

Postman验库（落库检查）

1、什么是验库？

验库，也称为落库检查；

2、为什么需要验库？

断言时，关心两个方面，一个是接口的返回值，一个是对数据库的影响，这两方面都要正确；

验库就是对数据库中的数据进行确认，比如通过查询数据库证明实际结果确实存在，包括表中的行和列的检查。

3、什么时候需要验库？

当接口访问时，对数据库有影响时，需要验库；

比如查询的结果需要从数据库获取，我们可以对数据库进行数据确认；

比如注册新账户后，通过查询数据库证明新的记录已经添加；

通过Postman访问数据库，可以进行数据库的自动化控制，进行批量数据添加，从而实现测试数据的初始化。

4、采用什么工具？

node.js提供的工具，xmysql提供了很多restful接口，实现对数据库的增删改查操作。

5、xmysql使用步骤

(1) 前提：安装cnpm工具，并安装xmysql（已经提前安装，无需重复）

在真机中安装 cnpm（淘宝资源源）：

```
npm install cnpm -g --registry=https://registry.npm.taobao.org
```

```
cnpm -v
```

安装 xmysql：

```
cnpm install -g xmysql
```

(2) 启动xmysql服务器

在安装有Postman和xmysql的计算机上启动

打开cmd窗口：

```
xmysql -h 数据库服务器地址 -u 数据库用户名 -p 数据库密码 -d 数据库名
```

比如：数据库服务器IP地址为 192.168.198.134，可以执行以下命令，大家可以修改一下IP地址试试：

```
xmysql -h 192.168.198.133 -u root -p 123456 -d wa_test
```

执行效果如下：

```
D:\>xmysql -h 192.168.198.134 -u root -p 123456 -d wa_test
```

Generating REST APIs at the speed of your thought..

```
-----  
Database           : wa_test  
Number of Tables   : 4  
  
REST APIs Generated : 84  
  
Xmysql took        : 0.3 seconds  
API's base URL     : localhost:3000  
-----
```

(3) 使用Postman新建请求，对MySQL进行访问

访问增删改查的接口：

接口地址：http://localhost:3000/api/表名/

请求方法：post

请求类型：表单类型

请求参数：数据库的表列名

注意：在body的x-www-form-urlencoded处输入列名和 数据作为参数即可

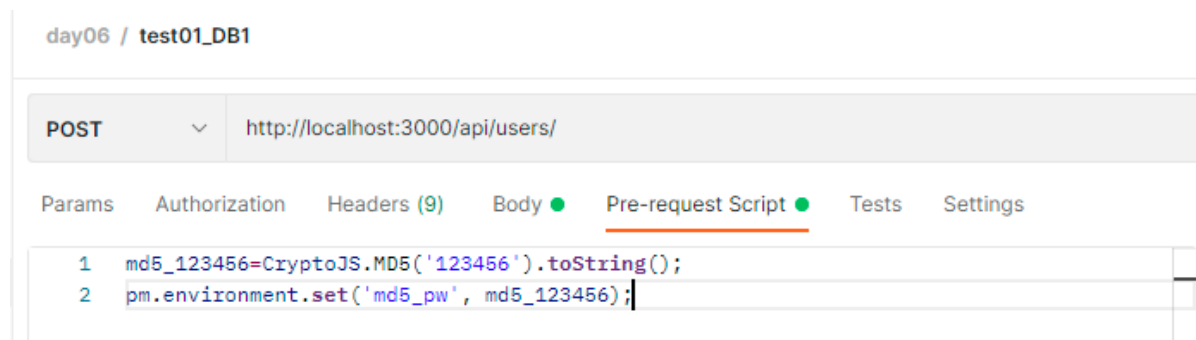
案例1：向users表中插入一条记录

技巧：密码需要经过md5加密

在pre-request-script中写代码：

```
md5_123456 = CryptoJS.MD5('123456').toString();  
  
pm.environment.set('md5_pw', md5_123456);
```

如图：



在Tests选项卡中添加

```
//向info表插入一条数据，id使用users生成的id
actual=JSON.parse(responseBody);
id=actual.insertId;
if(id!=null)
    pm.globals.set('id' , id); //插入成功
else
    pm.globals.set('id' , 0); //插入出错
```

配置请求地址和参数，发送请求，查看结果：

The screenshot shows a REST client interface with the following details:

- Request Method:** POST
- Request URL:** http://localhost:3000/api/users/
- Params:** id (4), username (test01), password ({{md5_pw}})
- Body:** JSON format, showing the response:


```
{
  "fieldCount": 0,
  "affectedRows": 1,
  "insertId": 4,
  "serverStatus": 2,
  "warningCount": 0,
  "message": "",
  "protocol41": true,
  "changedRows": 0
}
```
- Status:** 200 OK, Time: 153 ms, Size: 395 B

说明：关注响应体中的affectedRows（受影响的行数，表示实际向数据库的表中插入了几行数据）

查看一下数据库：

```
mysql> select * from users;
+----+-----+-----+-----+
| id | username | password | regtime |
+----+-----+-----+-----+
| 1 | admin | e10adc3949ba59abbe56e057f20f883e | 2020- 07- 15 15: 30: 47 |
| 2 | Dumb | 6013221752df892237d0fc99fd8a328f | 2020- 07- 15 15: 30: 47 |
| 3 | Superman | c6f42878ad2e794e6454d5894104675f | 2020- 07- 15 15: 30: 47 |
| 4 | test01 | e10adc3949ba59abbe56e057f20f883e | 2022- 05- 08 23: 23: 03 |
+----+-----+-----+-----+
4 rows in set (0.00 sec)
```

案例2：test02_查询users表所有行

Get方法：http://localhost:3000/api/users/

day06 / test02_查询users表所有行

GET http://localhost:3000/api/users/ Send

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION	Bulk Edit
Key	Value	Description	

Body Cookies Headers (8) Test Results Status: 200 OK Time: 13 ms Size: 717 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 1,
3   "username": "admin",
4   "password": "e10adc3949ba59abbe56e057f20f883e",
5   "regtime": "2020-07-15T07:30:47.000Z"
6 },
7 {
8   "id": 2,
9   "username": "Dumb",
10  "password": "6013221752df892237d0fc99fd8a328f",
11  "regtime": "2020-07-15T07:30:47.000Z"
12 },
13 }
```

案例3: test03_查询info表中所有行

Get方法: http://localhost:3000/api/info/

day06 / test03_查询info表中所有行

GET http://localhost:3000/api/info/ Send

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION	Bulk Edit
Key	Value	Description	

Body Cookies Headers (8) Test Results Status: 200 OK Time: 13 ms Size: 735 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 1,
3   "name": "管理员",
4   "gender": "男",
5   "birth": "1985-03-26T16:00:00.000Z",
6   "work_age": 10,
7   "major": "计算机科学与技术",
8   "introduce": "IT技术宅男, 此处省略400字"
9 },
10 {
11   "id": 2,
12   "name": "龙龙",
13   "gender": "女",
14 }
```

条件查询

get方法: `http://localhost:3000/api/表名?_where=(列名1,eq,值1)~and(列名2,eq,值2)`

注意: 逗号左右不能加空格

eq: equal, 相等

返回包含json数组的字符串

使用时, 需要把字符串转成json数组格式, 取其中的元素, 取其中的键的值

缺陷: 查询不到信息时会返回所有数据表中信息

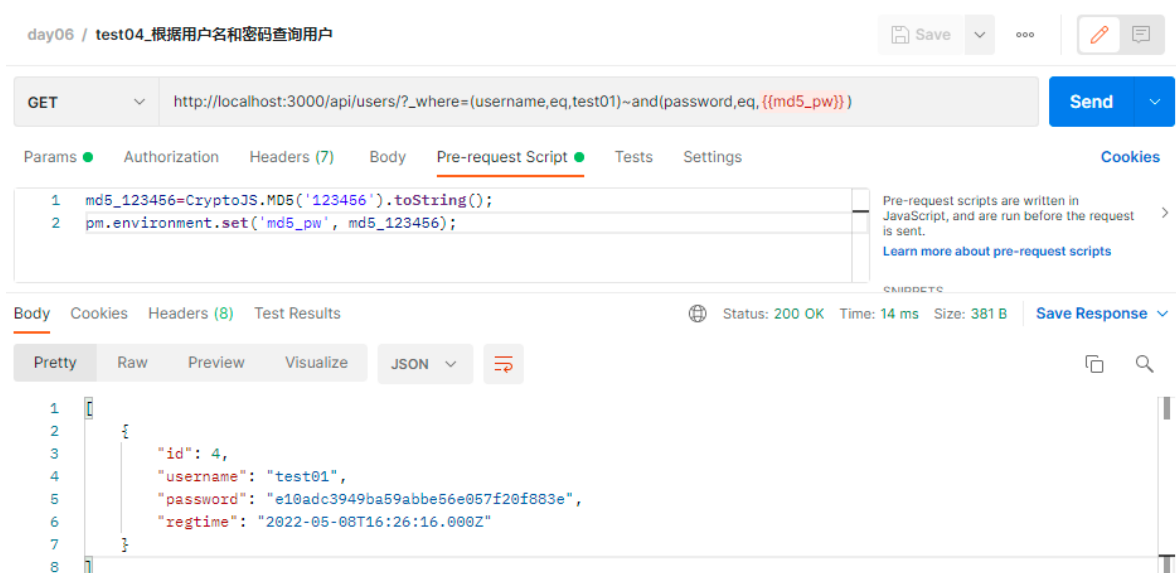
案例4: 查询用户名是test01且密码为123456的行 密码必须加密

请求名称: test04_根据用户名和密码查询用户

请求方法: Get

接口地址: [http://localhost:3000/api/users/?_where=\(username,eq,test01\)~and\(password,eq,{{md5_pw}}\)](http://localhost:3000/api/users/?_where=(username,eq,test01)~and(password,eq,{{md5_pw}}))

配置信息如下:



案例5: 查询info表中姓名为管理员的id号

请求名称: test05_查询info表中的id

请求方法: Get

请求地址: [http://localhost:3000/api/info/?_where=\(name,eq,管理员\)](http://localhost:3000/api/info/?_where=(name,eq,管理员))

day06 / test02_查询info表中的id

GET http://localhost:3000/api/info/?_where=(name,eq,管理员) Send

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

Query Params

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	_where	(name,eq,管理员)			
	Key	Value	Description		

Body Cookies Headers (8) Test Results Status: 200 OK Time: 12 ms Size: 446 B Save Response

Pretty Raw Preview Visualize JSON

```
1 [
2   {
3     "id": 1,
4     "name": "管理员",
5     "gender": "男",
6     "birth": "1985-03-26T16:00:00.000Z",
7     "work_age": 10,
8     "major": "计算机科学与技术",
9     "introduce": "IT技术宅男, 此处省略400字"
10  }
11 ]
```

在Tests选项卡中，为查询结果添加检查点：

day06 / test02_查询info表中的id

GET http://localhost:3000/api/info/?_where=(name,eq,管理员) Send

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

1 user_id = JSON.parse(responseBody)[0].id;
2 pm.globals.set('user_id', user_id);
3 console.log(user_id)

Test scripts are written in JavaScript, and are run after the response is received. [Learn more about tests scripts](#)

SNIPPETS

Body Cookies Headers (8) Test Results Status: 200 OK Time: 11 ms Size: 446 B Save Response

Pretty Raw Preview Visualize JSON

```
1 [
2   {
3     "id": 1,
4     "name": "管理员",
5     "gender": "男",
6     "birth": "1985-03-26T16:00:00.000Z",
7     "work_age": 10,
8     "major": "计算机科学与技术",
9     "introduce": "IT技术宅男, 此处省略400字"
10  }
11 ]
```

查询表的行数：

get方法: [http://localhost:3000/api/表名/count?_where=\(列名1,eq,值1\)~and\(列名2,eq,值2\)](http://localhost:3000/api/表名/count?_where=(列名1,eq,值1)~and(列名2,eq,值2))

案例6：查询用户表中总行数

请求名称: test06_查询user表的行数

请求方法: Get

请求地址: <http://localhost:3000/api/users/count>

day06 / test06_查询user表的行数

GET

http://localhost:3000/api/users/count

Send

Params

Authorization

Headers (7)

Body

Pre-request Script

Tests

Settings

Cookies

```

1 db_actual_rows = JSON.parse(responseBody)[0].no_of_rows;
2 tests[db_actual_rows] = true;

```

Test scripts are written in JavaScript, and are run after the response is received.
[Learn more about tests scripts](#)

Body

Cookies

Headers (8)

Test Results (1/1)

Status: 200 OK Time: 9 ms Size: 285 B

Save Response

Pretty

Raw

Preview

Visualize

JSON

```

1 [
2   {
3     "no_of_rows": 4
4   }
5 ]

```

案例7：查询不存在用户的id

请求名称：test07_查询不存在用户的id

请求方法：Get

请求地址：[{http://localhost:3000/api/users/?_where=\(username,eq,test00\)~and\(password,eq,{{md5_pw}}\)}](http://localhost:3000/api/users/?_where=(username,eq,test00)~and(password,eq,{{md5_pw}}))

day06 / test07_查询不存在用户的id

GET

http://localhost:3000/api/users/?_where=(username,eq,test00)~and(password,eq,{{md5_pw}})

Send

Params

Authorization

Headers (7)

Body

Pre-request Script

Tests

Settings

Coo

```

1 md5_123456=CryptoJS.MD5('123456').toString();
2 pm.environment.set('md5_pw', md5_123456);

```

Pre-request scripts are written in JavaScript, and are run before the request is sent.
[Learn more about pre-request scripts](#)

Body

Cookies

Headers (8)

Test Results (1/1)

Status: 200 OK Time: 10 ms Size: 267 B

Save Respon

Pretty

Raw

Preview

Visualize

JSON

```

1 []

```

添加检查点：

day06 / test07_查询不存在用户的id

GET

http://localhost:3000/api/users/?_where=(username,eq,test00)~and(password,eq,{{md5_pw}})

Send

Params

Authorization

Headers (7)

Body

Pre-request Script

Tests

Settings

Cool

```

1 actual = JSON.parse(responseBody);
2 if(actual.length > 0) {
3   user_id = actual[0].id;
4 } else {
5   user_id = 0;
6 }
7 tests[user_id] = true;

```

Test scripts are written in JavaScript, and are run after the response is received.
[Learn more about tests scripts](#)

SNIPPETS

[Get an environment variable](#)

[Get a global variable](#)

Body

Cookies

Headers (8)

Test Results (1/1)

Status: 200 OK Time: 10 ms Size: 267 B

Save Respon

All

Passed

Skipped

Failed

PASS 0

xmysql删除数据：

在postman中操作
接口地址: <http://localhost:3000/api/表名/主键值>
请求方法: delete
请求参数: 无

案例8: 删除users表中的一条数据

请求名称: test08_删除users表数据

请求方法: Delete

请求地址: <http://localhost:3000/api/users/4>

注意:关注响应体中的affectedRows (受影响的行数, 表示实际从数据库的表中删除了几行数据)

返回的结果:

```
{
  "fieldCount": 0,
  "affectedRows": 1,
  "insertId": 0,
  "serverStatus": 2,
  "warningCount": 0,
  "message": "",
  "protocol41": true,
  "changedRows": 0
}
```

day06 / test08_删除users表数据

DELETE <http://localhost:3000/api/users/4> Send

Params Authorization Headers (7) Body Pre-request Script Tests Settings

```
1 actual=JSON.parse(responseBody);
2 rows=actual.affectedRows;
3 if(rows!=0)
4   tests['删除了: ' + rows + '行'] = true; //删除成功
5 else
6   tests['未删除'] = true; //未删除
```

Test scripts are written in JavaScript, and are run after the response is received.
[Learn more about tests scripts](#)

SNIPPETS
[Get an environment variable](#)
[Get a global variable](#)
[Get a variable](#)

Body Cookies Headers (8) Test Results (1/1) Status: 200 OK Time: 8 ms Size: 395 B Save Respon

All Passed Skipped Failed

PASS 删除了: 1行

案例8: 修改users表中的一条数据

请求名称: test09_修改users表数据

请求方法: Patch

请求地址: <http://localhost:3000/api/users/4>

修改的数据: 设置在Body选项卡中, 选择x-www-form-urlencoded模式

day06 / test09_修改users表数据

Save

...

PATCH

http://localhost:3000/api/users/4

Send

Params

Authorization

Headers (9)

Body

Pre-request Script

Tests

Settings

Cookies

☐ none

☐ form-data

☒ x-www-form-urlencoded

☐ raw

☐ binary

☐ GraphQL

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	username	test001			
	Key	Value	Description		

Body

Cookies

Headers (8)

Test Results

Status: 200 OK

Time: 89 ms

Size: 436 B

Save Response

Pretty

Raw

Preview

Visualize

JSON

```
1  {
2    "fieldCount": 0,
3    "affectedRows": 1,
4    "insertId": 0,
5    "serverStatus": 2,
6    "warningCount": 0,
7    "message": "(Rows matched: 1  Changed: 1  Warnings: 0)",
8    "protocol41": true,
9    "changedRows": 1
10 }
```