1. apa itu firebase?

suatu database yang memakai konsep nosql dan digunakan untuk realtime database, data yang dikembalikan berupa json.

Sebelum memulainya kita membuat project di firebase console,

Ini adalah cth configurasinya,

// Set the configuration for your app  
  // TODO: Replace with your project's config object  
  var config = {  
    apiKey: "***apiKey***",  
    authDomain: "***projectId***.firebaseapp.com",  
    databaseURL: "https://***databaseName***.firebaseio.com",  
    storageBucket: "***bucket***.appspot.com"  
  };  
  firebase.initializeApp(config);  
  
  // Get a reference to the database service  
  var database = firebase.database();

Ini adalah cth jsonnya,

{  
  "users": {  
    "alovelace": {  
      "name": "Ada Lovelace",  
      "contacts": { "telpon": “123456789”},  
    },  
    "ghopper": { ... },  
    "eclarke": { ... }  
  }  
}

2. dalam menggunakannya memiliki 3 event.

onchild dan onvalue adalah event untuk memangil data dengan realtime. hanya saja perbedaannya.

1. onchild = data yang dipangil berupa satuan. cth: ketika kita memanggil 10 data,

data yang ditampilkan tidak lagsung diberikan 10 10nya,

jadi diberikan satu satu selama 10 kali,

tanpa perlu dilooping menggunakan for atau foreach.

Add = hanya mengembalikan data yang di insert, istilah firebasenya push atau set.

Change = hanya mengembalikan data yang di edit, istilah firebasenya update.

Delete = mengembalikan data yang di delete, istilah firebasenya remove.

{  
  "users": {  
    "alovelace": {//key  
      "name": "Ada123 Lovelace",  
      "contacts": “123456789”,  
    },  
    "ghopper": { ... },  
    "eclarke": { ... }  
  }  
}

var commentsRef = firebase.database().ref('users/’);  
commentsRef.on('child\_added', function(data) {  
  addCommentElement(data.key, data.val().name, data.val().contact);  
});  
  
commentsRef.on('child\_changed', function(data) {  
  setCommentValues(data.key, data.val().name, data.val().contact);  
});  
  
commentsRef.on('child\_removed', function(data) {  
  deleteComment(data.key);  
});

commentsRef.on('child\_moved', function(data) {  
  moveComment(data.key);  
});

2. onvalue = data yang dipanggil sekaligus, cth: ketika kita memanggil 10 data,

data yang di tampilkan harus menggunakan looping seperti foreach atau for.

ref.on('value', function(snapshot) {  
  snapshot.forEach(function(childSnapshot) {  
    var childKey = childSnapshot.key;  
    var childData = childSnapshot.val().name;  
    // ...  
  });  
});

3. oncevalue = memamngil 1 data dengan 1 kali panggil (tidak realtime) data yang sifatnya statis.

ref.once('value', function(snapshot) {  
  snapshot.forEach(function(childSnapshot) {  
    var childKey = childSnapshot.key;  
    var childData = childSnapshot.val().name;  
    // ...  
  });  
});

contoh inser, edit, delete

{  
  "users": {  
    "alovelace": {//key  
      "name": "Ada123 Lovelace",  
      "contacts": “123456789”,  
    },  
    "ghopper": { ... },  
    "eclarke": { ... }  
  }  
}

function writeUserData(key, name, email, imageUrl) {

firebase.database().ref('users/' + key).set({

name: name,

contacts: email

});

};

function deleteData(key){

firebase.database().ref('/users/' + key).remove();

}

function updateData(key,name){

firebase.database().ref('/users/' + key).update({ name: name });

}

3. didalam onchild memiliki 3 event.

1. childadded = event untuk memaggil data saat tiap kali di add.

2. childchanged = merubah data di tampilan.

3. childremove = menghapus data saat mendelete data saat delete.

4. event order

Orderbychild

Orderbykey

Orderbyvalue

5. where name = "dian"

ref("users").orderbychild("name").equalto("jo");

6. limit ada 2

LimitToFirst (ambil data dari atas)

LimitToLast ambil data dari bawah.

Contohnya

ref("users").orderbychild("name").limitToFirst(10);

5. ketika lagi offline data transaksi bisa di padding jika online bisa di running kembali.

Aplikasi Firebase tetap berfungsi meskipun aplikasi Anda kehilangan koneksi jaringan untuk sementara. Kami menyediakan beberapa fitur bantuan untuk memonitor kehadiran dan melakukan sinkronisasi status lokal dengan status server, yang diperkenalkan dalam dokumen ini.