Storing the Data for Longer Time and Maintain Security API using Java

K. Govinda[1] N. Pratap[2] Prashant D. Patil[3]
[1]Assistant Professor
[1, 2, 3] SCSE,VIT University

Abstract---In recent years the evaluation of Application Programming interfaces (API's) increases for web-based applications. It describes how few components of software interacted with one another. It is a collection of functions that perform a specified task to interact for components of software it can be defined as standard library and it can be considered as specification of Remote Calls. In this API we are maintaining the data for longer period of time and with security based mechanism using java which provides authentication also. In this API the data can be taken from transactional and relation databases. In this additionally packages of security can be added for data to maintain the data over un-trusted parties. In this API the queries can be handled and maintained accurately with the Google App's Engine.

Keywords: security, Remote Calls, relation database, web-based applications, API

I. INTRODUCTION

In the development of web the API can be represented in terms of request-response messages by using the HTTP and along considered the JSON or XML format for response. In the web-services they used the SOA and SOAP. But now a day they use the ROA and REST which is related to semantic-web and RDF. The collection of web-API is known as mash-ups. Google applications is the service provided by the Google using with domain based names suggested by costumers and Google application engine is a platform which builds web-based applications. In this API store huge amount data and maintain the security for the data. When the user wants certain data can take through the API as service based mechanism. In this API the relations consists of columns and rows for table each entity can be taken as a class for java which is light weight can be given as a xml file. The security for this API can accesses the data with some cryptography based algorithms which can be encrypted and decrypted. The data can be classified as object and this objects can be accesses though this API. This API can handle different databases.

A. Steps To Create an API

1. Add the essential imports
2. Define the subclass for API
3. Determine the method what the data expects for request-response
4. Create desired name for API
5. Add the server API code

After this define the API and allow the multiple clients and call the API methods to implement the service for various clients to access the data. In addition to this add the security packages to the current API

II. LITERATURE SURVEY

In previous java API's the data can be stored and retrieved through JDBC derby according to java persistence API [1] it will store the data for longer time. Different java API'S are used for storing the data for java web based applications and for the database connection to retrieve the data values of the user by using java database connectivity. The blob store [2] application engine which provides the persistence and management the huge the data another data store API it also an application engine which provides the storage interfaces and data access objects (DAO). Memory cache interface using java which used for storing the data in the cache memory but it stores the unreliable data. Search API's [3] are used to retrieve the data using java based queries. Files API is used for accessing and storing data from the files. Java task based API [3] which is maintain the queue for user based request and response mechanism and the channel based API which provides the to maintain the connectivity for the persistence API's in order to messaging passing between java clients. In search API's the data or the documents can be retrieved with indexing mechanism and queries and search checkers API [5] which provides the to check the values of queries, indexes, expressions which are in sorting order and document.java security based API's [4] are used in many areas which provides services like security, authentication and packages for the key generation on client and server side. In the platform of java security will consists of API's, tools packages and classes which are applicable to provide the service in efficient manner. Java securities based API's will majorly include in the area of security like authentication, communication secure and control access. The security API's [4] will be designed with the following principles they are implementation of interoperability, independence and extensibility of algorithm. In java security can be provided by set of packages and selected with priority based service. In security cryptography can be done in java which includes API's and algorithms such as message digest algorithm, encryption based algorithm like symmetric and asymmetric stream and bulk encrypted algorithms, key generation, pseudo random generation, password (authentication) based encryption algorithm. In java the security can be done in two different packages like java security and java crypto which consists of cipher based and agreement by using the keys while providing the keys it should have package like java security key store and it will provide certificate as java security cert certificate store which provides security over un- trusted network. Authentication can be provided by the package java security authentication login which has login process. This login process [5] will have authentication based frame work and communication can be done securely with the package as java security net secure socket engine which has normally for stream based socket this packages can be run at java run-time environment. The control access mechanism can be done by using the package java language security manager. It will check all the control access of the data and to access the system properties with the package as java utility property permission and to access the resources of the files the package is java input-output file permission.

To open the sockets the package is java net socket permission and for the authentication and to handle the streams by using the package as java net permission. Security which is related to access the internal commands it can be done by using the package as java security permission. Among all this API's [2] the java storing and security based API will provide authentication and persist the data for longer time.

III. DESCRIPTION

Storing the data for longer time and to maintain the security API can be done by using the java platform. It is based on the mapping the objects to the tables in the database the mapping can be called as object relation mapping where we can retrieve the data. JSSA API can be used for mapping, update, retrieve, store from relation tables and objects and vice versa. JSSA can be implemented by using persist provider and it can be used in java-SE and java-EE applications. JSSA API can be connected directly with the objects instead of Structured Query Language statements. It uses the metadata for the correct operations of database. This metadata can be containing the XML configuration. This XML configuration overwrites the annotations. JSSA will support the dynamic and static queries in the database. It automatically creates the database schema which on basis of metadata. It is entity based relation for this table is created it can be done by using the package java persistence entity. All the entities should have the Primary key and the instances are row for the table. JSSA will automatically generate the Primary key by using the annotation @ Generated value. It will store the fields and to access the data by using the getter () and setter () are used. It will persists the data default when the table is created the notations can be defined as @ ID indicates unique Id for the data entity, @ Generated value indicates generates the Id automatically and @ Transient the field which is not saved in the database. Relational Mapping can be done between objects and entities. This can be represented in four ways as the

@ one-one
@ One-many
@ many-one
@ many-many

And the entity manager provides the operations to and from the database. To synchronize the objects we use merge () and to manage the entities entity manager can be used. It can execute the query statements by using the commit and after this disconnect the database by using disclose () and at last it provides the security for the data values in the database and for this API it uses different packages and classes. It can provide the accurately which can be stored in the database. The instances can have the values without any duplication which has the Unique key and to avoid the redundancy in the database and this security field can provide the confidentially, authorization from third party networks and overcome attacks. JSSA API can be done in four steps they are

1. Entity
2. Storing the entities
3. Relational mapping method
4. Security

A. Implementation

The following are the steps to implement the JSSA API:

1. Create our first JSSA entity.
2. Interaction with the JSSA entity with Entity Manager.
3. Generating the forms from the JSF Pages framework from JSSA entities.
4. Generating the JSSA entities from the existing schema database.
5. JSSA based query language.
6. Generating the complete JSF Applications from the JSSA entities.

In this the data values can be stored in the database this entities are plain old java objects (POJOs). After this create the new web application as a JSSA entity under java faces framework which is already build with glassfish server as default and create entity class. After this we should give the class name and the package name and then create the data storage unit and give the name after this go for the new data source connection and the data source can be done in the package javax.sql and after this data connection can be done with the JNDI which has the host name and port number and give the username and password after this create the additional property and pass the parameter and this username and password are stored in the XML file and for the data connection. In this click the java transactions API for the data transaction like retrieve and store.

@ Id
@ Generated value (strategy1 = Generation type. AUTO)

Indicating that it will provide default Primary key if no primary key is specified in the database.

Adding the storing fields to the JSSA entity

Adding the attributes to the table to retrieve the values to store the data which can be done by getter () and setter ().

Creating to access the database objects

In the NetBeans create the new java class with the extension of faces.xml. After creating this one click on this one add the managed Bean and create the as the session. Create the java file for the user access and create the JSP file and for the access the data it can represented as follows

<Managedbean1>
<Managedbean1name>customer1DAO</
Managedbean1name >
< Managedbean1class>
com.ensode.jpaweb.customer1DAO
</ Managedbean1class >
<Managedbean1scope>session</ Managedbean1scope >
</Managedbean1>

After creating this we have to connect to database by using the JDBC it can be done by using the following steps.

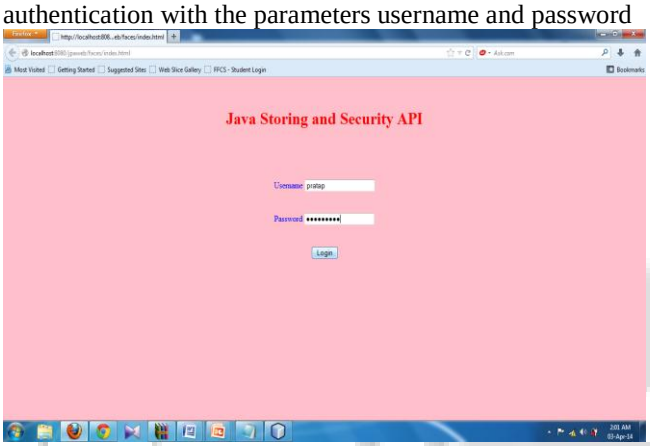
1. Establish the connection
   a. Load the specific driver
   b. Making the connection in this creating the object and pass the parameters username and password
2. Creating the JDBC statements
   Statement stmt1 = con.createstatement ();
   It will provides the creating the objects to send the Sql statements to the database.
3. Executing the Sql statements.
4. Get the resultset
   String Queryabc = select * from ABC;
   It will execute the query and the result values through the database.


```
Resultset rsl = statement.executeQuery (queryabc);
While (rs1.next ()) {
        Int ssn1 = rs1.getint ("SSN1");
        String name1 = rs1.getstring ("NAME1");
        Int marks1 = rs1.getint ("MARKS1");
}
```

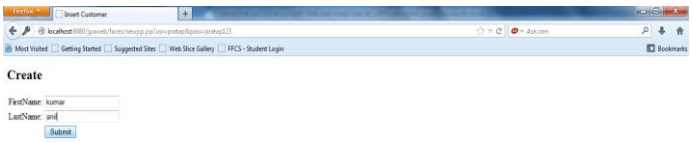
5. Close the connection
   It can done by two methods they are
   Statement.close ().
   Connection.close ();
   The transactions can be done it will allow all the Sql statements together to execute as a single transaction. In the transaction the connection can be done default as the command using auto commit command. In the JDBC each sql statement can be consider as the transaction. By using the auto commit false statement we can disconnect connection.

B. Screen shots

In this JSS API it will provide the security and authentication with the parameters username and password



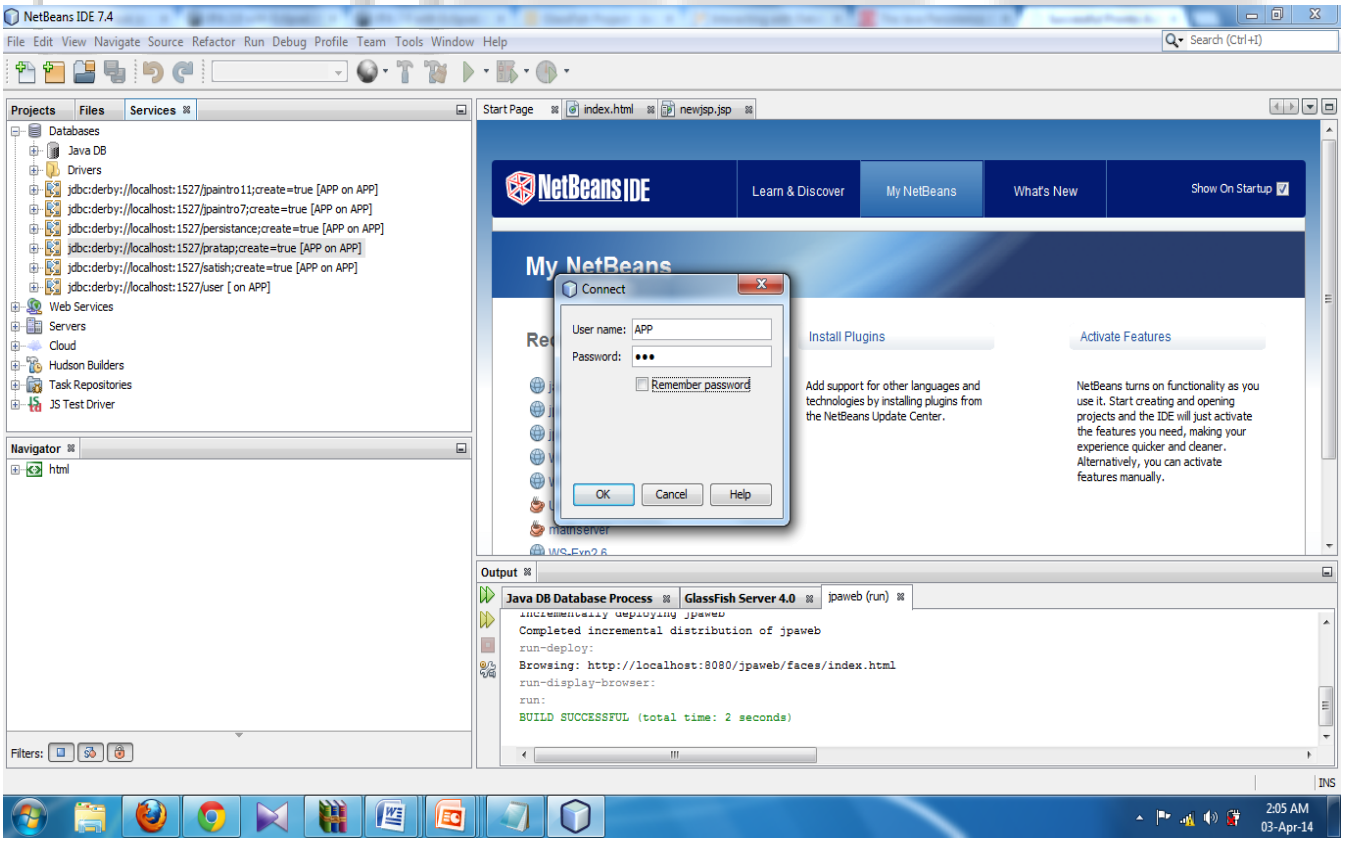
IV. INSERT THE VALUES IN TO THE DATABASE



After giving the login process we have specify the values of the user to insert in to the database when the values are created it stores in to the database

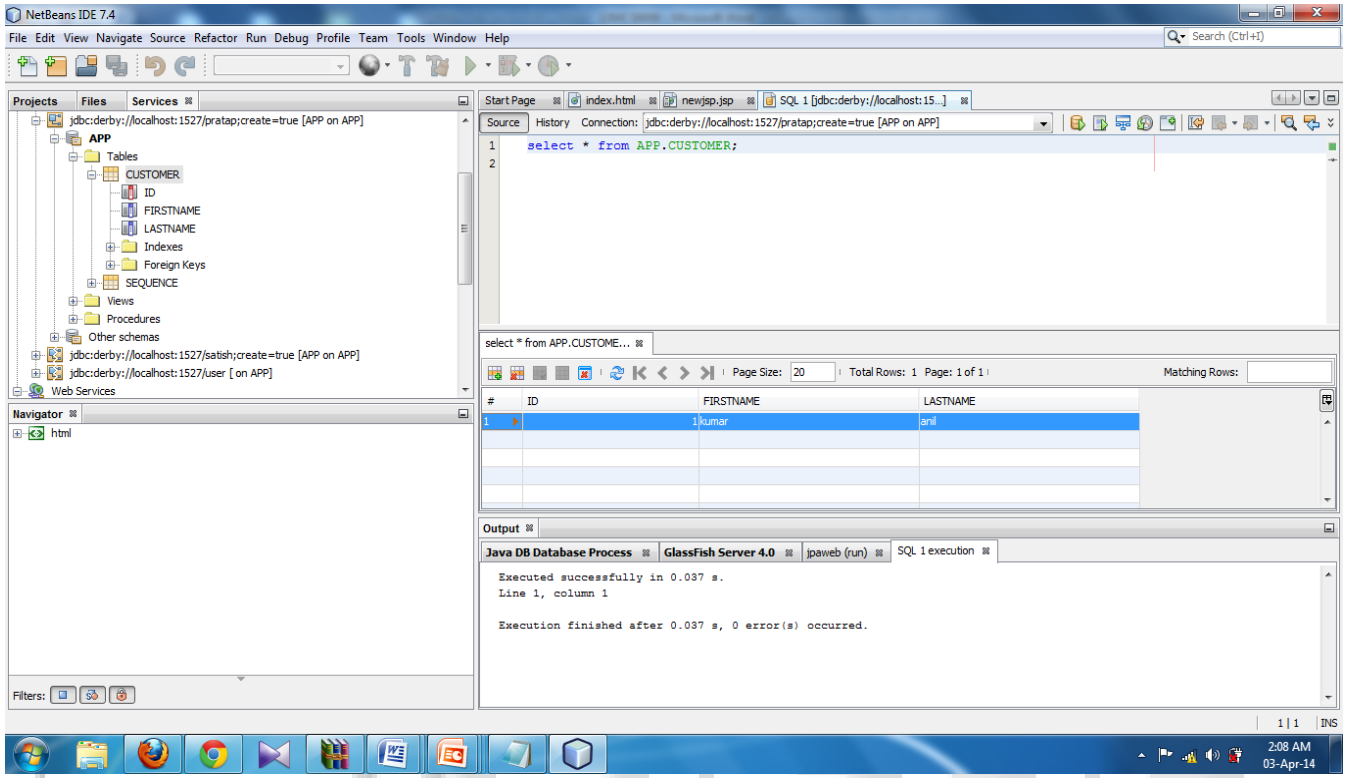
V. CONNECTION TO THE DATABASE

In this one after storing the values we can retrieve the values from the database and connect to the database with port number and host name after this provide the username and password for the database connectivity



VI. STORING THE VALUES IN TO THE DATABASE

After connecting to the database the values can be viewed through the table which values enter in to the database that can be stored for longer time with security based mechanism.



VII. IMPLEMENTATIONS TOOLS

JDK 5.0
Oracle 11g
Net beans IDE 7.4

VIII. CONCLUSION

The data can be store and retrieved accurately and also maintain security by using API. It will also provide the authentication while accessing the data to end-users and it is very difficult to retrieve the data from the third party attacks.

REFERENCES

[1] http.//www.hil.ber.tinc.com./white.papers./Java.%202.%20.Security.pdf
[2] http.//.developers.google.com./app.engine./docs./java./java.doc/com./google/.appengine/api/search/.checkers/.package-.summary
[3] http.://www..oracle.com/.technetwork./articles./javaee/.jpa-137156.html
[4] http.://www..vogella.com/.tutorials/.JavaPersistence.API/.article.html
[5] https.://..java.net/.downloads/.satjug/.ESAPI.%20.JUG.pdf

