

App and Database Integration

Almost all good apps need a database

Device and Network Storage

- ✦ Two type of Device Storage
 - ✦ Local (Temporary Cache)
 - ✦ Database

Network Storage

- ✦ Database or Static files hosted on a server accessed via the network connection.
- ✦ Multiple Technology Solutions

Possible Solutions

- ✦ Requires a Scripting Language and Database

- ✦ SCRIPTING
LANGUAGE

- ✦ PHP

- ✦ Python

- ✦ node.js

Databases

- ✧ Relational

- ✧ mySQL

- ✧ postGIS

- ✧ many others

- ✧ Document Based

- ✧ mongoDB

- ✧ rethinkDB

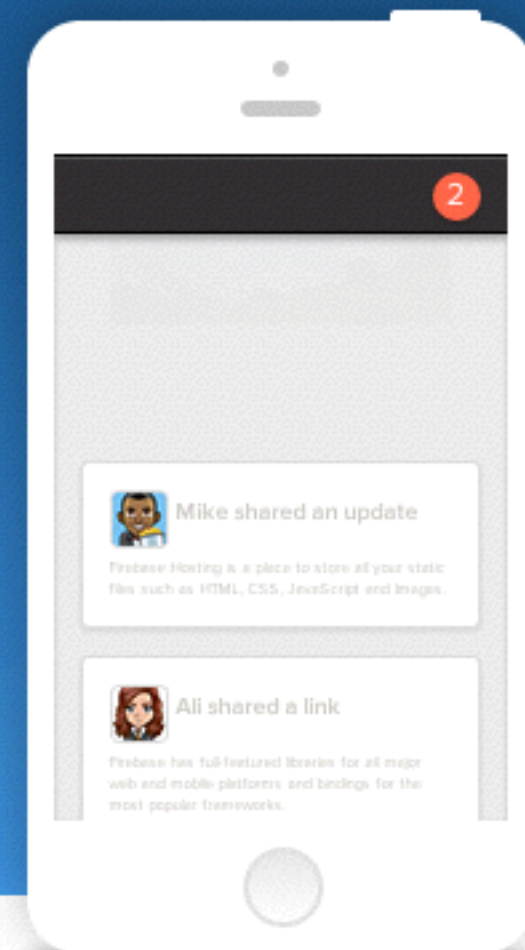
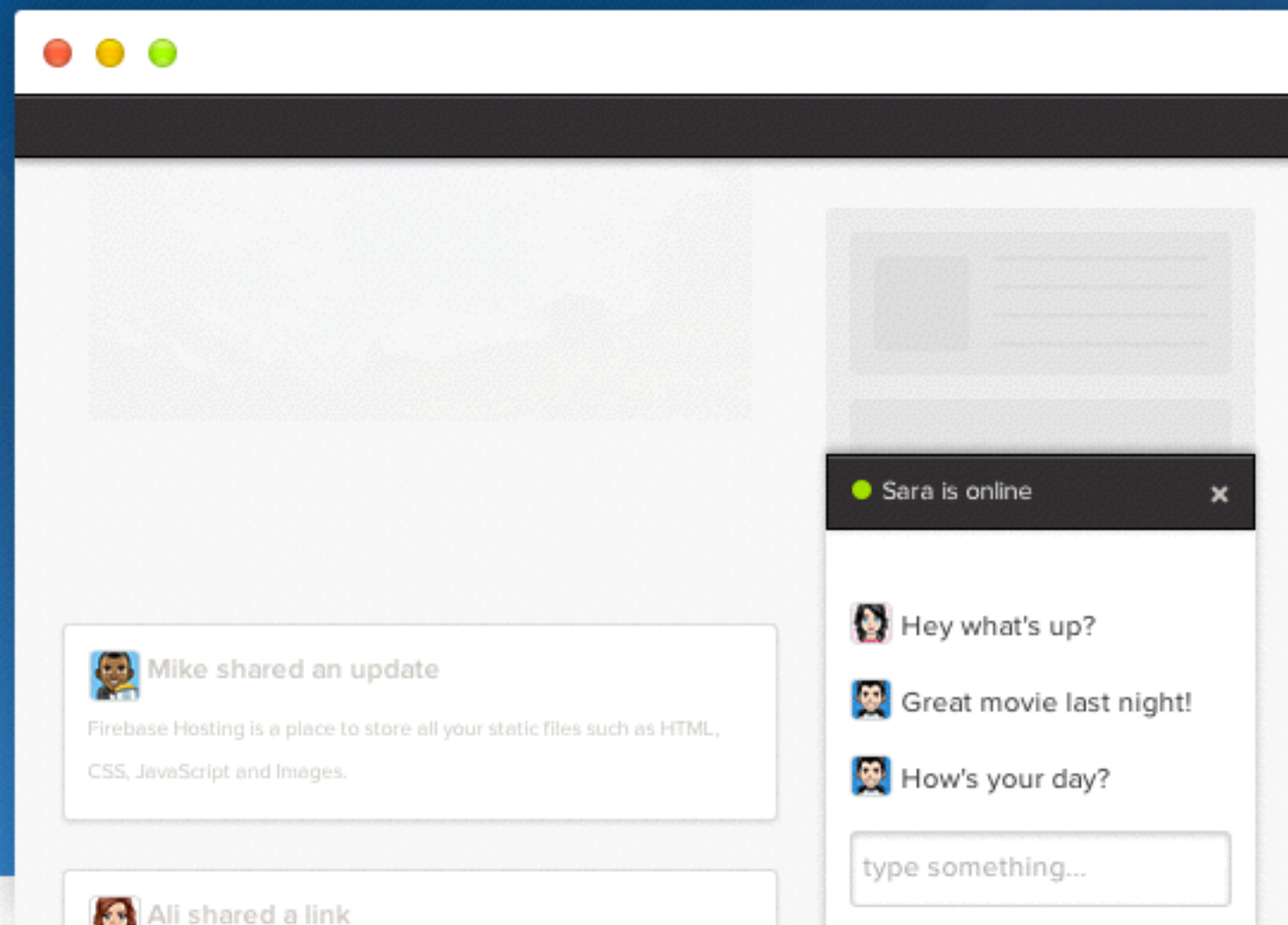
- ✧ few others

Another option...

- ✧ Data Services
 - ✧ a hosted API to a database
 - ✧ Firebase

Build Realtime Apps

A powerful API to store and sync data in realtime.



THE REALTIME APP PLATFORM

Store & Sync Data Instantly

Signup and Create an App

The screenshot shows the Firebase Account Dashboard in a web browser. The browser's address bar displays `https://www.firebase.com/account/#/`. The dashboard header includes the 'Dashboard' title and an 'Account Settings' link. The main content area features a large 'Welcome to Firebase' message, stating 'Your account has been created successfully! Take a look at these resources to help you get started.' Below this message are four buttons: '5 min. tutorial', 'Quickstarts', 'Examples', and 'API Docs'.

Below the welcome message, there is a 'CREATE NEW APP' form and two existing app cards. The 'CREATE NEW APP' form has fields for 'APP NAME' and 'APP URL', and a 'CREATE NEW APP' button. The two existing app cards are for 'PHOTOTRIPS' and 'TESTINGAPP'. Both cards show the plan as 'DEVELOPMENT ONLY - HACKER PLAN' and provide the app's Firebase URL and a 'Set up Hosting' button. Each card also has 'Manage App' and 'Upgrade Plan' buttons, and an 'Add a developer' link at the bottom.

App Name	Plan	Firebase URL	Set up Hosting	Manage App	Upgrade Plan	Add a developer
PHOTOTRIPS	DEVELOPMENT ONLY - HACKER PLAN	phototrips.firebaseio.com	Set up Hosting NEW	Manage App	Upgrade Plan	Add a developer
TESTINGAPP	DEVELOPMENT ONLY - HACKER PLAN	sweltering-heat-2568.firebaseio.com	Set up Hosting NEW	Manage App	Upgrade Plan	Add a developer

The Data

The screenshot shows the Firebase console interface for a project named 'Forge: Firebase Graphical'. The browser address bar displays 'https://phototrips.firebaseio.com'. The left sidebar contains navigation links for Dashboard, Data, Security Rules, Simulator, Analytics, Login & Auth, Hosting, and Secrets. The main content area is titled 'VIEWING PHOTOTRIPS' and includes buttons for 'Expand Data', 'Collapse Data', 'Import JSON', and 'Export JSON'. A tree view shows the database structure under the 'phototrips' root, with three child nodes: 'locations', 'North Korea', 'Thailand', and 'UAE'. Each location node contains fields for 'cost', 'date', 'desc', and 'photos'. A legend on the right indicates that yellow represents 'Changed', green represents 'Added', red represents 'Deleted', and blue represents 'Moved'.

```
graph LR
    phototrips --> locations
    phototrips --> North_Korea[North Korea]
    phototrips --> Thailand
    phototrips --> UAE
```

phototrips

- locations**
 - North Korea**
 - cost: 9000
 - date: 20140901
 - desc: "See a place few westerners have ever seen."
 - photos: "https://c1.staticflickr.com/3/2257/1706418822_6..."
 - Thailand**
 - cost: 2500
 - date: 20150314
 - desc: "Travel to Chiang Mai and Bangkok with internati..."
 - photos: "https://c1.staticflickr.com/3/2257/1706418822_6..."
 - UAE**
 - cost: 4500
 - date: 20150205
 - desc: "See and experience the arab culture during the ..."
 - photos: "https://c1.staticflickr.com/3/2371/2432735649_5..."

Legend

- Changed
- Added
- Deleted
- Moved

The Data (manual and load json)

The screenshot shows the Firebase Graphical interface for a database named 'PHOTOTRIPS'. The interface includes a sidebar with navigation options: Dashboard, Data, Security Rules, Simulator, Analytics, Login & Auth, Hosting, and Secrets. The main content area displays the database structure, which is organized into a tree view. The 'phototrips' node contains three sub-nodes: 'locations', 'North Korea', 'Thailand', and 'UAE'. Each sub-node has associated metadata such as 'cost', 'date', 'desc', and 'photos'. A legend on the right side of the interface indicates the status of the data: Changed (yellow), Added (green), Deleted (red), and Moved (blue).

Dashboard VIEWING PHOTOTRIPS Account Settings

Expand Data Collapse Data Import JSON Export JSON

PHOTOTRIPS

phototrips

- locations
 - North Korea
 - cost: 9000
 - date: 20140901
 - desc: "See a place few westerners have ever seen."
 - photos: "https://c1.staticflickr.com/3/2257/1706418822_6..."
 - Thailand
 - cost: 2500
 - date: 20150314
 - desc: "Travel to Chiang Mai and Bangkok with internati..."
 - photos: "https://c1.staticflickr.com/3/2257/1706418822_6..."
 - UAE
 - cost: 4500
 - date: 20150205
 - desc: "See and experience the arab culture during the ..."
 - photos: "https://c1.staticflickr.com/3/2371/2432735649_5..."

Legend

- Changed
- Added
- Deleted
- Moved

JavaScript Based.

- ✦ open index.html
- ✦ open app.js

```
24
25 function loadTrips(){
26     // Create our Firebase reference
27     var tripsToDisplay = 3;
28     var tripListRef = new Firebase('https://phototrips.firebaseio.com//locations');
29     var tripListView = tripListRef.limit(tripsToDisplay);
30
```


Data Connection API

```
25 function loadTrips(){
26     // Create our Firebase reference
27     var tripsToDisplay = 3;
28     var tripListRef = new Firebase('https://phototrips.firebaseio.com//locations');
29     var tripListView = tripListRef.limit(tripsToDisplay);
30
31     tripListRef.once('value', function(dataSnapshot) {
32         var listItems = "";
33         // store dataSnapshot for use in below examples.
34         tripListView = dataSnapshot.val();
35         console.log(tripListView);
36     });
}
```


SnapShot with Loop

```
31 tripListRef.once('value', function(dataSnapshot) {  
32     var listItems = "";  
33     // store dataSnapshot for use in below examples.  
34     tripListView = dataSnapshot.val();  
35     console.log(tripListView);  
36  
37  
38     $.each(tripListView, function(key, val) {  
39         var location = [];  
40  
41         console.log('Key: ' + key + ' Val: ' + val)  
42         location.push(key);  
43         $.each(val, function(key, val) {  
44             location.push(val);  
45         });  
46     });  
47  
48  
49  
50  
51  
52     });  
53  
54     console.log(listItems);  
55  
56 });  
57
```


Write some HTML to page

```
37
38 $.each(tripListView, function(key, val) {
39     var location = [];
40
41     console.log('Key: ' + key + ' Val: ' + val)
42     location.push(key);
43     $.each(val, function(key, val) {
44         location.push(val);
45     });
46
47
48
49     listItems += '<li><a href="#">'
50     listItems += '<h2>' + location[0] + "</h2>";
51     listItems += '<p>' + location[3] + '</p></a>'
52
53
54
55 });
56
57 console.log(listItems);
58 $("#trip-list").html(listItems);
59 $("#trip-list").listview("refresh");
60 });
61
62
```


It works! Dynamic App!

Test Query

```
73  
74 function go() {  
75     var searchTerm = prompt('Country?', 'Thailand');  
76     checkIfUserExists(searchTerm);  
77 }  
78
```


Test if in data?

```
78 function go() {  
79     var searchTerm = prompt('Country?', 'Thailand');  
80     checkIfUserExists(searchTerm);  
81 }  
82  
83 var DATA_LOCATION = 'https://phototrips.firebaseio.com//locations';  
84  
85 function termExistsCallback(searchTerm, exists) {  
86     if (exists) {  
87         alert('user ' + searchTerm + ' exists!');  
88     } else {  
89         alert('user ' + searchTerm + ' does not exist!');  
90     }  
91 }  
92  
93 // Tests to see if /users/<searchTerm> has any data.  
94 function checkIfUserExists(searchTerm) {  
95     var termRef = new Firebase(DATA_LOCATION);  
96     termRef.child(searchTerm).once('value', function(snapshot) {  
97         var exists = (snapshot.val() !== null);  
98         console.log(exists);  
99         termExistsCallback(searchTerm, exists);  
100     });  
101 }  
102
```


Great job!

Now for your own learning....

Do Firebase Leaderboard tutorial

[https://www.firebase.com/tutorial/#session/
rtkdkdn77wt](https://www.firebase.com/tutorial/#session/rtkdkdn77wt)

Homework

- ✦ Plan and outline your data structure
- ✦ Input the starting data for your app into Firebase.
- ✦ Write a Firebase query to display all records on an html page just like the example in class.