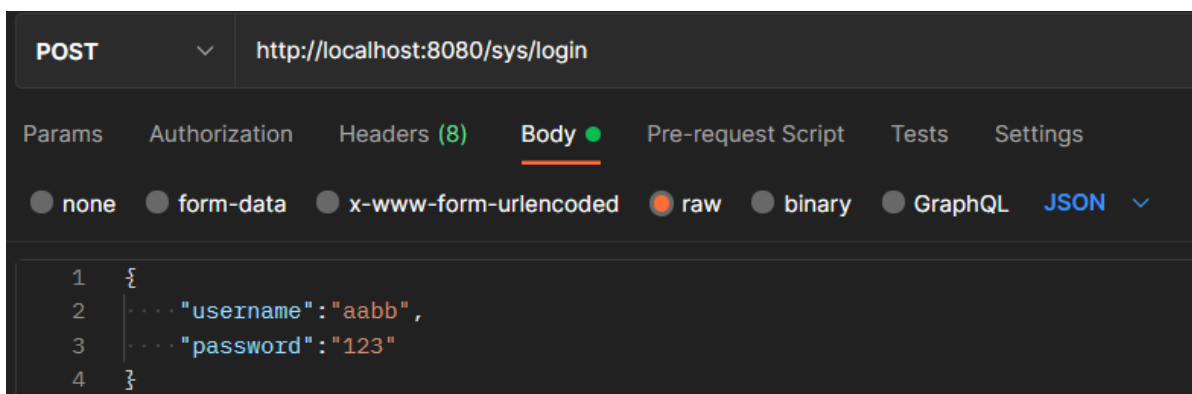


系统管理员登录

接口地址：/sys/login

请求方式：POST

请求参数：



返回正确json示例：

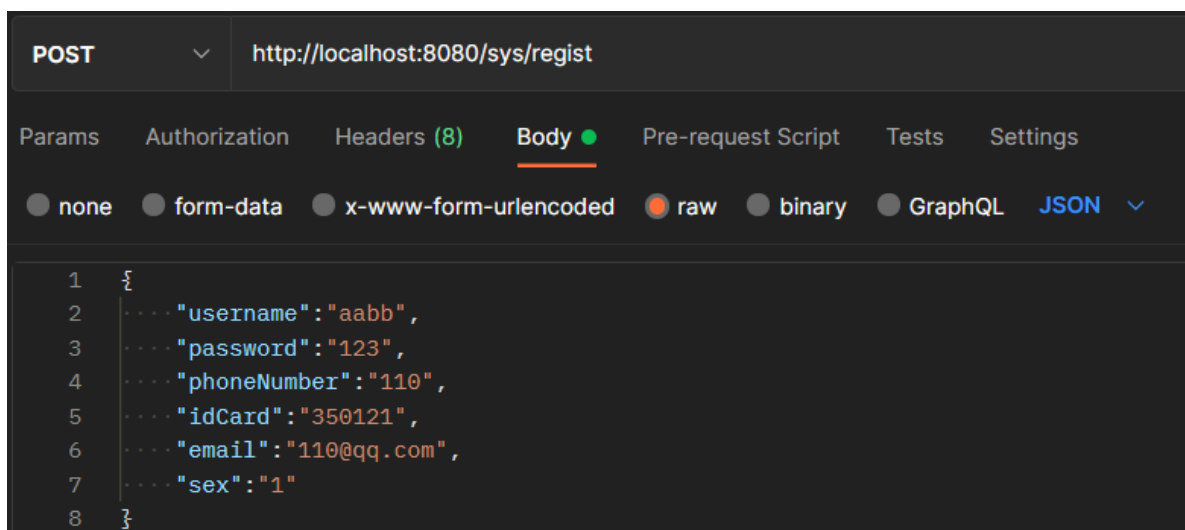


系统管理员注册

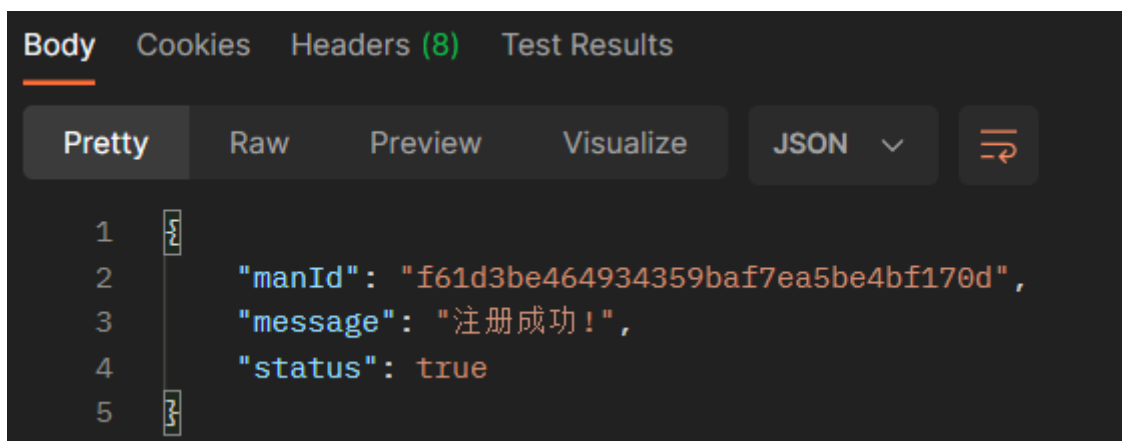
接口地址: /sys/regist

请求方式: POST

请求参数:



返回正确json示例:

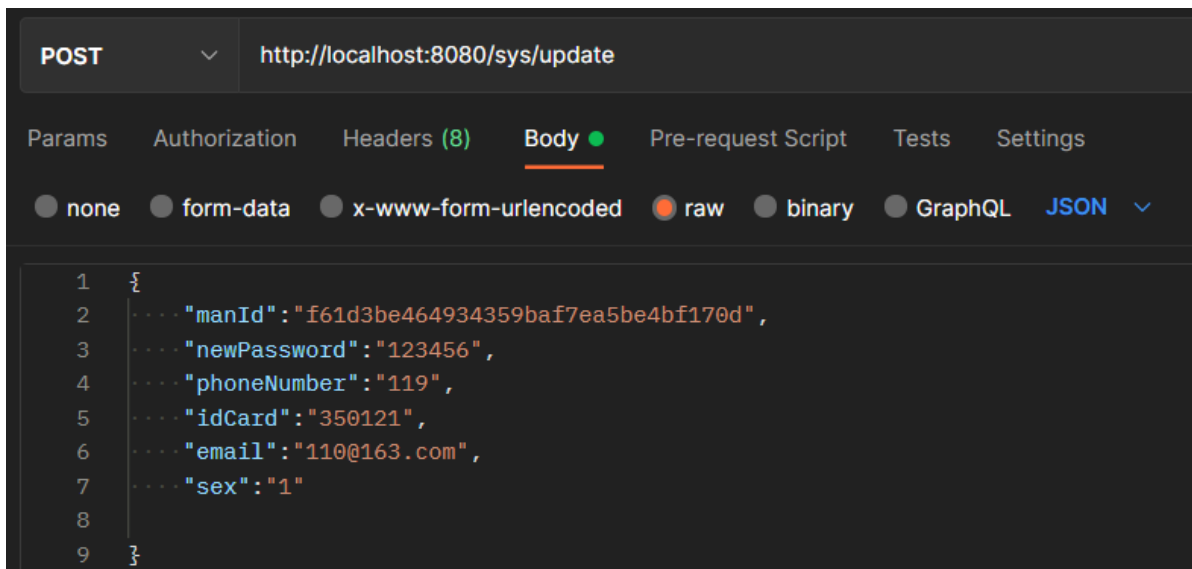


系统管理员完善信息

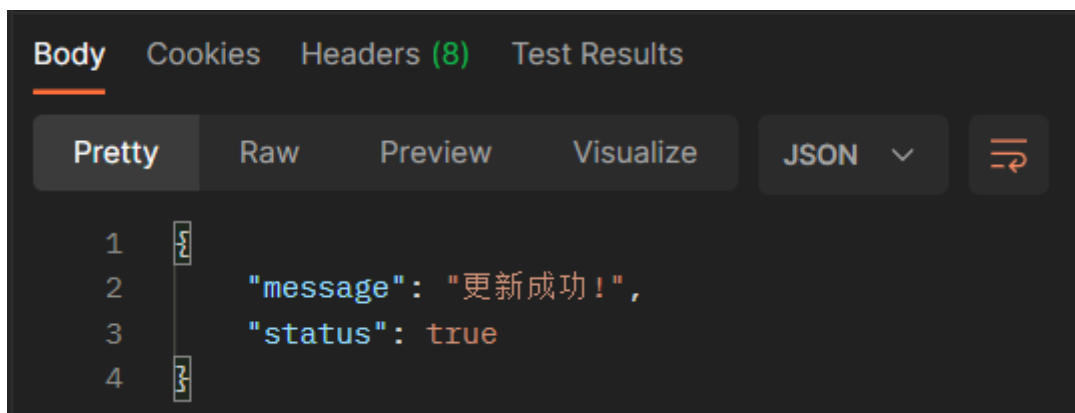
接口地址: /sys/update

请求方式: POST

请求参数:



返回正确json示例:

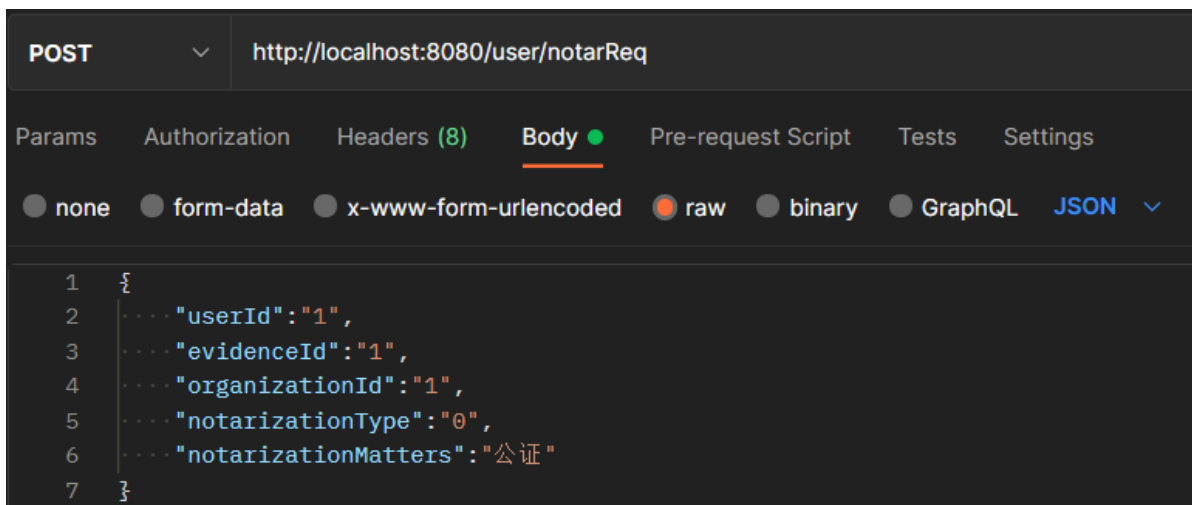


公证申请

接口地址: /user/notarReq

请求方式: POST

请求参数:



返回正确json示例:

Body Cookies Headers (8) Test Results

Pretty Raw Preview Visualize JSON ↗

```
1 {
2   "notarizationMoney": "100",
3   "notarizationStartTime": "2021-08-11T22:31:46.000+00:00",
4   "transactionStatus": "0",
5   "message": "申请公证成功",
6   "notarizationStatus": "1",
7   "transactionId": "7fb2acc72a044f47b4c9febe8e481e4d",
8   "status": true
9 }
```

公证缴费

接口地址: /user/notarPay

请求方式: POST

请求参数:

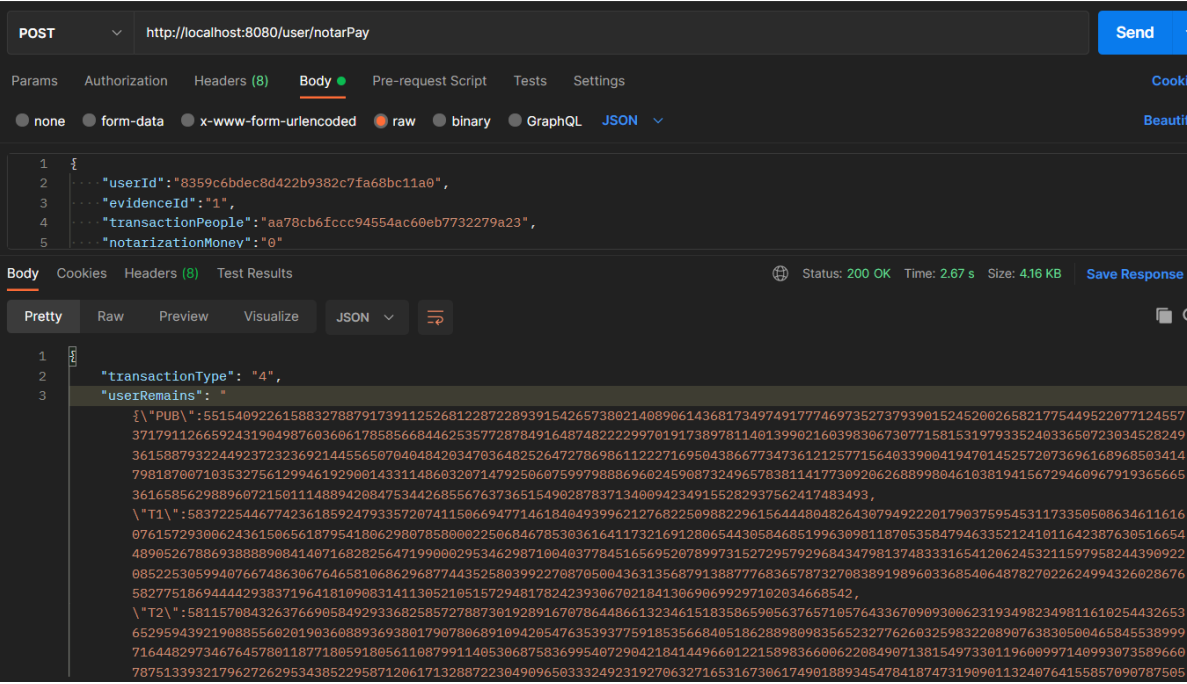
POST http://localhost:8080/user/notarPay

Params Authorization Headers (8) Body ● Pre-request Script Tests Settings

● none ● form-data ● x-www-form-urlencoded ● raw ● binary ● GraphQL JSON ▼

```
1 {
2   ... "userId": "8359c6bdec8d422b9382c7fa68bc11a0",
3   ... "evidenceId": "1",
4   ... "transactionPeople": "aa78cb6fccc94554ac60eb7732279a23",
5   ... "notarizationMoney": "0"
6
7 }
```

返回正确json示例:

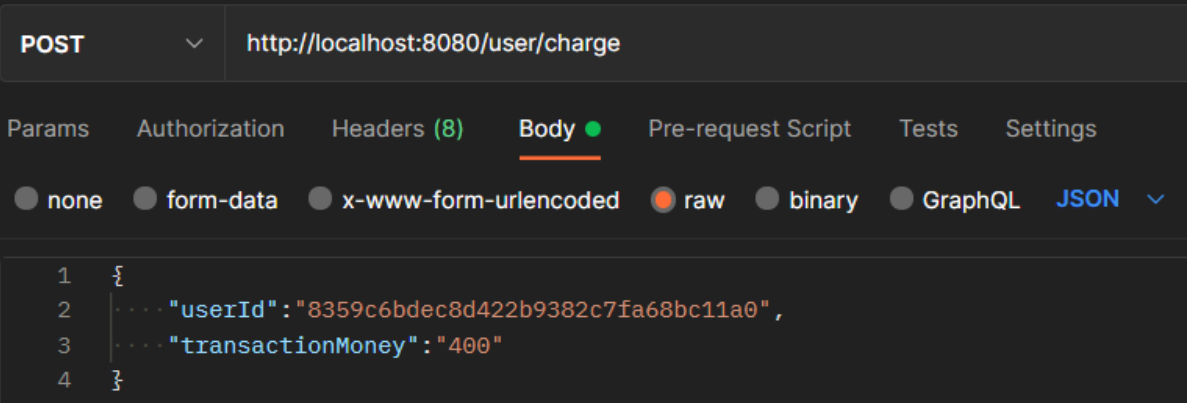


积分商店充值

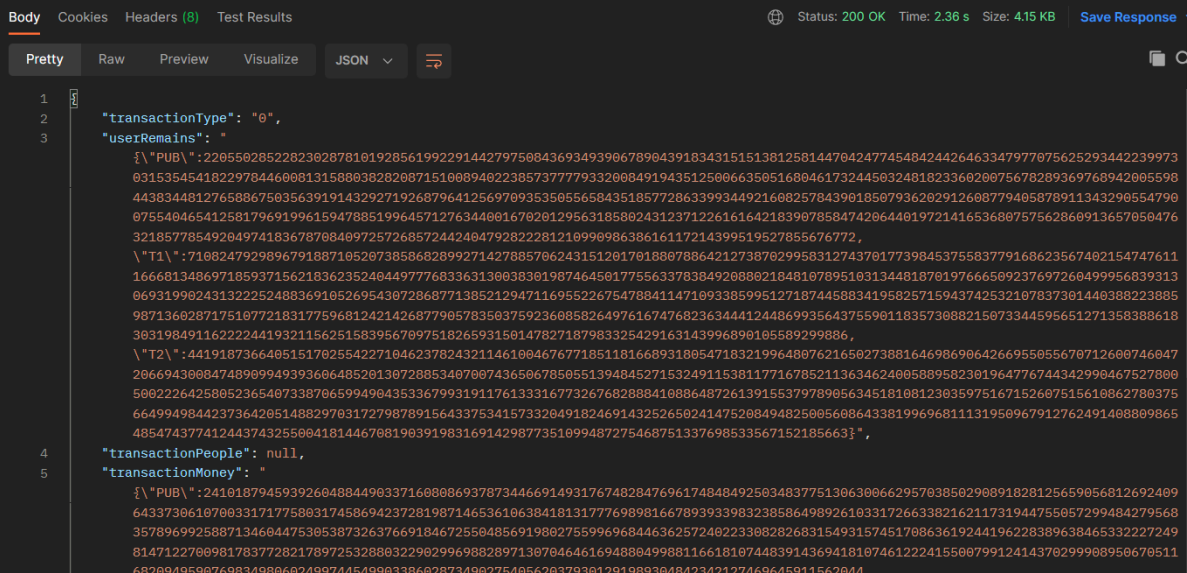
接口地址: /user/charge

请求方式: POST

请求参数:



返回正确json示例:



积分商店转赠

接口地址： /user/give

请求方式： POST

请求参数：

POST

▼

http://localhost:8080/user/give

Params

Authorization

Headers (8)

Body

Pre-request Script

Tests

Settings

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

▼

1

{

2

... "userId": "8359c6bdec8d422b9382c7fa68bc11a0",

3

... "transactionPeople": "aa78cb6fccc94554ac60eb7732279a23",

4

... "transactionMoney": "200"

5

}

返回正确json示例：

Body

Cookies

Headers (8)

Test Results

⌕

Status: 200 OK

Time: 2.32 s

Size: 4.15 KB

Save Response

Pretty

Raw

Preview

Visualize

JSON

▼

⌕

1

2

3

"transactionType": "1",

"userRemains": "

{ "PUB": "2409035046841783322606125419290362631127798853989646512584469114247331856395168870082003693523889196292480717310057183098630

9911798248092190245303373481545696322158062765691813524073896497281509890695667104113041948082957627937860479879669252521599319271549

7347047605947609065515110861124664376815253890403530694129538833136027390352687808427545143227612317926638374974428552613821184173198

82441073024188077246963877267831080822162212085351547089624869878000736086516632331799031790790984852473900108414234830506134410075

849706430466738414358388405693747272176152503802496279113009831258562338045917809680393308818,

"T1": "637247863507453355627877218963524576154606851785426474684670858170604291998362670763800853096573637378850236501576170451446164

4203256473061932028455287427633910995514201521525965115938317137277266598005481667502491951405391023438614198283826805770056213010068

4037197933626483175478417048137853578862050075195922287941798188303529852035009008364348900037900219303411140488637609136468887795203

5381200103820826046275266346169349424863738375244268214181071627083692854034071542365795939075398747928415817644322736629726489978847

9896454279672579728076913724089742679421977174296990936031190075610644420242707006691855509,

"T2": "401036114745201252779746676705229790603759360110642550220500161393835686113680541260386835289125357123939852229642907057276438

6569587825485182404975260734681876530294259022101372678769880536706408688003005382824047510761849167378898266882924754879467441238325

1202359036392623639785613870548999969395511449088608745045522249189854475473488895420256050492304982302983338504545603312295939078876

7205851917708272648194326418242994596453692304870140152386033758209445523945079581108050970837846486135474600364772240797815687877344

9538383475732413603812798129187387772155354246939717507841914754230906458897402128734755377",

积分商店提现

接口地址： /user/withdraw

请求方式： POST

请求参数：

POST

▼

http://localhost:8080/user/withdraw

Params

Authorization

Headers (8)

Body

Pre-request Script

Tests

Settings

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

▼

1

{

2

... "userId": "8359c6bdec8d422b9382c7fa68bc11a0",

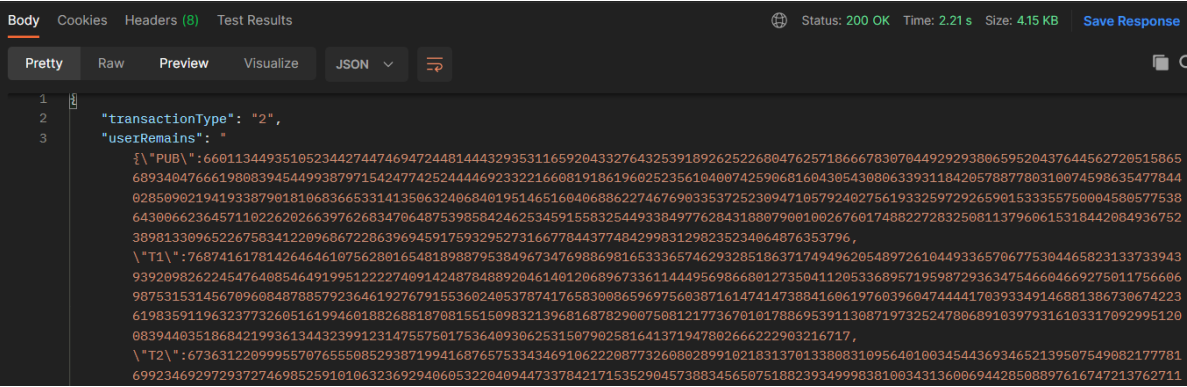
3

... "transactionMoney": "20"

4

}

返回正确json示例：

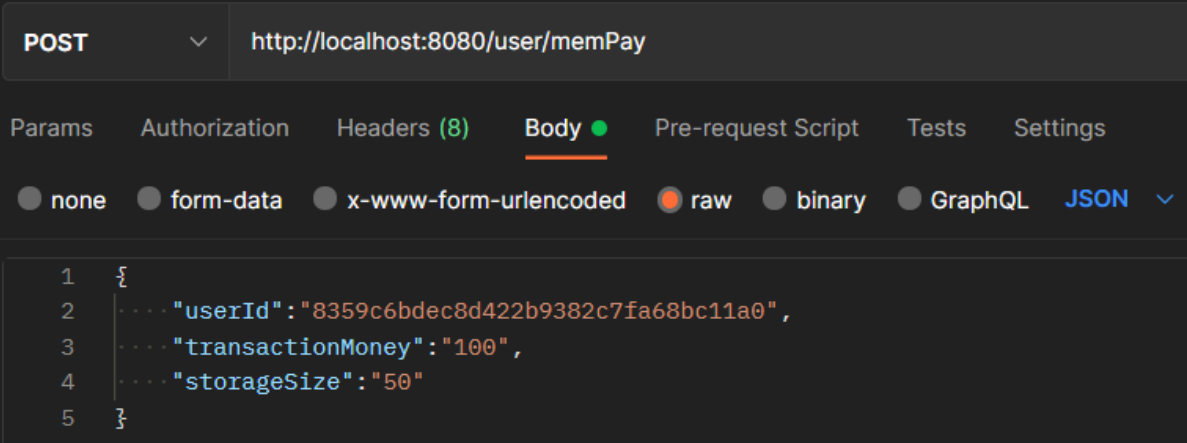


积分商店购买存储空间

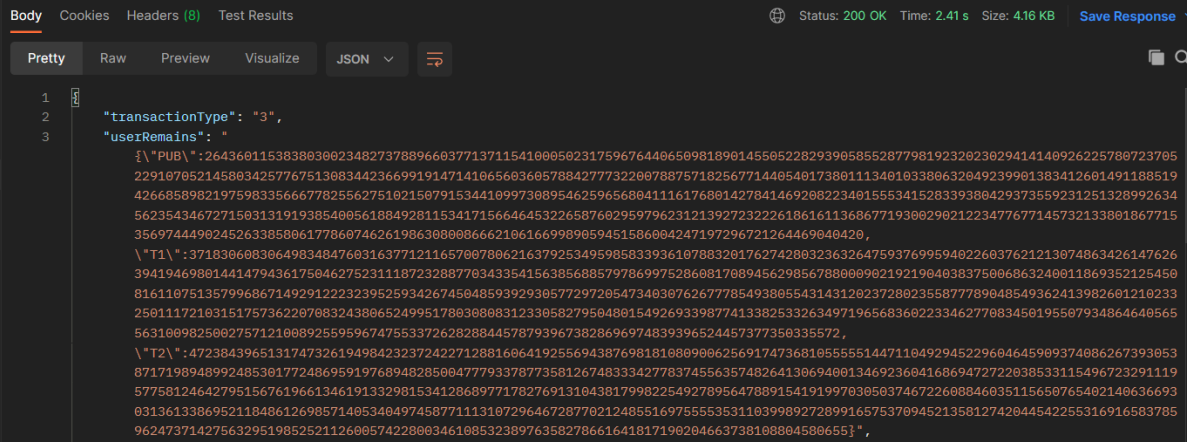
接口地址: /user/memPay

请求方式: POST

请求参数:



返回正确json示例:

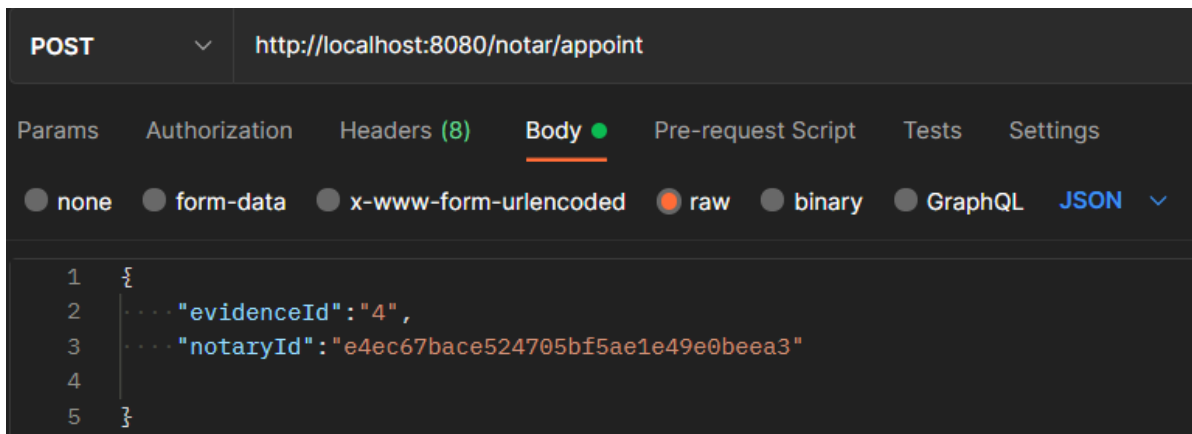


预约公证

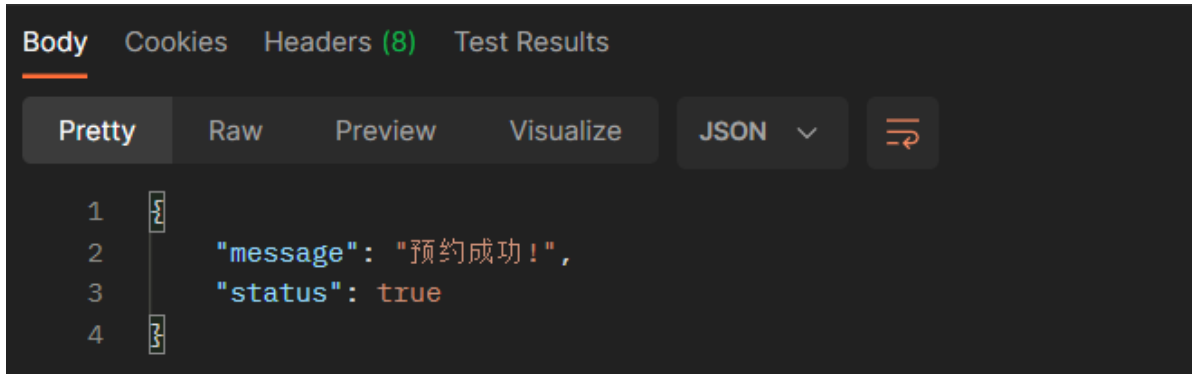
接口地址: /notar/appoint

请求方式: POST

请求参数:



返回正确json示例:

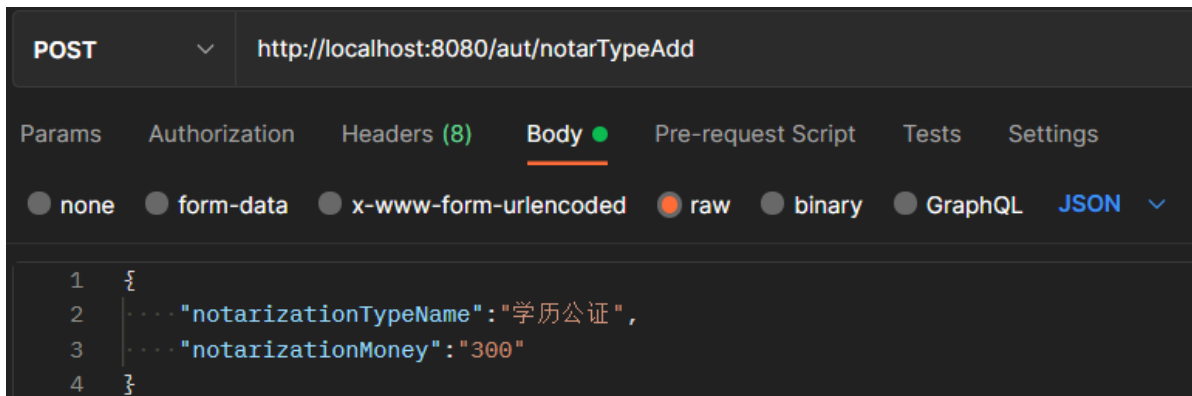


新增公证类型

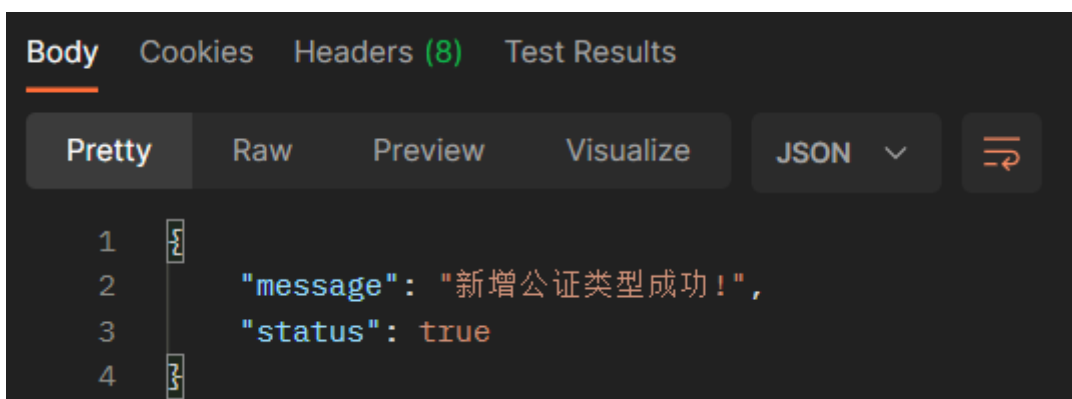
接口地址: /aut/notarTypeAdd

请求方式: POST

请求参数:



返回正确json示例:

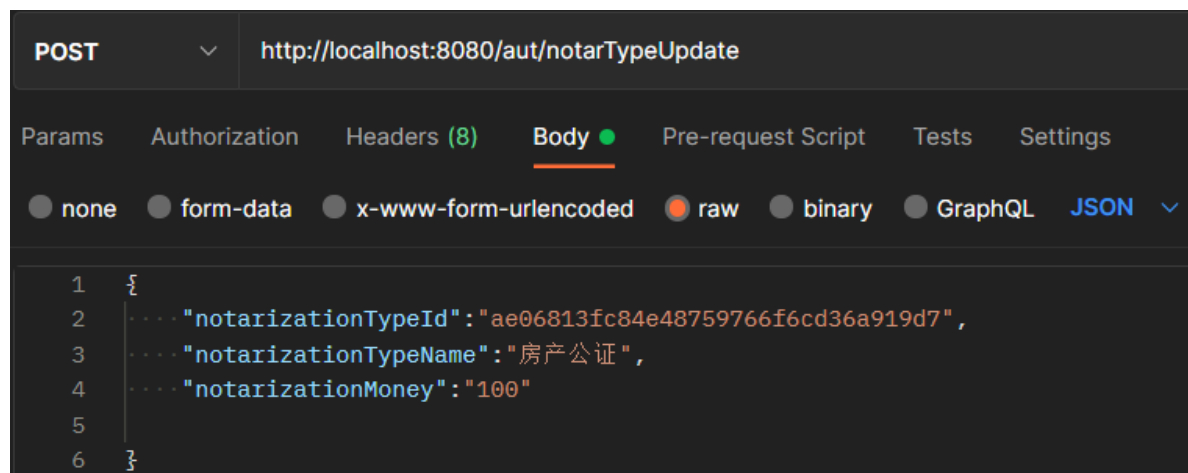


更新公证类型

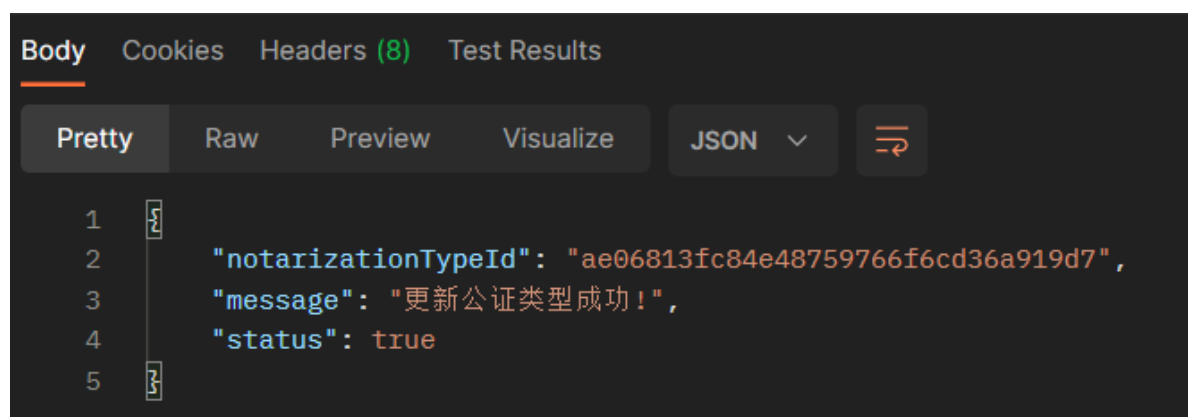
接口地址: /aut/notarTypeUpdate

请求方式: POST

请求参数:



返回正确json示例:

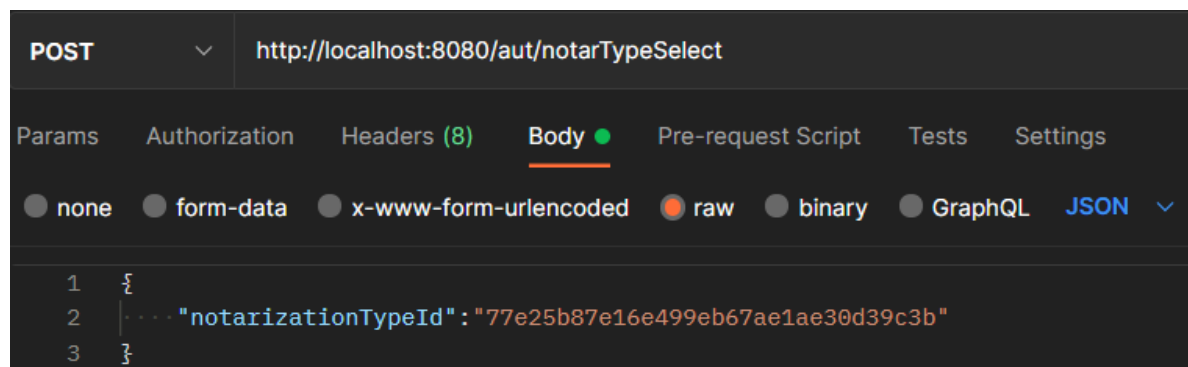


查找公证类型名称和公证金额

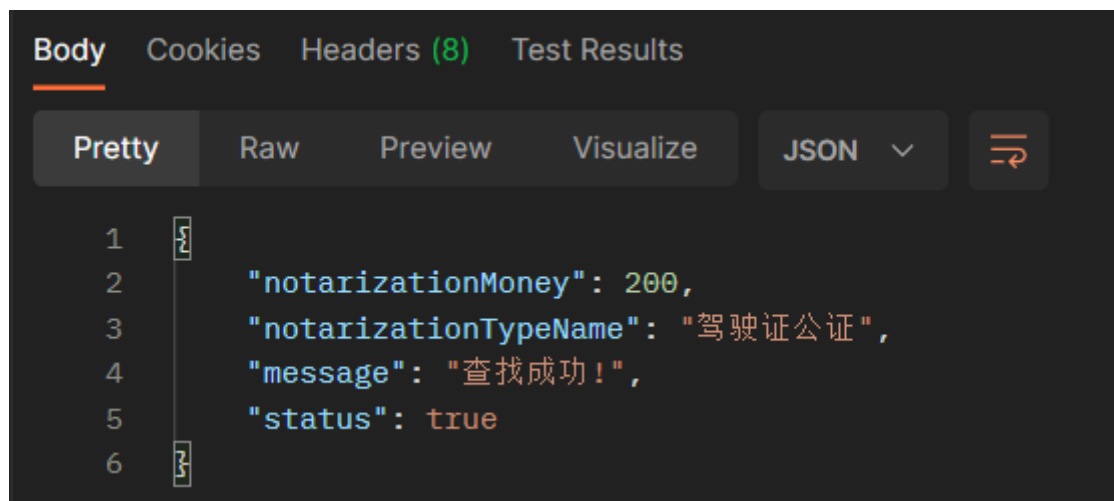
接口地址: /aut/notarTypeSelect

请求方式: POST

请求参数:



返回正确json示例:



接口地址: /user/regist

请求方式: POST

请求参数:

返回正确json示例:

返回错误json示例: