

# HTML5 Web Storage, IndexedDB and File System

IndexedDB: Keys, Capacity, Performance and Versions

**Craig Shoemaker**  
craigshoemaker.net  
@craigshoemaker



**pluralsight**   
hardcore developer training

# Contents

Introduction

Web  
Storage

IndexedDB

File System

Libraries

Introduction &  
Concepts


Initialization & CRUD  
Operations

Cursors, Indexes &  
Ranges

Keys, Capacity,  
Performance &  
Versions

# Unique Identifiers

# autoIncrement



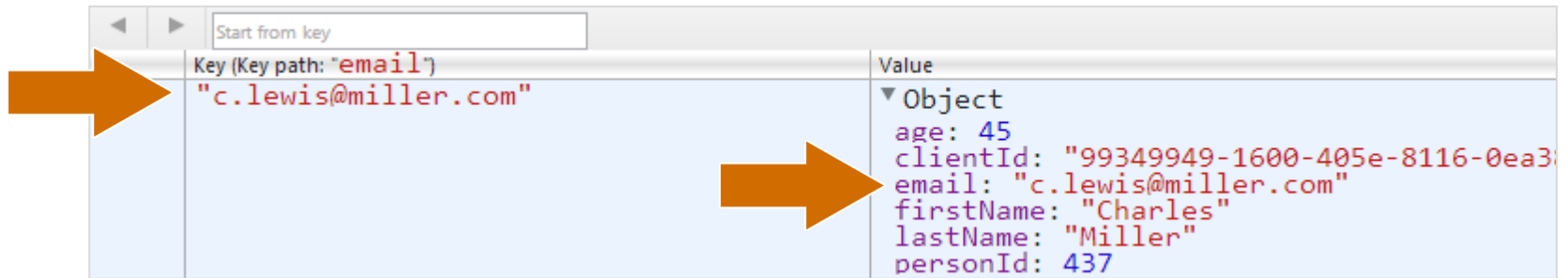
Start from key	
Key	Value
1	<div>▼ Object</div> <div>age: 64</div> <div>clientId: "b2ff00b2-ba28-4a51-816a-8ca7"</div> <div>email: "r.perez@jackson.com"</div> <div>firstName: "Ronald"</div> <div>lastName: "Wilson"</div> <div>personId: 356</div>

## RESULT



- [Walker] Gary Walker, 29
- [Walker] William Walker, 29
- [Walker] Paul Walker, 39
- [Walker] John Walker, 29
- [Walker] Brian Walker, 56

# keyPath

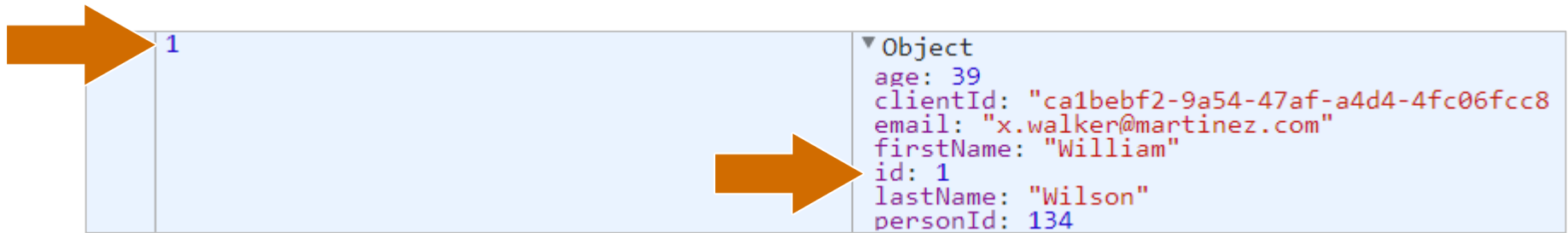


The diagram illustrates the concept of a keyPath in a JSON object. It shows a table with two columns: 'Key (Key path: "email")' and 'Value'. The 'Key' column contains the string '"c.lewis@miller.com"', and the 'Value' column contains a JSON object. An orange arrow points from the left to the 'Key' column, and another orange arrow points from the 'Value' column to the 'email' property within the object.

Key (Key path: "email")	Value
"c.lewis@miller.com"	<pre>Object   age: 45   clientId: "99349949-1600-405e-8116-0ea3"   email: "c.lewis@miller.com"   firstName: "Charles"   lastName: "Miller"   personId: 437</pre>

**The best of both  
worlds**

# keyPath + autoIncrement




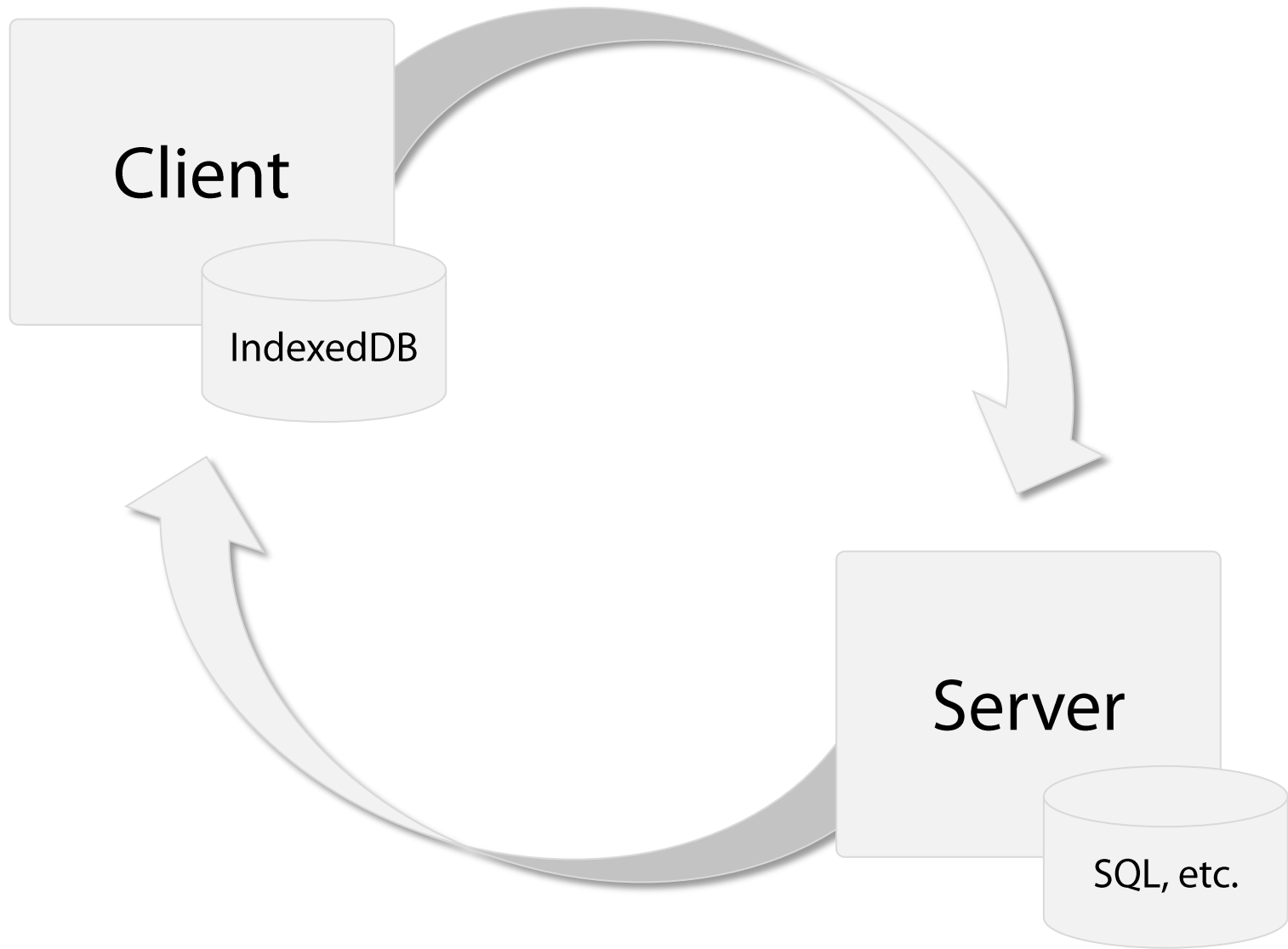
The diagram illustrates a table with one row and its corresponding object structure. An orange arrow points from the left to the first column of the table, which contains the value '1'. Another orange arrow points from the first column to the 'id' property of the object structure.

1	<div>▼ Object age: 39 clientId: "ca1bebf2-9a54-47af-a4d4-4fc06fcc8" email: "x.walker@martinez.com" firstName: "William" id: 1 lastName: "Wilson" personId: 134</div>
---	--



# Universally Unique Identifier

Start from key		
#	Key	Value
0	1	 ▼ Object age: 64 clientId: "b2ff00b2-ba28-4a51-816a-8ca7" email: "r.perez@jackson.com" firstName: "Ronald" lastName: "Wilson" personId: 356



# clientId

Start from key	
#	Value
Key (Key path: "clientId") "1b5c9c51-5b6a-447d-9b26-6b3265e19725"	▼ Object age: 26 clientId: "1b5c9c51-5b6a-447d-9b26-6b3265e19725" email: "w.garcia@white.com" firstName: "Jeffrey" lastName: "Garcia" personId: 479

# Summary



Storage ▾

IndexedDB ▾

File System ▾

Concepts ▾

Libraries ▾

## Performance

**Note:** Since there are so many objects loaded into this store make sure you give the browser a sufficient amount of time to seed the store the first time you run this demo. It's possible the database could take perhaps 30 seconds or so to open the database once data is in the store.



Range (numeric)

Range (string)

```
var
  parts = $('#input-box').val().split('-'),
  startAge = 0,
  endAge = 0;

startAge = Number(parts[0]);

if (isNaN(startAge)) {
  $$result.log('You must enter a numeric value.');
```

### RESULT

- Creating `people` store
- Attempting to seed data store with 500K objects...
- 500000 people objects are now added to the `people` store.
- **Make sure to wait for the open notification (after this line) to appear before you proceed.**
- The `PerformanceDB` database open and ready.

# Up Next...



Storage ▾

IndexedDB ▾

File System ▾

Concepts ▾

Libraries ▾

## Add Home

Add a new home for listing.

Address

1234 Main Street

City

Anytown

State

CA

Zip  
Code

90210

Comments

Nice corner lot

Address	City	Action
345 Plumb Avenue	Whittier	<button>Select</button> <a href="#">Delete</a>