



# CouchDB, DayBed, etc.

Classic [List](#) [Threaded](#)

7 messages [Options](#) ▾

[Pyyhtiä Christer via use-livecode](#)

[Reply](#) | [Threaded](#) | [More](#) ▾

Aug 23, 2017; 10:28am **CouchDB, DayBed, etc.**



11427 posts

Just a quick thank you to Greg for the info on his Daybed library, and for the Udemy course on CouchDB. I found the course fascinating - and I watched it first so I had the overview before watching the Daybed specific talk in Livecode Global.

(And I have to admit I watched it all on 1.5x speed, and skipped one of each of the 'Futon'/'Fauxton' parts, so it was only only about 2-1/2 hours :-)) I wish we had that speeding capability on the Livecode Global webinar replays rather just on the (crippled) Youtube version.)

I found the Daybed library fascinating - and as soon as it can cover Couchant (so I have an easy, free way to try it out :-)), I will be playing with it.

BUT - I have a fairly fundamental problem that I don't know how to resolve - so I'll throw it out here and see if there's an obvious solution I've missed.

With the exception of a couple of things that basically scraped data off some web pages, every app I've done in the last couple of years (all for personal use by myself or friends - I'm a hobbyist) have shared one characteristic : they have local storage, and that data is subsequently synched to "the cloud" to be shared with other devices/users, or at the very least backed-up to the cloud (i.e. my on-rev or HostM server account).

I haven't yet done anything on mobile - but because of where I live and other circumstances, even for laptops I *\*need\** to have things work without an Internet connection and synch up when available.

This doesn't seem to fit well with, say, CouchDB; the REST API lets me do great things when I talk to the server - but when working locally I have to forego all those features, and, for instance, revert to an array which I 'manually' search / filter / etc.

Currently, what I do is:

- local SQLite database, and use Andre's DBLib, including selecting etc. for filtering
- remote MySQL accesses through my own server page (with a few added

columns)  
- home-grown synch process that asynchronously uploads / downloads changes (\*).

Is there a NoSQL equivalent ?

Or has anyone written a LC library that does array searching / filtering / etc. that would provide similar, even if less extensive, features ?

Thanks for any comments, suggestions, etc.  
Alex.

(\*) my homegrown synch process:

- I use a "data model layer", and all changes to the DB go through it. As well as writing to the SQLite database, it adds an update record to the end of a queue of changes (each held in a separate textfile).
- each record has an UUMID (Universally Unique Meaningful ID) - made up of the device name where it is being created plus the date/time to the millisec
- each record has a 'modified' timestamp (local time on the device, so not universally comparable)
- each record has a 'accepted' timestamp, which is the time \*on the server\* when the change is synched on to the server (and so they are all universally comparable).
- the synch handler runs continuously, is triggered at start-up to download changes, or by any DB update or periodically for downloading new updates - and re-sends to itself while there are remaining synch's to do; i.e. non-blocking, and if there is no Internet connection, it simply waits a while and tries again
- for now, conflict resolution is \*only\* "server wins"

Unfortunately, while the skeleton of the synch handler, and of the server page, are reusable, they need to be fleshed out each time. One of these days I'll find time to look at this properly, and see how close it can become to a fixed, standard library for the synch capability (and look at conflict resolution more closely).

---

use-livecode mailing list

[\[hidden email\]](#)

Please visit this url to subscribe, unsubscribe and manage your subscription preferences:

<http://lists.runrev.com/mailman/listinfo/use-livecode>

**[Pyyhtiä Christer via use-livecode](#)**

[Reply](#) | [Threaded](#) | [More](#) ▾

Aug 23, 2017; 5:43pm **Re: CouchDB, DayBed, etc.**



Alex,

- > local storage, and that data is subsequently
- > synched to "the cloud"

You might want to take a look at Mark Talluto's LiveCloud.

<http://livecloud.io>

11427 posts

And I agree Greg's Daybed library and CouchDB course are very good.

Jim Lambert

---

use-livecode mailing list

[\[hidden email\]](#)

Please visit this url to subscribe, unsubscribe and manage your subscription preferences:

<http://lists.runrev.com/mailman/listinfo/use-livecode>

[pink](#)

[Reply](#) | [Threaded](#) | [More](#) ▾

Aug 23, 2017; 8:38pm **Re: CouchDB, DayBed, etc.**



60 posts

In reply to [this post](#) by Pyyhtiä Christer via use-livecode

Daybed's built in local storage system is the save stack, works well on all platforms. I am actually fiddling around with using SQLite as a save file as well, just haven't had a lot of free time. You can set things to sync whenever possible rather easily as well.

CouchDB does have a local save and sync format called PouchDB. Unfortunately I have yet to find a way to get it working in Livecode. It is largely javascript based, so I believe there should be way of using a browser widget to do the heavy lifting.

Thanks for taking the course! I just added a bunch of lessons on replication, a lesson on creating a production ready server with SSL, and I am planning on a section about setting up a cluster.

---

---

Greg (pink) Miller  
mad, pink and dangerous to code

[Pyyhtiä Christer via use-livecode](#)

[Reply](#) | [Threaded](#) | [More](#) ▾

Aug 24, 2017; 1:59am **Re: CouchDB, DayBed, etc.**



In reply to [this post](#) by Pyyhtiä Christer via use-livecode

Alex

> On 23 Aug 2017, at 22:28, Alex Tweedly via use-livecode <[\[hidden email\]](#)> wrote:

>

> With the exception of a couple of things that basically scraped data off some web pages, every app I've done in the last couple of years (all for personal use by myself or friends - I'm a hobbyist) have shared one characteristic : they have local storage, and that data is subsequently synched to "the cloud" to be shared with other devices/users, or at the very least backed-up to the cloud (i.e. my on-rev or HostM server account).

>

> I haven't yet done anything on mobile - but because of where I live and other circumstances, even for laptops I *\*need\** to have things work without an Internet connection and synch up when available.

>

> This doesn't seem to fit well with, say, CouchDB; the REST API lets me do great things when I talk to the server - but when working locally I have to forego all those features, and, for instance, revert to an array which I 'manually' search / filter / etc.

You can easily run CouchDB on a laptop or PC. There is a little more admin than SQLite but not very much. One of the strengths of CouchDB is its database synchronisation (one-way or both-ways) -

<http://docs.couchdb.org/en/2.1.0/replication/index.html> <<http://docs.couchdb.org/en/2.1.0/>

There is also a mobile embedded Couchbase that should sync with CouchDB -

<https://developer.couchbase.com/mobile> <<https://developer.couchbase.com/mobile>> - I guess that would need a LiveCode Builder wrapper before it could be accessed from LiveCode.

Peter

---

use-livecode mailing list

[[hidden email](#)]

Please visit this url to subscribe, unsubscribe and manage your subscription preferences:

<http://lists.runrev.com/mailman/listinfo/use-livecode>

---

## [Pyyhtiä Christer via use-livecode](#)

[Reply](#) | [Threaded](#) | [More](#) ▾

Aug 24, 2017; 12:09pm **Re: CouchDB, DayBed, etc.**



In reply to [this post](#) by pink

pink wrote:

- > Daybed's built in local storage system is the save stack, works well
- > on all platforms.

Have you considered LSON files (LiveCode encoded arrays)?

Where I used to use stack files for some storage options, I've been migrating to LSON with two benefits:

- Being raw data outside of a stack object, I never need to think about the IDE's limitation on stack names in memory.
- The syntax for working with the resulting variable is a tad simpler than with object properties.

As a collection of nested key-value pairs it would seem LSON is a good fit for CouchDB's way of working with data.

--

Richard Gaskin

Fourth World Systems

Software Design and Development for the Desktop, Mobile, and the Web

---

[[hidden email](#)]

<http://www.FourthWorld.com>

---

use-livecode mailing list

[[hidden email](#)]

11427 posts

**pink**

[Reply](#) | [Threaded](#) | [More](#) ▾

Aug 27, 2017; 10:37am **Re: CouchDB, DayBed, etc.**



60 posts

I've used "LSON" in the past and had issues. Not with save files, but with socket transmission...

I may play around a bit and see how things go

---

Greg (pink) Miller  
mad, pink and dangerous to code

**[Pyyhtiä Christer via use-livecode](#)**

[Reply](#) | [Threaded](#) | [More](#) ▾

Aug 27, 2017; 11:14am **Re: CouchDB, DayBed, etc.**



11427 posts

pink wrote:

- > I've used "LSON" in the past and had issues. Not with save files, but
- > with socket transmission...
- >
- > I may play around a bit and see how things go

LSON is binary, so set headers accordingly. When in doubt, base64.

I have two systems that use LSON almost exclusively to move data back and forth between the client and server. It's a great format, almost like BSON but simpler, and more compact than JSON. Too bad LC is the only program that uses it. But when you have LC on both ends, it's pretty sweet.

--

Richard Gaskin  
Fourth World Systems  
Software Design and Development for the Desktop, Mobile, and the Web

---

[\[hidden email\]](#) <http://www.FourthWorld.com>

---

use-livecode mailing list

[\[hidden email\]](#)

Please visit this url to subscribe, unsubscribe and manage your subscription preferences:  
<http://lists.runrev.com/mailman/listinfo/use-livecode>