

Title	API Automation
Created Date	19 Jan. 2017
Created By	Vipul Jain
Reviewed By	Shankar Garg
Current Version	1.0

## API Automation

### Rest-Assured Framework:

For validating various fields from response we use Rest Assured Framework. It follows Behavior Driven Development i.e.: given when then approach.

1. **Given** part describes the state before you begin the behavior that you will specify in your scenario. In this part you will pass query parameters, cookie map, content type (XML or JSON) etc.
2. **When** part describes the behavior that you specify, like hitting a url (API).
3. **Then** section describes the changes you expect due to the specified behavior usually include the assertion part here.

Example of behavior driven development:

```
given()  
    .cookies(cookieMap)  
    .contentType("application/json")  
    .queryParams("json", postData)  
.when()  
.get(url);
```

**\*\*Note:** Use the following dependency in your Maven project's POM for including Rest Assured library.

```
<dependency>  
    <groupId>com.jayway.restassured</groupId>  
    <artifactId>rest-assured</artifactId>  
    <version>2.9.0</version>  
</dependency>
```

Rest assured works with both XML and JSON.

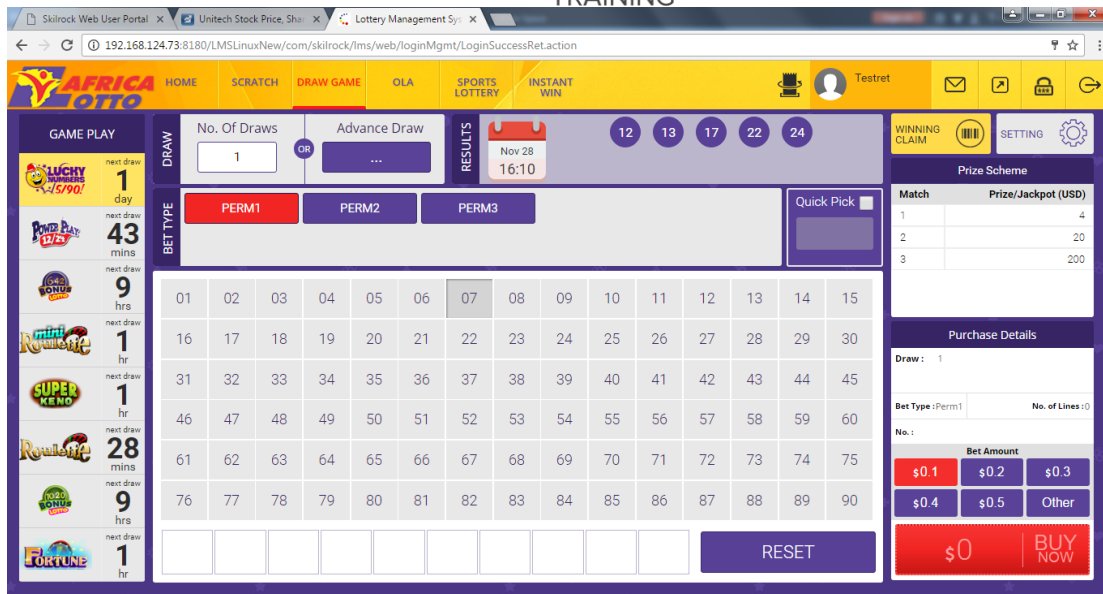
### Hybrid Framework :

This involves requesting APIs, accessing the response and then validating this response from the UI

### Retrieving the API URLs through Charles proxy:

Navigate to the LMS site and observe the Charles proxy

Login API:



**AFRICA 1070**

HOME | SCRATCH | DRAW GAME | OLA | SPORTS LOTTERY | INSTANT WIN

Testret

**GAME PLAY**

- LUCKY 5/90! next draw 1 day
- POWERBALL next draw 43 mins
- Mega Millions next draw 9 hrs
- Rolling 28 next draw 1 hr
- SUPER KENO next draw 1 hr
- Rolling 28 next draw 28 mins
- LIVE KENO next draw 9 hrs
- FORTUNE next draw 1 hr

**DRAW**

No. Of Draws: 1 OR Advance Draw

**RESULTS**

Nov 28 16:10

12 13 17 22 24

**BET TYPE**

PERM1 PERM2 PERM3 Quick Pick

01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40	41	42	43	44	45
46	47	48	49	50	51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70	71	72	73	74	75
76	77	78	79	80	81	82	83	84	85	86	87	88	89	90

RESET

**Prize Scheme**

Match	Prize/Jackpot (USD)
1	4
2	20
3	200

**Purchase Details**

Draw : 1

Bet Type: Perm1 No. of Lines: 10

No. :

Bet Amount

\$0.1 \$0.2 \$0.3 \$0.4 \$0.5 Other

\$0 BUY NOW

Charles: 40.1 - Session 1\*

File Edit View Proxy Tools Window Help

Structure Sequence

- http://192.168.124.73:8180
  - LMSLinuxNew
    - com
      - skilrock
        - lms
          - web
            - loginMgmt
              - newLogin.action?request\_locale=en&oldHttp=00
              - afterLogin.action
              - Login.action?data=00B4E4BAA7329AD132EABE9E
              - LoginSuccessRet.action
              - Header2.jsp
              - LoginSuccessRet.action
              - Header2.jsp
              - fetchNamenLimit.action?14816888245000B4E4B4
              - fetchNamenLimit.action?1481688824624000B4E4B4
              - fetchNamenLimit.action?148168882464000B4E4B4
              - fetchNamenLimit.action?1481688824654000B4E4B4
              - openSLEPage.action?userType=RETAILER
              - Logout.action
              - showLogin.action
              - login.jsp
              - newLogin.action?request\_locale=en&oldHttp=00
              - afterLogin.action
              - Login.action?data=00B4E4BAA7329AD132EABE9E
              - LoginSuccessRet.action
              - Header2.jsp
              - LoginSuccessRet.action
              - Header2.jsp
              - fetchNamenLimit.action?1481688971608000B4E4B4
              - fetchNamenLimit.action?1481688971620000B4E4B4
              - fetchNamenLimit.action?1481688971804000B4E4B4

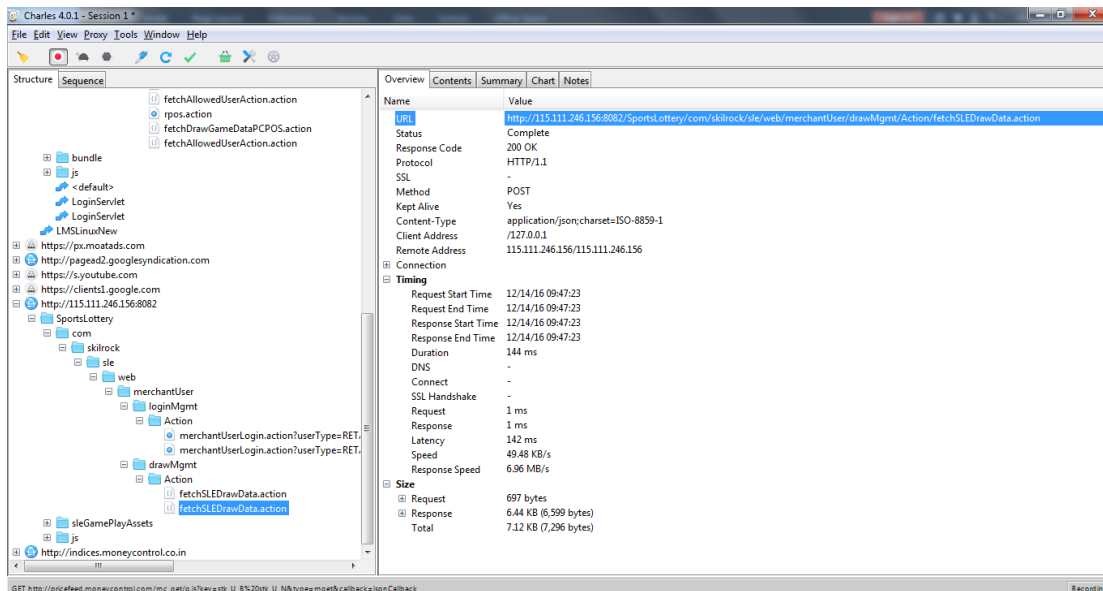
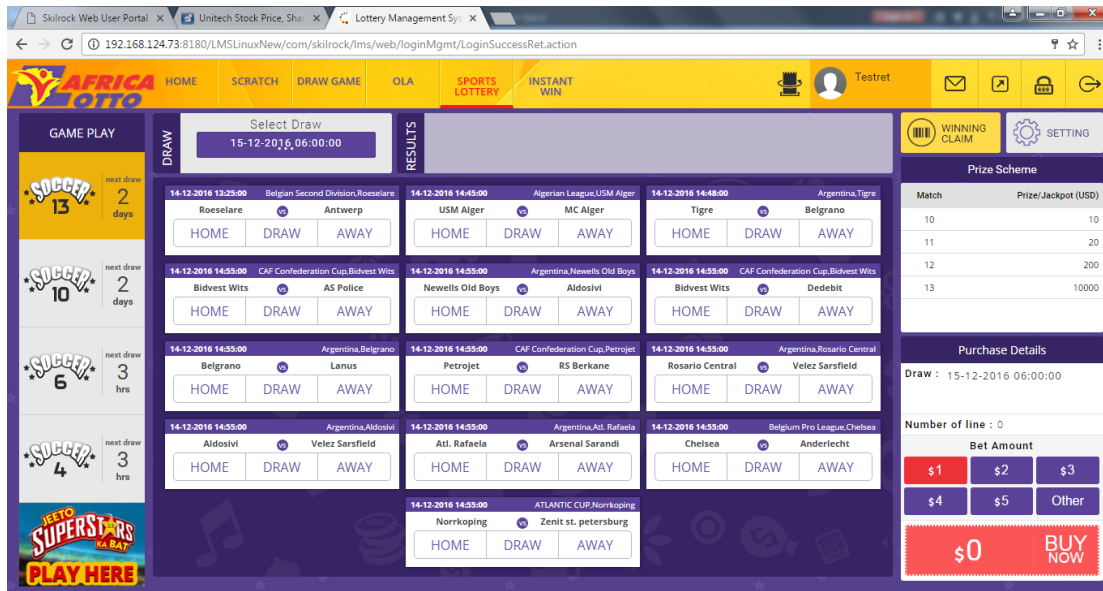
Overview Contents Summary Chart Notes

Name	Value
URL	http://192.168.124.73:8180/LMSLinuxNew/com/skilrock/lms/web/loginMgmt/newLogin.action?request_locale=en&oldHttp=00B4E4BAA7329AD132EABE9E
Status	Complete
Response Code	200 OK
Protocol	HTTP/1.1
SSL	-
Method	GET
Kept Alive	Yes
Content-Type	text/html; charset=UTF-8
Client Address	/127.0.0.1
Remote Address	192.168.124.73/192.168.124.73
Connection	-
Timing	-
Request Start Time	12/14/16 09:43:37
Request End Time	12/14/16 09:43:37
Response Start Time	12/14/16 09:43:37
Response End Time	12/14/16 09:43:37
Duration	26 ms
DNS	-
Connect	-
SSL Handshake	-
Request	1 ms
Response	15 ms
Latency	10 ms
Speed	255.82 KB/s
Response Speed	443.42 KB/s
Size	-
Request	601 bytes
Response	6.06 KB (6,210 bytes)
Total	6.65 KB (6,811 bytes)

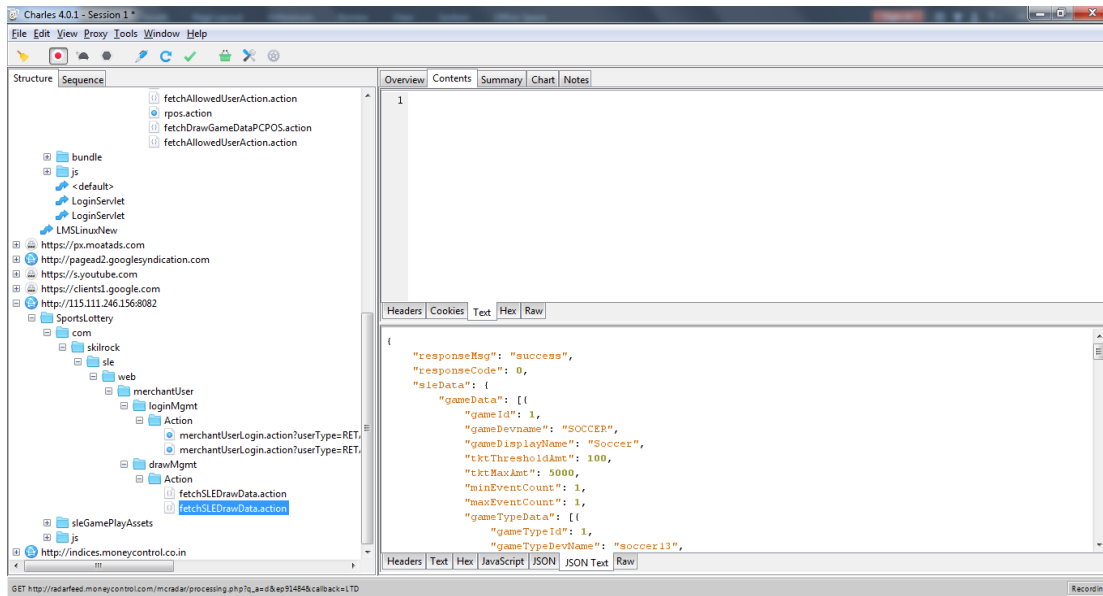
GET http://pricefeed.moneycontrol.com/mc\_get/g.js?key=stk\_U\_8%20stk\_U\_Nbtype=mgntcallback=jsonCallback

Recording

## Fetch SLE API:



## Checking Response from Fetch SLE Draw Data under text tab in Charles Proxy :



## Accessing some of the data from the API response:

Suppose we want to access value for a field called ticketNo in the response which is present under another field with key tktData

We can retrieve the ticketNo with the following code:

```
String temp=response.jsonPath().get("tktData.ticketNo").toString();
```

References for JSONPath: <http://goessner.net/articles/JsonPath/>

## Hitting the API and encapsulating the response :

Minimum requirement for generating a response from an API is to pass a queryParameter which we accomplish through the following statement:

```
response=given()
```

```

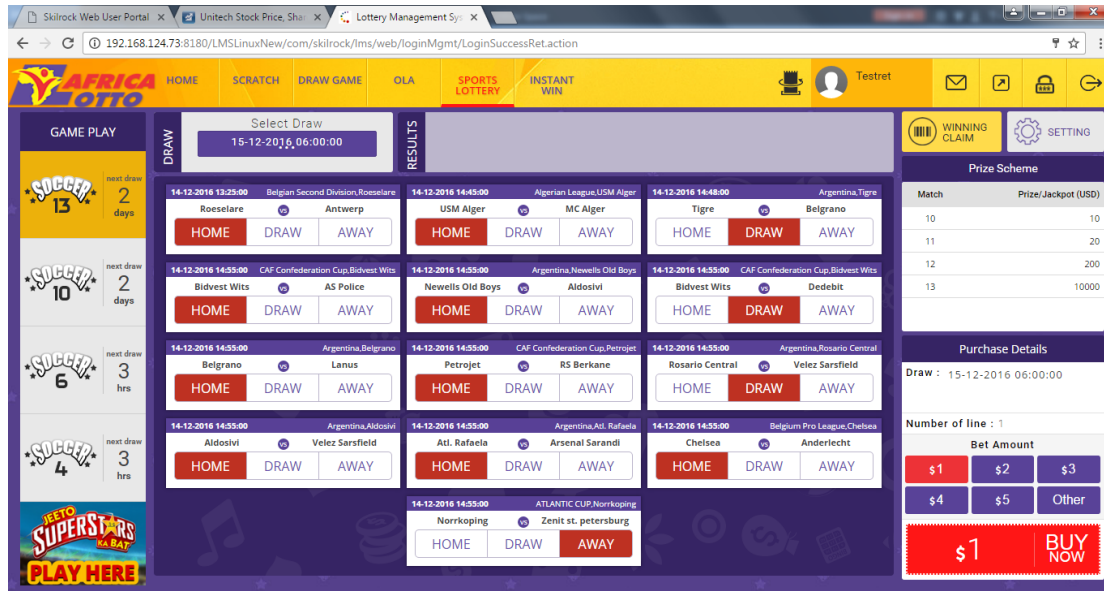
    .contentType("application/json")
    .queryParams("requestData", inputData)
    .get(url1);

```

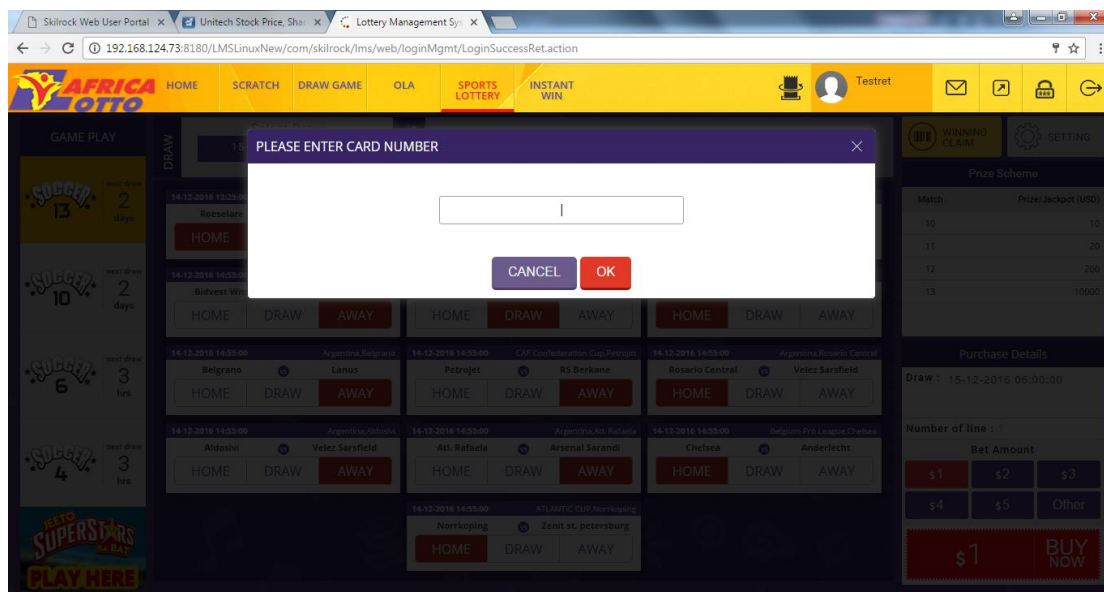
Where requestData is the key name for the request parameter passed before hitting the API. This requestData can be retrieved from charles proxy

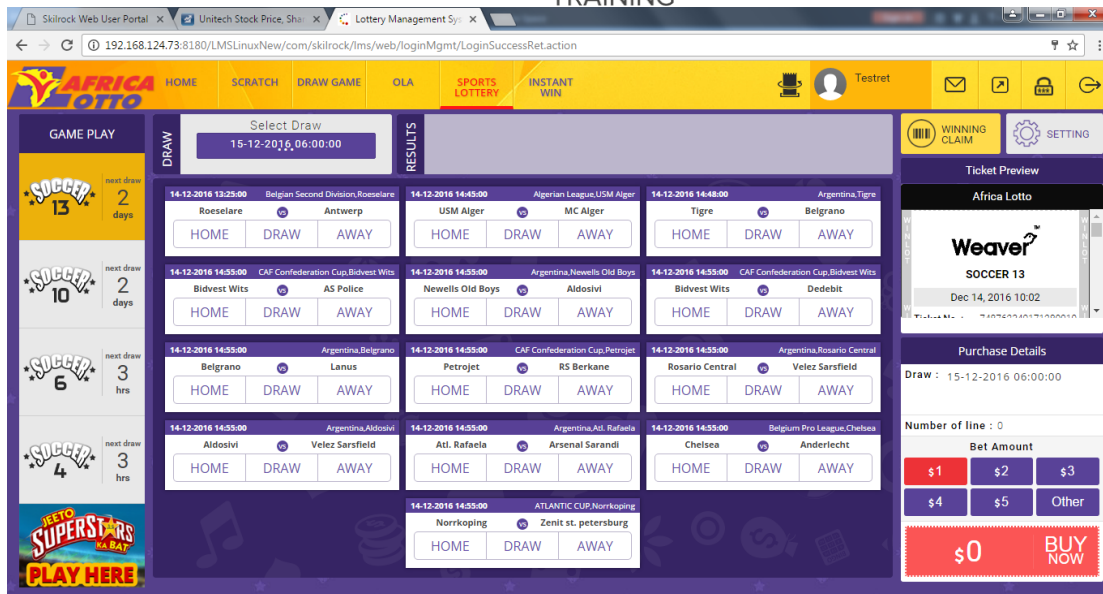
## Steps to see how requestData(in json form) looks like in charles :

1. Select at least one games from all the 13 events present under Soccer13

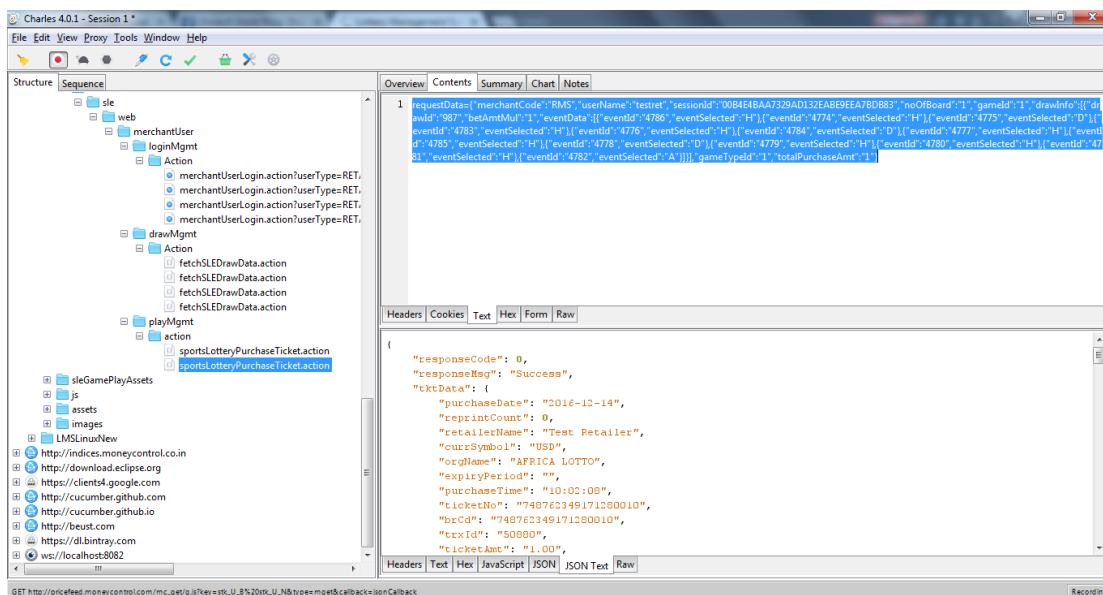


Perform a sale by clicking the buy button





Check charles proxy to see which API got hit for performing soccer 13 sale:



The above highlighted portion represents the requestData

requestData={"merchantCode":"RMS","userName":"testret","sessionId":"00B4E4BAA7329AD132EABE9EEA7BDB83","noOfBoard":"1","gameId":"1","drawInfo":[{"drawId":"987","betAmtMul":"1","eventData":[{"eventId":"4786","eventSelected":"H"}, {"eventId":"4774","eventSelected":"H"}, {"eventId":"4775","eventSelected":"D"}, {"eventId":"4783","eventSelected":"H"}, {"eventId":"4776","eventSelected":"H"}, {"eventId":"4784","eventSelected":"D"}, {"eventId":"4777","eventSelected":"H"}, {"eventId":"4785","eventSelected":"H"}, {"eventId":"4778","eventSelected":"D"}, {"eventId":"4779","eventSelected":"H"}, {"eventId":"4780","eventSelected":"H"}, {"eventId":"4781","eventSelected":"H"}, {"eventId":"4782","eventSelected":"A"}]}],"game1psd":"1","totalPurchaseAmt":"1"}

```
77","eventSelected":"H"}, {"eventId":"4785","eventSelected":"H"}, {"eventId":"4778","eventSelected":"D"}, {"eventId":"4779","eventSelected":"H"}, {"eventId":"4780","eventSelected":"H"}, {"eventId":"4781","eventSelected":"H"}, {"eventId":"4782","eventSelected":"A"}]]], "gameTypeId":"1", "totalPurchaseAmt":"1"
}
```

This request data can be changed if we want to change the input/query parameter being passed to the API.

### **Another Example of Retrieving/Extracting Data from a Response:**

Sample Response:

```
{
  "isSuccess": true,
    "refundAmount": 2,
    "ticketNumber": "349962351171280290",
    "gameTypeName": "Soccer 13",
    "Balance": "84953.12",
    "mode": "Cancel",
    "cancelDateTime": "2016-12-16 12:12:11.486",
    "orgName": "AFRICA LOTTO",
    "currSymbol": "USD"
}
```

1.Retrieving organization's name:

```
response.jsonPath().get("orgName").toString()
```

2.Retrieving currency symbol value:

```
response.jsonPath().get("currSymbol").toString()
```

### **Another Example of Retrieving/Extracting Data from a Response:**

Sample Response:

```
{
  "responseCode": 0,
  "responseMsg": "Success",
  "tktdData": {
    "purchaseDate": "2016-12-16",
    "reprintCount": 0,
    "retailerName": "Test Retailer",
  }
}
```



```
"currSymbol": "USD",  
"orgName": "AFRICA LOTTO"  
}  
}  
1.Retrieving reprint count:  
response.jsonPath().get("ticketData.reprintCount").toString()
```

### **Steps to getting a response:**

1. Make sure the cookies, content type and input data(query parameter) is mentioned in the given() part

Eg :

given()

.contentType("application/json")

.queryParams(key,val)

.cookies(cookie map object)

2. After mentioning the given part hit the url of the api as below

.when()

.post(url)

### **Maintaining session via cookies:**

\* save the cookie during the authentication process to maintain same session throughout a given flow which requires the same session to be maintained.

Map<String,String> cookieObject=response.getCookies()

\* the cookie map returned in the above step can be passed as input within the given() clause/method of the rest assured to maintain a given session.

### **API AUTOMATION:**

This involves validating the response from the API against the Database

**\*\*Note:** All the methods hitting APIs are present under PCPOSApi.java file under src/main/java/apis

**\*\*Note:** All the methods involving validating response against the data from the Database can be found under ApiCommonValidationPage.java

**\*\*Note:** For all other games and bet the same above flow

**[\\*\\*Refer to Rest Assured](#)**

