PostMan的使用

本章大纲

- PostMan介绍
- JSON介绍
- 使用PostMan测试GET接口
- 使用PostMan测试POST接口测试
- 面向场景的接口测试

PostMan介绍

使用PostMan工具测试

它是一款功能强大的网页调试与发送网页HTTP请求的工具。

PostMan能够发送任何类型的HTTP请求(GET, HEAD, POST,PUT..)，附带任何数量的参数和HTTP headers。支持不同的认证机制（basic, digest,OAuth，接收到的响应语法高亮（HTML，JSON或XML）。

PostMan的相关资料

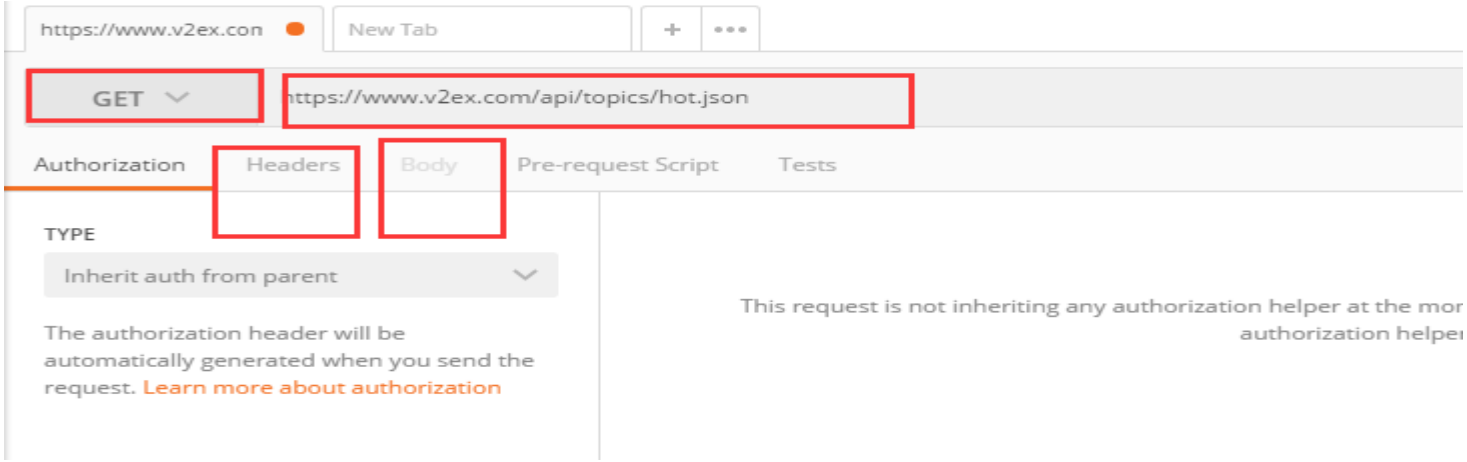
- 官网及下载地址：
  https://www.getpostman.com/
- 官方文档
  https://www.getpostman.com/docs/
- 社区：
  https://www.getpostman.com/community

Postman介绍

- 主要功能包括：
- 模拟各种http requests
- Collection功能
- 人性化的Response整理
- 内置测试脚本管理
- 设定变量与环境

PostMan简单介绍

- HTTP请求： method， url ， headers， body



PostMan简单介绍

- HTTP请求的body分类

  使用Content-Type来指定不同格式的请求信息，在请求头里设置，默认为text/html

Content-Type	Postman中body格式	备注
multipart/form-data	form-data	将表单的数据处理为一条消息，由boundary隔离，既可以上传多个文件（包括二进制文件），也可以上传键值对
application/x-www-from-urlencoded	x-www-from-urlencoded	将表单内的数据转换为键值对，比如，name=tom&age = 23
text/plain	raw	可以上传任意格式的文本，可以上传text、JSON、xml、html等
application/octet-stream	binary	只可以上传二进制数据，一次只能上传一个文件

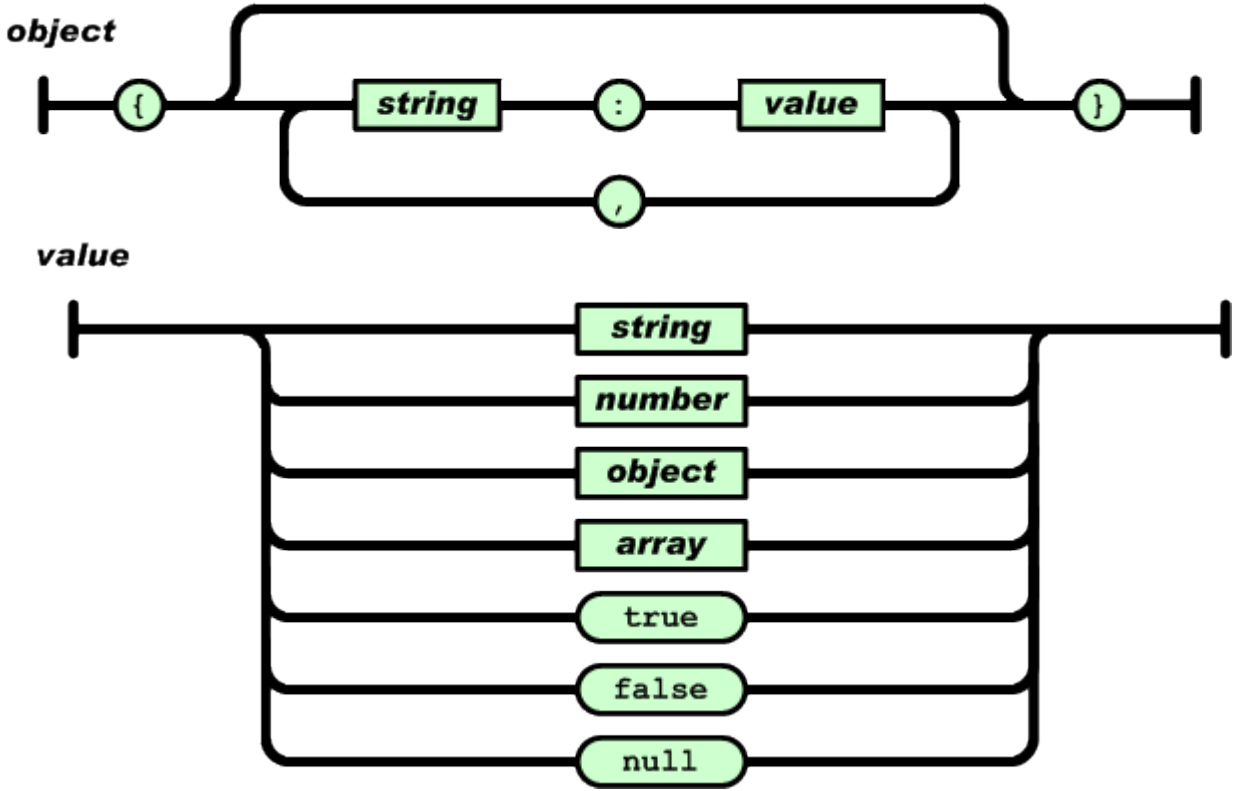
本章大纲

- PostMan介绍
- JSON介绍
- 使用PostMan测试GET接口
- 使用PostMan测试POST接口测试
- 面向场景的接口测试

JSON介绍

- JSON（JavaScript Object Notation ）轻量级的数据交换语言，以文字为基础，且易于阅读。
  - 对象（object）:一个对象以“{”开始，并以“}”结束。每个对象包含一系列排序的名称/值对，每个名称/值对之间使用“,”区分。
  - 名称/值（collection）：名称和值之间使用“:”隔开，一般的形式是：{key1:value,key2:value2}

JSON介绍


object
{
string
:
value
}
,
value
string
number
object
array
true
false
null

JSON介绍

- JSON数据类型

数值（整数或浮点数）

示例：”age”:86

”price”:123.78

“temperature “:-3.5

“speed_of_light”:3.12e12

JSON介绍

- JSON数据类型

null（空值）

示例：”name”:”tom”

”name”:null

”name”:””

注意：””表示的是空字符串，而null表示的才是空值

JSON介绍

- JSON数据类型

逻辑值（true或false）

示例：”student”:true

JSON介绍

- JSON数据类型

对象（在花括号中）

示例：”address”:{

“line”: 123 yuhua road”,

“city”:”shijiazhuang”,

“postalCode”:”051220”,

“country”:”China”

}

JSON介绍

- JSON数据类型

数组（在方括号中）

示例："employees": [

{ "firstName":"Bill" , "lastName":"Gates" },

{ "firstName":"George" , "lastName":"Bush" },

{ "firstName":"Thomas" , "lastName":"Carter" }

]

JSON语法规则

数据在键值对中	{“name”:”tom”}
数据由逗号分隔	{“name”:”tom”, ”phone”:”13899008877”}
方括号保存数组	{“name”:”tom”, ”phone”:”13899008877”, “Education”:[{“school”:”AAA”,”profession”:”BB”}]}

花括号保存对象	{“name”:”tom”,”phone”:”13899008877”,
	”address”:{
	“line”: 123 yuhua road”,
	“city”:”shijiazhuang”,
	“postalCode”:”051220”,
	“country”:”China”
	}
	}

JSON举例

```
var employee={
        "name":"tom",
        "phone":"13899008877",
        "address":{
                "line": 123 yuhua road",
                "city":"shijiazhuang",
                "postalCode":"051220",
                        }
        "education":[
        {
        "school":"河北师范大学“,
        "info":[
            {"diploma":"本科","profession":"软件工程":"grade":2015},
            {“diploma”:“本科”,“profession”:“数学":"grade":2015}
        ]
        },
        {
        “school”:“北京大学“,
        "info":{"diploma":"研究生","profession":"软件工程":"grade":2018}
  }
        ]
}
```

JSON举例

在JavaScript中，使用下面这样的代码访问数据：

- ✓ 读取name值：employee.name
- ✓ 读取phone值：employee.phone
- ✓ 读取address对象的city值：employee.address.city

返回的内容是：shijiazhuang

JSON举例

在JavaScript中，使用下面这样的代码访问数据：

读取education数组

- ✓ 第一个school值：employee.education[0].school
- ✓ 第二个school值：employee.education[1].school

返回的内容是：北京大学

JSON举例

在JavaScript中，使用下面这样的代码访问数据：

读取education数组

- ✓ 第一个元素的第一个diploma值：

  employee.education[0].info[0].diploma

- ✓ 第二个元素的第一个diploma值：

  employee.education[1].info[0].diploma

返回的内容是：研究生

JSON举例

在JavaScript中，使用下面这样的代码修改数据：

- ✓ 修改名字：

  employee.name=“jerry”

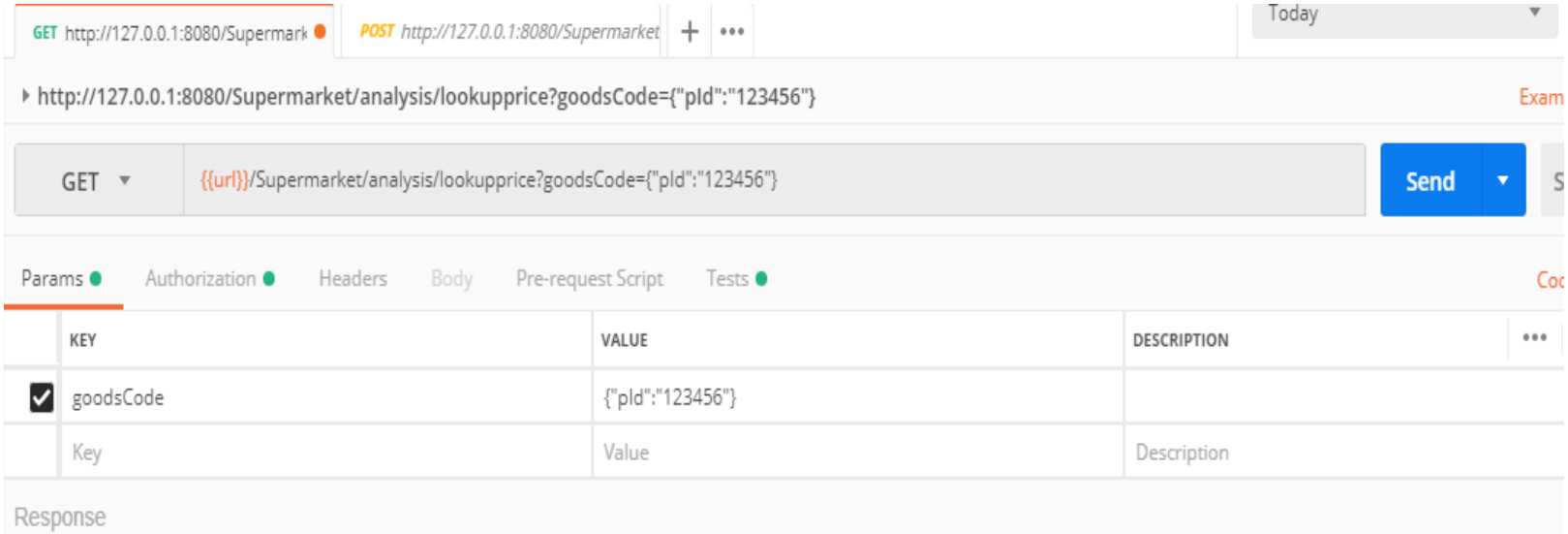
- ✓ 修改专业：

  employee.education[0].info[0].profession=“数学”

本章大纲

- PostMan介绍
- JSON介绍
- 使用PostMan测试GET接口
- 使用PostMan测试POST接口测试
- 面向场景的接口测试

GET请求接口测试


GET http://127.0.0.1:8080/Supermark
POST http://127.0.0.1:8080/Supermarket
Today
http://127.0.0.1:8080/Supermarket/analysis/lookupprice?goodsCode={"pId":"123456"}
Exam
GET
{{url}}/Supermarket/analysis/lookupprice?goodsCode={"pId":"123456"}
Send
Params
Authorization
Headers
Body
Pre-request Script
Tests
KEY
VALUE
DESCRIPTION
goodsCode
{"pId":"123456"}
Key
Value
Description
Response

本章大纲

- PostMan介绍
- JSON介绍
- 使用PostMan测试GET接口
- 使用PostMan测试POST接口测试
- 面向场景的接口测试

HTTP请求格式-POST方法

请求（Request）

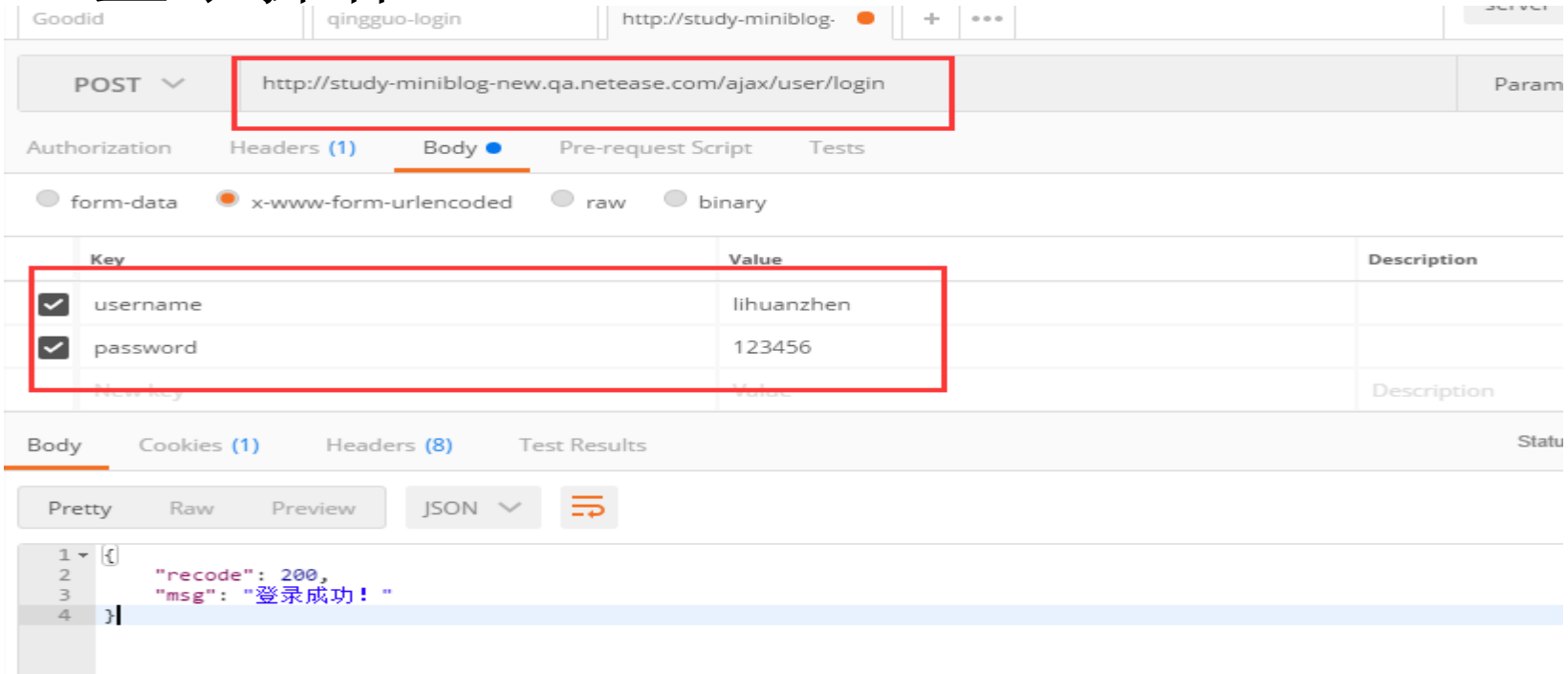
请求行	POST **/xinhu/index.php?a=check&m=login** HTTP/1.1
请求头	Accept： ext/html,application/xhtml+xml Accept-Encoding: gzip, deflate Accept-Language:zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3 Connection:keep-alive Host： localhost:8032
请求正文	adminuser=YWRtaW4%3A&=123456&rempass=0&button=&jmpass=false&device=1517376146707&adminpass=MTIzNDU2

响应（Response）

状态行	**HTTP1.1 200 OK**
响应头	Connection： Keep-Alive Content-Encoding： gzip Content-Length： 1234 Content-Type： text/html;charset=utf-8 Date： Mon, 05 Feb 2018 02:43:40 GMT
响应正文	{"success":true,"face":"http:\/\/localhost:8032\/xinhu\/upload\/face\/1.jpg"}

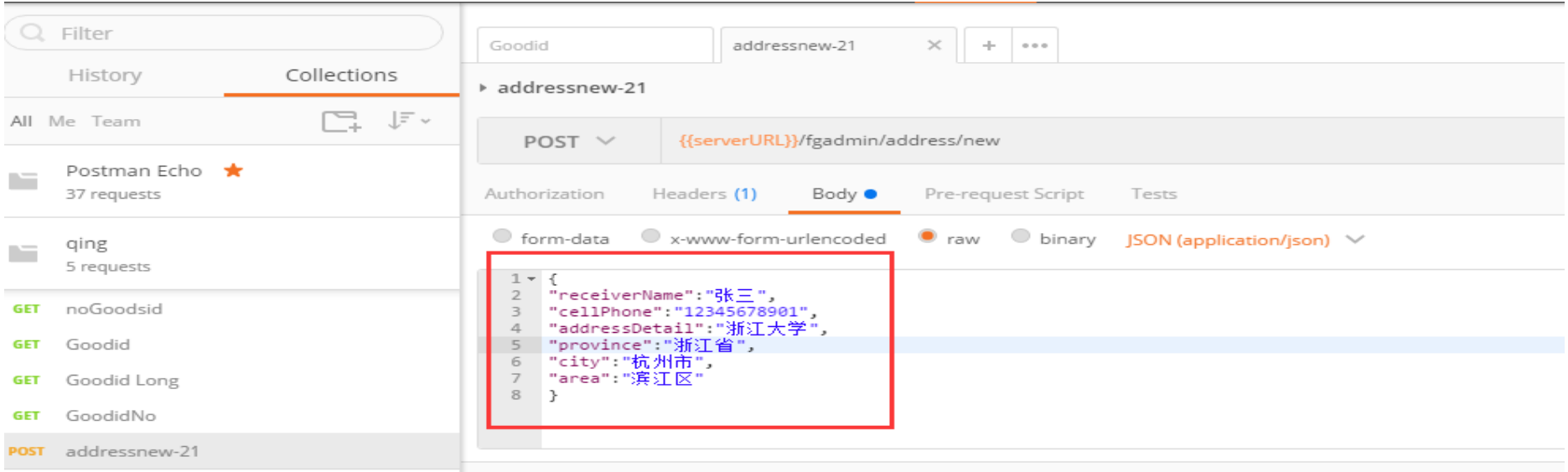
POST请求接口测试

- 登录操作



POST请求接口测试

- 从商品详情登录后添加地址



POST请求接口测试

- 什么是cookie

  跟踪会话，弥补HTTP协议无状态的不足

POST请求接口测试

- 如何带上用户Cookie信息

  发送一个请求时，可能会要求发送这个请求前，先做用户认证（登录）

本章大纲

- PostMan介绍
- JSON介绍
- 使用PostMan测试GET接口
- 使用PostMan测试POST接口测试
- 面向场景的接口测试

面向场景的接口测试

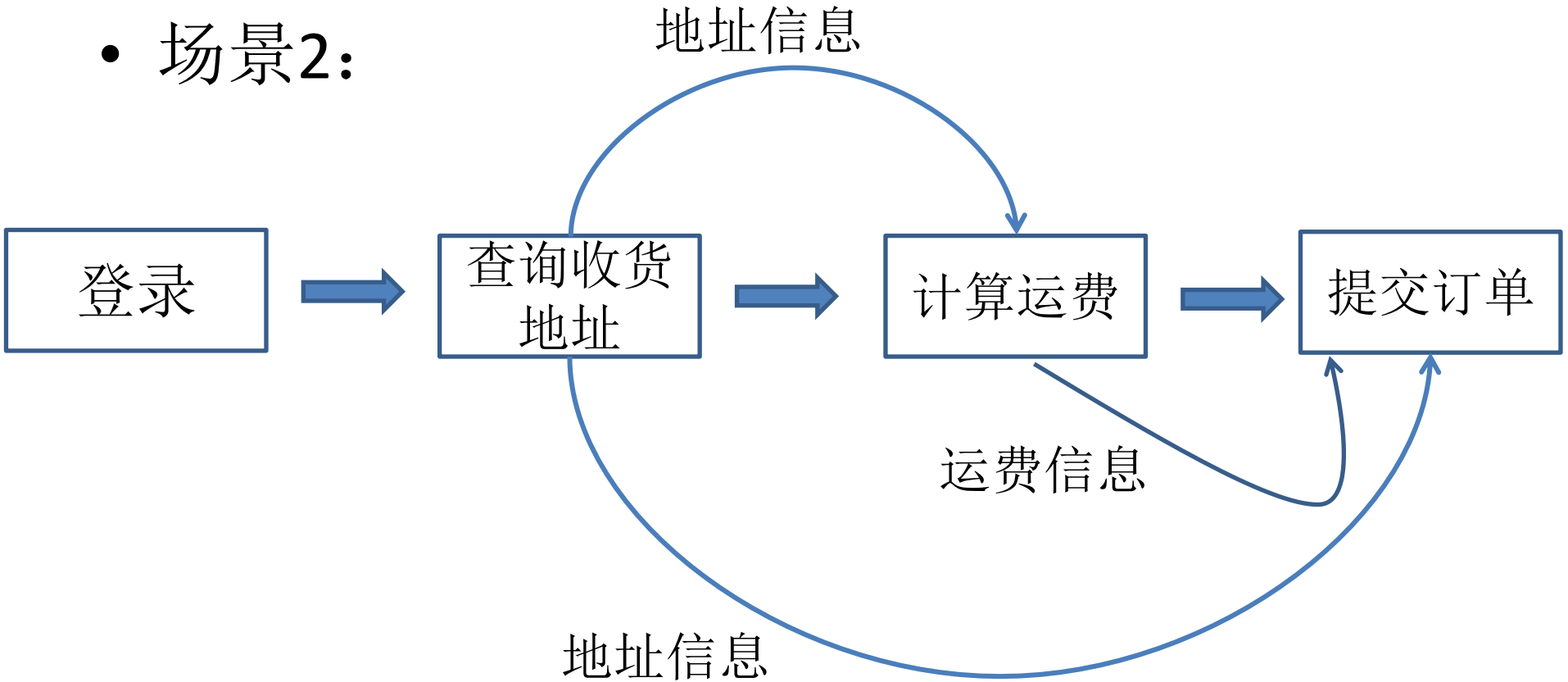
- 场景2：

  完整下单流程(登录、有收货地址)

- 场景3：

  完整下单流程(登录、无收货地址)

- 场景2：



- 场景3：