ISSN (Print) : 2320 – 9798
ISSN (Online): 2320 – 9801

International Journal of Innovative Research in Computer and Communication Engineering
*Vol. 1, Issue 1, March 2013*

# Mock Manager Tool

Priyanka Bansal[1], Mamta Vadhariya[2], Apurva Narasimhan[3], Ankit Me hta[4] ,Mrs.Shweta Tiwaskar[5]

Student, Dept.of Computer Engineering, V.I.I.T College, Pune-48, India[1,2,3,4]

Associate Professor, Dept.of Computer Engineering, V.I.I.T College, Pune-48, India[5]

**ABSTRACT***:* We implemented "THE MOCK MANAGER TOOL" which deals with the problems stated below and overcome the difficulties. Developers face difficulty when it comes to develop and test an application when there is no actual data available and they are asked to test it with mock data. With mock people try to replicate the Data from database and data from the Web Services.And from many other sources which is time consuming. The practice till now is to prepare mock through java mock class or property file or XML structured text file that developer needs to develop or maintain manually, and depends upon the other team members may be working at different location or in different shifts to validate its consistency. The various Issues are that many time while developing or defect fixing, though data is available on the production server, but developers or testers are requested not to trigger too many queries (As each query execution and the time it is taking is charged).So there is need to have an application that interrupts the request from the user, trigger it at the backend and cache the result so that when the user makes same request again, it shouldn't result in the execution of query on database. To solve this issue, we developed an application which will improve the response time of the dummy data required during developing and testing phase by dynamic caching [8]the data in a <key-value> pair once it is retrieved from the database and next time retrieving it using the <key-value> map if it is required again. Also facilities to provide the data in the required format which tester or developer wants like XML to JSON either for databases or web services. Also the response time is consecutively increased.Additionally, the data required by the tester is got from the cache or database and can be directly imported in the method for which testing is to be carried out and provide the result. Thus as a conclusion, all the above facilities is proposed through the usage of Mock Manager.

**Keywords**: Annotation, Reflection, Cache , Dummy Data

## I. INTRODUCTION

The paper proposes a tool named " Mock Manager" .As the major problem is availability of data for testers during testing cum development phase, as at times a query hit at the server located at remote location ,takes hours and days to give response. Presently, the traditional practice was is to prepare mock through java mock class or property file or XML structured text file that developer needs to develop or maintain manually, and depends upon the other team members may be working at different location or in different shifts to validate its consistency.

Studying this in detail, it was found that few of the Issues faced while developing application are as stated ahead. Many times ,while developing or defect fixing though data is available on the production server but requested not to trigger to many queries ( As each query execution and the time it is taking is charged).Thus, there arouses a need to have an application which is named as "MOCK MANAGER" that will interrupt the request from the user , trigger it at the backend and cache the result so that in case the user makes the same request again it shouldn't result in the execution of query on database .Another issue was , Validation of mock requires the availability both the developer and the other member who has knowledge of the data structure. Thus, the need emerged to have an application that can be managed through remote system so that even if one member is not available other member can access it and can validate the data and its structure.

To suffice all the need as stated above, this paper proposes the tool MOCK MANAGER with features like:-
1. Retrieval of data with faster response time
2. Availability of data in the format required by the tester
3. Direct execution of the method by reflecting the values in the method call
4. Provision of query forming automatically without the tester having deep knowledge of database

## II. PROPOSED SYSTEM

The Mock Manager tool facilitates to help out the tester or developer in the testing cum development phase. As it is very time consuming and inefficient to wait for the response from the server located at remote location for the test data, thus we propose to prepare mock data readily available to the tester at the client location itself . As soon as the tester hits the query for the respective data or record , the application first checks if it is available in its cache , if yes then it responds to the tester else the query is hit at the server side and it is send back. And the caching of result in key-value pair will make the response time more efficient when the query is fired for the next time again.

ISSN (Print) : 2320 – 9798
ISSN (Online): 2320 – 9801

International Journal of Innovative Research in Computer and Communication Engineering
*Vol. 1, Issue 1, March 2013*

At the same time this response from the server is also cached. The cache is maintained with the technique of LEAST RECENTLY USED algorithm which deletes the records or data which have been least recently used. Also, the query thus received is then parsed through the DOM parser .The parser then parses the query and displays it to the user in the structured tabular format which was initially stored in the XML format . Apart from this, a second phase of the proposed system is testing of the method automatically by reflecting the data acquired and passing it as parameters. The details of this provided in the implementation section.

Thus , the proposed tool provides facilities of helping out the tester to use application even if he or she doesn't have great knowledge about writing queries through provision of excellent GUI and also provide data in the required converted format the tester wants and also to test the method through use of annotations and reflection[1].

## III. IMPLEMENTATION AND ALGORITHM

Firstly, as it reflects a client server application,server is connected to many clients . In this case , to 7 clients as we are using 32 bit Operating system. We see , that the client send requests to the server at a known server port (say port 1200 ). Now the server services the client by re-directing it to the required port say 1201 for the first client.The redirection of port is necessary so that the server can facilitate other clients in queue. Thus, the client then keeps working at the port directed by the server so that the port 1200 (server port) now services the second client and redirects it to port 1202.

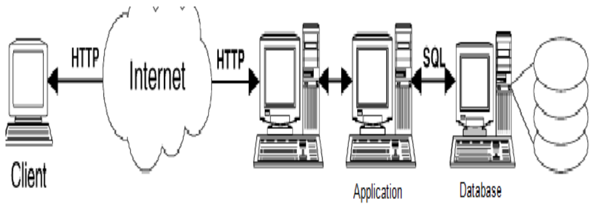
As soon as the client provides the required ID in the GUI and the type of data it request ,a query is formed and hit at the backend. First time , the query is brought from the server and send to the client as well as stored in the cache. The cache contains the records that have been most frequently used.

For Caching the algorithm is:

1. Declaring a global array for ID & RECORDs and declaring a static global variable count to keep track of both the arrays.
2. While retrieving a record from server insert each ID and its corresponding record into the array and increase the count variable.
3. While retrieving a record from cache, query is not hit back at server and the records are fetched and displayed from the cache only.

Least Recently used Record concept is use which deletes the record which has been least recently used and replaces it with new record when the cache is full.

Now as the records are stored in XML format , we use the Document Object Model parser.



With the help of this, the records are parsed and also displayed in tabular format to the client.

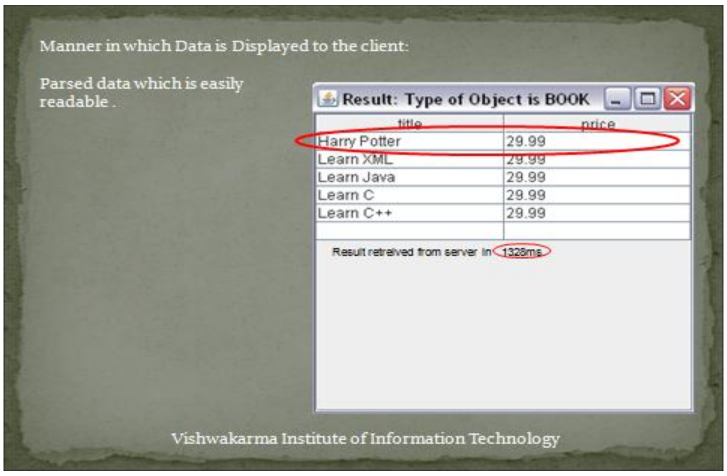
This available data can be converted in any format using standard java commands as for converting XML to JSON. The tester is provided the data in JSON format by clicking on the data conversion option on the GUI.

Now further, we make use of concept of annotations[2][3] and reflection in java to get the parameters of the ID passed by the user, invoke it in the method required to test.

For example , if there is a method "add(a, b)" in the tester's code , then the values of parameters 'a' and 'b' are automatically passed through annotation and reflection and the tester's code is tested and results are provided.

## IV.RESULTS

The snapshots of the result have been shown as :-



Manner in which Data is Displayed to the client:

Parsed data which is easily readable .

Result: Type of Object is BOOK

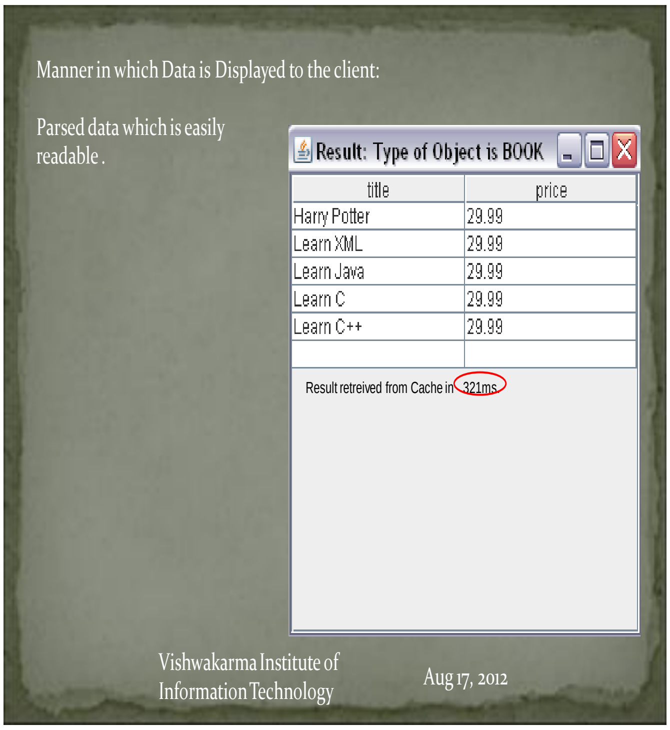
| title | price |
|---|---|
| Harry Potter | 29.99 |
| Learn XML | 29.99 |
| Learn Java | 29.99 |
| Learn C | 29.99 |
| Learn C++ | 29.99 |

Result retreived from server in 1328ms

Vishwakarma Institute of Information Technology

Fig 1. Result of data retrieval from server

As shown in fig I ,the time taken for the response from the server is 1328 ms whereas as shown in fig II, time taken to retrieve from the cache is 321ms.Thus, there is a great improvement in the response time so achieved.



Manner in which Data is Displayed to the client:

Parsed data which is easily readable .

Result: Type of Object is BOOK

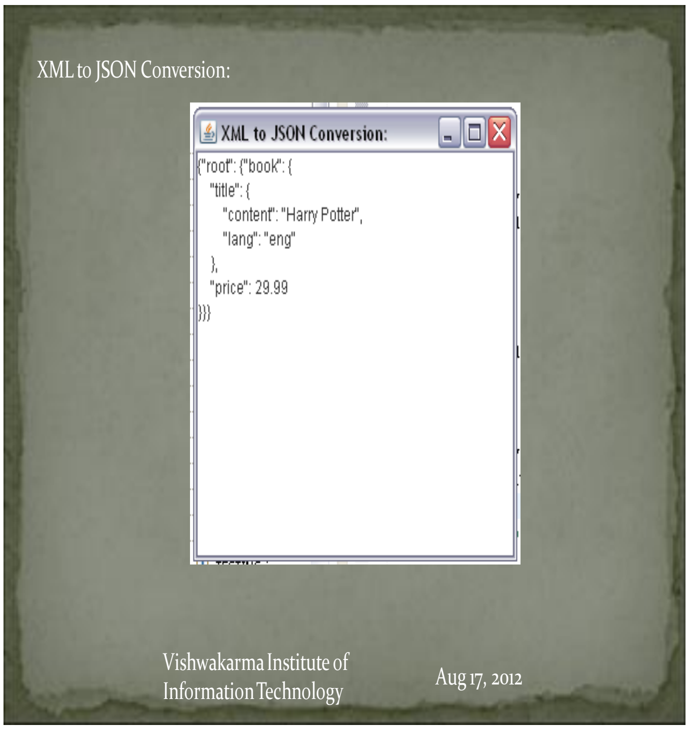
| title | price |
|---|---|
| Harry Potter | 29.99 |
| Learn XML | 29.99 |
| Learn Java | 29.99 |
| Learn C | 29.99 |
| Learn C++ | 29.99 |

Result retreived from Cache in 321ms

Vishwakarma Institute of Information Technology

Aug 17, 2012

Fig 2. Result of data retrieval from Cache prepared


XML to JSON Conversion:

XML to JSON Conversion:

```
{"root": {"book": {
   "title": {
      "content": "Harry Potter",
      "lang": "eng"
   },
   "price": 29.99
}}}
```

Vishwakarma Institute of Information Technology

Aug 17, 2012

Fig 3. Result of data conversion from XML to JSON

## V.TESTING

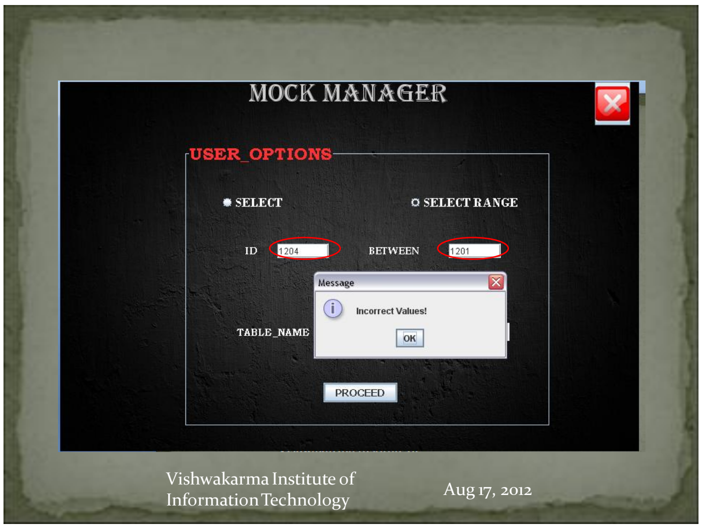


Fig 4. Testing for incorrect values of input for data retrieval/conversion

## VI.CONCLUSION AND FUTURE SCOPE

Thus, we conclude that mock manager [4] tool facilitates with rretrieval of data with faster response time from the cache as compared to server, also provides availability of data in the format required by the tester along with the provision  of query forming automatically without the tester having deep knowledge of database  and additionally, direct execution of the method by reflecting the values in the method call which tester wishes to test. Future scope can be provision of various other data conversions and also reflecting more than one parameter in the annotations for method testing.

## VII. ACKNOWLEDGMENT

We would like to express our heartfelt gratitude to our guide , Mrs. Shweta Tiwaskar, Assistant Professor at Vishwakarma Institute Of Information Technology for her valuable technical inputs and constant guidance.

## REFERENCES

1. K. Amiri, S. Park, R. Tewari, and S. Padmanabhan. Scalable template-based query containment checking in web semantic caches. In *Proceedings of the IEEE International Conference on Data Engineering (ICDE)*, Bangalore, India, 2003.
2. A. Iyengar and J. Challenger. Improving web server performance by caching dynamic data. Dec. 1997.
3. Kathy Sierra, SCJP ( Sun Certified Java Programming) (TEXT book)
4. Herbert Schildt, Complete reference JAVA 2 (Text book)
5. http://tutorials.jenkov.com/java-reflection/index.html
6. http://isagoksu.com/2009/development/java/creating-custom-annotations-and-making-use-of-them/
7. http://javapapers.com/core-java/java-annotations/
8. http://www.shivlearningjava.com