

# Offline First

Making Your App Awesome  
When the Network Isn't

Teri Chadbourne

@EventT

# Offline First

- A **development approach** that plans for the most constrained network conditions first
- A **user experience** that just works, from the developing world to the urban commute, offline or on
- A **social movement** aimed at democratizing access to data

# Offline Camp

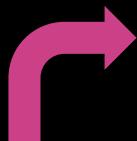


Building the Offline First community, one campfire at a time.

# The Offline Camp Medium Publication



@EventTeri #OfflineFirst



The screenshot shows the Offline Camp Medium publication website. The header features the Offline Camp logo and the tagline "Building the Offline First community, one campfire at a time." Below the header, there's a navigation bar with links like "DATA & SERVICE WORKERS", "COUCHCAMP & POUCHCAMP", "UX", "DECENTRALIZED-WEB", "DEVELOPING WORLD", "PASSION TALKS", and "RESOURCES". The main content area includes a photo of people gathered around a campfire, a section titled "Offline First as a Social Movement" with a sub-section about balancing social power, and a "Latest" section featuring three articles: "Blockchains: Beyond Bitcoin" by Bradley Holt, "Mapping the Internet" by Bradley Holt, and "Is the Offline First Community Ready for an Offline First Foundation?" by Key Guly.

Offline Camp  
Building the Offline First community, one campfire at a time.

DATA & SERVICE WORKERS COUCHCAMP & POUCHCAMP UX DECENTRALIZED-WEB DEVELOPING WORLD PASSION TALKS RESOURCES

Offline First as a Social Movement

Balancing social power, regardless of level of connectivity

Latest

Blockchains: Beyond Bitcoin

Mapping the Internet

Is the Offline First Community Ready for an Offline First Foundation?

# The editorial process

- Process starts while offline at camp
- Lots of back and forth with authors back home
- Multiple editors need a shared tool for collaboration
- Steps change based on type of article and availability of resources

**TEMPLATE - PASSION TALK <Name> Session Title**

In list List

Members Labels

Passion Talk Oregon

Edit the description...

Checklist

0%  

- Added as writer
- Sent rough draft of passion talk video
- Confirmed name for slide
- Confirmed talk title for slide and updated here
- Confirmed twitter handle for slide

Add an item...

Add Comment

Write a comment...

Save

Activity

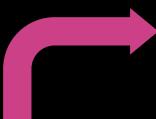
Teri Chadbourne copied this card from [TEMPLATE - SESSION RECAP](#) <Name> Session Title in list List Nov 29, 2017 at 3:09 PM

<TBD> Governance of Decentralized / P2P Networks

Feb 23 5 3/3

Add a card...

# A custom-built project management tool



## Blog Tracker

Add an Article

View Articles  
Select an article below to view or edit.

P2P Networks (Authorlicious)  
Fabulous Article (Authorlicious)  
Amazing Article (Super Author)

Filtered By: All Articles

All Status: Completed Editor: Bradley Gregor Steven Teri  
In progress Needs Author

Sync Status: Currently syncing to remote database.

This app works Offline First! The first time you load the page, a service worker caches the resources necessary for you to work offline in the future. The app saves data locally first using PouchDB, then syncs to a remote CouchDB database (in this case, Cloudant) whenever you have a connection. Using multiple devices or browsers? It will all sync up automatically!

Created by [Téri Chabourne](#)

@EventTéri #OfflineFirst

## Blog Tracker

Go Back (Discard)

### Edit Article

Article Title: P2P Networks

Article Type: Passion Talk

Presenter(s): Joe

Event: Offline Camp Oregon, November 2017

Author Detail:

Author:

#### Session Resources Available

Rough Cut Video  
 Session Notes  
 Pictures of session in Progress

#### Publication & Sharing Status:

Target Date:  /mm/dd/yyyy

Draft Received

Save & Close

## Getting Started on Your Blog Post

Way to be awesome, Super Author!

Thanks so much for volunteering to write up a recap of our unconference session on Amazing Article (and for all your great contributions at camp!).

Here's how to get started (and finished):

- Please send us your Medium username so that we can add you as a writer to the Offline Camp Medium publication. If you haven't used Medium before, note that you can sign in with another social media account to [get started](#). (Try clicking the extra Sign In button for more social media/network options.)
- Below you'll find links to any session audio, video, notes, or pics available, which you can use to jog your memory as needed.
- Feel free to write as short or as long a story as you'd like.
- You can find examples of other session recap posts in our Medium publication.
- If you're working on an Offline First project, feel free to mention it in your story. You might also want to link to relevant resources that provided added background on the topic at hand.
- Please add the tag "Offline First," as well as any other relevant tags for your topic, to your story using [These Instructions](#).
- Every Medium story needs at least one Featured Image (which cannot be a video), and you may add additional images if you'd like. Please be sure to add captions and credit the source of each image. If you can't think of what to use for a featured image, just reach out, and we can help brainstorm.
- Once we've added you as writer, you'll be able to add your draft post to the Offline Camp publication using [These Instructions](#) when you're ready. (Don't worry, that won't actually publish your article; just make it available to our editorial team for review.)
- Please let me know when you expect to be able to submit your draft. We've found that it's easiest to write within the first couple of weeks after Offline Camp while things are fresh in your memory.

Here are the resources you can use to jog your memory:

- Rough-Cut Video: <https://www.vimeo.com>
- Session Notes: [www.vanson.com](http://www.vanson.com)

Thank you so much in advance for taking the time to work on this. Feel free to reach out with any questions, either by email or on Slack!

The Offline Camp Team

...that  
works  
offline

# A beginner's guide to...

- Enabling offline data storage, sync, and sharing with PouchDB and CouchDB
- Ensuring quick page loads—offline or not—with a service worker
- Avoiding gotchas

...all with only client-side (front-end) code

**“Stop treating offline as an error condition.**

**With Offline First, we make offline the baseline and work up from there.”**

John Kleinschmidt,  
GitHub

# Progressive enhancement

- The **data and resources** we choose to cache locally determine the **baseline functionality** that our app will have while offline
- Everything that requires a network connection is icing on the cake

# A beginner's guide to...

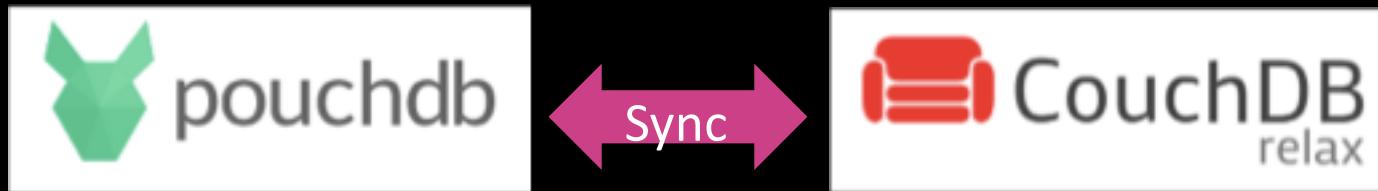


- Enabling offline data storage, sync, and sharing with PouchDB and CouchDB
- Ensuring quick page loads—offline or not—with a service worker
- Avoiding gotchas

...all with only client-side (front-end) code

# Data Storage, Sync, and Sharing

- Data stored **locally** is available **offline**
- Data stored **remotely** can be **shared** by multiple browsers, devices, or users





# DEMO

## Data Storage, Sync, and Sharing



# Defining Demo Success

```
const demoSuccess = function (networkStatus) {  
  if (networkStatus == 'awesome') {  
    proveOfflineFirstIsAwesome()  
  } else if (networkStatus == 'crappy') {  
    proveOfflineFirstIsNecessary()  
  }  
}
```

# Defining Demo Success

useHotspot()

# PouchDB

- In-browser database inspired by **Apache CouchDB™**
- Open source
- Has an asynchronous API (can use either promises or callbacks)
- Cross-browser
- JavaScript library



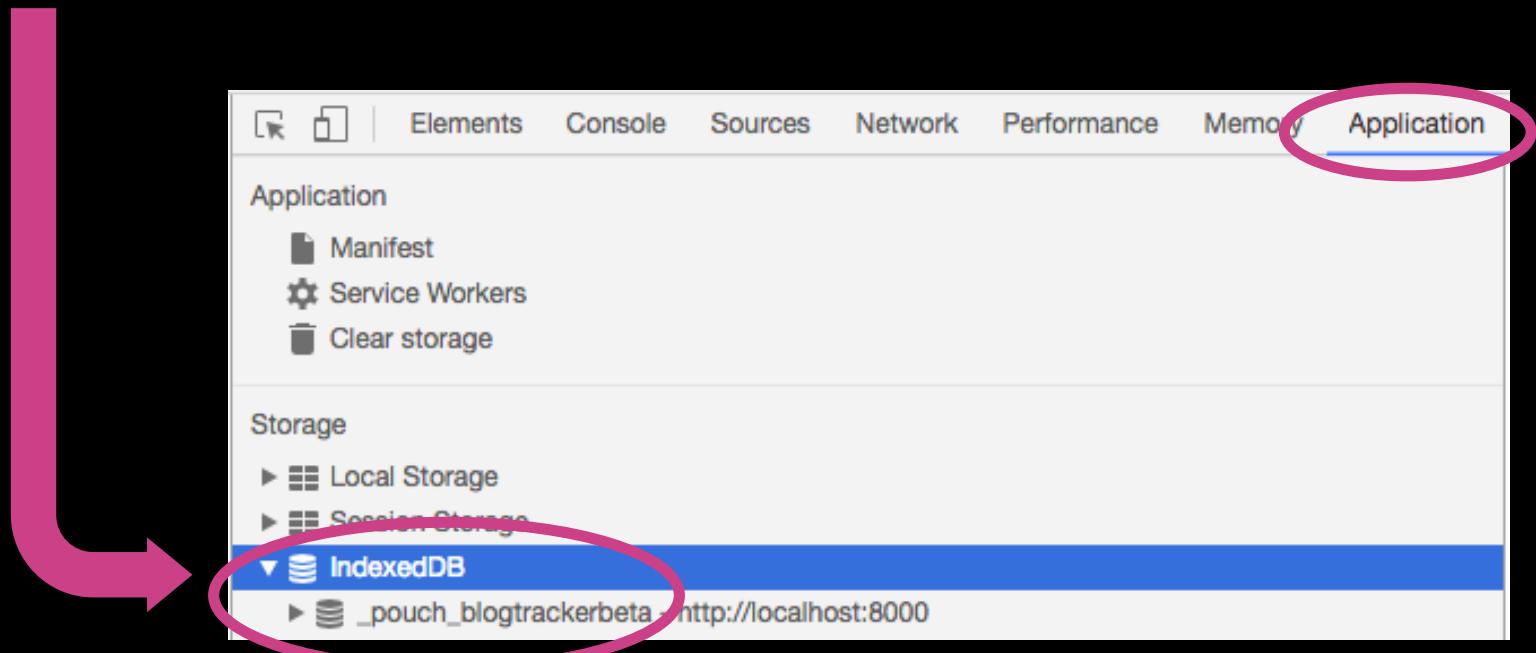
# Using the PouchDB library

- Reference the PouchDB script in your HTML, before the other JavaScript files that depend on it

```
<script src="js/pouchdb-6.3.4.js"></script>
<script src="js/jquery-3.2.1.js"></script>
<script src="js/project-manager.js"></script>
```

# Creating a local PouchDB database

```
var db = new PouchDB('blogtrackerbeta');
```



# Apache CouchDB™

- NoSQL document store
- Stores documents in JSON format
- Developer-friendly query language called Mango
- Couch Replication Protocol well-suited for offline applications



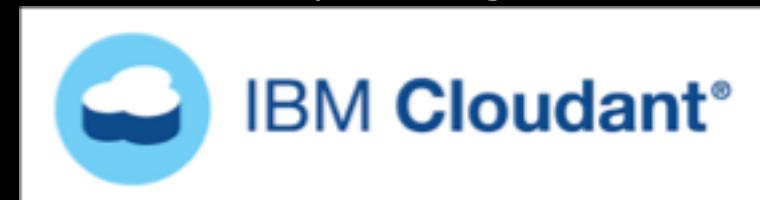
# Creating a remote Apache CouchDB™ database

- Generate an API key
- Enable Cross-Origin Resource Sharing (CORS)
- Note the URL used to access your database

Open Source



Fully-Managed



# A Cloudant database

The screenshot shows the Cloudant NoSQL Database interface. On the left is a sidebar with icons for different database management tasks: All Documents (selected), Query, Permissions, Changes, Design Documents, and a gear icon. The main area displays the 'All Documents' view for the database 'blogtrackerbeta'. At the top right are buttons for Document ID, Table, Metadata, JSON, Create Document, and a bell icon. Below these are buttons for Table, Metadata, and JSON. The table itself has columns for id, key, and value. Three document entries are listed:

	id	key	value
<input type="checkbox"/>	2017-12-13T23:43:54.775Z	2017-12-13T23:43:54.775Z	{"rev": "8-f53c54ccf7c54..."}
<input type="checkbox"/>	2017-12-13T23:48:10.552Z	2017-12-13T23:48:10.552Z	{"rev": "2-384fd11b7117..."}
<input type="checkbox"/>	2017-12-13T23:49:13.837Z	2017-12-13T23:49:13.837Z	{"rev": "2-76cecc54b338..."}

<https://username:password@username-bluemix.cloudant.com>

# Connecting to your CouchDB database

```
var remoteCouch = "YOUR_REMOTE_COUCHDB_URL";
```

```
if (remoteCouch) {  
    sync();  
}
```

# Syncing between PouchDB and CouchDB

- Set up continuous sync
- Listen for changes

```
function sync() {  
  var opts = {live: true};  
  db.sync(remoteCouch, opts, syncError)  
    .on('change', updateArticles);  
}
```

# Alerting the user to a sync error

- Offline status is a heads up, not an error
- Reassure the user that their data is safe offline

**Sync Status: Error syncing to remote database. Data will continue saving locally until connection is re-established.**

# Creating a JSON object

```
{  
  "_id": "mittens",  
  "name": "Mittens",  
  "occupation": "kitten",  
  "age": 3,  
  "hobbies": [  
    "playing with balls of yarn",  
    "chasing laser pointers",  
    "lookin' hella cute"  
  ]  
}
```

From the PouchDB guide

# Assigning a unique ID

- The `_id` must be **unique** and stay **consistent** across all updates
- Common practice is to use a timestamp

```
var ID = new Date().toISOString();
```

# Managing revision markers

- The `_rev` is generated by PouchDB and can't exist on first write
- On subsequent updates, you need to pass in the same `_rev` most recently assigned by PouchDB

```
if (location.hash !== "#new") {  
    var ID = $("#_id").val();  
    var REV = $("#_rev").val();  
} else if (location.hash == "#new") {  
    var ID = new Date().toISOString();  
    var REV = null;  
}
```

# Saving form entries as JSON

```
var article = {  
    //fields created and updated without user  
    input  
    _id: ID,  
    _rev: REV,  
  
    //fields copied from form entries  
    title: $("#title").val(),  
    author: $("#author").val(),  
    completed: $("#completed").prop("checked"),  
};
```

# Writing documents with db.put()

```
db.put(article
  ).then(function (response) {
    //...stuff to do if save succeeds...
  }).catch(function (err) {
    //...stuff to do if save fails...
});
```

# Fetching all documents with db.allDocs()

```
db.allDocs([options, callback])
```

```
function updateArticles() {
  db.allDocs({
    include_docs: true,
    descending: true
  }, function(err, doc) {
    redrawArticleList(doc.rows);
  });
}
```

# Data returned by db.allDocs()

```
{  
    "total_rows": 24,  
    "offset": 0,  
    "rows": [  
        {  
            "id": "2017-12-07T06:06:55.303Z",  
            "key": "2017-12-07T06:06:55.303Z",  
            "value": {  
                "rev": "1-3d5ae7e898a64e85a4d2f166e7472abc"  
            },  
            "doc": {  
                "_id": "2017-12-07T06:06:55.303Z",  
                "_rev": "1-3d5ae7e898a64e85a4d2f166e7472abc",  
                "objectType": "article",  
                "title": "added while offline but the page was already loaded",  
                "articleType": "session_recap",  
                "author": "teri",  
                "event": "teri"  
            }  
        },  
        {  
            "id": "2017-12-07T06:05:22.384Z",  
            "key": "2017-12-07T06:05:22.384Z",  
            "value": {  
                "rev": "1-ce2e270bc81c4dbd86dc4f2851233a8e"  
            },  
            "doc": {  
                "_id": "2017-12-07T06:05:22.384Z",  
                "_rev": "1-ce2e270bc81c4dbd86dc4f2851233a8e",  
                "objectType": "article",  
                "title": "a completed event recap for the catskills",  
                "articleType": "event_recap",  
                "author": "teri",  
                "event": "teri"  
            }  
        }  
    ]  
}
```

# Using the data returned

```
function redrawArticleList(articles) {  
  articles.forEach(function(currentArticle) {  
    $('#article-list').append("<li>" +  
      currentArticle.doc.title + " (" +  
      currentArticle.doc.author + "</li>");  
  });  
}
```

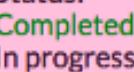
# The list of articles

**View Articles**

Select an article below to view or edit.

 <a href="#">P2P Networks</a>	 <a href="#">Fabulous Article</a> (Authorlicious)	 <a href="#">Amazing Article</a> (Super Author)
--	--	--

**Filtered By: All Articles**

All	
Status:	
Completed	
In progress	
Needs Author	

# Fetching individual documents with db.get()

```
db.get(_id).then(function(doc) {  
  //write data to page  
});
```



```
for (var d in doc){  
  if ($("#" + d).attr('type') == 'checkbox'){  
    $("#" + d).prop("checked", doc[d]);  
  } else {  
    $("#" + d).val(doc[d]);  
  }  
}
```

# The edit screen

Edit Article

Article Title

Article Type

Presenter(s)

Event

Author Detail:

Author

Medium Username:

Added as writer on Medium

Session Resources Available

Rough Cut Video available at:

Session Notes available at:



@EventTeri #OfflineFirst

# Deleting a record with db.remove()

```
db.get('mydoc').then(function(doc) {  
  return db.remove(doc._id, doc._rev);  
});
```

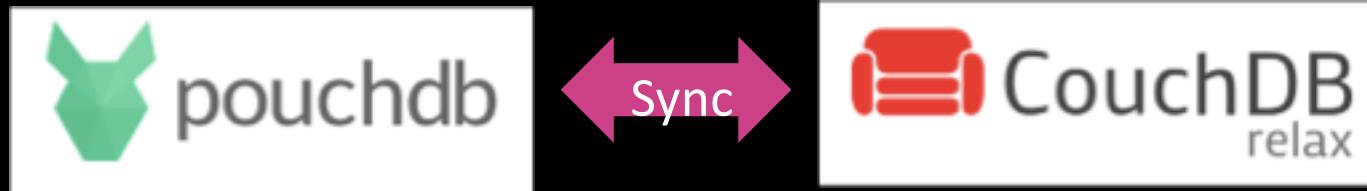


\_deleted: true

```
db.get($("#_id").val()).then(function(doc) {  
  return db.remove($("#_id").val(), $("#_rev").val());  
});
```

# Sync is the Offline First superpower

- Store and edit data locally while offline
- Sync data to the cloud when a network connection is available
- Share data from CouchDB across multiple users or devices, each with their own local PouchDB database



# A beginner's guide to...



- Enabling offline data storage, sync, and sharing with PouchDB and CouchDB
- **Ensuring quick page loads—offline or not—with a service worker**
- Avoiding gotchas

...all with only client-side (front-end) code

# Caching resources for quick page loads

- The **vital resources** that make a page functional (JavaScript, HTML, CSS) need to be available **offline**

Service  
Workers



# DEMO

## Caching Resources for Quick Page Loads



# No app can work offline on a first visit.

The “first” in Offline First means that we  
**prioritize user experience**  
on subsequent visits by  
**loading cached resources first**  
before trying to reach the network.

# Service workers

- Run in the background, separate from a web page
- Enable modern web features like **push notifications**, **background sync**, and **caching**
- Cache resources on a first visit to enable **lightning-fast** and **offline-capable** page loads on subsequent visits

# The service worker lifecycle

- **Register**
- **Install** (defining what to cache)
- **Activate**
- **Be incredible** (though also incredibly frustrating to debug)

# Registering a service worker

```
<script>
  if('serviceWorker' in navigator) {
    navigator.serviceWorker.register('sw.js')
      .then(function() {
        console.log("Service Worker Registered");
      });
  }
</script>
```

# Installing a service worker

```
self.addEventListener('install', function(e) {  
  e.waitUntil(  
    caches.open('blog-tracker').then(function(cache) {  
      return cache.addAll([  
        '/',  
        'index.html',  
        'css/style.css',  
        'js/project-manager.js',  
        'js/pouchdb-6.3.4.js'  
      ]);  
    })  
  );  
});
```

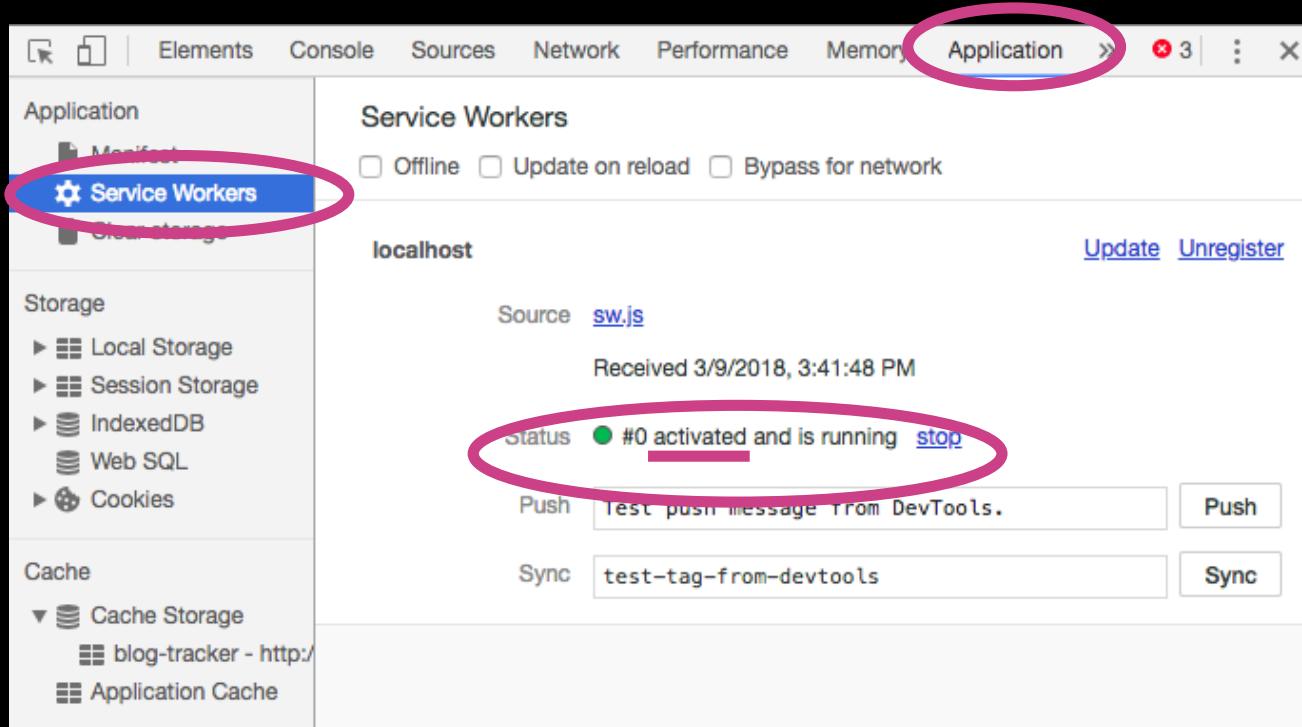
@EventTeri #OfflineFirst

# Cached resources

The screenshot shows the Chrome DevTools interface with the Application tab selected (indicated by a pink oval). On the left, the sidebar lists 'Application' (Manifest, Service Workers, Clear storage), 'Storage' (Local Storage, Session Storage, IndexedDB, Web SQL, Cookies), and 'Cache' (Cache Storage, Application Cache). Under 'Cache', 'Cache Storage' is expanded, showing a list of items including 'blog-tracker - http://'. The main panel displays a table of cached resources with columns: Path, Content-Type, Content-Length, and Time Cached. The table includes entries for '/', 'css/normalize-7.0.0.css', 'css/style.css', 'images/logo-450x450.png', 'index.html', 'js/credentials.js', 'js/jquery-3.2.1.js', 'js/pouchdb-6.3.4.js', and 'js/project-manager.js'.

Path	Content-T...	Content-...	Time Cac...
/	text/html; ...	14,676	3/9/2018, ...
css/normalize-7.0.0.css	text/css; c...	7,718	3/9/2018, ...
css/style.css	text/css; c...	3,068	3/9/2018, ...
images/logo-450x450.png	image/png	65,085	3/9/2018, ...
index.html	text/html; ...	14,676	3/9/2018, ...
js/credentials.js	application/...	133	3/9/2018, ...
js/jquery-3.2.1.js	application/...	268,039	3/9/2018, ...
js/pouchdb-6.3.4.js	application/...	416,557	3/9/2018, ...
js/project-manager.js	application/...	30,405	3/9/2018, ...

# Activating a service worker



# Intercepting network requests

- Access cached resources *first*, using the network as a fallback

```
self.addEventListener('fetch', function(event) {  
  event.respondWith(  
    caches.match(event.request).then(function(response) {  
      return response || fetch(event.request);  
    })  
  );  
});
```

use the cached file if found

otherwise find the file on the network

“If you want to build an application that has a really just **stunningly fast user experience**, the best thing you can do is go Offline First, which is preferring a local cache... at all costs to the network.”

Nolan Lawson,  
Microsoft Edge

@EventTeri #OfflineFirst



# A beginner's guide to...

- Enabling offline data storage, sync, and sharing with PouchDB and CouchDB
- Ensuring quick page loads—offline or not—with a service worker
- **Avoiding gotchas**

...all with only client-side (front-end) code

# Service workers are sticky.

@EventTeri #OfflineFirst

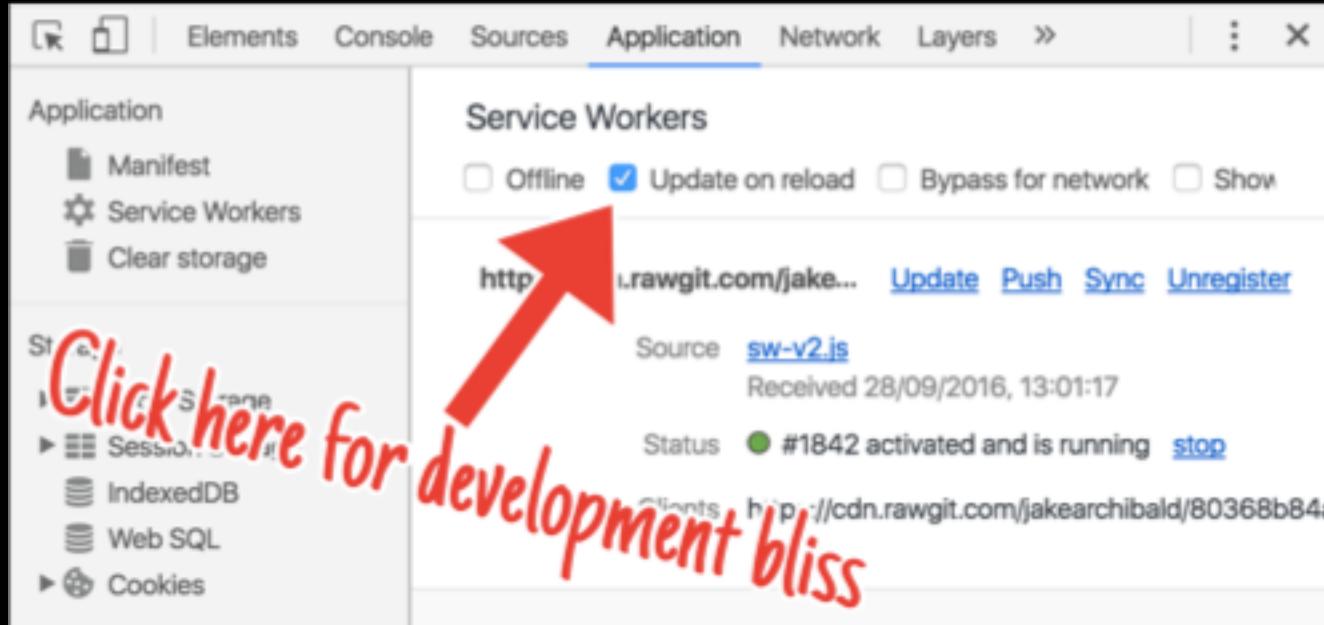


© Anders L. Damgaard | [www.amber-inclusions.dk](http://www.amber-inclusions.dk)

# Avoiding gotchas

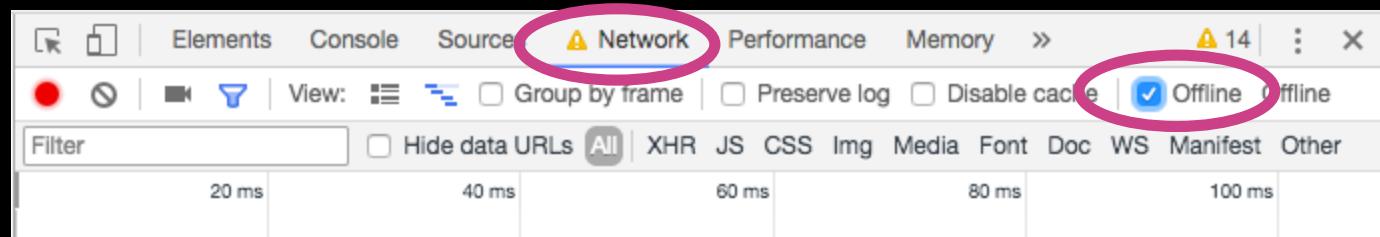
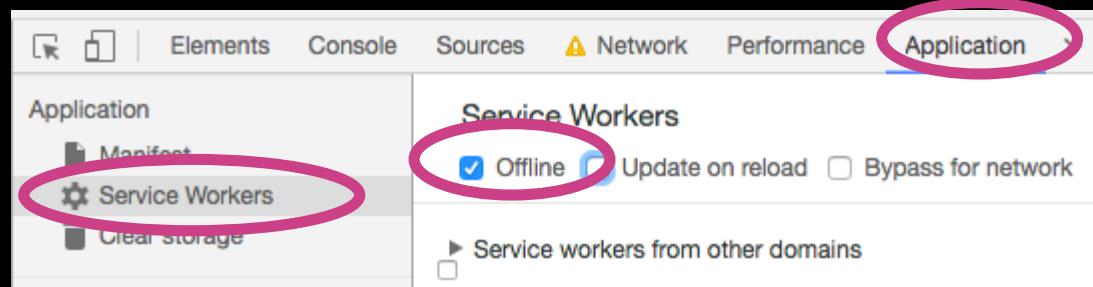
- Implement service workers last
- Learn how to do a hard refresh
- Use incognito browser windows
- Get familiar with Chrome Dev Tools
- Check out the resources at [developers.google.com](https://developers.google.com)
- Use “Update on reload” during development

# Debugging service workers

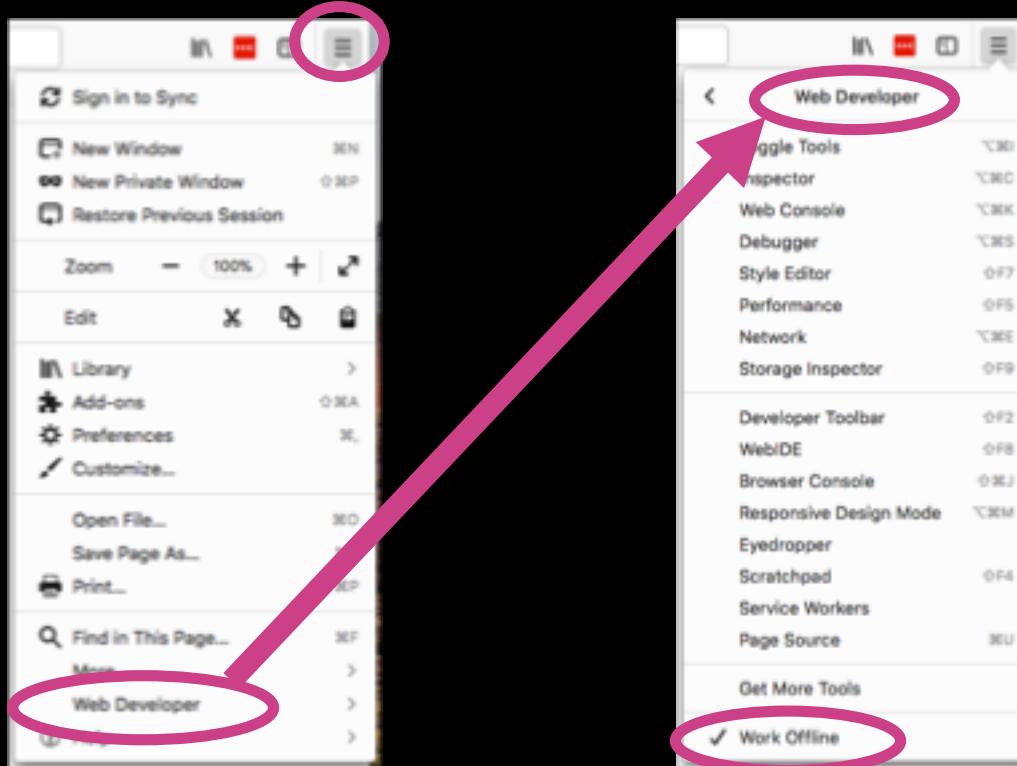


From [The Service Worker Lifecycle](#), by Jake Archibald

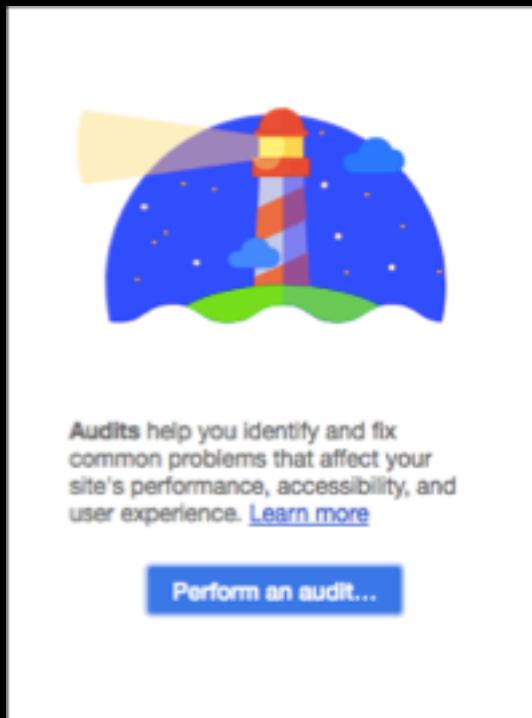
# Simulating an offline experience in Chrome



# Simulating an offline experience in Firefox



# PWA performance audits with Lighthouse

A screenshot of a web browser window titled "Home - SHOP" showing the Lighthouse audit results for the URL "https://shop.polymer-project.org". The browser's address bar also shows the date and time: "shop.polymer-project.org 9/26/2017, 4:59:10 P". The Lighthouse interface has a dark header with tabs for "Elements", "Console", "Sources", "Audits" (which is selected), and "Metrics". Below the header, there are four circular icons with scores: "Progressive Web App" (100), "Performance" (88), "Accessibility" (100), and "Best Practices" (85). The main content area displays audit results for the "Progressive Web App" category, which includes sections for "Passed Audits" (0 failed audits, 11 passed audits, and manual checks to verify), and "Metrics" (88 score).

Men's Outerwear  
SHOP NOW

Progressive Web App  
These audits validate the aspects of a Progressive Web App, as specified by the baseline [PWA Checklist](#).

0 failed audits

> 11 Passed Audits

> Manual checks to verify

Performance  
These encapsulate your app's performance.

Metrics  
These metrics encapsulate your app's performance across a number of dimensions.

[developers.google.com/web/tools/lighthouse/](https://developers.google.com/web/tools/lighthouse/)

Offline First isn't a specialized approach  
for low-bandwidth scenarios.

**It's just a great design pattern.**

Go make your app awesome.  
**You've got this.**

# Resources

- **GitHub Repo & Tutorial Articles:** [bit.ly/OfflineFirstProjectManager](https://bit.ly/OfflineFirstProjectManager)
- **PouchDB:** [pouchdb.com](http://pouchdb.com)
- **CouchDB:** [couchdb.org](http://couchdb.org)
- **Service Workers:**  
[developers.google.com/web/fundamentals/primers/service-workers/](https://developers.google.com/web/fundamentals/primers/service-workers/)
- **Offline First:** [offlinefirst.org](http://offlinefirst.org)
- **Offline Camp:** [offlinefirst.org/camp](http://offlinefirst.org/camp)
- **Offline First Resources:** [bit.ly/OfflineFirstResources](https://bit.ly/OfflineFirstResources)
- **Offline Camp Medium Publication:** [medium.com/offline-camp](https://medium.com/offline-camp)
- **Lighthouse:** [developers.google.com/web/tools/lighthouse](https://developers.google.com/web/tools/lighthouse)

# Offline First

Making Your App Awesome

When the Network Isn't



[bit.ly/OfflineFirstProjectManager](http://bit.ly/OfflineFirstProjectManager)



**Teri Chadbourne**

@EventTeri