

# 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



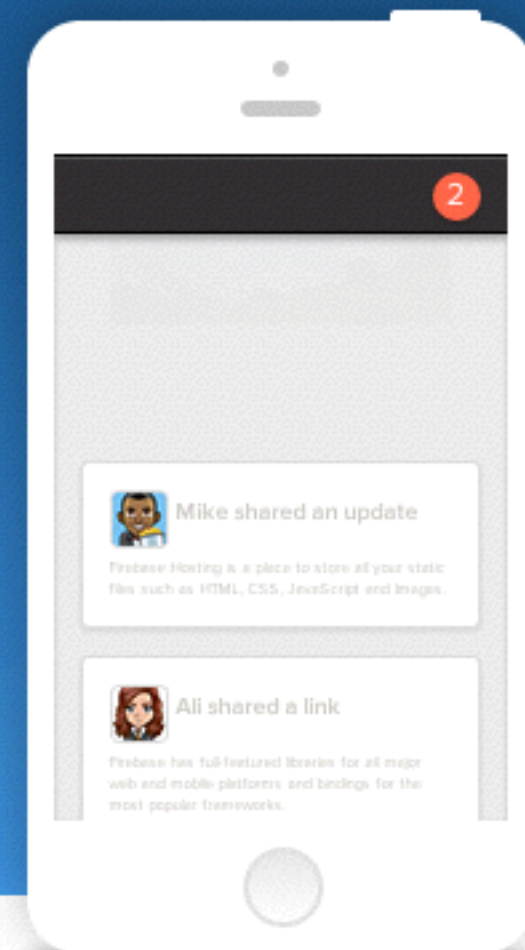
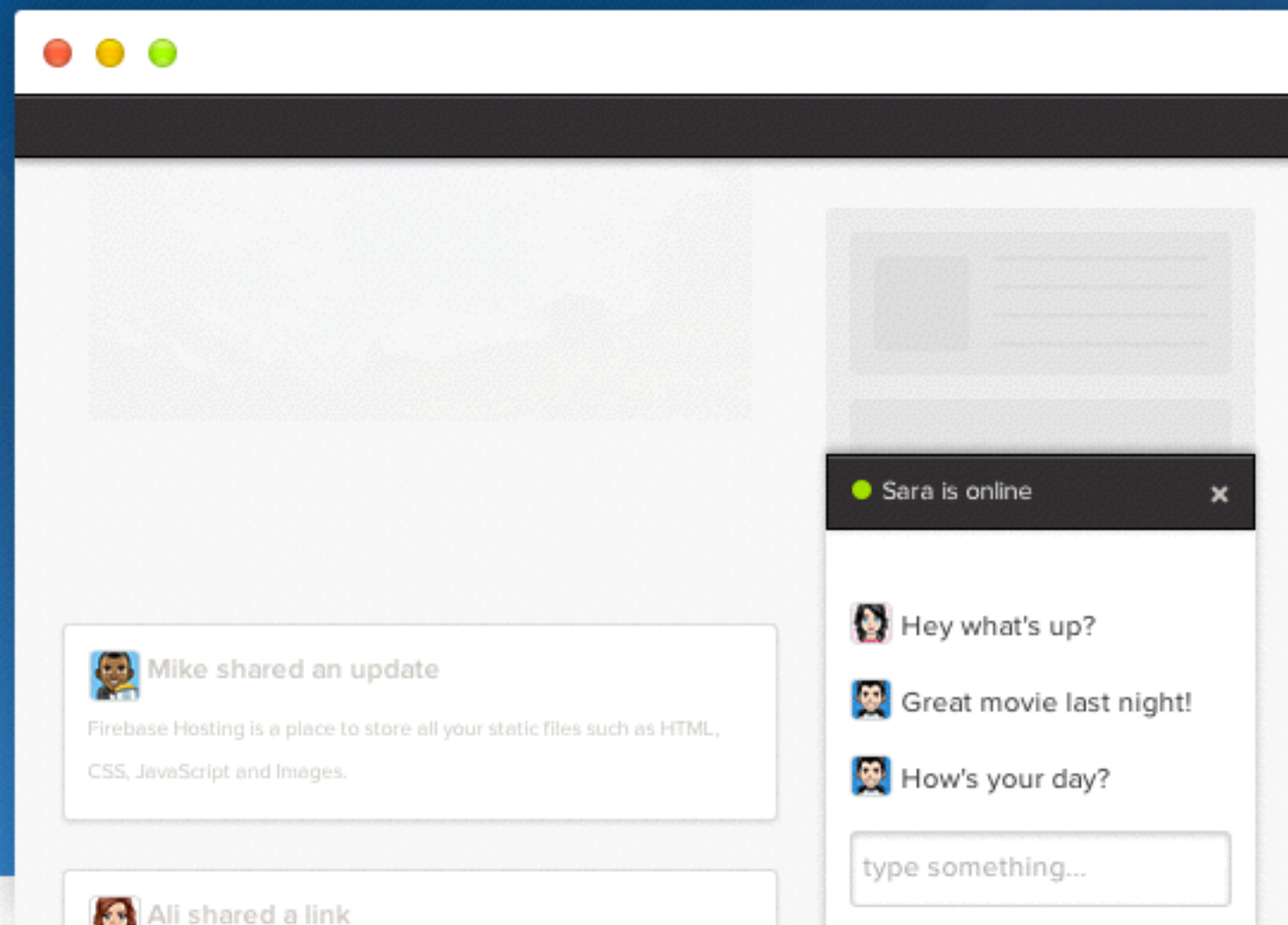
# Another option...

- ✧ 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 the app URL as `phototrips.firebaseio.com` and `sweltering-heat-2568.firebaseio.com` respectively. Each card has a 'Set up Hosting' button with a 'NEW' badge, a 'Manage App' button, and an 'Upgrade Plan' button. At the bottom of each card is an 'Add a developer' button.



# The Data

The screenshot shows the Firebase Realtime Database console for a project named 'Forge: Firebase Graphical'. The URL 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 area shows the 'PHOTOTRIPS' database structure. A legend on the right indicates changes: Changed (yellow), Added (green), Deleted (red), and Moved (blue).

```
graph LR
    phototrips --> locations
    locations --> North Korea
    locations --> Thailand
    locations --> UAE
    North Korea --> cost_9000["cost: 9000"]
    North Korea --> date_20140901["date: 20140901"]
    North Korea --> desc_1["desc: 'See a place few westerners have ever seen.'"]
    North Korea --> photos_1["photos: 'https://c1.staticflickr.com/3/2257/1706418822_6...']"]
    Thailand --> cost_2500["cost: 2500"]
    Thailand --> date_20150314["date: 20150314"]
    Thailand --> desc_2["desc: 'Travel to Chiang Mai and Bangkok with internati...']"]
    Thailand --> photos_2["photos: 'https://c1.staticflickr.com/3/2257/1706418822_6...']"]
    UAE --> cost_4500["cost: 4500"]
    UAE --> date_20150205["date: 20150205"]
    UAE --> desc_3["desc: 'See and experience the arab culture during the ...']"]
    UAE --> photos_3["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 under the 'PHOTOTRIPS' tab. The structure is as follows:

- 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..."

A legend on the right side of the interface indicates the status of the data:

- Changed (Yellow)
- Added (Green)
- Deleted (Red)
- Moved (Blue)



# 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.



# Ways to Update the Data



# 4 ways

## Ways to Save Data

<b>set( )</b>	Write or replace data to a <b>defined path</b> , like <code>messages/users/&lt;username&gt;</code>
<b>update( )</b>	Update some of the keys for a defined path without replacing all of the data
<b>push( )</b>	<b>Add to a list of data</b> in Firebase. Every time you call <code>push()</code> Firebase generates a unique ID, like <code>messages/users/&lt;unique-user-id&gt;/&lt;username&gt;</code>
<b>transaction( )</b>	Use our transactions feature when working with complex data that could be corrupted by concurrent updates



# Set up Function

```
78 function updateCountry(){  
79     var value = prompt('Change the cost to:', '5000');  
80  
81     var ref = new Firebase("https://phototrips.firebaseio.com//locations");  
82     console.log(ref);  
83  
84 }
```

```
18 $( document ).ready(function() {  
19     //for browser use only  
20     console.log( "ready!" );  
21     //function to populate menu  
22     loadTrips();  
23  
24     updateCountry();  
25  
26 });
```



# update()

update()

Update some of the keys for a defined path without replacing all of the data

```
78 function updateCountry(){
79     var value = prompt('Change the cost to:', '5000');
80
81     var ref = new Firebase("https://phototrips.firebaseio.com//locations");
82     console.log(ref);
83     var countryRef = ref.child("Thailand");
84
85     countryRef.update({
86         "cost" : value
87     });
88     console.log("done running updateCountry()")
89 }
90
```



# push()

push()

Add to a list of data in Firebase. Every time you call `push()` Firebase generates a unique ID, like `messages/users/<unique-user-id>/<username>`

```
91
92 function postCountry(){
93     ref.push("USA");
94     var countryRef = ref.child("USA");
95     countryRef.push({
96         cost: "0",
97         date: "20140401",
98         desc: "See your homeland!",
99         photo: "https://c1.staticflickr.com/3/2371/2432735649_518d41a061.jpg";
100     });
101
102
103 }
104
```



# Same by different

```
91 function postCountry(){
92     //ref.push("USA");
93     //var countryRef = ref.child("USA");
94     ref.push({
95         "USA":{
96             cost: "0",
97             date: "20140401",
98             desc: "See your homeland!",
99             photo: "https://c1.staticflickr.com/3/2371/2432735649_518d41a0
100         }
101     });
102
103
104 }
105
```



# Another way...

```
91 function postCountry(){
92     //ref.push("USA");
93     //var countryRef = ref.child("USA");
94     ref.push({
95         name: "USA",
96         cost: "0",
97         date: "20140401",
98         desc: "See your homeland!",
99         photo: "https://c1.staticflickr.com/3/2371/2432735649_518d41a06
100
101     });
102
103
104 }
105
```



# set()

set( )

Write or replace data to a defined path, like `messages/users/<username>`

```
105 function setCountry(){
106
107     ref.set({
108         "USA":{
109             cost: "0",
110             date: "20140401",
111             desc: "See your homeland!",
112             photo: "https://c1.staticflickr.com/3/2371/2432735649_518d41a06
113         }
114     });
115
116
117
118 }
119
```



# Homework

- ✦ Working on your app project, connect some element to your database.
- ✦ After Fall Break you will have the entire app connected to a database.