A hand holding a smartphone is the central focus. The phone's screen displays the text 'Storing data in the mobile browser'. The background is a dark blue gradient with glowing binary code (0s and 1s) and a faint, semi-transparent image of a computer keyboard. The overall aesthetic is high-tech and digital.

Storing data in the mobile browser

Andy Gup

Why attend this demo theater?

Have storage needs > 5MBs

Want to store data types other than Strings

Don't want to manually serialize/deserialize

Looking into offline JavaScript



Advanced browser storage
Coding patterns
Use Cases

Common questions...

Can I store more than 5MBs? **Yes!**

How much data can I store? **It varies!**

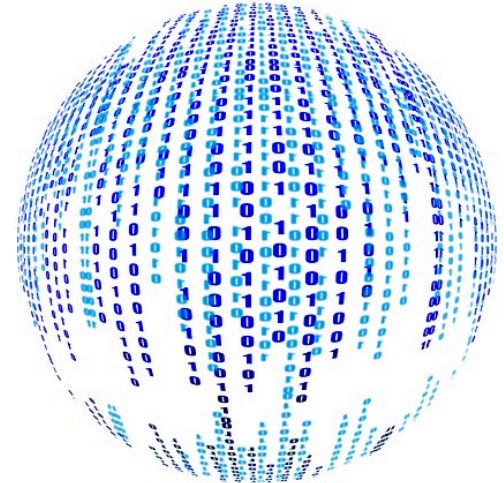
What type of data??

JSON Objects (not serialized)

Complex Objects (difficult serialize/deserialize)

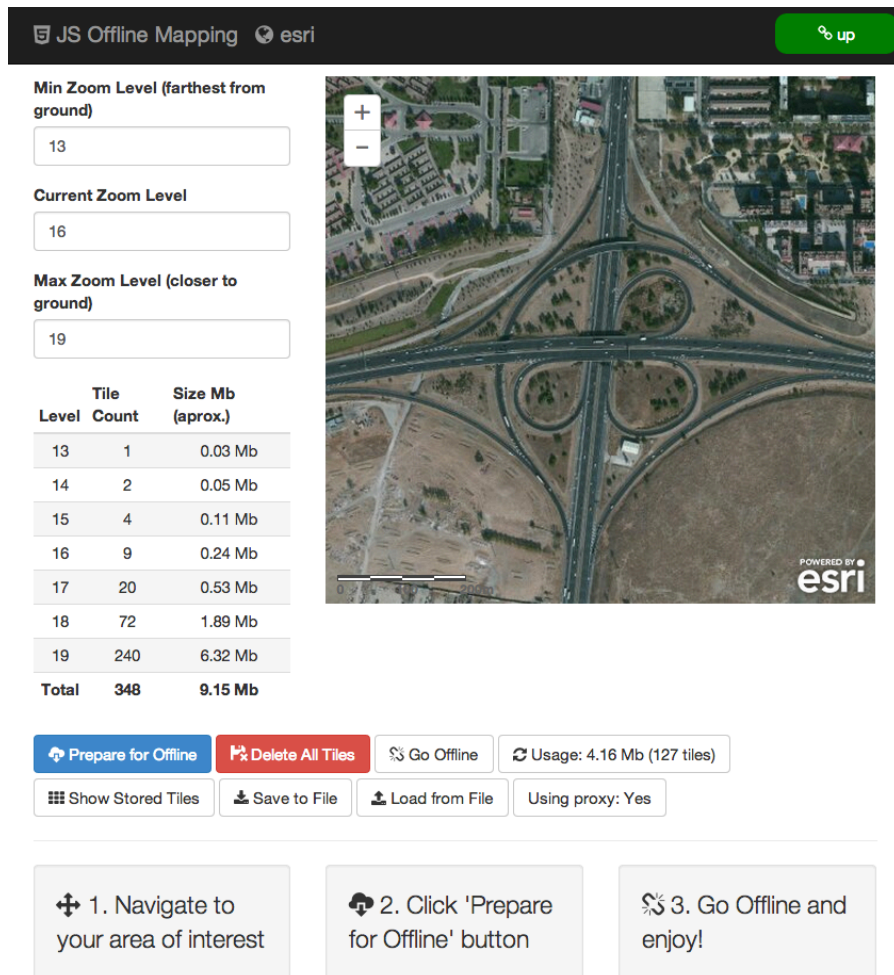
Binary data (e.g. images, files)

Arrays



Demo

[Github.com/esri/offline-editor-js](https://github.com/esri/offline-editor-js)



JS Offline Mapping esri up

Min Zoom Level (farthest from ground)
13

Current Zoom Level
16

Max Zoom Level (closer to ground)
19

Level	Tile Count	Size Mb (approx.)
13	1	0.03 Mb
14	2	0.05 Mb
15	4	0.11 Mb
16	9	0.24 Mb
17	20	0.53 Mb
18	72	1.89 Mb
19	240	6.32 Mb
Total	348	9.15 Mb

Map: Aerial view of a highway interchange with a scale bar and a 'POWERED BY esri' logo.

Buttons: Prepare for Offline, Delete All Tiles, Go Offline, Usage: 4.16 Mb (127 tiles), Show Stored Tiles, Save to File, Load from File, Using proxy: Yes

1. Navigate to your area of interest
2. Click 'Prepare for Offline' button
3. Go Offline and enjoy!

Tell me how??

IndexedDB - native

ApplicationCache – native (HTML, CSS, JavaScript)

~~WebSQL~~ – native

LocalStorage (Mozilla. LocalStorage fallback)

PouchDB (JS CouchDB + remote sync)

Polyfill?

Exposes IndexedDB API on Web SQL browsers

<https://github.com/axemclion/IndexedDBShim>

IndexedDB - CR

Method of storing data client-side, allows indexed database queries.

U.S.A.	51.28% + 30.37% =	81.65%
unprefixed:	51.16% + 29.81% =	80.97%
Global	60.39% + 17.77% =	78.17%
unprefixed:	60.16% + 17.18% =	77.34%

IE	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Chrome for Android
		36	5.1					
8	31	37	6.1				4.3	
9	34	38	7		7.1		4.4	
¹ 10	35	39	² 7.1		² 8		4.4.4	
¹ 11	36	40	² 8	27	² 8.1	8	37	40
TP	37	41		28				
	38	42		29				
	39	43						

[Notes](#)
[Known issues \(1\)](#)
[Resources \(7\)](#)
[Feedback](#)

¹ Partial support in IE 10 & 11 refers to a number of subfeatures **not being supported**.
² Partial support in iOS 8 refers to **seriously buggy behavior**.

How does IndexedDB work?

Key-Value pairs

Search Indexes

NoSQL

Cursors

Asynchronous via callbacks

Notifications via DOM events

Key/Value pairs?

Key

String, Number, Date, Array

Value (partial list)

String

Object

Array

Blob

ArrayBuffer

UintArray

File

How to use? IndexedDBShim.js

Required for Safari 7 on iOS and Mac

Safari 7 only comes with Web SQL

```
<script src="IndexedDBShim.js"></script>
```

How to use? Set Prefixes

```
window.indexedDB = window.indexedDB ||  
    window.mozIndexedDB ||  
    window.webkitIndexedDB ||  
    window.msIndexedDB;
```

```
var transaction = window.IDBTransaction ||  
    window.webkitIDBTransaction;
```

How to use? Validate functionality

```
this.isSupported = function(callback){
```

```
    if(!window.indexedDB && !window.openDatabase){  
        return callback(false);
```

```
    }
```

```
    else{
```

```
        testFunctionality(function(success,result){
```

```
            return callback(success,result);
```

```
        });
```

```
    }
```

```
};
```




How to use? Open database

```
var db = null;
```

```
var request =  
  indexedDB.open("customerDB", /*version*/ 2);
```



```
request.onsuccess = function(event){  
  db = event.target.result;  
  callback(true);  
}
```



```
request.onerror = function(error){  
  callback(false, error);  
}
```

How to use? Transaction requests

```
var transaction =  
  db.transaction(["customerDB"], "readwrite");
```



```
var objectStore =  
  transaction.objectStore("customerDB");
```



```
var request =  
  objectStore.put({test:1}, "myKey");
```



```
// Did transaction request work or not?  
request.onerror = function(error) { . . . }  
request.onsuccess = function(event) { . . . }
```



How to use? Write data

add(anyValue, /*optional*/ key)

Write only unique values

Duplicate entries fail with error

put(anyValue, /*optional*/ key)

Overwrites existing entries with same key

How to use? Write data

```
request =  
  objectStore.put({test:1}, "myKey");  
  
request.onsuccess = function(event){  
  callback(true, event.target.result);  
}  
  
request.onerror = function(error){  
  callback(false, error);  
}
```

How to use? Get data

```
request =  
  objectStore.get("aUniqueKey");
```

```
request.onsuccess = function(event){  
  callback(true, event.target.result);  
}
```

```
request.onerror = function(error){  
  callback(false, error);  
}
```

Will IndexedDB work offline?

YES!

Can also be used with Application Cache.

Database size – how big??

General suggestion (smartphones/tablet):

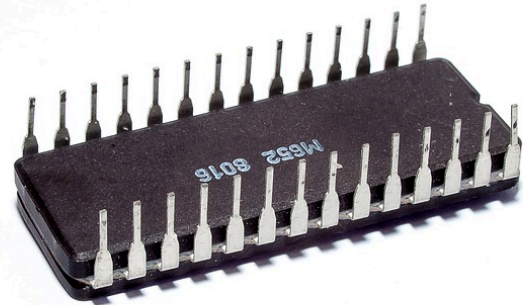
- Less than 100 MB

Why? Depends on:

- Remaining free memory
- How much memory used by browser
- How many tabs are already open
- If other applications already running

Can I use too much memory?

YES!



The device OS will shutdown the browser

Greater potential for data corruption/loss

- If shutdown occurs during a 'write' operation

Andy Gup, Esri
Developer Evangelist
<http://www.andygup.net>
agup@esri.com
[@agup](#)

