

O'REILLY®



Client-Side Data Storage

KEEPING IT LOCAL

Raymond Camden

Client-Side Data Storage

by Raymond Camden

Copyright © 2016 Raymond Camden. All rights reserved.

Printed in the United States of America.

Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.

O'Reilly books may be purchased for educational, business, or sales promotional use. Online editions are also available for most titles (<http://safaribooksonline.com>). For more information, contact our corporate/institutional sales department: 800-998-9938 or corporate@oreilly.com.

Editor: Meg Foley

Production Editor: Nicholas Adams

Copyeditor: Rachel Monaghan

Proofreader: James Fraleigh

Indexer: Wendy Catalano

Interior Designer: David Futato

Cover Designer: Randy Comer

Illustrator: Rebecca Demarest

January 2016: First Edition

Revision History for the First Edition

2015-12-18: First Release

See <http://oreilly.com/catalog/errata.csp?isbn=9781491935118> for release details.

The O'Reilly logo is a registered trademark of O'Reilly Media, Inc. *Client-Side Data Storage*, the cover image, and related trade dress are trademarks of O'Reilly Media, Inc.

While the publisher and the author have used good faith efforts to ensure that the information and instructions contained in this work are accurate, the publisher and the author disclaim all responsibility for errors or omissions, including without limitation responsibility for damages resulting from the use of or reliance on this work. Use of the information and instructions contained in this work is at your own risk. If any code samples or other technology this work contains or describes is subject to open source licenses or the intellectual property rights of others, it is your responsibility to ensure that your use thereof complies with such licenses and/or rights.

978-1-491-93511-8

[LSI]

Table of Contents

Preface.....	v
1. A Gentle Introduction to Client-Side Data Storage.....	1
2. Working with Cookies.....	3
Cookies? Seriously?	3
Working with Cookies	4
Reading Cookies	5
Deleting Cookies	6
Demos	6
Inspecting Cookies Within Developer Tools	10
Support and Recommended Usage	11
3. Working with Web Storage.....	13
Web Storage, AKA Local Storage	13
Working with Web Storage	14
Demos	15
Listening for Storage Changes	19
Inspecting Web Storage with Dev Tools	23
Support and Recommended Usage	24
4. Working with IndexedDB.....	27
Welcome to Deep Data	27
Key IndexedDB Terms	28
Checking for IndexedDB Support	29
Working with Databases	29
Working with Object Stores	31
Making Object Stores	32

Defining Primary Keys	34
Defining Indexes	36
Working with Data	37
Creating Data	38
Reading Data	42
Updating Data	45
Deleting Data	47
Getting All the Data	48
Working with Ranges and Indexes	51
Even More with IndexedDB	54
Storing Arrays	54
Counting Data	59
Inspecting IndexedDB with Dev Tools	59
Support and Recommended Usage	61
5. Working with Web SQL.....	63
Dead Spec Walking	63
Basic Database Terms	64
Checking for Web SQL Support	64
Working with Databases	64
Working with Transactions	66
Inspecting Web SQL with Dev Tools	72
Support and Recommended Usage	73
6. Making It Easier with Libraries.....	75
“Use the Library, Luke...”	75
Working with Lockr	75
Simplifying IndexedDB with Dexie	80
Working with localForage	88
More Options	90
7. Building a Sample Application.....	91
Let’s Build Something!	91
Our Sample Data	92
The Application	95
The Code	97
Wrap-up	103
Index.....	105

A Gentle Introduction to Client-Side Data Storage

Browsers have—slowly and with a lot of growing pains—evolved over the past decade to become powerful workhorses. Enhanced layout controls, 3D graphics and gaming, and even music are now within the realm of possibility of the little old browser. One more exciting, although relatively less flashy (no pun intended), feature is client-side data storage. But what do we mean by this?

The “typical” process by which you navigate the Web hasn’t changed since time began: your browser asks for a URL, a web server sends stuff back, you then ask for more stuff, and the web server sends more stuff to you.

You can certainly get more complex and add JavaScript and AJAX to the mix. But even in a fancy web 2.0 application, your browser may be requesting information from the server again and again and again. The reason for this is that—for all intents and purposes—the browser is an amnesiac. Everything it knows it has to learn from the server.

While this is true in general, it overlooks a powerful alternative—storing data on the browser itself. This enables the browser to skip asking the server for information and to simply retrieve it locally from the user’s machine. It even affords the chance to manipulate that data for whatever purposes may make sense. That data could then be sent back to the server for updating later.

In summary, this gives the browser:

- Immediate access to data. Even with AJAX being typically much quicker for fetching data, by having the data local to the machine itself access will be even quicker.

- Less network traffic. Instead of constantly fetching data from the server, data can be fetched once and stored for as long as it makes sense. This leads to...
- Less strain on your server. If your server is constantly responding to requests and fetching stuff from a database server, you could overtax your server. By reducing the amount of calls you make, your server does less work.
- Finally, with data stored locally, the possibility of a fully offline application becomes much more feasible.

Of course, it isn't all sunshine and roses. Moving data to the browser can also have the following negatives:

- There isn't any support for synchronization. Imagine you've copied data from a server to a browser. How do you handle keeping that data in sync? What happens if there are conflicts? None of the core technologies covered in this book support any concept of handling sync. You can, however, find libraries like **PouchDB**, that have this built in.
- Storage limits are fuzzy. As developers, we hate fuzzy. We want to know exactly how much of a resource we can use. Unfortunately, for many of the technologies this book will cover, the limits—as well as what happens when you break them—are a bit vague.
- Finally, while the technologies covered here are powerful, they are not a replacement for a full database server. Database servers are extremely fine-tuned to the job of handling huge amounts of data and providing a way to search them. The solutions we'll cover are certainly capable of storing data, but at the same time, they aren't like an embedded Oracle server. (Although that may be a good thing.)

In this book, we'll discuss various client-side storage techniques. For each one we'll also cover, plainly and honestly, how well they are supported in the wild. You'll see examples of the APIs as well as demos you can use to help you learn how to use the APIs yourself. Finally, we'll look at a few libraries that aim to make client-side storage even easier. Buckle up—this is going to be a fun, and sometimes rough, tour through some of the more useful aspects of the Web!