

SYNCING FILES FROM ELECTRONIC SLATE, ENCRYPT AND STORE IN DATABASE

P Surya teja¹ & M.Koteswara rao²

¹M.Tech student, CNIS (IT), VNRVJIET, Hyderabad, TS, India.

² Assistant Professor IT,VNRVJIET, Hyderabad, TS, India.

Received: August 28, 2018

Accepted: October 19, 2018

ABSTRACT

Currently there is no easy interface to web or database that simulates a slate electronically (e-slate) where a user can write anything on the e-paper and save retrieve from the database. In this project aim to develop a system that helps to interface an electronic slate with computer applications to be able to receive the data written on e-slate. There can be number of applications utilizing this system like online examination for essay type questions in any language, ability to draw pictures etc even by a novice computer user. This can be used by doctors to write prescriptions and store as a medical record against a patient.

Keywords:

I. INTRODUCTION

Currently there is no easy interface to web or database that simulates a slate electronically (e-slate) where a user can write anything on the e-paper and save retrieve from the database. In this project aim to develop a system that helps to interface an electronic slate with computer applications to be able to receive the data written on e-slate. There can be number of applications utilizing this system like online examination for essay type questions in any language, ability to draw pictures etc. even by a novice computer user. This can be used by doctors to write prescriptions and store as a medical record against a patient.

A Boogie board is e- slate where it allows to write or scribble on with a pen stylus (hardware pen) or even with a finger. every models of this boogie board are same, except the Sync board which is new project going to be introduced here which we can write and read in it . The electronic slate is not even a smart gadget and nothing will bestored and view on it. nothing can be edited exceptional on writing more on it. If the written part is erased it cannot be regain the data written on it. There is a button named as erase which provides to erase the screen. This is actually an e- paper which can be used as an scratch pad. In the provided models a removable battery and a few of them doesn't but the battery level of them will last for a long time.

The scribbled information are saved in image format in the form of a PDF file and which can be copied to a web application(database) sing Bluetooth. Such a way that notes and lines are stored in the database. E-slate board itself is only a visual presentation. These drawings can be scribbled on this slate only using the stylus pen.

This interface helps the user to write any input language and will be able to read in web application(database), computer or in a mobile. Whatever is written on the electronic slate will be encrypted and saved in the database and the data can also be retrieved from it.

II. RELATED WORK

2.1 Boogie board:

Create: Writes and draws any type of data which is useful for students and lecturers fornotes.

Save:The board can savehundreds and thousands of pages and drawings to this e-slate internal memory with the button provided. The buttons which are on the e-slate are save and Erase.

Save and Share:In this Bluetooth is used to save and share the files from the e-slate to the mobile..

Sync, which originally debuted in 2013, features ImproveElectronics' eWriting technology, as well as a microSD card slot for storage of notes, microUSB charging, and Bluetooth support for connection to mobile devices. That last bit allows you to wirelessly sync with other devices like displays, phones, etc.

Files Transfer

The files will be transferred from the e-slate to the app is by using the API of the e-slat through the Bluetooth.

The library is broken up into two essential parts so that , On one side we will have the Streaming API where you can get erase/save button pushes as well as real-time paths that are drawn on the e-slate.

On the other is the File Transfer API where you can delete, download files from the e-slate.

Both of these APIs extend a single service to facilitate long running operations.

Typically, we first bind to the SyncStreamingService or SyncFtpService. When we receive a callback for onServiceConnected() we will get an Ibminder. From there we cast the Ibminder to the specific binder class and then retrieve reference to the actual service object.

AUTH AND USER MANAGEMENT

- Drop-in support for login with.
- Email & Password.
- Common OAuth providers.
- Google
- Facebook
- Twitter
- Github
- Your own custom auth tokens Auth and user management.

2.1 Firebase Database and Storage:

Firebase allows to build apps which need authentication, database, file storage, analytics and server side functionality without having to own and manage infrastructure and software required for server side support.

For example, if we want to develop an app which allows user to save certain data and allows user to access it from any device. For this, we need to implement authentication and store data in remote data base. This requirement can be implemented using Firebase products. All we need to do is register and start using its products. There is a free quota for all resources it provides.

Firebase provides firestore and also realtime database. Using firebase, we can access google cloud store for storing images from our app. Using cloud functions, we can build server side functionality for you app.

To push notifications to specific device or to all users of our app, we can use firebase cloud messaging.

we can configure the app without having to release new version using firebase remote configuration. we can monitor performance and view crash reports using firebase products.



Figure 1. Example of Firebase Real time Database

Collaboration of all the devices easily

Realtime matching up creates it very easy for our clients to retrieve their data from any type of gadget, webpages or portable, and it is available to the clients to perform the results together with everyone.

Key capabilities

How it works?

This Database gives access to enhance smooth and effective, communitarian apps by enabling safety precautions to the database straightforwardly from client side code. The data is taken on locally, and although when offline, Realtime databases occasionally keep on termination, results in end user a response or deal. during the device recaptures connection, the Database syncs the local data or text changes during the remote updates that required while the client went offline, combination of any contention automatically. This Database results in adaptable, articulation base guidelines dialect, called Firebase Real-time Database Security polices, to differentiate the data or text might to be performed and during the data or text can be taken from it. At some point associating with Firestore Authentication, designers can be differentiate who approaches what does the data, and how they can access it.

This Real-time Database is a NoSQL database and all things considered has different enhancements and use of contrasted with a traditional database. This Database API is intended to just give access to activities that performed fastly. This empowers you to fabricate an extraordinary real-time encounter that can serve a great many clients without trading off on responses. Along these lines, it is critical to make sure how the clients need to get data and after that algorithm it in like manner.

Why to prefer Real-time Database?

- The reasons behind to prefer realtime database are given below.
- Basically the apps require to update frequently to get bet performance so for all the it is cost efficient.
- It has been present from the long time so it has tried thousands of apps run on it and knows how to manage them..
- It has a very good latency and . it works much better if we need to work on low-latency in real time database.
- Especially for the new apps going to be developed firestore is more preferable on the other real time databases because it

stores any type of data whether PDF or work .

2.3 NoSQL

NoSQL is the only way to deal with databases that represents to a move far from traditional relational database management systems (RDBMS). To characterize NoSQL, it is useful to begin by portraying SQL, which is an query language used by RDBMS. traditional databases depends upon tables where it consists of rows and columns or schemas to arrange and regain the data. Interestingly, NoSQL databases cannot depend upon these structure and to utilize increasingly adapting data models. NoSQL can signify "not SQL" or "not only SQL." As RDBMS have progressively ignored to meet the implementation, versatility, and adaptability needs that next-Gen to come, data concentrated apps need, NoSQL databases can beget by standard endeavours. NoSQL is especially helpful for putting away unstructured information, which is becoming unquestionably more faster than the organized information and it doesn't fits in the social blueprints of RDBMS. Basic classification of unstructured data contains user and session data; chat, messaging, and log data; time series data more like IoT and device data; and large objects more like as multimedia.

III. CLASSIFICATION OF NOSQL DATABASES:

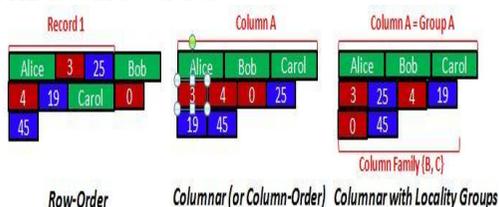
There are many types variants of these type of databases introduced to support particular needs and use cases. These are divided mainly into four primary classes.

Key-value data stores: In this type of NoSQL database accentuate straightforwardness and are exceptionally valuable in speed or boostan applications to help fast peruse and making or handling of non-transactional data. Stores away values these can be any sort of binary object like plain text, video content , JSON , and so on. these gotten to through a key. This application has to be finished on commanding over what to be stored away in the esteem, making this the most adaptable NoSQL model. Data is divided and repeated over a group or cluster to get scalability and availability. Hence, key value stores frequently doesn't supports transactions. In any case, they are exceptionally powerful at scaling applications that bargain with high-speed, non-transactional based data. The Keys are mapped to (possibly) more difficult value like lists .these Keys can be stored in a hash table and can be distributed easily Such stores typically support regular CRUD (create, read, update, and delete) operations .That is, no joins and aggregate functions

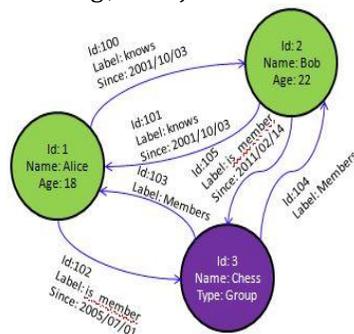
Document stores: Document databases generally store themselves portraying JSON, XML, and BSON reports. These are same as previous type of stores , however for this situation, an esteem is a solitary records all these information and also stores all information identified with an explicit key. Famous fields in the record can be listed to give quick recovery without knowing the key. Each report can have the equivalent or an alternate structure. Documents will be saved in some standard format or encoding for example PDF or MS word. Basically these most are of preferred as Binary Large Objects (BLOBs). These can be easily indexed. It permits these document storage to outperform traditional file system. E.g., MongoDB and CouchDB these databases can also be used to query MapReduce.

Wide-column stores: Columnar databases are a collaboration of RDBMSs and Key-Value stores type of NoSQL databases. these Values will be saved in bunch of null or more columns, but in this Column-Order as reflected to the Row-Order Values are matched keys when query is taken place.

E.g., HBase and Vertica



Graph stores: In this type of NoSQL database the Data in here are represented as vertices and edges. these Graph type of databases are strong and robust for graph-like queries for example to detect the shortest path among two or more elements or nodes E.g., Neo4j and VertexDB



Advantages Of NoSQL

These provides endeavors imperative preferences on traditional RDBMS. Given that we then comment on MongoDB specifically vs SQL even if both auto-shard. PostgreSQL in particular has made a lot of strides in terms of getting unstructured data usable (JSON/JSONB types) not

to mention the power you can get from something like PLV8, it's probably the most suited to handling the types of loads we might throw at a document store with the advantages of NoSQL. Where it happens to fall down is that replication, sharding and failover are bolted on solutions not really in the box.

For small to medium loads sharding really isn't the best approach. Most scenarios are mostly read so having a replica-set where you have additional read nodes is usually better when you have 3-5 servers. MongoDB is great in this scenario, the master node is automatically elected, and failover is pretty fast. The only weirdness I've seen is when Azure went down in late 2014, and only one of the servers came up first, the other two were almost 40 minutes later. With replication any given read request can be handled in whole by a single server. our data structures become simpler, and our chances of data loss are reduced.

Again for a mediums sized classifieds site, the vast majority of data belongs to a single collection. it is searched against, and displayed from that collection. With this use case a document store works much better than structured/normalized data. The way the objects are stored are much closer to their representation in the application. There's less of a cognitive disconnect and it simply works.

The fact is that SQL JOIN operations kill performance, especially when aggregating data across those joins. For a single query for a single user it's fine, even with a dozen of them. When you get to dozens of joins with thousands of simultaneous users, it starts to fall apart. At this point you have several choices.

- **Caching** - caching is always a great approach, and the less often the data changes, the better the approach. This can be anything from a set of memcache/redis instances to using something like MongoDB, RethinkDB or Elasticsearch to hold composite records. The challenge here comes down to updating or invalidating our cached data.
- **Migrating** - migrating the data to a data store that better represents your needs can be a good idea as well. If you need to handle massive writes, or very massive read scenarios no SQL database can keep up. we could never handle the likes of Facebook or Twitter on SQL.

Exchange of Data

- During the exchange of the data between a server and a browser, the data should only be the information in text.

- JSON is text, and it should only be changed into any type of JavaScript method into JSON, and sends this same JSON to the server model.
- It can be easily changed into any type of JSON format received at the server into JavaScript objects or methods.
- In This type of method we should only work with the data which is text here as JavaScript..

Sending of Data

- When the text is going to be saved in a JavaScript method, then it can easily change it into the JSON format, and then it sends the data to the server.

Receiving Data

- When we receive the data in the format of JSON, we can again easily change it to a pure JavaScript method.

Storing the Data

- When the information or text is being stored the text or information, this text is to be in only particular format, and without any problems in choosing to save it this is the best and only legal way format to be stored.
- JSON format can only make it work to save JavaScript objects as information or data.

Implementation

This execution clarifies how the boogie board application was interacting with Firebase. These lines are put away as vector pictures in a PDF document and can be exchanged to a web application(database) by means of Bluetooth. Along these lines the notes and illustrations can be protected in the database. The e-slate board itself is just a visual portrayal. anything can be written on the e-slate however except if the exceptional stylus pen is utilized it cannot be recorded.

This interface causes the client to compose any information dialect will have the capacity to re promotion in web .

Application(database), PC or in a portable. Whatever is composed on the electronic slate will be encrypted and saved in the database and the information can likewise be recovered from it.

5.1 FIREBASE

Implementation of Firebase:

- Add firebase support In build.gradle dependencies in android studio
{compile 'com.firebase: firebase-package-name-android:3.2.1+' }
- In the Android manifest file

Syntax is <uses-permission android:name="android.permission.INTERNET" />

- In Android apps or activity@Override
- These creates overriding of the activity

public void onCreate() { // creates the content for this syntax

```
super.onCreate();
Firebase.setAndroidContext(this);
```

}

Data store

Data store Creating a reference Firebase rootRef = new Firebase("https://docs-examples.firebaseio.com/web/data");

- Directly access child nodes in the data

```
Firebase rootRef = new
Firebase("https://docs-examples.firebaseio.com/web/data/users/mchen/name");
```
- Using the Child() api method for calling

```
Firebase rootRef = new
Firebase("https://docs-examples.firebaseio.com/web/data");
reference adding to syntax
rootRef.child("users/mchen/name");
```
- **Syntax for storing mStorageReference**

```
=
FirebaseStorage.getInstance().getReference();
```
- **Syntax for database reference**

```
nDatabaseReferences=
Firebase.Databases.getInstance().getReference
```

(Constants.DATABASE_PATH_UPLOADS);

Saving Data

- setValue() is the method for writing or to replace data to a specified path, for example messages/users/<username>.
- updateChildren() is the method for Updating few keys for a specified path without replacing data at all.
- push() is a method used to Add a list consists data in our Firebase database or storage.
- runTransaction() is a method for Using our transactions feature when complex data is taken under considerations.

Retrieve Data

- Attach an asynchronous listener to a Firebase reference.
- This listener will be triggered when the initial state of the data is being processed and again the data can be changed when ever we wanted.

ENCRYPTION:

Transmitting confidential information, for example, plain text like an useful data .text which is not to be shared to others except to the authorized users through a wire is constantly powerless against security. It is constantly prescribed to encrypt such data as text or files. And use SSL to transfer this confidential and sensitive data , text or files. Java language gives numerous encryption calculation to this. In this paper it is examined about an standard encryption technique AES i.e., Advanced Encryption Standard. this is an symmetric encryption standard using java this AES with CBC mode of encryption is quicker and more secured than the 3DES.

Encryption Type

We knew that basically there are two main fundamental sorts of encryption techniques they are Asymmetric and Symmetric encryption. Uneven encryption utilizes two unique keys they are public and private keys .Here, we can encrypt confidential data with a key called as public key and for decrypt the same text an private key is utilized. Asymmetric encryption is all of the part utilized during when there are of two different end-points are included, for example, VPN model client and server model , SSH, etc.

In this AES the input can be 128 bit, 192 bit, 256 bit and relating bit of figure content is created.

What is AES Encryption

AES represents Advanced Encryption System and which is a symmetric encryption type of algorithm. It is a particular for the encryption where an electronic information built up by the U.S. National Institute of Standards and Technology (NIST) in 2001. This AES algorithm considers a plain-text with an secret key which is used for encryption so now the plain text is converted to the cipher text. After his for decrypting the cipher text to the same plain text the same secret key is used.

AES Encryption in Java

In the Following example the program in java that shows how AES encryption is going to be utilized in the paper .In this example it is utilizing AES with CBC method so that to encrypt a plain text as an ECB method which isn't secure at all . This IV mode ought to likewise be randomized for CBC mode.

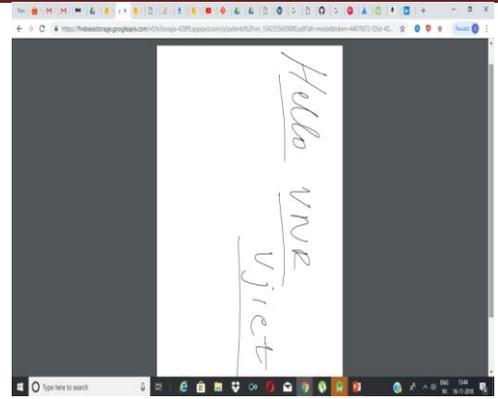
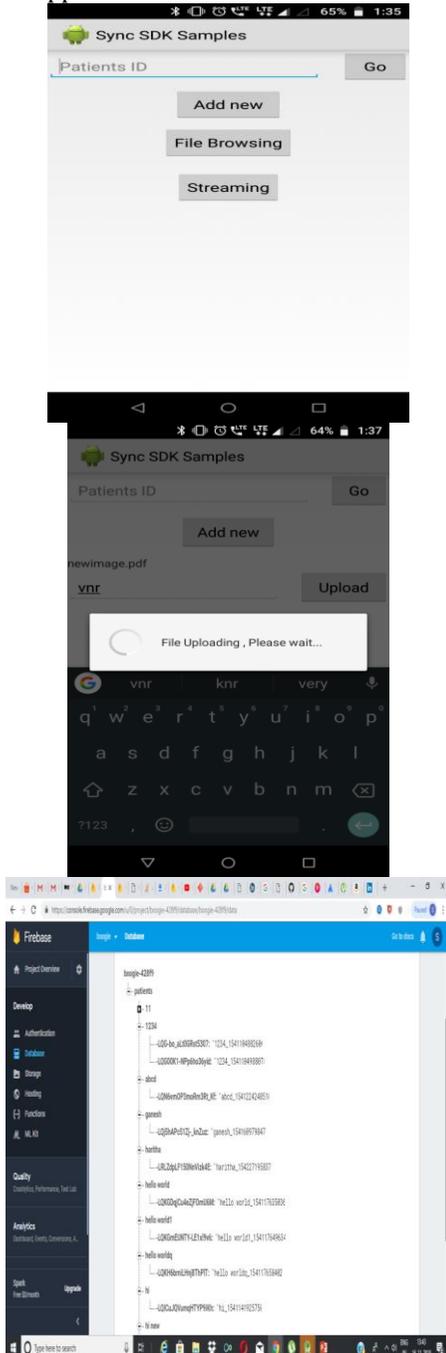
On the off chance that a similar key is utilized to encrypt all of the message here which is as the text before encrypting and on the off chance that an assailant discovers the key, the figure can be decrypt in the comparative way.

AES Decryption in Java

For the decryption of the cipher text , the exact reverse procedure is done. Because it is symmetric encryption method the same secret key is the only possible way to decrypt the cipher text to the plain and readable text. The same 14 rounds of decryption procedure is followed in this method for the 256 bit key length.

IV. Results

Below there are results from our project of an android application and Firebase Database.



V. Conclusion and future work

This interface helps the user to write any input language and will be able to read in web application (database), computer or in a mobile. Whatever is written on the electronic slate will be encrypted and saved in the database and the data can also be retrieved from it.

The future work is **boogie kit** is a syncing e-slate that connects to **Android phone**. It also supports photo notes, voice notes, and checklists. It can also be used for lectures for presenting the notes on the projector and automatically notes can be send to the students through boogie board.. If you're not using a syncing note-taking app yet, you love Android, and **boogie kit** is your default e-slate there are hundreds of other apps that do what boogie kit does.

REFERENCE

1. K. Dehdouh, O. Boussaid and F. Bentayeb. Columnar NoSQL star schema benchmark. Model and Data Engineering, LNCS 8748, Springer, pp. 281-288, 2014.
2. Strauch, C. NoSQL Databases. Selected Topics on Software-Technology, Stuttgart Media University.

3. P. Chodowiec and K. Gaj. Very Compact FPGA Implementation of the AES Algorithm. In C. D. Walter, C. etin Kaya Ko,c, and C. Paar, editors, Cryptographic Hardware and Embedded Systems - CHES 2003, 5th International Workshop, Cologne, Germany, September 8-10, 2003, Proceedings, volume 2779 of Lecture Notes in Computer Science, pages 319–333. Springer, 2003.
4. NoSQL. <http://nosql-database.org/>
5. <https://www.guru99.com/system-testing.html>
6. <https://www.softwaretestinghelp.com/types-of-software-testing/>
7. <https://www.tutorialspoint.com/firebase/>
8. <https://www.oreilly.com/library/view/introduction-to-javascript/.../ch01.html>