



Firestore

Audréa Malialin, Juliana Leclaire, Maxime Guyaux

Université Claude Bernard



Lyon 1

Les attentes utilisateur

- Fonctionnement
- Vitesse

Les attentes développeur

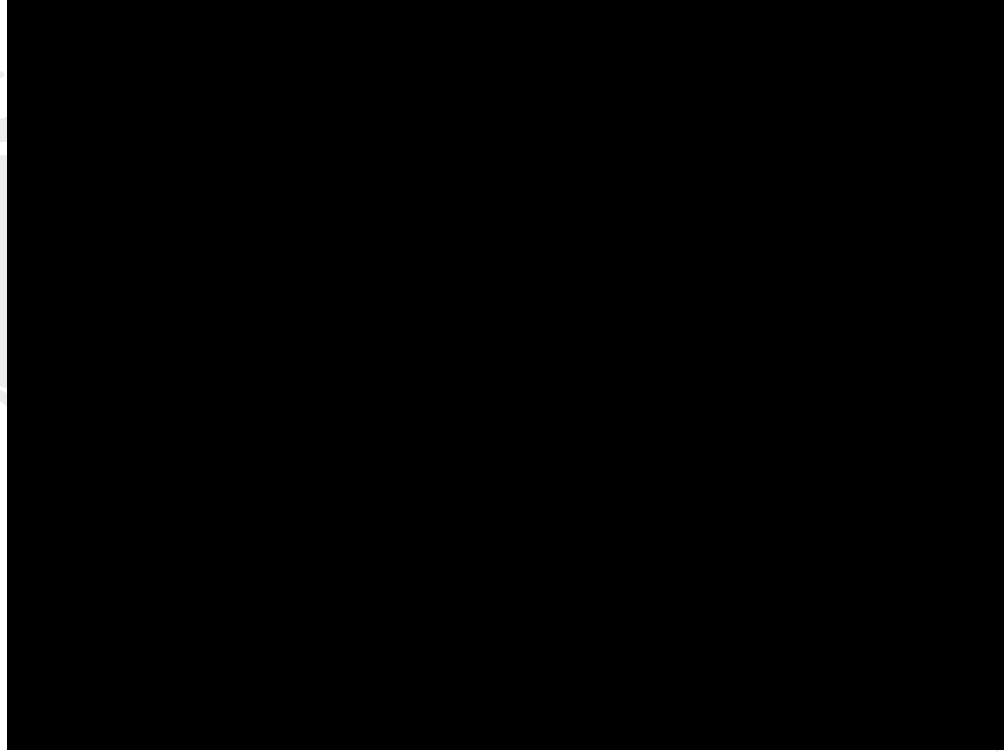
- Efficacité
- Sécurité
- Facilité

Les promesses

- Mise en place rapide
- Facile d'utilisation
- "Temps réel"
- Sécurité
- Faire travailler le client



Firestore



Stockage des données

- NoSQL
- Utilise JSON
- Chaque ressource a une URL
 - API REST
- Local storage
- Références / DataSnapshot
 - Objet contenant les données de Firebase

Firebase API (Web)

Références Firebase

- new Firebase(url)

```
var firebaseUrl = "https://torrid-torch-2690.firebaseio.com/addressbook";  
var firebase = new Firebase(firebaseUrl);
```

Navigation

- child(path)

```
firebase.child("new");
```

- parent()



Firestore API (Web)

Persistence

- `push(object)`
 - Génération automatique de clés

```
firebase.push({ firstname: "prenom", lastname: "nom", email:
```

- `set(object)`

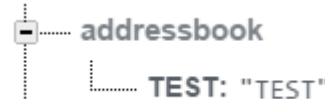
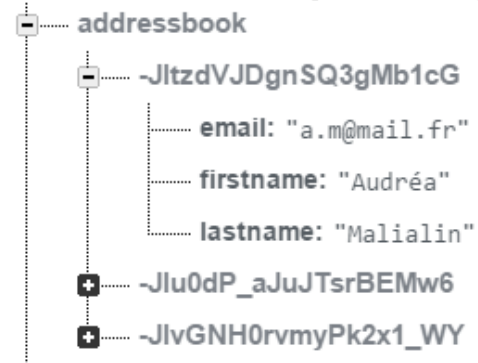
```
firebase.set({ TEST: "TEST" });
```

- `update(object)`

```
firebase.child(key).update({firstname: "Françoise"});
```

- `remove()`

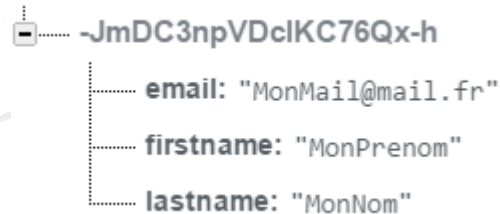
```
firebase.child(key).remove();
```



Firestore API (Web)

Méthodes DataSnapshot

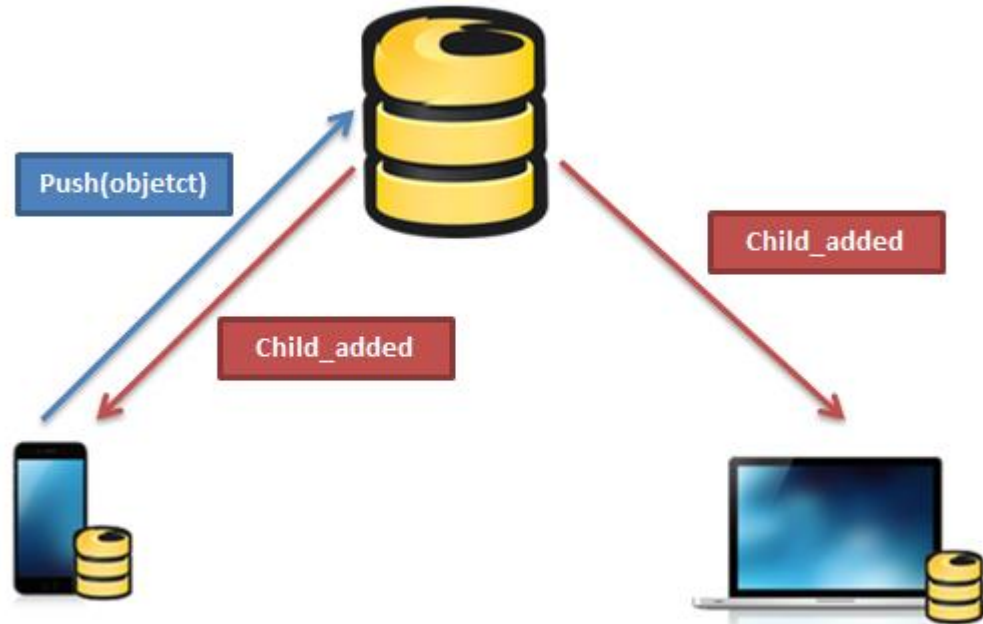
- `key()`
 - Récupère la clé
- `val()`
 - Récupère l'objet JavaScript
- `child(path)`
 - Récupère l'objet DataSnapshot



```
console.log(snapshot.key());  
-JlqUp6KPLHIO-NcURDe  
console.log(snapshot.val());  
Object {email: "MonMail@mail.fr", firstname: "MonPrenom", lastname: "MonNom"}  
console.log(snapshot.child('firstname').val());  
MonPrenom
```

Synchronisation de données

- child_added



Synchronisation de données

- child_added

```
firebase.limitToLast(10).on('child_added', function (snapshot) {
    addAddressOnView(snapshot.key(), snapshot.val());
});

function addAddress(firstname, lastname, email) {
    if(firstname !== "" && lastname !== "" && email !== "") {
        firebase.push({ firstname: firstname, lastname: lastname, email:
email });
    }
}
```

Synchronisation de données

Autres évènements possibles:

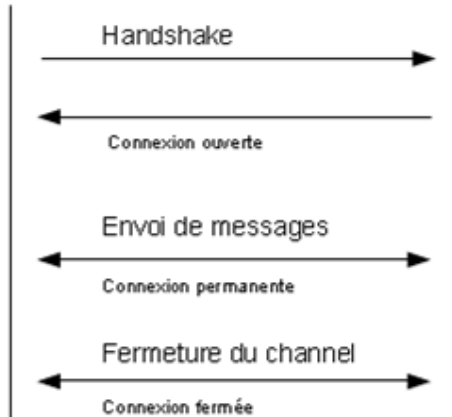
- `child_changed`
- `value`
- `child_removed`
- `child_moved`

Synchronisation de données

- Synchronisation clients-serveur Firebase
- Communication full-duplex
 - Utilisation websocket

Client

Serveur



Protocole HTTP

Protocole WebSocket

▼ General

Request URL: wss://s-dal5-nss-30.firebaseio.com/.ws?v=5&ns=torrid-torch-2690

Request Method: GET

Status Code: 🟢 101 Switching Protocols

▼ Response Headers [view source](#)

Connection: Upgrade

Sec-WebSocket-Accept: bI3Y42x3zJ4VNFTQJsBs//nXWEg=

Upgrade: websocket

▼ Request Headers [view source](#)

Accept-Encoding: gzip, deflate, sdch

Accept-Language: fr-FR,fr;q=0.8,en-US;q=0.6,en;q=0.4

Cache-Control: no-cache

Connection: Upgrade

Sec-WebSocket-Extensions: permessage-deflate; client_max_window_bits

Sec-WebSocket-Key: r1sCpp8Boon+TipJ48t2hw==

WebSocket

- Protocole réseau RFC 6455
- Canal de communication full-duplex
- Utilisation TCP
- Handshake
 - Une requête HTTP: option upgrade
 - Connexion client-serveur

Requêtes

- **Ordre**
 - by child, by key, by value
 - by priority
- **Limit**
 - to first, to last
- **Range**
 - Start at, End at (inclusive)

Requêtes

- Équivalent au WHERE
 - Pour l'ID
 - Par adresse / child
 - Pour la valeur ...
 - `startAt("----").endAt("----").once()`

Et les jointures ?

- Vous pouvez les faire ... vous même

```
userDonnees = baseDonnees.child('bob');
userCommandes = baseCom.child('bob');

userDonnees.once('value', function(donSnap){
    userCommandes.once('value', function(comSnap){
        res = extend( {}, donSnap.val(), comSnap.val() );
    });
});
```

Accès concurrent

- “J’aime”, Nombre de vues
- Problème d’accès concurrent
- Solution : fournir une fonction

```
db.transaction(function (current_value) {  
    return (current_value || 0) + 1;  
});
```

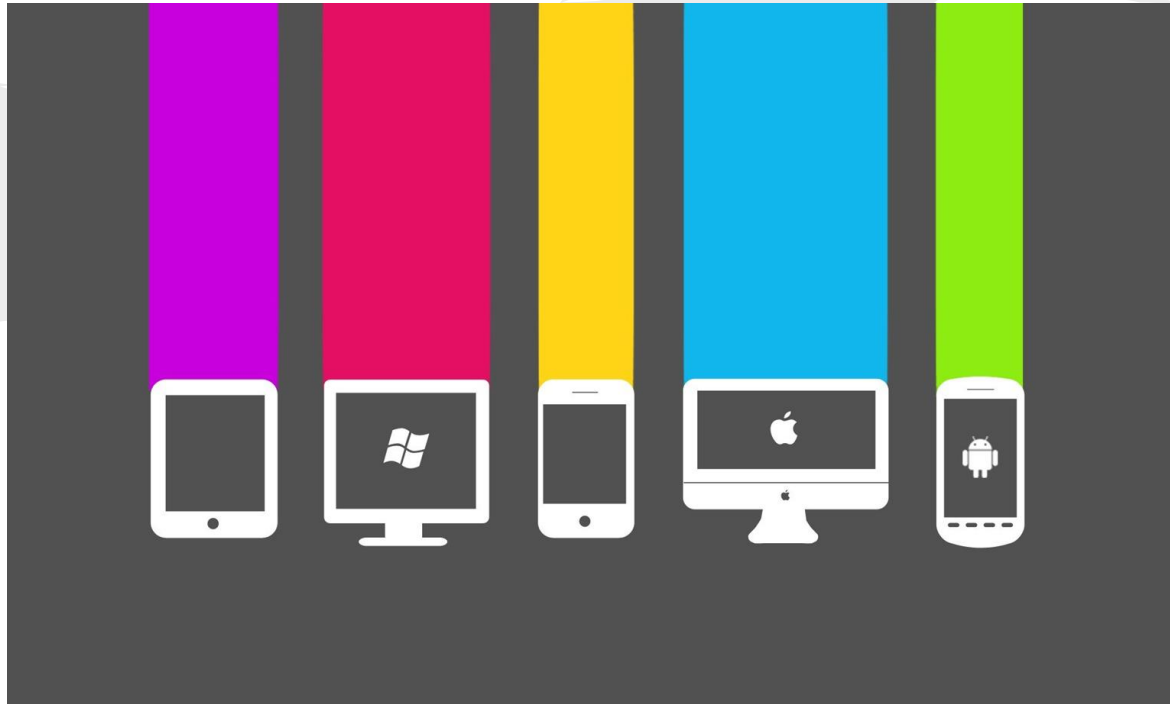
Et si j'ai pas internet ?

```
db.onDisconnect().set("Vous avez été déconnecté");  
db.onDisonnect().cancel();  
new Firebase("https://sharedpad.firebaseio.com/.info/connected");
```

- Firebase locale
- Best-effort



Multiplateforme



Différences

Mise en place



- `<script
src="https://cdn.firebase.com/js/client/2.2.3/firebase.js"></script>`
- `var db = new Firebase('https://sharedpad.firebaseio.com');`



- `compile 'com.firebase:firebase-client-android:2.2.3+'`
- `<uses-permission android:name="android.permission.INTERNET" />`
- `Firebase.setAndroidContext(this);`
- `Firebase firebase = new
Firebase("https://sharedpad.firebaseio.com/");`



- Télécharger la SDK : avec CocoaPods : `pod 'Firebase', '>= 2.2.0'`
- Ajouter au projet : `libcucore.dylib, libc++.dylib, CFNetwork.framework, Security.framework, SystemConfiguration.framework`
- Ajouter dans build settings dans "other linker flag" : `-ObjC`
- `#import <Firebase/Firebase.h>`
- `Firebase *myBase = [[Firebase alloc] initWithUrl @"https://sharedpad.firebaseio.com/"];`

Différences

Écriture / Lecture



- `set()`, `update()`, `push()`
- `db.on("event", action)`



- `set()`, `update()`, `push()`
- `db.addChildEventListener(new ChildEventListener() {`
 - `on event ...`

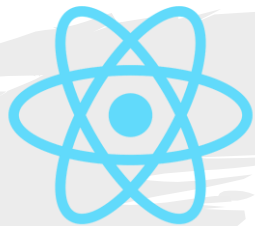


- `[db setValue/updateChildValues/childByAutoId value];`
- `[db observeEventType:FEventTypeValue`
`withBlack:^(FDataSnapshot *snapshot) ...action... }];`

Exemple

- Application “sharedpad”
 - google play store
 - Visualisation:
 - <https://sharedpad.firebaseio.com/>
 - Une seule entrée dans la base
 - Rafraîchissement de tout le pad à chaque changement de caractère
 - Test de vitesse

Compatible



Exemple de BackboneFire

```
<script src="https://cdn.firebase.com/libs/backbonefire/0.5.1/backbonefire.min.js"></script>
```

```
var MaCollection = Backbone.Firebase.Collection.extend({  
  model: monModel,  
  url: "url de la db"  
});
```

```
maCollection.create({cle1 : valeur1, cle2 : valeur2});
```

Coté serveur

- Basé sur l'API REST
- PHP, Python, Java, Perl, Ruby ...
- Utilisation de librairies

Exemple coté serveur

- Java : Firestore4j alpha
 - `Firestore db = new Firestore("adresse");`
 - `db.put/post(HashMap)`
 - `db.get()`
- PHP : firebase-php
 - `$fb = new Firestore(url, token);`
 - `$fb-> set / get / update / push / delete`

Sécurité

- Authentification
- Permissions
- Validation des données

Authentication

Dashboard VIEWING MY FIRST APP Account Settings

User Login & Authentication

Authorized Domains for OAuth Redirects

For third-party authentication (Facebook, Twitter, GitHub or Google), your domain must be whitelisted below for OAuth redirects. Default authorized domains include: localhost, 127.0.0.1, respndent-heat-5777.firebaseio.com

Session Length ⓘ 24 Hours

Email & Password Facebook Twitter GitHub Google Anonymous Custom

Enable Email & Password Authentication

Configuring Email & Password

Firebase makes it easy to integrate email and password authentication into your app. [Learn more](#)

Password Resets

From: noreply Password Reset Emails

Registered Users [refresh list](#) Add User

User ID	Email
1	the.unick@hotmail.com

Search for user...

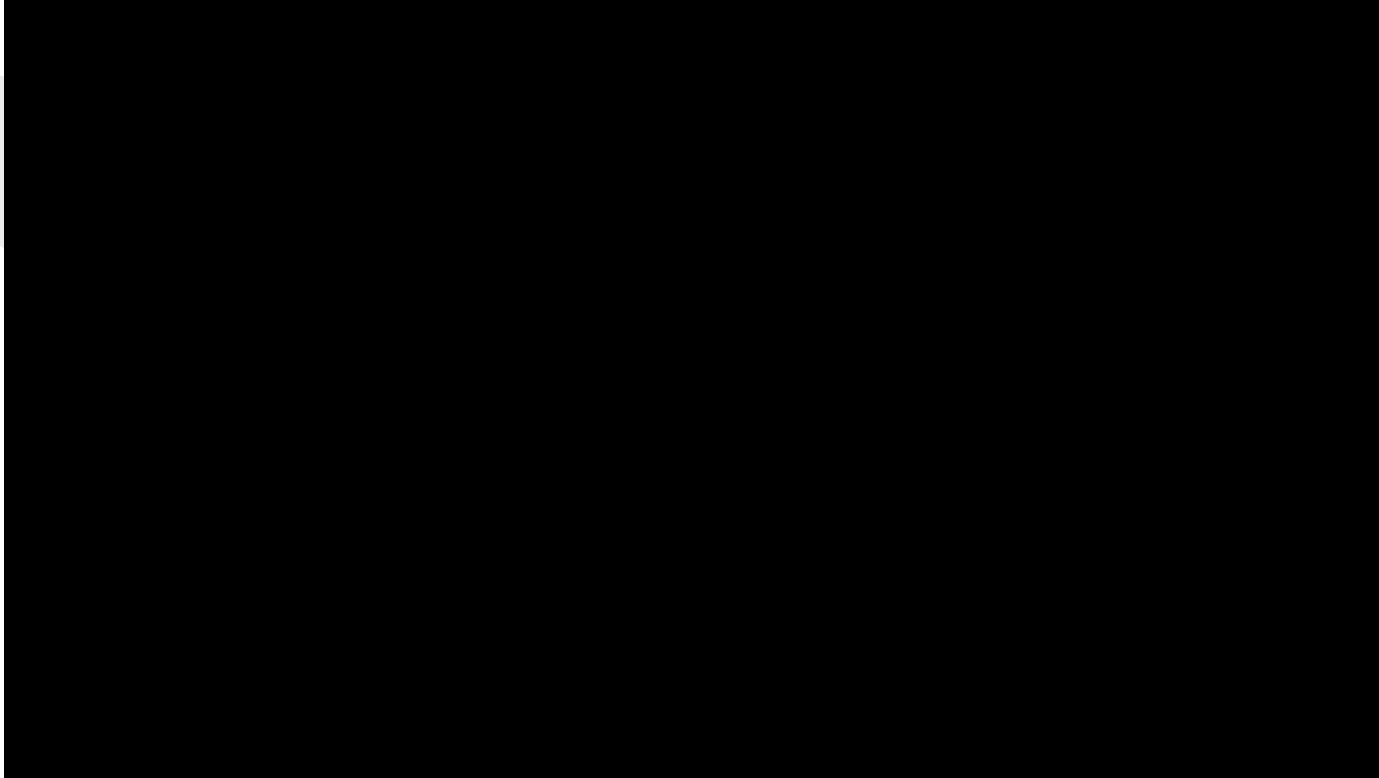


Authentication

- Simple
- Anonyme

```
ref.authWithPassword({
  email: req.body.email,
  password: req.body.mdp
}, function(error, authData) {
  if (error) {
    console.log("Erreur lors de l'authentification!", error);
  } else {
    console.log("Bienvenue :) ", authData);
  }
});
```

Authentication



Authentication

- Via autres API
- Personnalisé

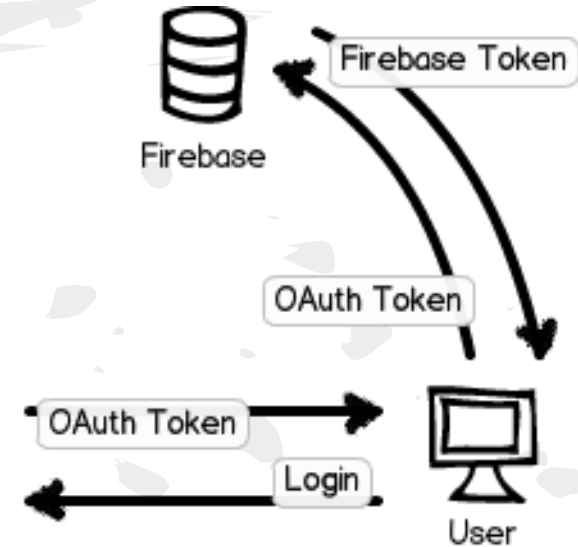


Email & Password **Facebook** Twitter GitHub Google Anonymous Custom

Enable Facebook Authentication

Facebook App Id:

Facebook App Secret:



Permissions

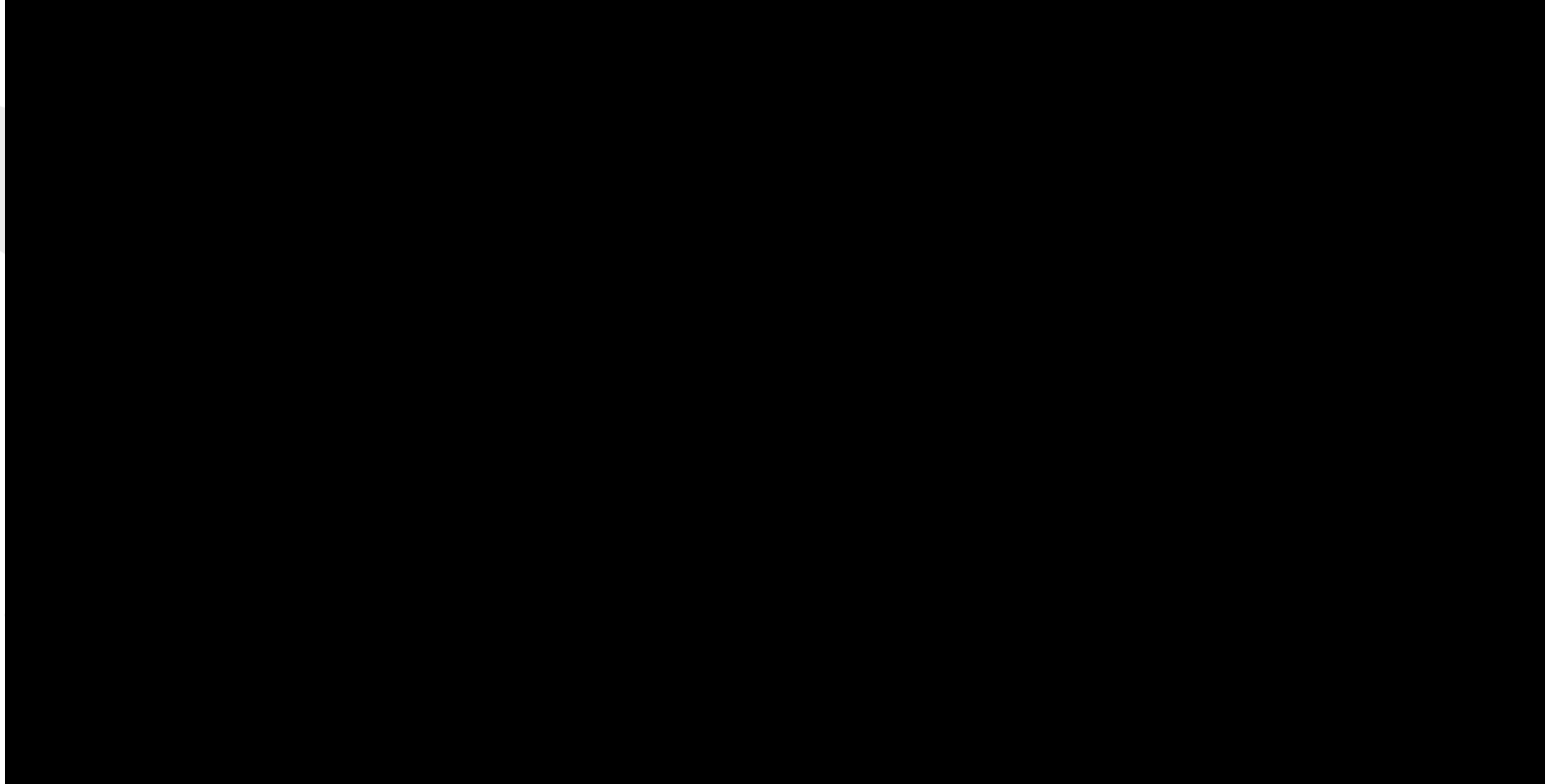
- Règles de sécurité
 - Gestion des accès lecture/écriture
- Gestion des règles en cascade



FIREBASE RULES

```
1 {  
2   "rules": {  
3     ".read": true,  
4     ".write": "auth != null" //seul les utilisateurs loggés ont le droit d'écrire  
5   }  
6 }
```

Permissions



Validation des données

- Règles de sécurité
 - Contraintes
 - Données cohérentes
- Pas en cascade

```
"author": {  
  ".validate": "newData.val() ==auth.uid"  
},  
"content": {  
  ".validate": "newData.isString()  
}
```

Construction de règles



.read
.write
.validate

Les variables:

now

root

data

newData

auth

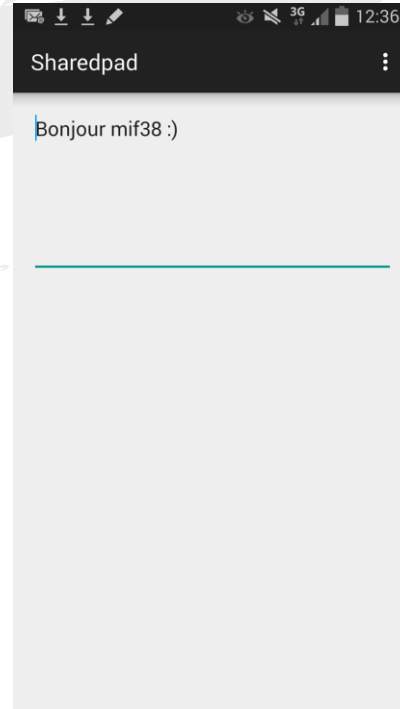
$\$<maVariable>$

Conclusion/résumé



Google play

- “Sharedpad”



Sharedpad, partie lecture

```
firebase.addChildEventListener(new ChildEventListener() {  
    // Retrieve new posts as they are added to Firestore  
    @Override  
    public void onChildAdded(DataSnapshot snapshot, String previousChildKey) {  
        editTextPad.setText((String)snapshot.getValue());  
    }  
  
    @Override  
    public void onChildChanged(DataSnapshot dataSnapshot, String s) {  
        int pos = editTextPad.getSelectionStart();  
        editTextPad.setText(dataSnapshot.getValue().toString());  
        if(pos <= editTextPad.length())  
            editTextPad.setSelection(pos);  
        else  
            editTextPad.setSelection(editTextPad.length());  
    }  
}
```

SharedPad, partie écriture

```
TextWatcher watcher= new TextWatcher() {  
    public void afterTextChanged(Editable s) {  
        firebase.child("pad").setValue(editTextPad.getText().toString());  
    }  
    public void beforeTextChanged(CharSequence s, int start, int count, int after) {  
        //Do something or nothing.  
    }  
    public void onTextChanged(CharSequence s, int start, int before, int count) {  
        //Do something or nothing  
    }  
};  
  
editTextPad.addTextChangedListener(watcher);
```

Recherches

- Pas de LIKE
- Utilisation de elasticSearch avec node-elasticsearch-client
 - installation de l'API
 - indexer les données
 - envoyer la requête avec elasticSearch
 - écouter la réponse sur mafb.firebaseio.com/search

Hosting

- Uniquement coté client
- CDN “global”



```
npm install -g firebase-  
tools  
firebase init  
firebase deploy
```

La concurrence

PubNub®



PUSHER
POWERING REALTIME

Pour aller plus loin ...

- Applications sans côté serveur



Ressources

Firestore:

- <https://www.firebaseio.com/docs/>

RFC6455:

- <https://tools.ietf.org/html/rfc6455#page-6>

WebSocket:

- http://liris.cnrs.fr/~lmedini/enseignement/MIF38/CM/CM/CM_WebSockets.html
- <http://www.jmdoudoux.fr/java/dej/chap-websockets.htm>

Requêtes “complexes”

- <https://www.firebaseio.com/blog/2013-10-01-queries-part-one.html>